OKI

Oki, Network Solutions for a Global Society

FEUL66525-01

ML66525 Family User's Manual

CMOS 16-bit microcontroller

Issue Date: Feb. 8, 2002

Preface

This user’s manual describes the hardware of Oki-original CMOS 16-bit microcontrollers ML66525 family. In addition to this manual, Oki also provides the following manuals which should be read with regard to the ML66525 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

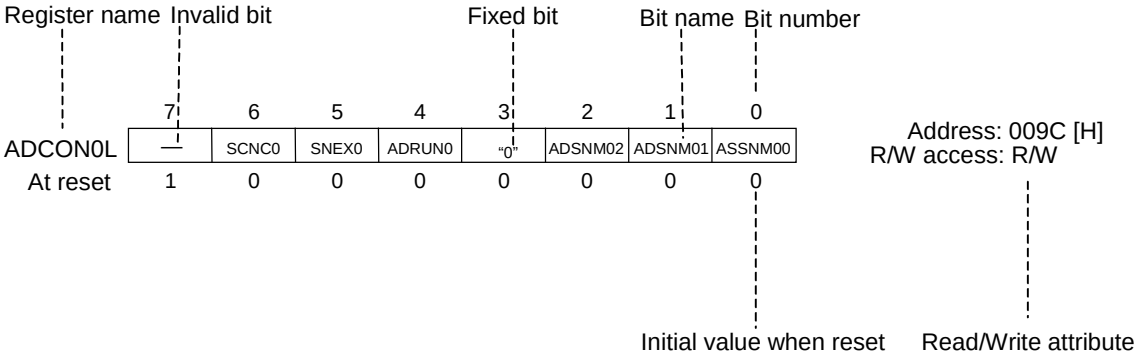
PW66K Flash Writer System User’s Manual
- PW66K Flash Writer System operation

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	1 KB = 1 kilobyte = 1024 bytes
	MB	1 MB = 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 4 Memory Control Functions

Chapter 5 Port Functions

Chapter 12 A/D Converter Functions

Chapter 13 Peripheral Functions

Chapter 14 External Interrupt Functions

Chapter 15 Interrupt Processing Functions

Chapter 18 Media Control Function

Chapter 19 Internal DMA Control Function

Chapter 20 Flash Memory

Chapter 1

Overview

1. Overview

1.1 Overview

The ML66525 family devices are high-performance 16-bit CMOS microcontrollers that utilize the nX-8/500S, Oki's proprietary CPU core.

Data from a personal computer with a USB connector can be automatically, quickly written or read to and from NAND type Flash Memory via USB I/F and NAND Flash Memory I/F.

The ML66525 family devices support clock gear functions, a sub-clock and HALT/STOP mode, which are suitable for low power applications.

The ML66525 family devices are provided with interfaces to external devices such as a 4-channel multi-functional serial interface with internal 32-byte FIFO and a high-speed bus interface that has separate address and data buses and does not require external address latches.

A wide variety of internal multi-functional timers enable various timing controls such as periodic and timed measurements.

With a 16-bit CPU core that enables high-speed arithmetic computations and a variety of bit processing functions, these general-purpose microcontrollers are optimally suited for Digital Audio devices such as MP3 players, voice recorders, handy games, and PC peripheral control systems (to control devices that can be connected to USB and store data into memory).

The ML66525 family devices also include the flash ROM version device (ML66Q525A) that is programmable with a single 3 V power supply (2.4 to 3.6 V).

1.2 Features

The ML66525 family has the following features.

- Instruction set with a wide variety of instructions
  - Dual-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
  - 83 ns at 24 MHz (2.4 to 3.6 V)
  - 61 µs at 32.768 kHz (2.4 to 3.6 V)

- Clock oscillation circuits
  - Main clock: 24 MHz (max.) crystal oscillator or ceramic resonator oscillator circuit
  - Subclock : 32.768 kHz crystal oscillator circuit

- Program memory (ROM)
  - Internal 128 KB
  - External 1 MB (in the case of EAn pin activated)

- Data memory (RAM)
  - Internal 6 KB + external 1016 KB

- I/O ports
  - Input ports 6 ports (secondary function is an analog input port)
  - Output port 1 port
  - I/O ports 64 ports max. (with programmable pull-up resistors)

- Timers
  - General-purpose auto reload timer 16 bits × 2, 8 bits × 1
  - 8-bit auto reload timer that also functions as a serial communication baud rate generator × 3
  - 8-bit auto reload timer that also functions as a watchdog timer × 1

- 8-bit PWM × 2

  (can also be used as one 16-bit PWM)

- 8-bit serial ports
  - UART/Synchronous × 2
  - Synchronous × 1
  - Synchronous with 32-byte FIFO × 1

- A/D converter
  - 10-bit resolution, 4 channels

- USB controller
  - Conforms to USB ver 1.1
  - High speed data transfer at 12 Mbps (Full-Speed)
  - 6 endpoints

    Controller EP 1
    Bulk/interrupt EP 3
    Isochronous/bulk/interrupt 2

- Media controller
  - NAND Flash memory I/F
  - ECC circuit

- Interrupts
  - Non-maskable × 1
  - Maskable: 6 external (5 vectors), 24 internal (23 vectors)
  - Three levels of priority

- ROM Window function

- Standby modes
  - HALT mode
  - STOP mode

- Separate power supplies
  - V_{DD}_CORE: Core power supply (excluding the USB controller section)
  - V_{DD}_IO: I/O power supply (excluding the following USB I/F pins: D+, D– and PUCTL)
  - VBUS: USB power supply (Vbus input pin)
  - V_{REF}: Power supply for the 10-bit A/D converter analog section; analog reference voltage

- Package
  - 100-pin plastic TQFP (TQFP100-P-1414-0.50-K)
  - 144-pin plastic LFBGA (P-LFBGA144-1111-0.80)
    (For external dimensions, refer to Chapter 23)

Table 1-1 ML66525 Family of Products

Parameter	ML66525
Operating temperature	–30 to +70°C
Power supply voltage/maximum operating frequency	V_{DD} = 2.4 to 3.6 V/f = 24 MHz
Minimum instruction execution time	83 nsec @ 24 MHz
	61 µsec @ 32.768 kHz
Internal ROM size (max. external)	128 KB (1 MB)
Internal RAM size (max. external)	6 KB (1 MB)
I/O ports	64 I/O pins (with programmable pull-up resistors), 6 input-only pins, 1 output-only pin
Timers	8-bit auto-reload timer × 1ch
	16-bit auto-reload timer × 2ch
	8-bit auto-reload timer (also functions as serial communication baud rate generator) × 3ch
	8-bit auto-reload timer (also functions as watchdog timer) × 1ch
	Watch timer × 1ch
	8-bit PWM × 2ch (16-bit PWM × 1ch)
Serial port	Synchronous (with 32-byte FIFO) × 1ch
	Synchronous (Shift register type) × 1ch
	Synchronous/UART × 2ch
A/D converter	10-bit × 4ch
External interrupts	Non-maskable × 1ch, Maskable × 7ch
USB control	Compliant with USB spec. version 1.1
	High-speed transfer at 12 Mbps
	Internal PLL (× 2, × 3, × 4) -> 48 MHz
	Internal transceiver
	Vbus detection circuit (connection to USB host : detect/non-detect)
	Bus power available
	EP0 (IN 32 bytes, OUT 32 bytes), control transfer
	EP1 (64 bytes × 2), bulk/interrupt transfer
	EP2 (64 bytes × 2), bulk/interrupt transfer
	EP3 (32 bytes), interrupt transfer
	EP4 (64 bytes × 2), bulk/isochronous/interrupt transfer
	EP5 (64 bytes × 2), bulk/isochronous/interrupt transfer
	Automatic, high-speed data transfer
Media control	ECC circuit
	Automatic, high-speed 512-byte data transfer
Interrupt priority	3 levels
Others	External bus interface (separate address and data buses)
	Dual clocks function
	Clock gear function
	Different powers available for USB, CPU core, and I/O port
Flash ROM version	ML66Q525A

1.3 Block Diagram

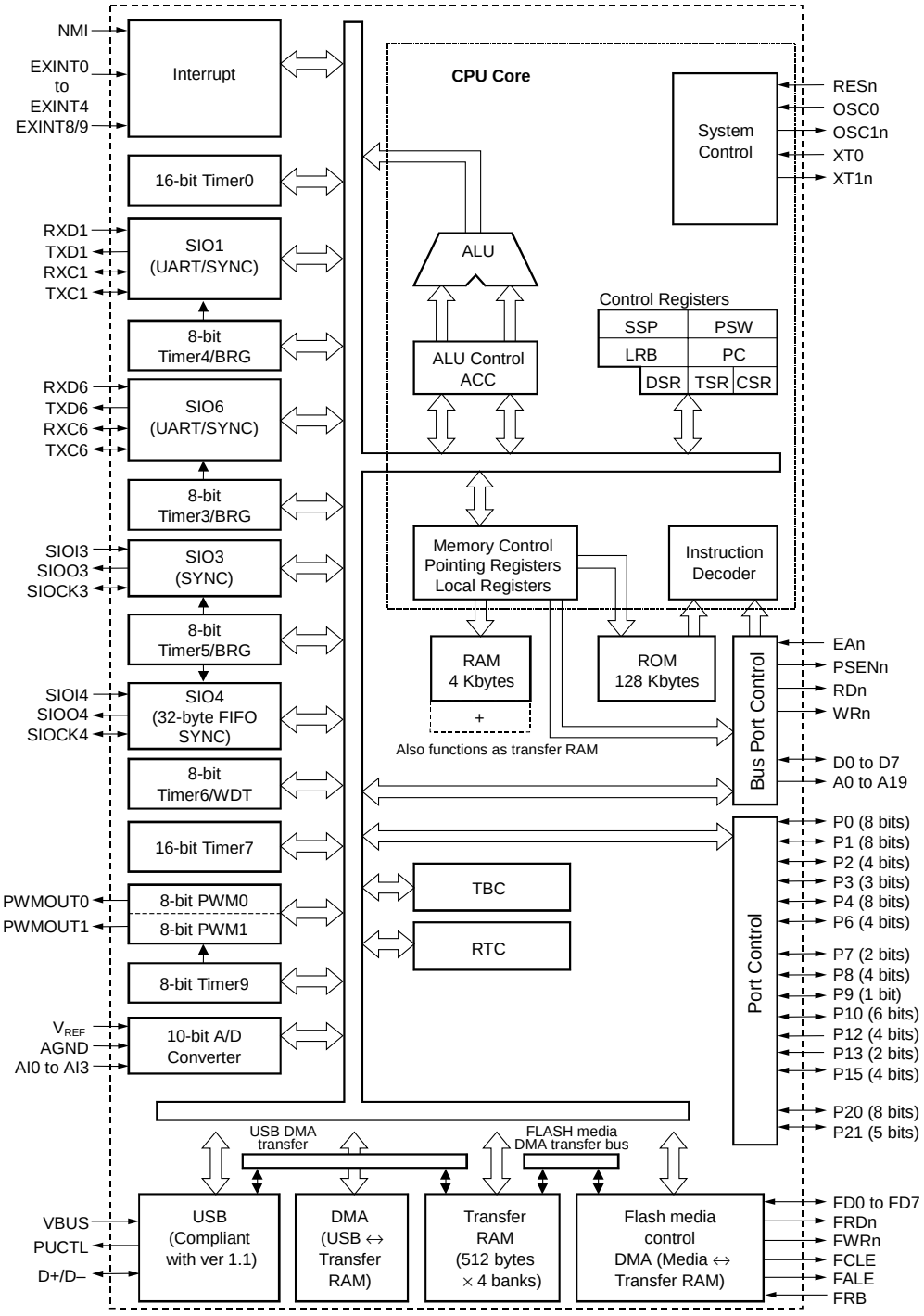
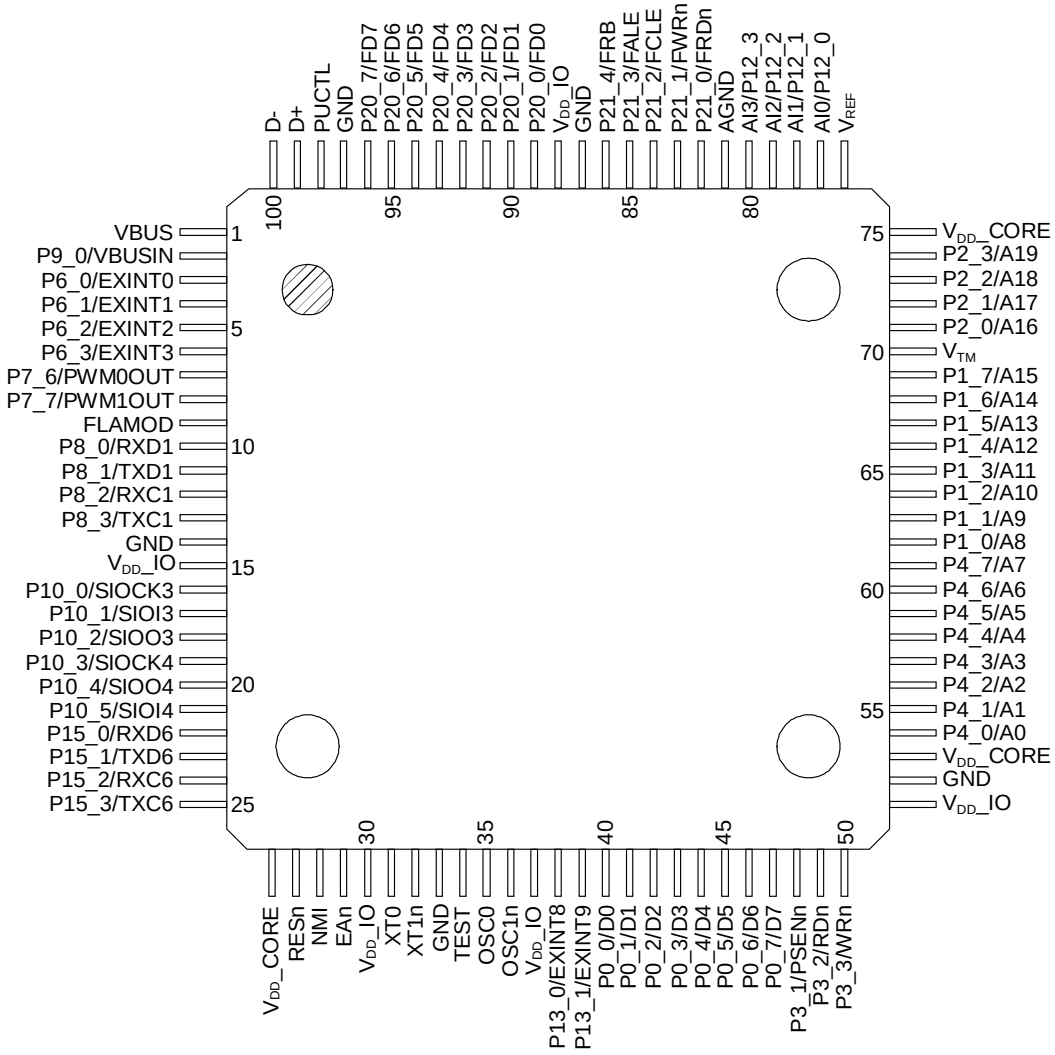


Figure 1-1 ML66525 Family Block Diagram

1.4 Pin Configuration (Top View)



100-Pin Plastic TQFP

A symbol with "n" suffixed indicates an active Low pin.

Figure 1-2 ML66525 Family Pin Configuration (TQFP)

* For the external dimensions of the package, refer to Chapter 23, "Package Dimensions". For the connections of unused pins, refer to Section 1.5.3.

NC	V_{DD}_IO	P3_2/ RDn	NC	P0_5/ D5	P0_3/ D3	P13_1/ EXINT9	OSC0	GND	XT0	NMI	V_{DD}_ CORE	NC	**N**
GND	P3_3/ WRn	P3_1/ PSENn	P0_4/ D4	P0_2/ D2	P0_1/ D1	V_{DD}_IO	OSC1n	TEST	XT1n	V_{DD}_IO	P15_2/ RXC6	P15_3/ TXC6	**M**
P4_0/ A0	NC	V_{DD}_ CORE	P0_7/ D7	P0_6/ D6	P0_0/ D0	P13_0/ EXINT8	NC	NC	EAn	RESn	P15_0/ RXD6	P15_1/ TXD6	**L**
P4_2/ A2	NC	P4_1/ A1	NC	NC	NC	NC	NC	NC	NC	P10_4/ SIOO4	P10_2/ SIOO3	P10_5/ SIOI4	**K**
P4_4/ A4	P4_5/ A5	P4_3/ A3	NC						NC	P10_3/ SIOCK4	NC	NC	**J**
P4_6/ A6	P4_7/ A7	P1_0/ A8	NC						NC	V_{DD}_IO	P10_0/ SIOCK3	P10_1/ SIOI3	**H**
NC	P1_1/ A9	P1_2/ A10	NC						NC	P8_3/ TXC1	P8_2/ RXC1	GND	**G**
P1_5/ A13	P1_4/ A12	P1_3/ A11	NC						NC	P8_1/ TXD1	P8_0/ RXD1	NC	**F**
NC	NC	P1_7/ A15	NC						NC	P7_6/ PWM0 OUT	FLAMOD	P7_7/ PWM1 OUT	**E**
NC	P1_6/ A14	V_{TM}	NC	NC	NC	NC	NC	NC	NC	P6_2/ EXINT2	NC	P6_3/ EXINT3	**D**
P2_1/ A17	P2_0/ A16	V_{REF}	P12_1/ AI1	P12_3/ AI3	P21_4/ FRB	V_{DD}_IO	P20_1/ FD1	P20_7/ FD7	NC	P6_0/ EXINT0	NC	P6_1/ EXINT1	**C**
P2_3/ A19	P2_2/ A18	NC	AGND	P21_1/ FWRn	P21_3/ FALE	GND	P20_2/ FD2	P20_3/ FD3	P20_5/ FD5	PUCTL	D-	P9_0/ VBUSIN	**B**
NC	V_{DD}_ CORE	P12_0/ AI0	P12_2/ AI2	P21_0/ FRDn	P21_2/ FCLE	P20_0/ FD0	P20_4/ FD4	P20_6/ FD6	GND	D+	VBUS	NC	**A**
13	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	

A symbol with "n" suffixed indicates an active Low pin.

[Note] Don't connect NC pins with others.

Figure 1-3 ML66525 Family Pin Configuration (LFBGA)

* For the external dimensions of the package, refer to Chapter 23, "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 ML66525 family.

In the I/O column, "I" indicates an input pin, "O" indicates an output pin, and "I/O" indicates an I/O pin. A symbol with "n" suffixed indicates an active Low pin.

Table 1-2 Pin Descriptions (1/3)

Classification	Symbol	Function			
		Type	Primary function	Type	Secondary function
Port	P0_0/D0 to P0_7/D7	I/O	8-bit I/O port Pull-up resistors can be specified for each bit.	I/O	External memory access data I/O port
	P1_0/A8 to P1_7/A15	I/O	8-bit I/O port Pull-up resistors can be specified for each 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 bit.	O	External memory access address output port
	P3_1/PSENn	I/O	1-bit I/O port Pull-up resistors can be specified for each bit.	O	External program memory access read strobe output pin
	P3_2/RDn	I/O*	1-bit output port	O	External program memory access read strobe output pin
	P3_3/WRn	I/O	1-bit I/O port Pull-up resistors can be specified for each bit.	O	External program 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 bit.	O	External memory access address output port
	P6_0/EXINT0	I/O	8-bit I/O port Pull-up resistors can be specified for each 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
	P7_6/PWM0OUT	I/O	2-bit I/O port Pull-up resistors can be specified for each bit.	O	PWM0 output pin
	P7_7/PWM1OUT			O	PWM1 output pin
	P8_0/RXD1	I/O	4-bit I/O port Pull-up resistors can be specified for each bit.	I	SIO1 receive data input pin
	P8_1/TXD1			O	SIO1 transmit data output pin
	P8_2/RXC1			I/O	SIO1 receive clock I/O pin
	P8_3/TXC1			I/O	SIO1 transmit clock I/O pin

* P3_2/RDn is an output-only port when used in the emulator.

Table 1-2 Pin Descriptions (2/3)

Classification	Symbol	Function			
		Type	Primary function	Type	Secondary function
Port	P9_0/VBUSIN	I/O	1-bit I/O port	I	Vbus detect input/non-detect external interrupt pin. 5 V tolerant input
	P10_0/SIOCK3	I/O	6-bit I/O port Pull-up resistors can be specified for each bit.	I/O	SIO3 transmit-receive clock I/O pin
	P10_1/SIOI3			I	SIO3 receive data input pin
	P10_2/SIOO3			O	SIO3 receive data input pin
	P10_3/SIOCK4			I/O	SIO4 (with internal 32-byte FIFO) transmit-receive clock I/O pin
	P10_4/SIOO4			O	SIO4 (with internal 32-byte FIFO) transmit data output pin
	P10_5/SIOI4			I	SIO4 (with internal 32-byte FIFO) receive data output pin
	P12_0/AI0 to P12_3/AI3	I	4-bit input port	I	A/D converter analog input port
	P13_0/EXINT8	I	2-bit input port	I	External interrupt 8 input pin
	P13_1/EXINT9			I	External interrupt 9 input pin
	P15_0/RXD6	I/O	4-bit I/O port Pull-up resistors can be specified for each bit.	I	SIO6 receive data input pin
	P15_1/TXD6			O	SIO6 transmit data output pin
	P15_2/RXC6			I/O	SIO6 receive clock I/O pin
	P15_3/TXC6			I/O	SIO6 transmit clock I/O pin
	P20_0/FD0 to P20_7/FD7	I/O	8-bit I/O port Pull-up resistors can be specified for each bit.	I/O	NAND Flash Memory access data I/O port
	P21_0/FRDn	I/O	5-bit I/O port Pull-up resistors can be specified for each bit.	O	NAND Flash Memory access read strobe output pin
	P21_1/FWRn	I/O		O	NAND Flash Memory access write strobe output pin
	P21_2/FCLE	I/O		O	NAND Flash Memory access CLE strobe output pin
	P21_3/FALE	I/O		O	NAND Flash Memory access ALE strobe output pin
	P21_4/FRB	I/O		I	NAND Flash Memory access Ready/Busy input pin

Table 1-2 Pin Descriptions (3/3)

Classification	Symbol	Type	Function
Power supply	V_{DD}_IO	I	IO Power supply pin Connect all the V_{DD}_IO pins.*
	V_{DD}_CORE	I	Core Power supply pin Connect all the V_{DD}_CORE pins.*
	VBUS	I	USB power supply pin (Vbus input pin)
	GND	I	GND pin Connect all the GND pins to GND.*
	V_{REF}	I	Analog reference voltage pin (Connect to the V_{DD} pin when A/D converter is not used.)
	AGND	I	Analog GND pin (Connect to the GND pin when A/D converter is not used.)
Oscillation	XT0	I	Sub-clock oscillation output pin Connect to a crystal of f = 32.768 kHz.
	XT1n	O	Sub-clock oscillation output pin Connect to a crystal 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.
	OSC1n	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.
USB I/F	PUCTL	O	External control output pin
	D+	I/O	D+ pin
	D–	I/O	D– pin
Reset	RESn	I	Reset input pin
Others	NMI	I	Non-maskable interrupt input pin
	TEST	I	Test pin Connect to the GND pin for normal operation.
	V_{TM}	I	Test pin Connect to the GND pin for normal operation.
	FLAMOD	I	Flash ROM programming mode input pin When the FLAMOD pin is set to "L", the device enters a programming mode. Connect to the V_{DD} pin when using as normal operation.
	EAn	I	External program memory access input pin When the 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.

* Be sure to supply power supply voltage to all the V_{DD}_IO and V_{DD}_CORE pins and ground voltage to all the GND pins. If any of those pins has no voltage supply, normal operation of the device is not guaranteed.

1.5.2 Pin Configuration

A simplified pin configuration for each pin of the ML66525 family is shown in Table 1-3 and Figure 1-4.

Table 1-3 Configuration of Each Pin

Pin name	Type	Pin name	Type
P0_0 to P0_7	5	P9_0	6
P1_0 to P1_7	5	P10_0 to P10_5	5
P2_0 to P2_3	5	P12_0 to P12_3	3
P3_1	4	P13_0, P13_1	1
P3_2, P3_3	5	P15_0 to P15_3	5
P4_0 to P4_7	5	P20.0 to P20_7	5
P6_0 to P6_7	5	P21.0 to P21_4	5
P7_6, P7_7	5	RESn	2
P8_0 to P8_3	5	NMI, EAn	1

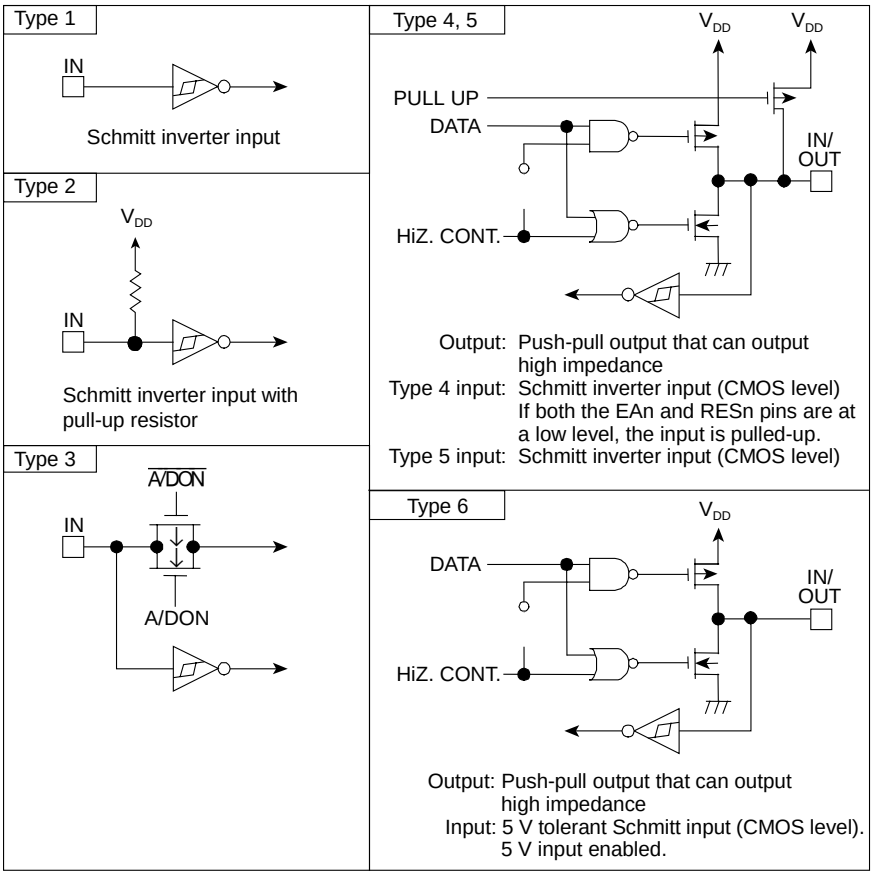


Figure 1-4 Types of Pin Configurations

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 P0_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 P1_7	
P2_0 to P2_3	
P3_1 to P3_3	
P4_0 to P4_7	
P6_0 to P6_3	
P7_6, P7_7	
P8_0 to P8_3	
P9_0*1	
P10_0 to P10_7	
P15_0 to P15_3	
P20_0 to P20_7	
P21_0 to P21_4	
P13_0, P13_1	High or Low level
P12_0 to P12_3	V_{REF} or AGND
V_{REF}	V_{DD}_CORE
AGND	GND
NMI	High or Low level
EAn	High level
XT0	GND*2
OSC1n, XT1n	Open

*1 Since P9_0 does not have the programmable pull-up function, its pin handling should be:
When input is set: High or Low level
When output is set: Open

*2 If the subclock (XT0, XT1n) is not used, in addition to connecting the XT0 pin to GND and leaving XT1n unconnected, the peripheral control register (PRPHCON) must be set. For details refer to Chapter 6, "Clock Oscillation Circuit."

1.6 Basic Operational Timing

The ML66525 family is configured such that one pulse of the main clock (CLK) is one state. In other words, one state is 41.7 ns (at 24 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 1-5 through Figure 1-8 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 16, "Bus Port Functions."

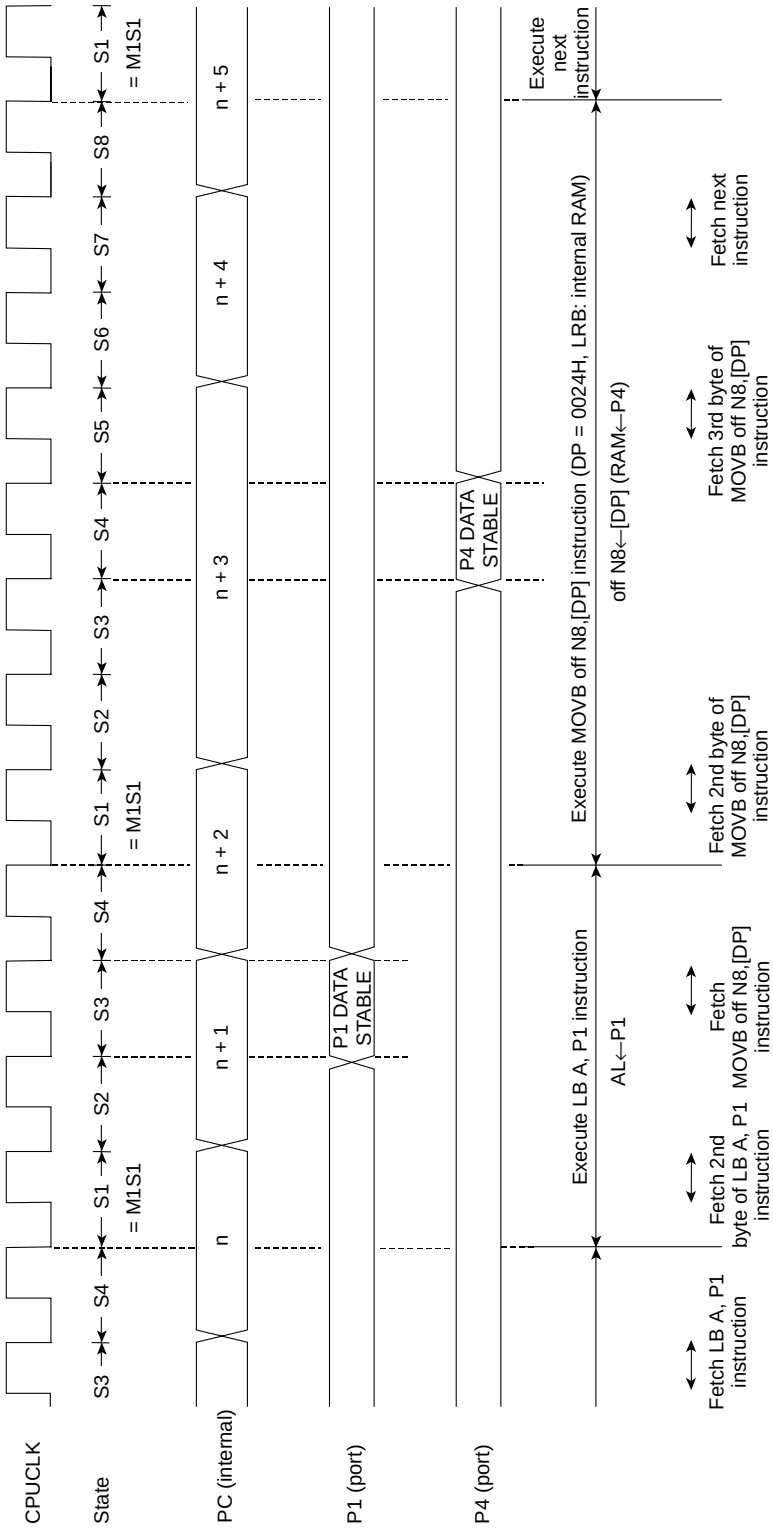


Figure 1-5 Basic Operation Timing Example (Reading Port Data)

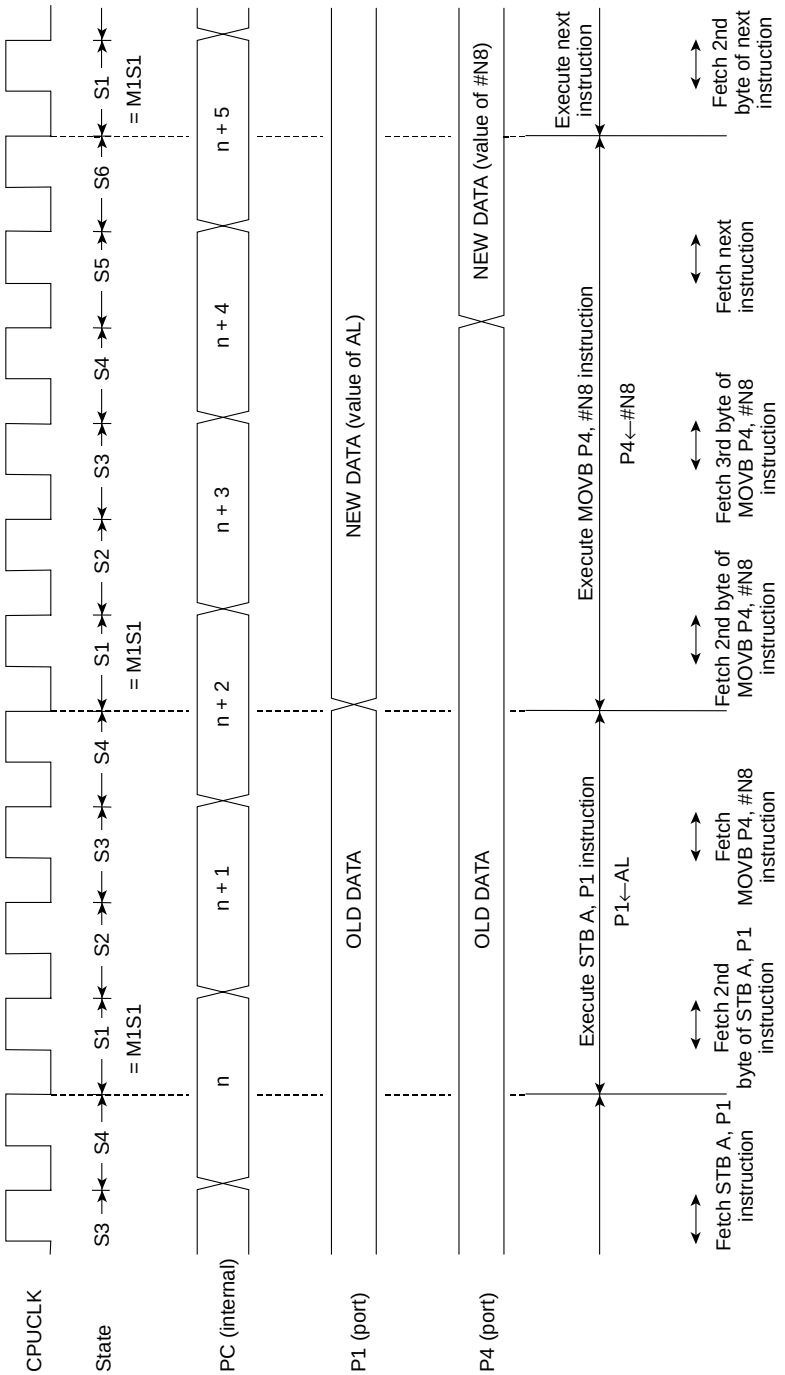
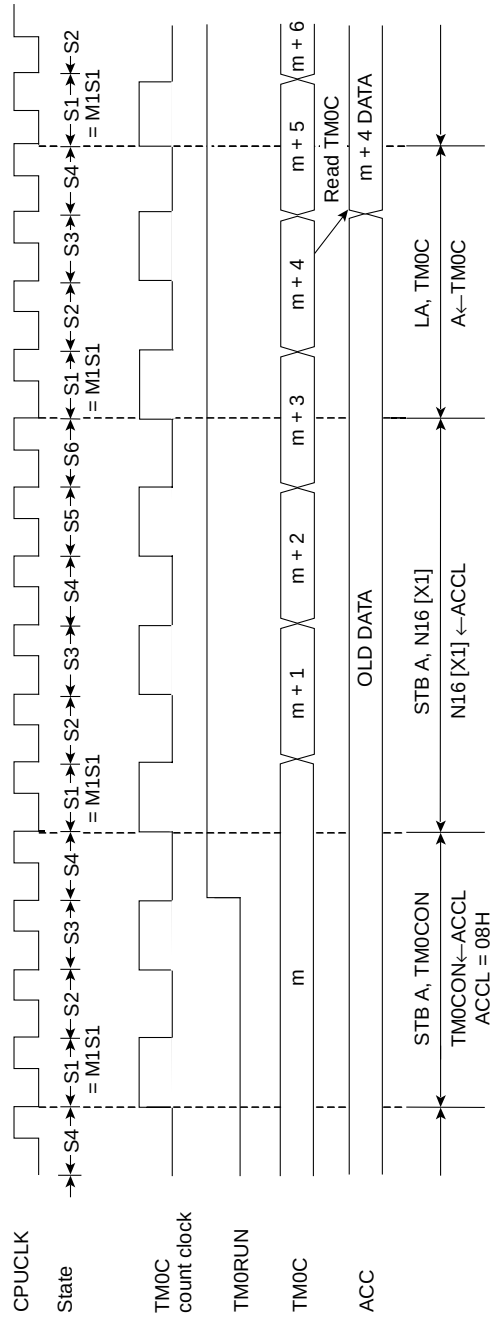


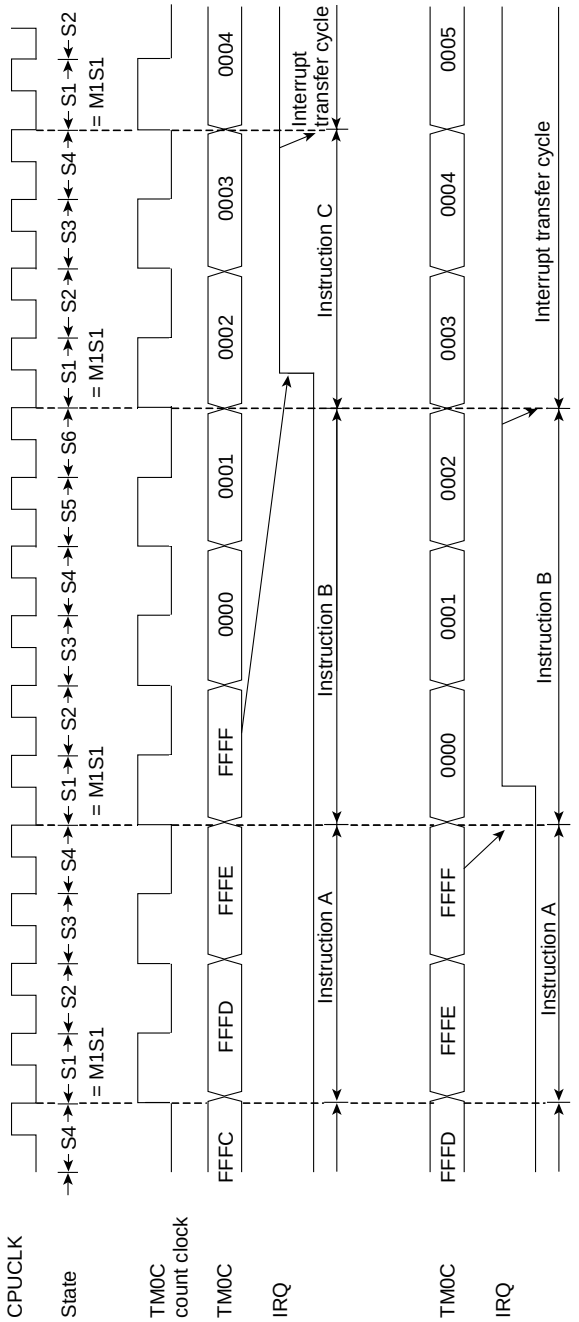
Figure 1-6 Basic Operation Timing Example (Writing Port Data)



[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 1-7 Timer 0 Operation Timing Example



[Note]

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

Figure 1-8 Interrupt Transfer Timing Example

Chapter 2

CPU Architecture

2. CPU Architecture

2.1 Overview

The ML66525 microcontroller family utilizes 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 1 MB. 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 1 MB.

2.2.1 Memory 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 1 MB by setting the LROM bit (bit 1) to "1". The data memory space can be expanded to 1 MB by setting the LRAM bit (bit 0) to "1".

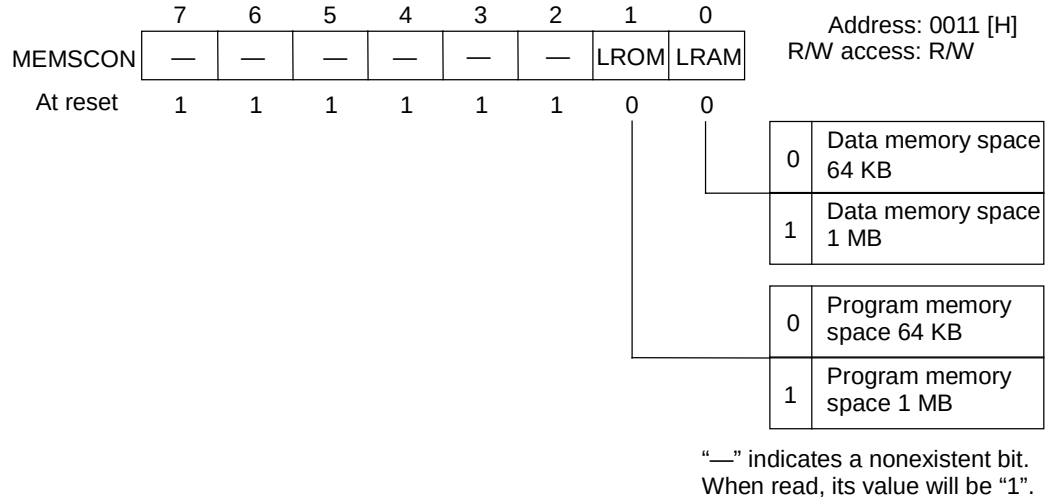


Figure 2-1 MEMSCON Configuration

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).

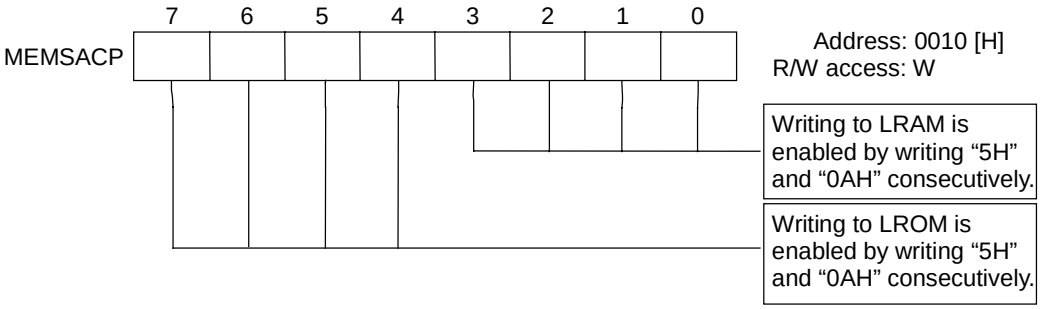


Figure 2-2 MEMSACP Configuration

Note: If the FJ, FCAL, or FRT instruction is executed while the LROM bit is being reset to "0", the opcode 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 (64 KB program memory space, 64 KB data memory space)
  MOVB MEMSACP, #05H
  MOVB MEMSACP, #0AH
  MOVB MEMSCON, #00H (initial value)

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

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

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

MEMSCON can be written only once after reset (due to a RESn 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". A maximum of 1 MB (1,048,576 bytes) of program memory can be accessed in 64 KB (65,536 bytes) unit segments from segment 0 to 15. However, if more than 64 KB (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 128 KB (131,072 bytes) area in segments 0 and 1 constitutes the internal ROM area and the 64 KB areas in segments 2 to 15 from 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.

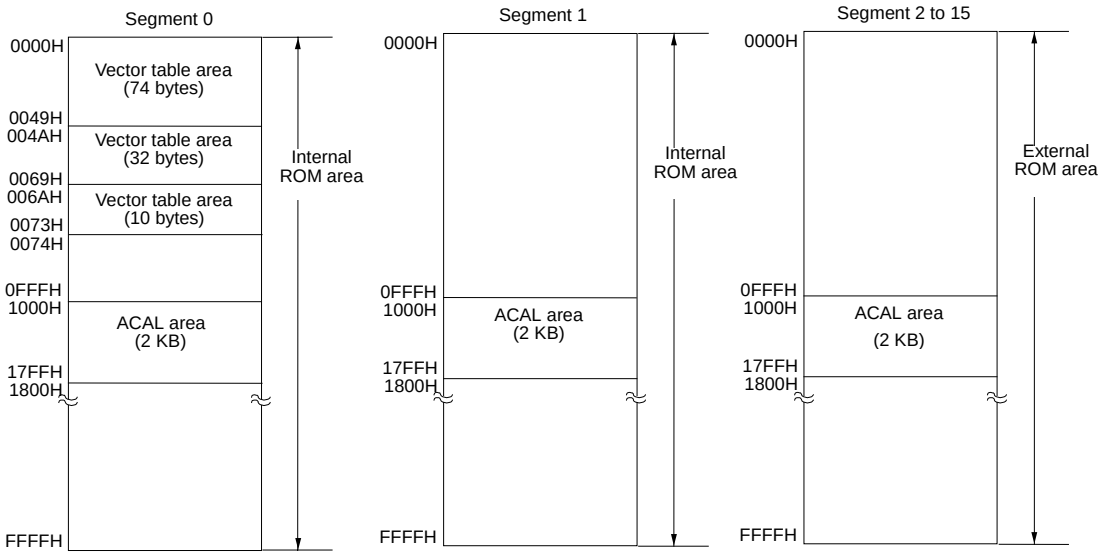


Figure 2-3 Memory Map of Program Memory Space

(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 EAn pin.

When a high level is input to the EAn 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 and FFFFFH. When the external ROM area is accessed, the secondary functions of the external memory control pins (ports 0, 1, 2, 3 and 4) must be set.

The area from 0000H to 1FFFDH can be fetched by the internal program. Therefore, 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 EAn pin, the external program memory area is accessed for all program addresses.

If the external memory area of the program memory space is accessed, 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/$\overline{\text{PSEN}}$ pin becomes 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 RESn 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 RESn pin input
0002	Reset by execution of BRK instruction
0004	Reset by overflow of watchdog timer
0006	Reset by opcode trap
0008	NMI pin input (non-maskable interrupt)
000A	EXINT0 pin input (external interrupt 0)
001A	Timer 0 overflow
001C	EXTINT1 pin input (external interrupt 1)
001E	EXTINT2 pin input (external interrupt 2)
0020	EXTINT3 pin input (external interrupt 3)
0026	Timer 3 overflow
002A	EXTINT4 pin input (Vbus detection interrupt)
002C	Interrupt from USB controller
002E	Interrupt from DMA controller
0030	Interrupt from Media controller
0032	Timer 7 overflow
0036	Timer 4 overflow
0038	SIO1 transmit buffer empty, transmit completion, receive completion
003A	Timer 5 overflow
003E	SIO3 transmit/receive completion SIO1 transmit buffer empty, transmit completion, receive completion
0040	SIO4 transfer completion
0042	Timer 6 overflow
0044	AD conversion, select mode cancelled
0046	EXINT8/EXINT9 pin inputs (external interrupts 8, 9)
0048	Real time counter output
006A	PWC0 overflow, PWC0 and PWR0 match
006C	PWC1 overflow, PWC1 and PWR1 match
0072	Timer 9 overflow

(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 1 MB, 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 64 KB (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 1 MB (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 2 KB 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 (1000 to 17FFH).

2.2.3 Data Memory Space

A maximum of 1 MB (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), EXPANDED RAM area (3,584 bytes) and an external memory area (1,040,384 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 (2000H to 0FFFFH of segment 0, 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.

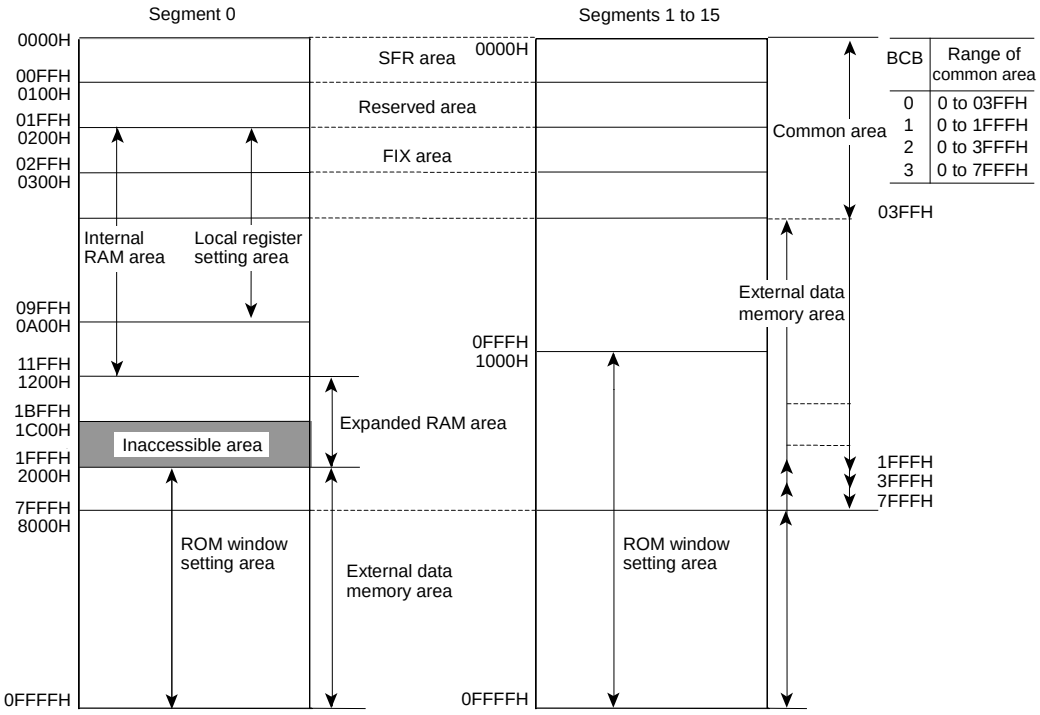


Figure 2-4 Memory Map of Data Memory Space

(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 22, "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. The reserved area is not available.

(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.

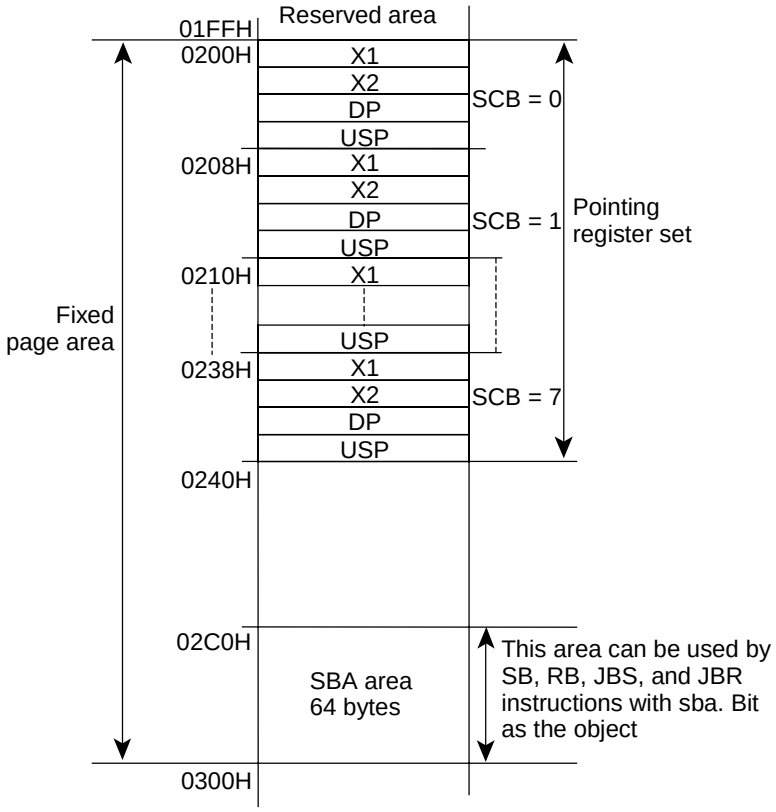


Figure 2-5 Map of Fixed Page Area

(5) Local register setting area

The local register setting area is the 2 KB 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.

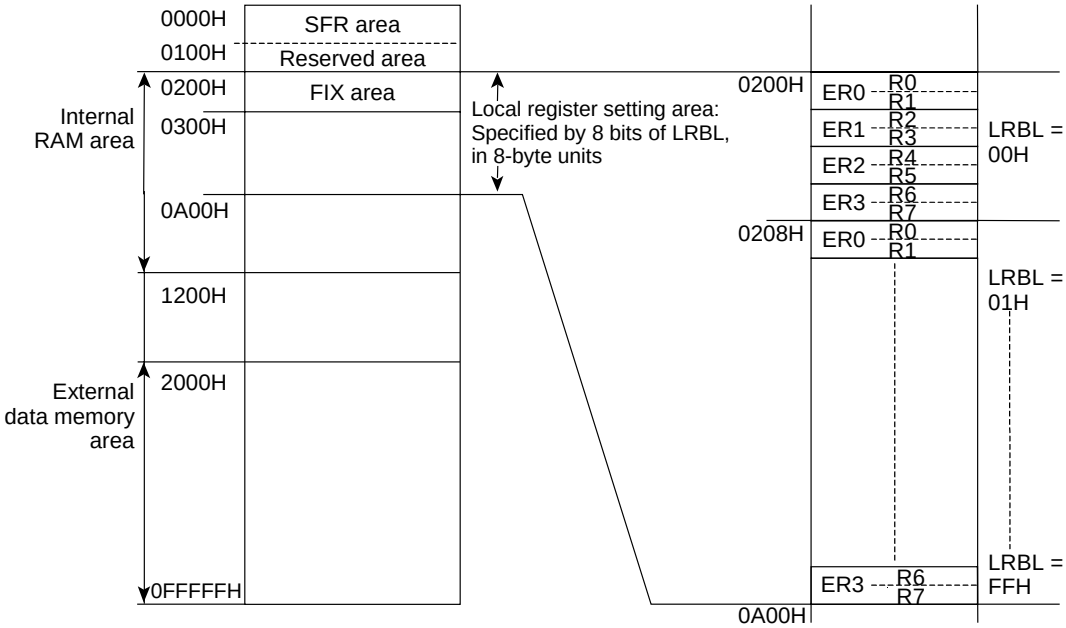


Figure 2-6 Map of Local Register Setting Area

(6) External data memory area

The external memory area is the 1,040,384-byte area of data memory from 2000H to 0FFFFFH. If this external memory is to be accessed, the secondary functions of memory related pins (ports 0, 1, 2, 3 and 4) must be set. The external 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), and the P3_3/WRn (write strobe output function) and P3_2/RDn (read strobe output function) signals.

The 1,040,384-byte area from 2000H to 0FFFFFH of data memory is the external 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 2000H 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.

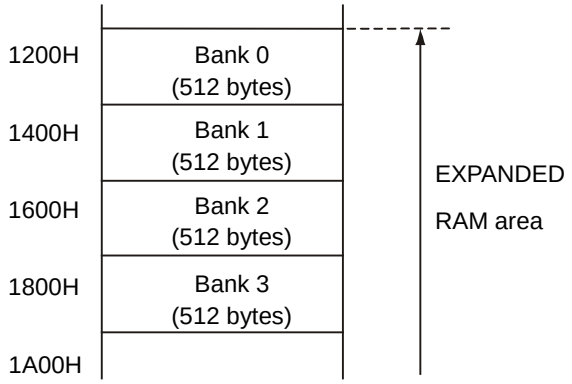
(7) Common area

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.

BCB		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

(8) EXPANDED RAM area

The EXPANDED RAM area is assigned to the 3,584-byte area of data memory from 1200H to 1FFFH. The EXPANDED RAM includes the DMA transfer buffer RAM (512 bytes × 4 banks), USB controller register, DMA transfer register, NAND Flash Memory transfer control register, and NAND Flash Memory I/F I/O ports. The memory from 1C00H to 1FFFH is an inaccessible area.



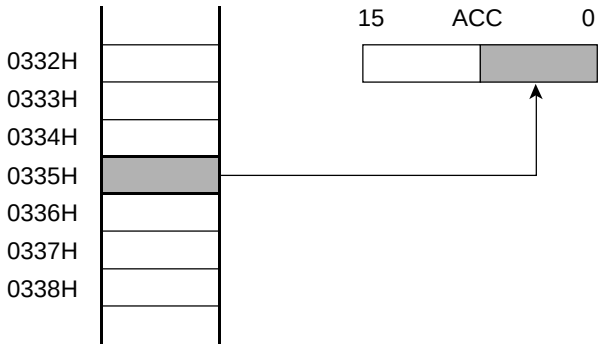
2.2.4 Data Memory Access

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



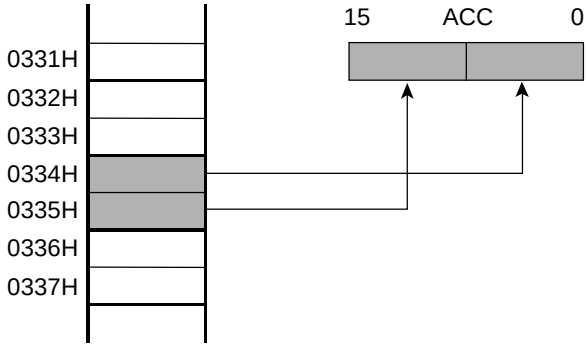
(2) Word operations

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.

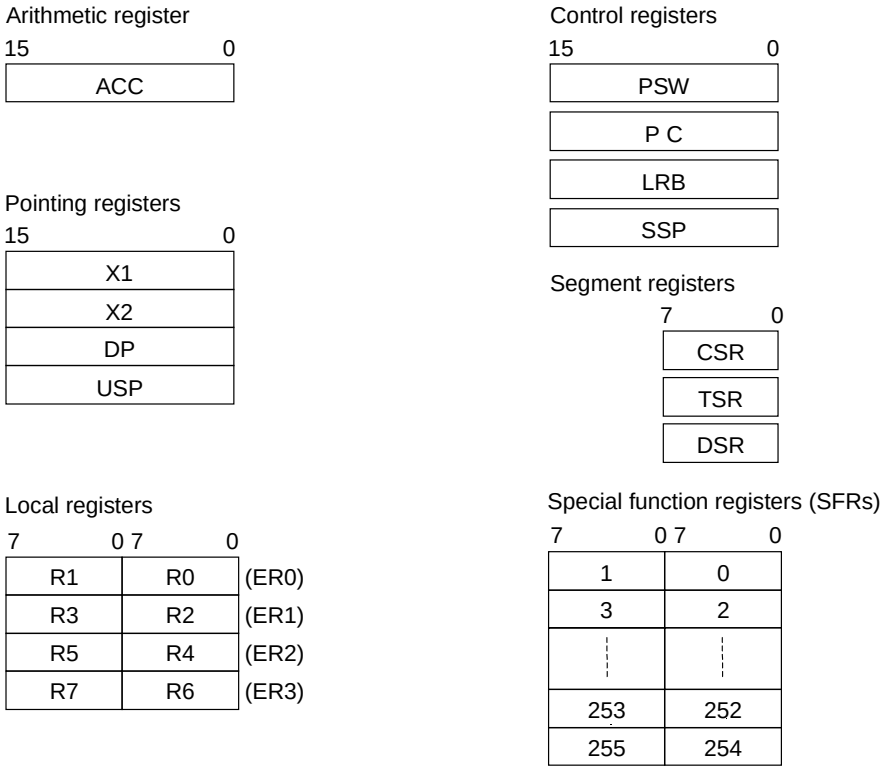


Figure 2-7 Register Configurations

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 RESn 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.

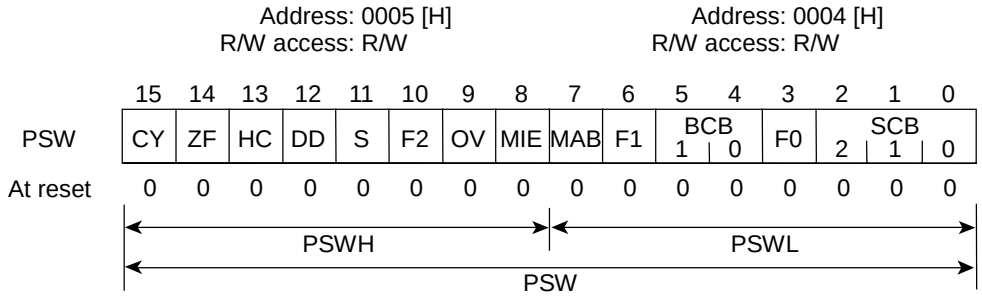


Figure 2-8 PSW Configuration

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 RESn 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 ML66525 family does not have the product-sam function. 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.

BCB		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.

SCB			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 RESn 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 2 KB data memory space from 0200H to 09FFH in 8-byte units (local register addressing). The upper 8 bits (LRBH) specify the 64 KB 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 RESn input, BRK instruction execution, watchdog timer overflow, or opcode trap), their value is undefined.

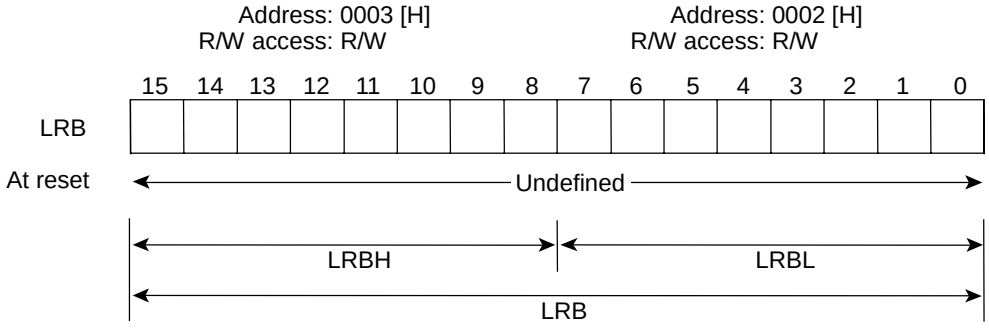
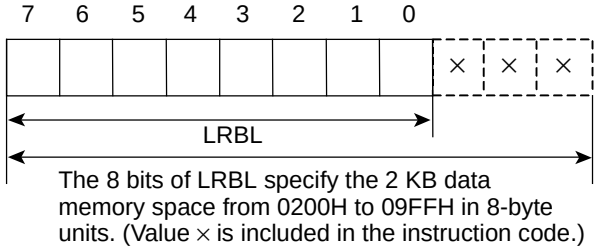
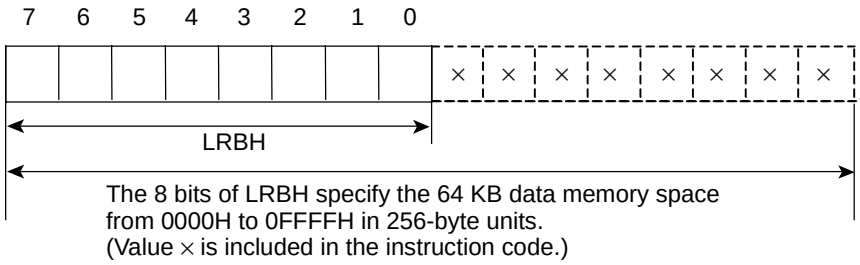


Figure 2-9 LRB Configuration

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



- The 8 bits of LRBH specify 64 KB 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 RESn input, BRK instruction execution, watchdog timer overflow, or opcode trap), the contents of SSP become 0FFFFH.

2.3.3 Pointing Register (PR)

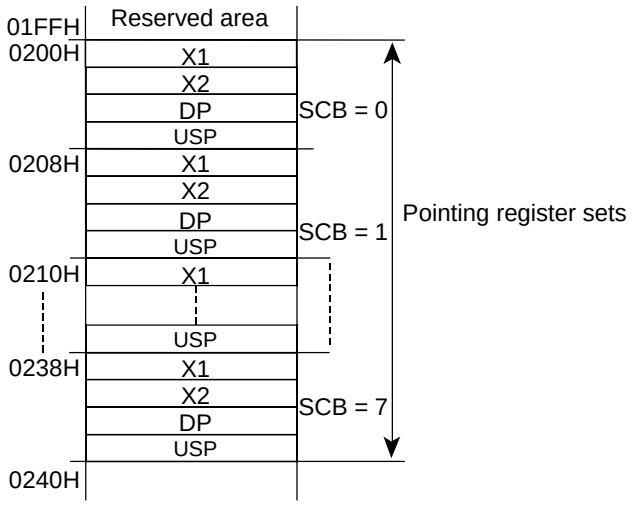
The PR has 8 sets of registers. One set consists of the following four 16-bit registers.

- 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.

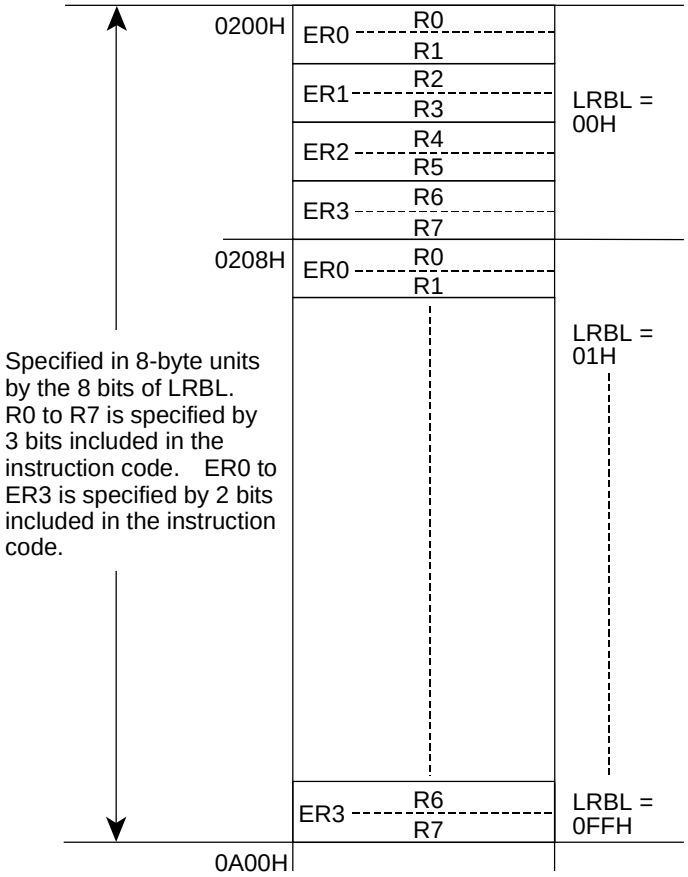
If the PR function is not used, this area can be used as normal internal RAM.

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 2 KB 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. 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 is 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 1 MB, 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"					Address: 0009 [H] R/W access: R/W
At reset	0	0	0	0	0	0	0	0	

"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 ML66525 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 data 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

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

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

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

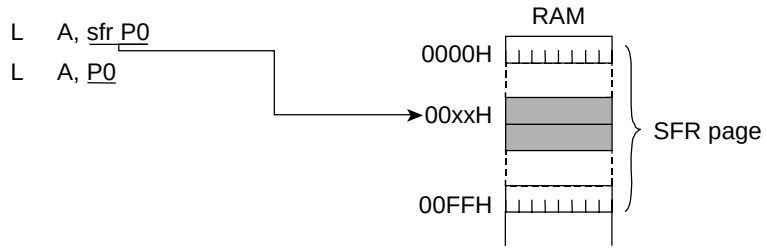
A. SFR page addressing

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.

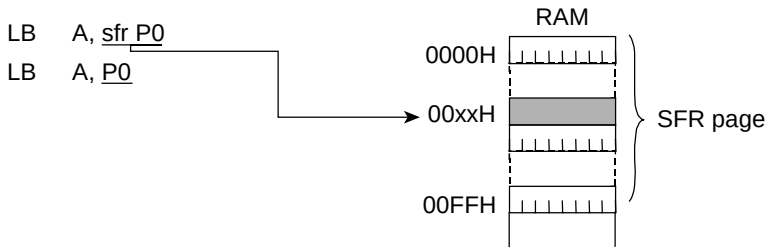
The SFR has address symbols for each type of device. These symbols are normally used for addressing the SFR.

[Word format]

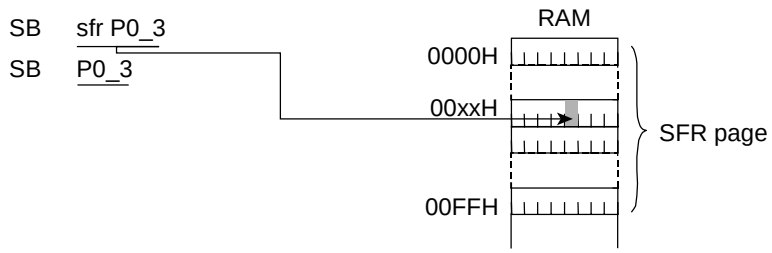


If an odd address is specified, word-format data is accessed starting at the following even address. (→word boundary) However, depending upon the SFR, there are some exceptions.

[Byte format]



[Bit format]

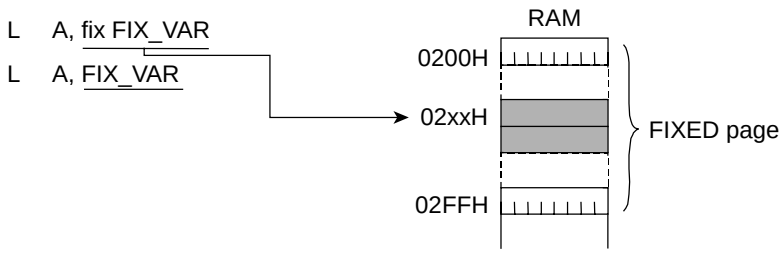


B. FIXED page addressing

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.

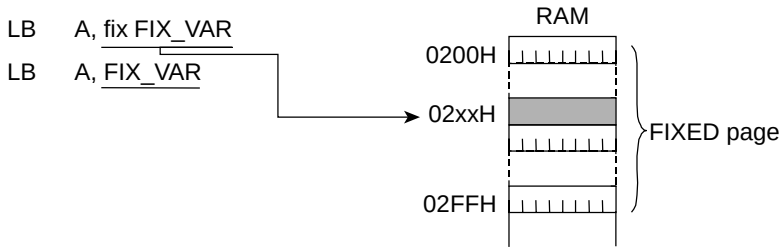
The operand is described using a format that has a fix addressing descriptor. The fix descriptor can be omitted, however in that case, the assembler will use FIXED page addressing only when it recognizes an address within the FIXED page area.

[Word format]

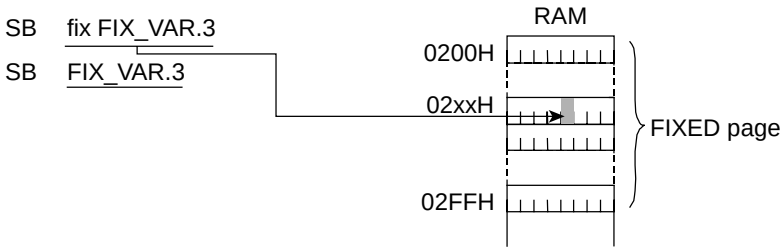


If an odd address is specified, word-format data is accessed starting at the following even address.

[Byte format]



[Bit format]

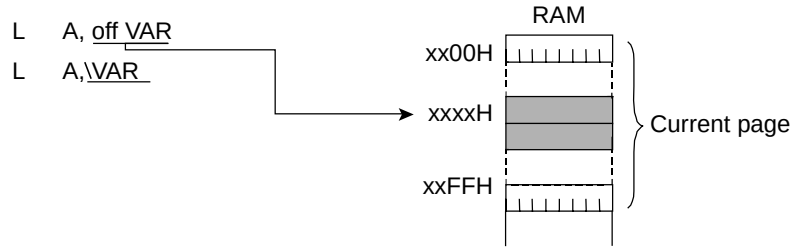


C. Current page addressing

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.

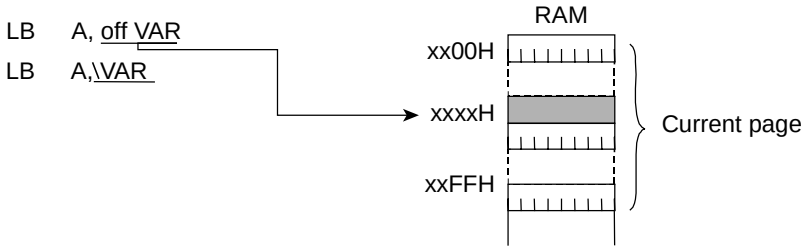
The operand is described using a format that has an off addressing descriptor. \ can be used instead of the off descriptor, however if bit-format data is accessed in the SBA area, operation will be slightly different. (sbaoff Badr)

[Word format]

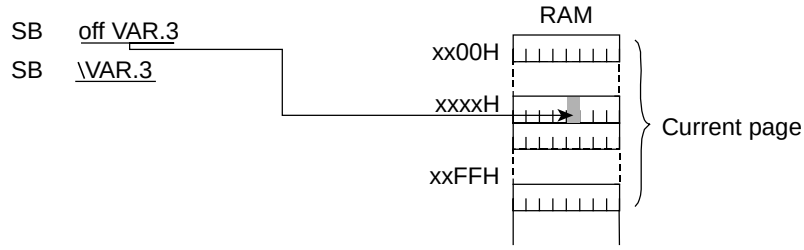


If an odd address is specified, word-format data is accessed starting at the following even address.

[Byte format]



[Bit format]

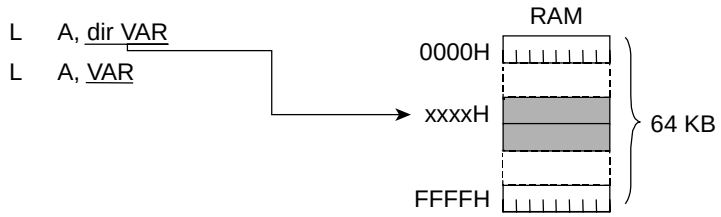


(3) Direct data addressing

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

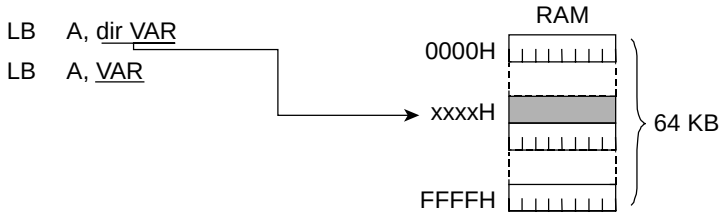
The operand is described using a format that has a dir addressing descriptor. The dir descriptor can be omitted, however in this case, if an address in a SFR page or FIXED page is specified, the assembler may interpret direct data addressing as SFR page addressing or FIXED page addressing.

[Word format]

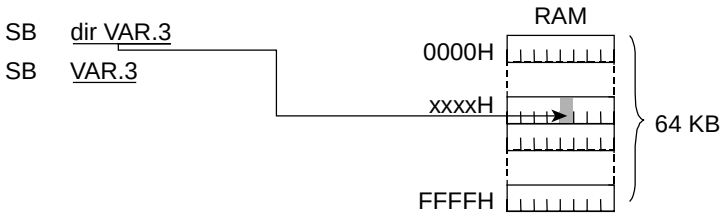


If an odd address is specified, word-format data is accessed starting at the following even address.

[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]

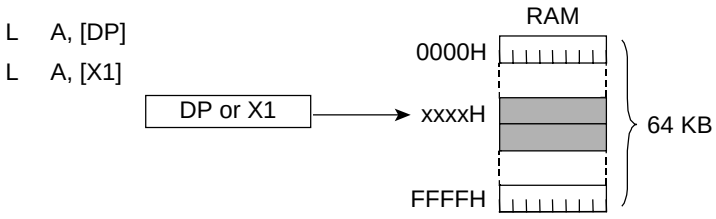
A. DP/X1 indirect addressing

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

[DP]: DP indirect addressing

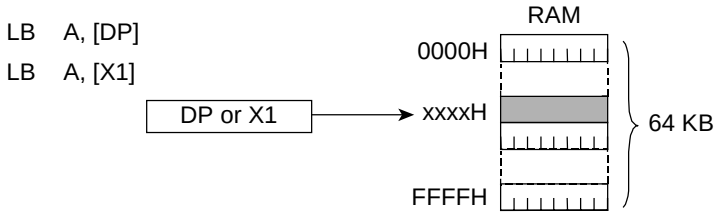
[X1]: X1 indirect addressing

[Word format]

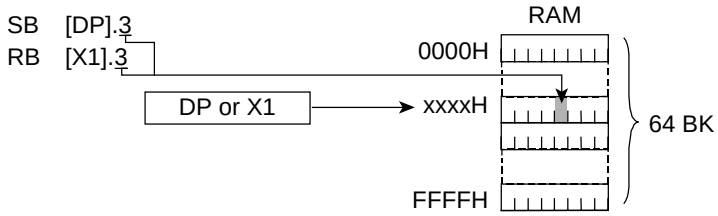


If an odd address is specified, word-format data is accessed starting at the following even address.

[Byte format]



[Bit format]



B. DP indirect addressing with post increment

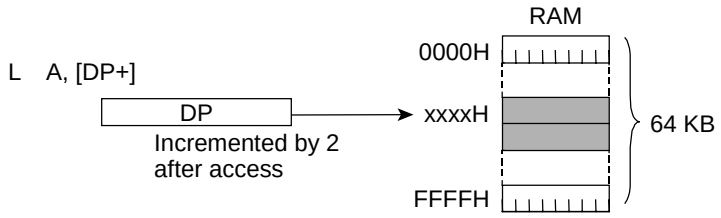
The contents of the pointing register specify an address in the current physical segment of data memory (address 0 to 0FFFFH: 64 KB). Word-format, byte-format or bit-format data at the specified address is accessed.

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.

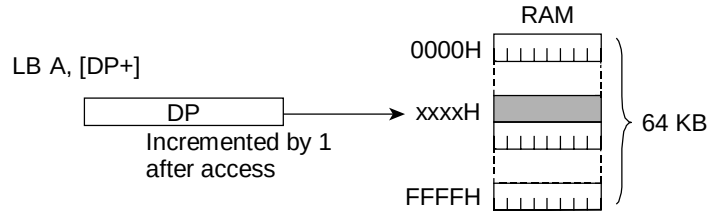
[DP+]: DP indirect addressing with post increment

[Word format]

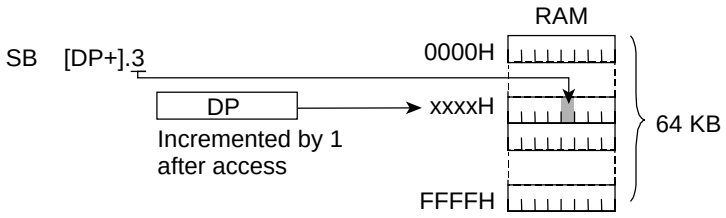


If an odd address is specified, word-format data is accessed starting at the following even address.

[Byte format]



[Bit format]



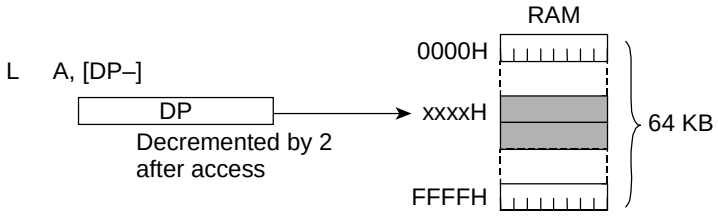
C. DP indirect addressing with post decrement

The contents of the pointing register specify an address in the current physical segment of data memory (addresses 0 to 0FFFFH: 64 KB). 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.

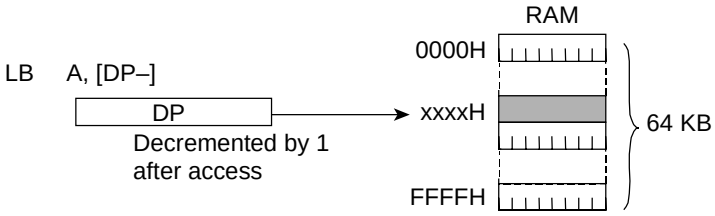
[DP–]: DP indirect addressing with post decrement

[Word format]

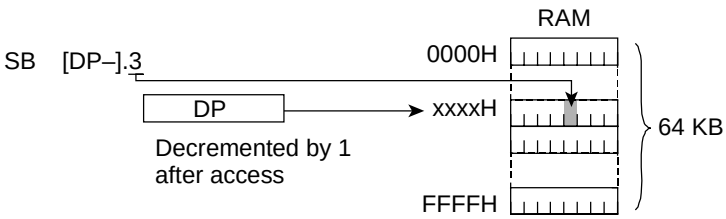


If an odd address is specified, word-format data is accessed starting at the following even address.

[Byte format]



[Bit format]

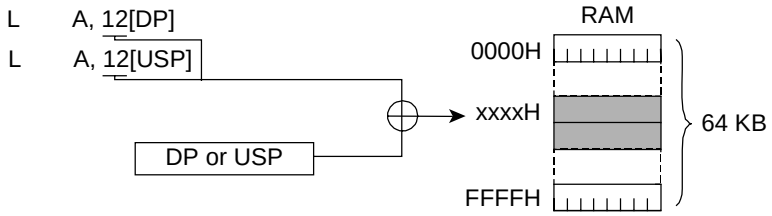


D. DP/USP indirect addressing with 7-bit displacement

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: 64 KB). The accessible range is –64 to +63 from the content of the pointing register. Word-format, byte-format or bit-format data at the specified address is accessed.

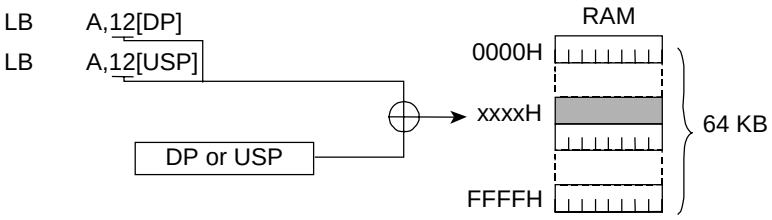
Numerical expression[DP]: DP indirect addressing with 7-bit displacement
Numerical expression[USP]: USP indirect addressing with 7-bit displacement
The numerical expression has a value in the range of –64 to +63.
DP and USP can be used as pointing registers.

[Word format]

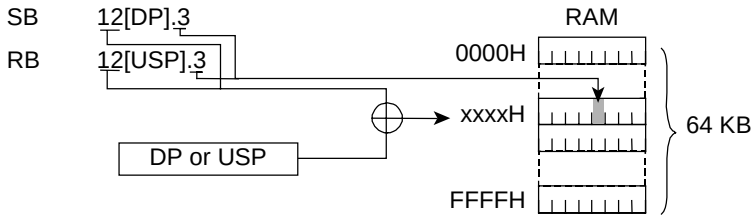


If an odd address is specified, word-format data is accessed starting at the following even address.

[Byte format]



[Bit format]

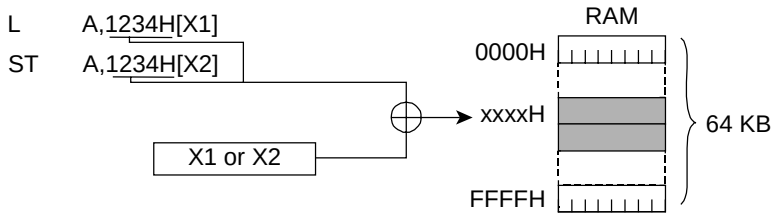


E. X1/X2 indirect addressing with 16-bit base

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: 64 KB). 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.

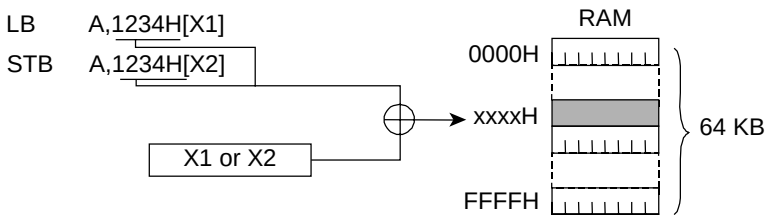
Address expression[X1]: X1 indirect addressing with 16-bit base
Address expression[X2]: X2 indirect addressing with 16-bit base
The address expression has a value in the range of 0 to 0FFFFH. However, the assembler allows values in the range of −8000H to +0FFFFH. This means that D16 can also be regarded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

[Word format]

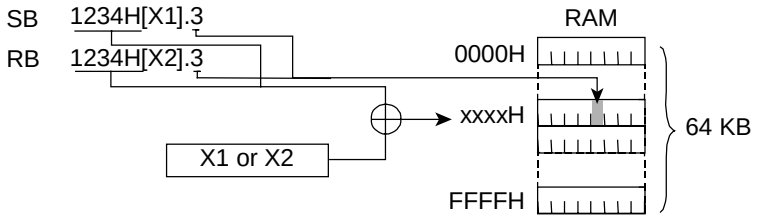


If an odd address is specified, word-format data is accessed starting at the following even address.

[Byte format]



[Bit format]



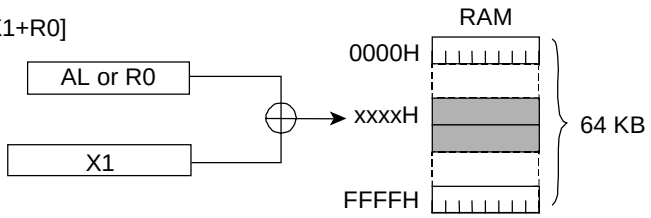
F. X1 indirect addressing with 8-bit register displacement

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: 64 KB). 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]

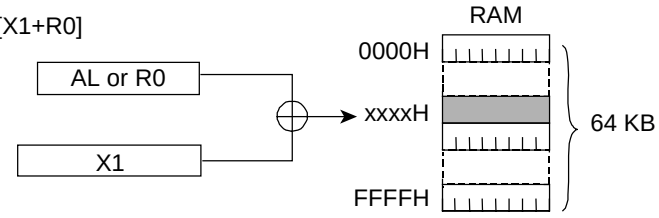
```
L     A, [X1+A]
L     A, [X1+R0]
```



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

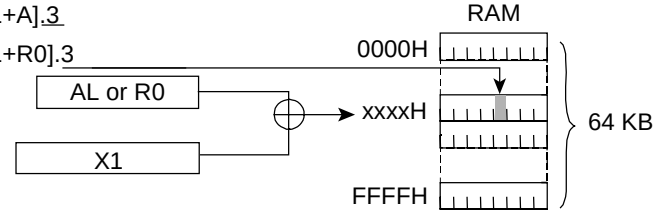
[Byte format]

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



[Bit format]

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



(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

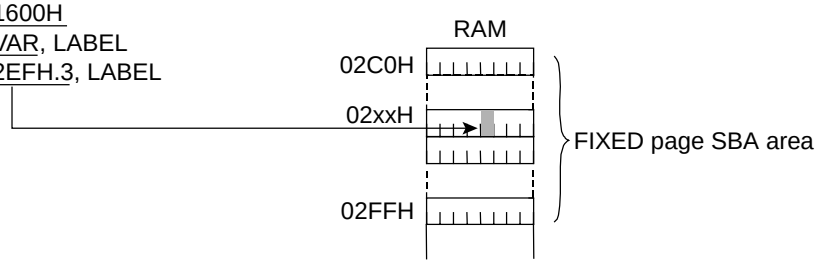
This addressing mode specifies a bit address in the 512-bit SBA area (2C0H.0 to 2FFH.7) located in a FIXED page. Bit format data at the specified address is accessed.

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

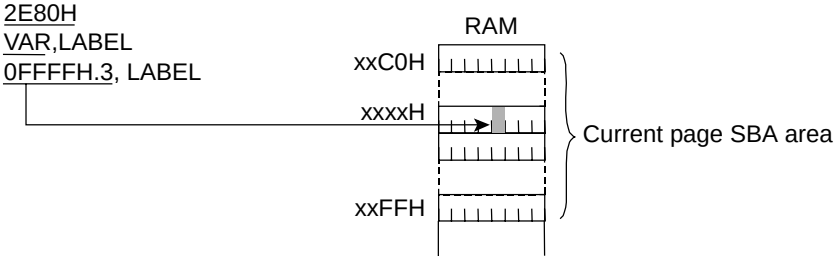
This addressing mode specifies a bit address in the 512-bit SBA area (xxC0H.0 to xxFFH.7) located in the current page. Bit format data at the specified address is accessed.

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 64 KB 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: 64 KB) 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: 64 KB). 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: 64 KB) 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: 1 MB) 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: 64 KB). 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: 2 KB) 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 ML66525 family has two CPU control functions, a standby function and a reset function.

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

The reset function is activated by the RESn 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.

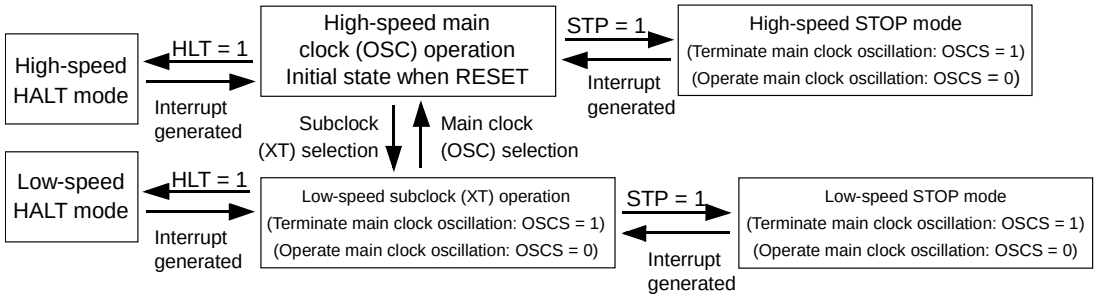
3.2 Standby Functions

The ML66525 family has two types of standby functions.

- HALT mode: activated by software, 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.



(Notes)
- Oscillation operation or termination is for the main clock (OSC) only. The subclock (XT) is not terminated.
- The initial value of OSCS (bit 3 of SBYCON) is "1".
- Stopping the main clock (OSC) with OSCS is effective when P9_0/VBUSIN is at a "L" level for input; alternatively, when P9_0/VBUSIN is at a "H" level for input and the USB control function is in the power down state (refer to Chapter 17).

Figure 3-1 Transition Diagram of CPU Operating States

Table 3-1 Standby Mode Summary

Standby mode		HALT mode	STOP mode *1	
Set conditions		Bit 1 (HLT) of SBYCON is set to "1"	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 RESn pin input WDT	Interrupt RESn pin input	Interrupt RESn pin input
Output pin states	P0 to P2, P4 (primary function)	No change	High impedance	No change
	P0 to P2, P4 (secondary function)	Pull-up	Pull-up	Pull-up
	P6 to P21	No change	High impedance *2	No change
	P3_1 (primary function)	No change	High impedance	No change
	P3_1 (secondary function)	High level	High impedance	High level
	P3_2, P3_3 (primary function)	No change	High impedance	No change
	P3_2, P3_3 (secondary function)	Pull-up	High impedance	Pull-up
Operation of internal functions	Time base counter (TBC)	Operate	Terminate	
	8/16-bit timers (including WDT)	Operate	Terminate	
	SIO1, SIO6	Operate	Terminate	
	SIO3	Operate	Operate during slave mode	
	SIO4	Operate	Terminate	
	Real-time counter	Operate	Operate	
	A/D converter	Operate	Terminate	
	PWM	Operate	Terminate	
	Media control	Operate	Terminate	
	Internal DMA control	Operate	Terminate	
	USB control	Operate	Ready to operate	

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

*2 If P10_0 is a secondary function output, then P10_2 (secondary function output) will be high impedance. If P10_0 is not a secondary function output, then P10_2 (secondary function output) will be the SIO3 data output.

3.2.1 Standby Function Registers

Table 3-2 lists 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	13-2

[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 22, "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 RESn input, BRK instruction execution, watchdog timer overflow, or opcode trap), STPACP is reset to "0".

The STPACP configuration is shown in Figure 3-2.

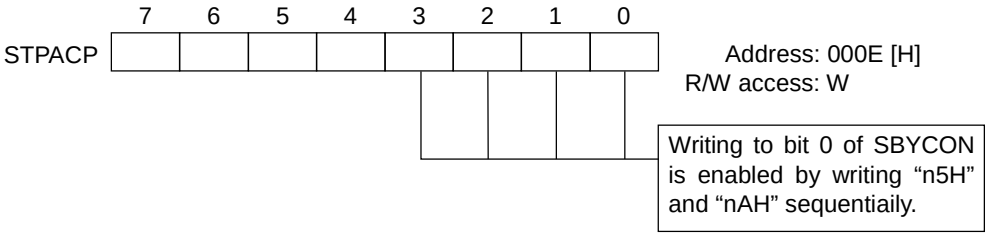


Figure 3-2 STPACP Configuration

(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 RESn input, BRK instruction execution, watchdog timer overflow, or opcode trap), SBYCON is 08H.

Figure 3-3 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 RESn 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 RESn 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 : External interrupt pins (EXINT0 to EXINT4)
  - P10_0 : SIO3 transmit-receive clock input pin
  - P10_1 : SIO3 receive data input pin

  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 or an SIO3 data reception. For details, refer to Section 3.2.4, "Operation of Each Standby Mode," (3) STOP Mode.

  For the ML66525 Family, the P9_0 pin is being configured as an output Setting FLT to "1" will cause the output ports to go to a high impedance state when the STOP mode is entered. However, at the input ports, the circuit to prevent current flow will not operate. Since the P9_0 pin is the Vbus detect input pin, it should not be used as an output.

- 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).

[Note]
To stop the main clock (OSCCLK) with OSCS, the following conditions also are required.
1. When P9_0/VBUSIN is at a "L" level for input, that is, when the USB is unconnected. Alternatively,
2. When P9_0/VBUSIN is at a "H" level for input and the USB control function is in the power down state (refer to Chapter 17).
   An attempt to terminate the main clock (OSCCLK) with OSCS will fail when the USB control function is in operation, because a clock must be supplied to the USB control function.

- 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 ms 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.

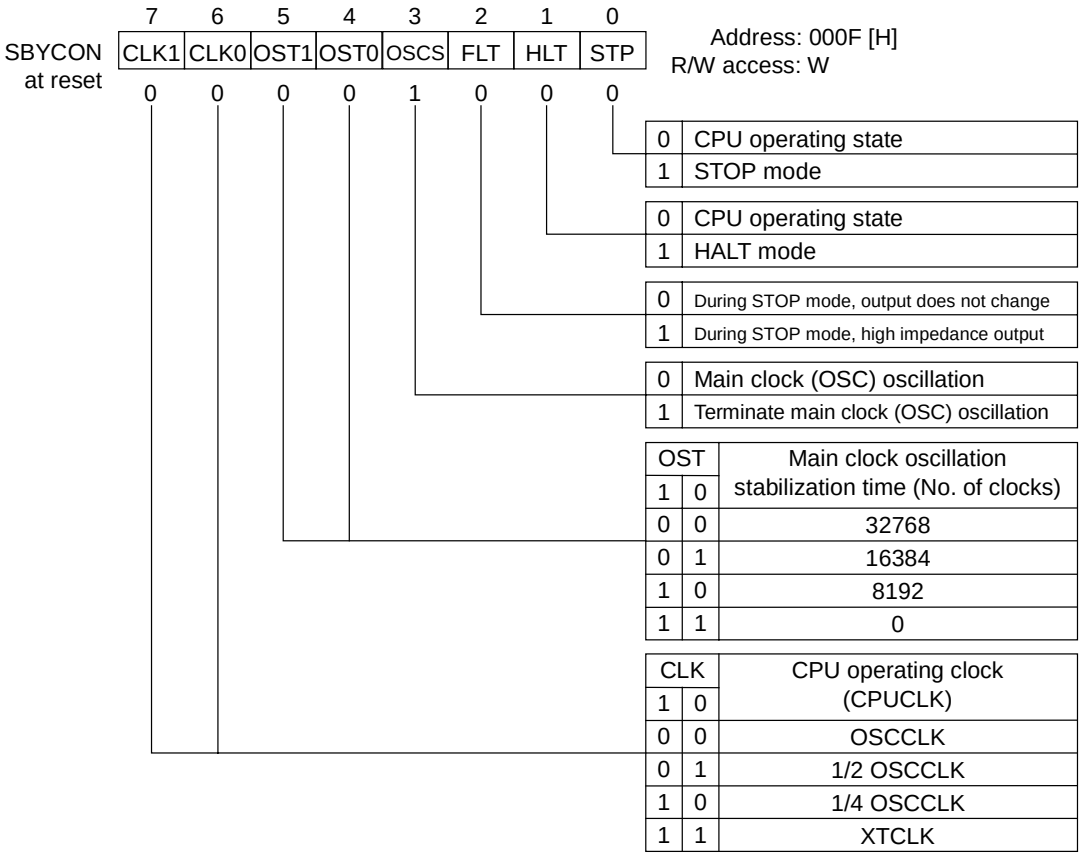


Figure 3-3 SBYCON Configuration

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.

- 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 occurs: an interrupt request, reset by the RESn 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 RESn pin input or overflow of the watchdog timer, the CPU will perform the reset processing.

(2) 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, SIO3 will operate if slave mode has been selected. Also, 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 RESn 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 EXINT4)

Interrupt caused by completion of an SIO3 transfer (during slave mode)

Interrupt caused by real-time counter output (when dual clocks are used)

Interrupt from the USB control function

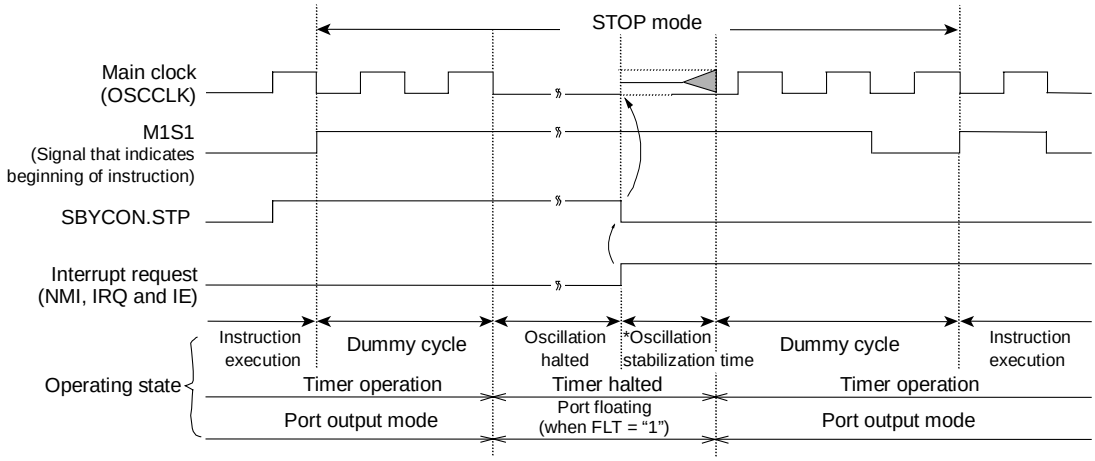
After STOP mode is released, if the master interrupt enable flag (MIE in PSW) has been set to "1", processing of the requested maskable interrupt is performed.

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 RESn pin input, the CPU will perform the reset processing. If the RESn pin input is to be used to release the STOP mode with main clock oscillation halted, apply a low level to the RESn 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 ML66525 family is reset by the following four factors.

- Low-level input to the RESn 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 RESn 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 22, "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.

Figure 3-5 shows an example of reset pin connection.

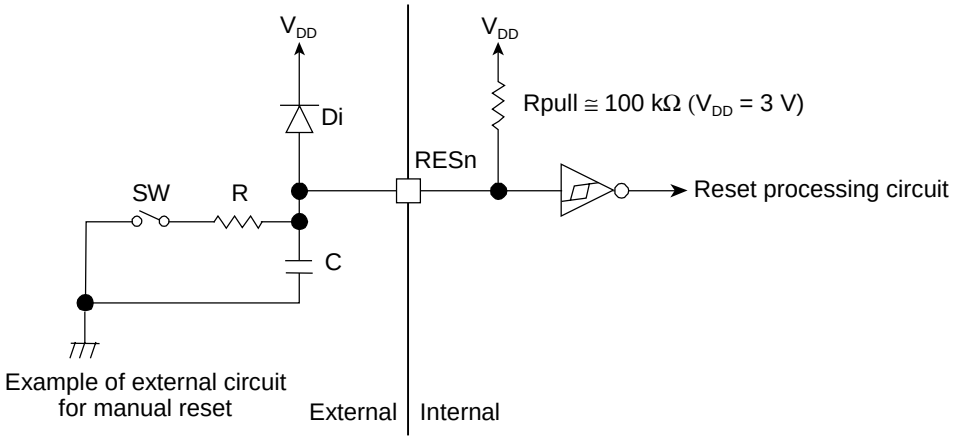


Figure 3-5 Reset Pin Connection Example

3.3.1 Reset Operation

The device requires self-resetting in the cases of 1) and 2) that follow:

1)
- When power-on reset is executed, or
- When the STOP mode is released when it is so specified that the main clock oscillation will be stopped during the STOP mode

2) When the main clock oscillation has stabilized

The figures below show the time required for a low level to be applied to the RESn pin (a high level needs to be applied to the V_{DD}_CORE and V_{DD}_IO pins) and the operation timing at reset. Figure 3-6 corresponds to case 1) above and Figure 3-7 case 2).

Table 3-4 shows the I/O port status during input of the reset signal.

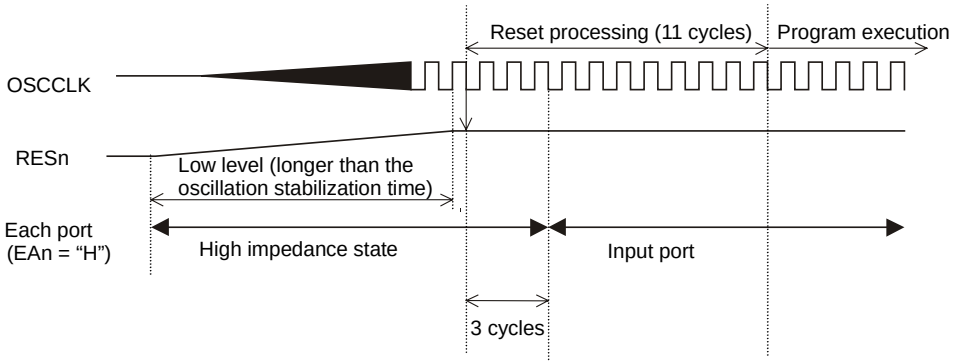


Figure 3-6 Time Required for a Low Level to be Applied to the RESn Pin (1)

[Notes]

1. For the reset operation on a Flash ROM product, the processing time of a maximum of 5 μs will be added to the internal reset processing time (11 cycles).
2. The internal state and the output state of the device at the time of power-on reset are undefined. Be sure to reset the device after power is turned on.
3. When power is applied to, or is disconnected from, the V_{DD}_CORE pins, V_{DD}_IO pins, and V_{REF} pin, apply or disconnect the power at the same time for all the pins.

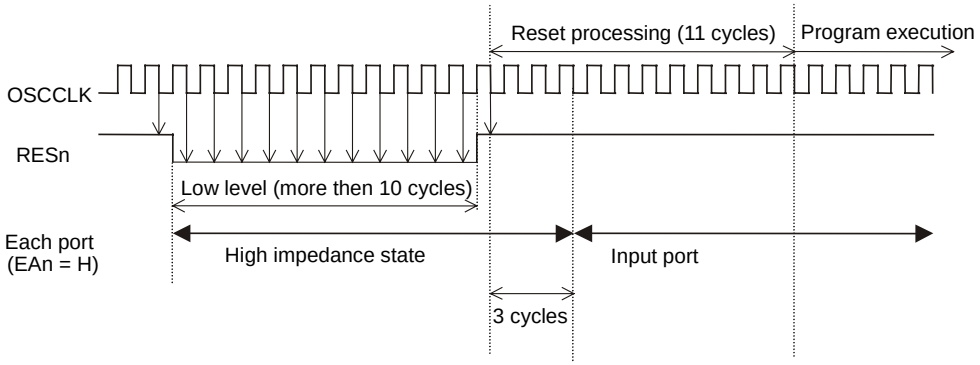


Figure 3-7 Time Required for a Low Level to be Applied to the RESn Pin (2)

[Note]

For the reset operation on a Flash ROM product, the processing time of a maximum of 5 μs will be added to the internal reset processing time (11 cycles).

Table 3-4 I/O Port Status During Reset Signal Input

Name	Low level EAn pin P3_1	High level EAn pin P3_1	Other ports
Status	Pulled-up	High impedance	High impedance

[Note]

If the EAn pin is at a low level, P0, P1, P2, P3_1 and P4 automatically change to secondary function output states (bus port function) after reset.

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.

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
000A ☆	EXPANDED RAM Ready Control Register	XPDRDY	—	R/W	8	FF	4-7
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	13-2

[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 22, "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 64 KB 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.

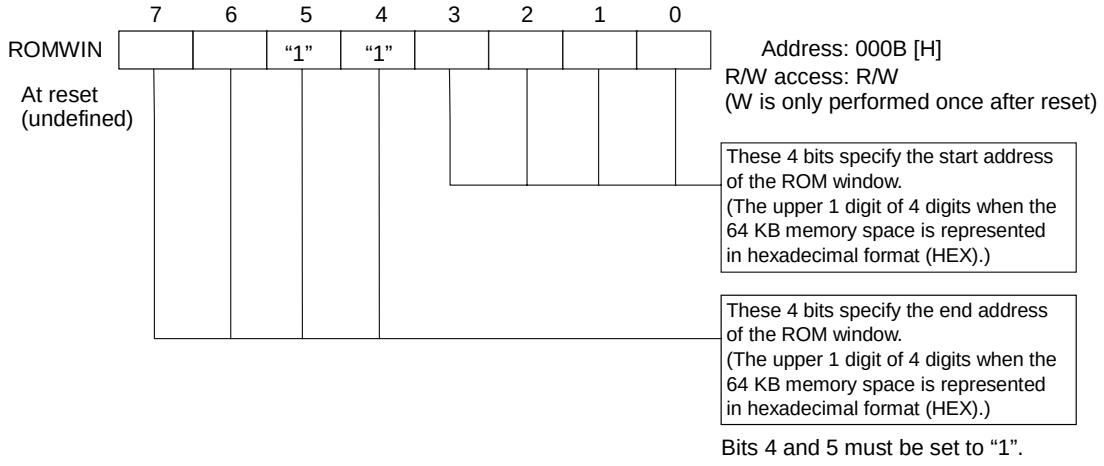


Figure 4-1 ROMWIN Configuration

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 2000H 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 (RESn 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.

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).

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 2, 4, 5, and 6, they must be written as "0". When read, bits 3 and 7 are always "1" and bits 2, 4, 5, and 6 are "0".

When reset (RESn 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.

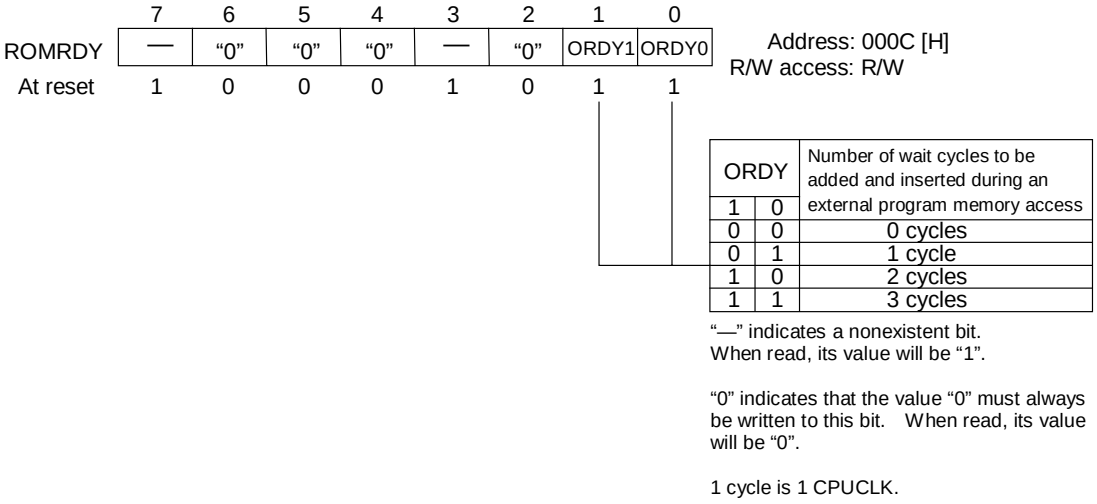


Figure 4-2 ROMRDY Configuration

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 2000H 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 (RESn 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.

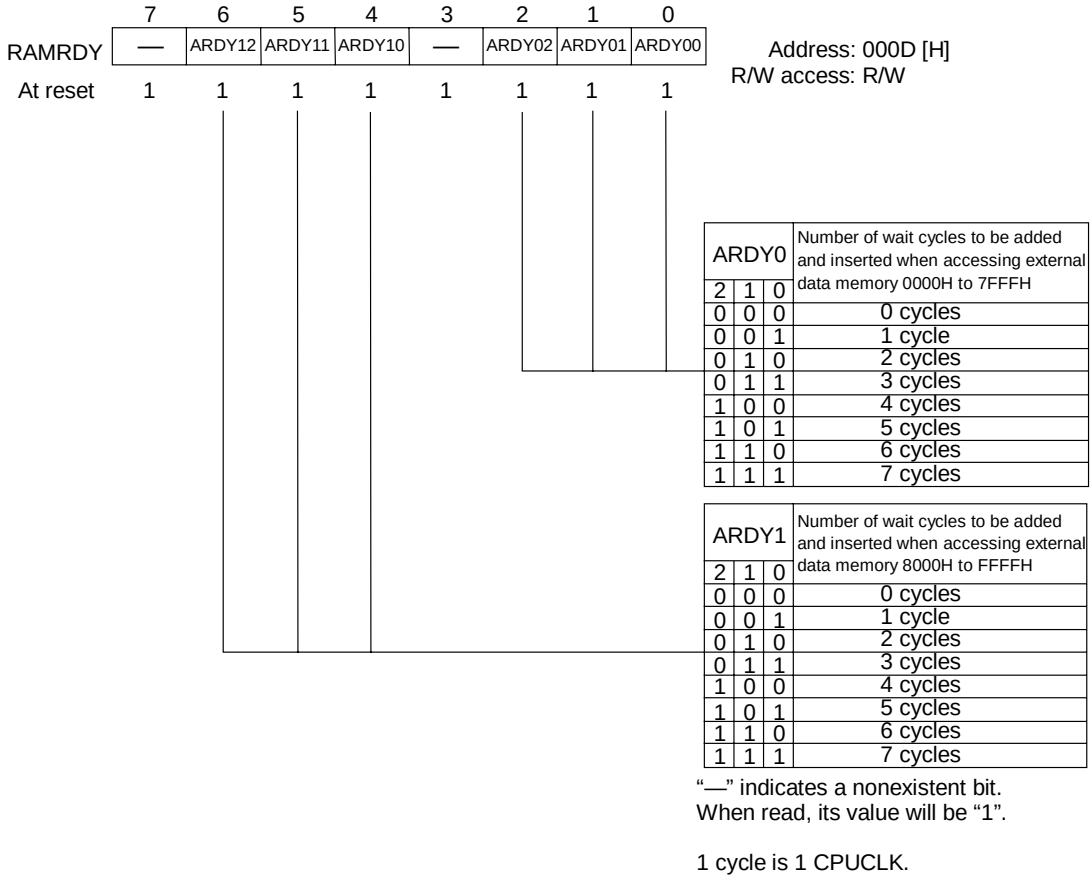


Figure 4-3 RAMRDY Configuration

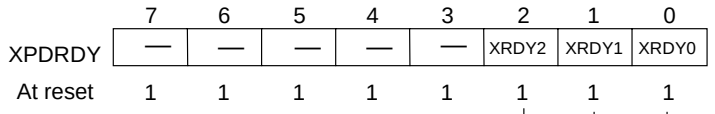
4.5 EXPANDED RAM Ready Control Register (XPDRDY)

The EXPANDED RAM ready control register (XPDRDY) consists of 3 bits. Bits 0 to 2 (XRDY0 to XRDY2) of XPDRDY specify the number of wait cycles for the EXPANDED RAM area from 1200H to 1FFFH.

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

When reset (RESn 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 the EXPANDED RAM area is accessed.

Figure 4-4 shows the configuration of XPDRDY.



XRDY			Number of wait cycles to be added and inserted when accessing EXPANDED RAM area 1200H to 1FFFH
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 4-4 XPDRDY Configuration

[Notes]
(In the explanation below, read XRDY as (XRDY2, XRDY1, XRDY0).)

1. When the main clock oscillates at 12 MHz: XRDY = 1H is recommended.
   Set XRDY to a value from 1H to 7H if the media control function is used and data is read.
2. When the main clock oscillates at 16 MHz: XRDY = 1H is recommended.
   Set XRDY to a value from 1H to 7H. Setting to 0H is disabled.
3. When the main clock oscillates at 24 MHz: XRDY = 2H is recommended.
   Set XRDY to a value from 1H to 7H. Setting to 0H is disabled.
   In the emulator, if the media control function is used and data is read, set XRDY to a value from 2H to 7H.

Chapter 5

Port Functions

5. Port Functions

5.1 Overview

The ML66525 family has 64 pins of I/O port, 6 pins of input-only port and 1 pin of output-only port.

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. (Excluding P9_0)

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 D0 to D7 (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_1	B	1	I/O	External program memory access PSENn (output)
	P3_2*	F	1	I/O	External data memory access RDn (output)
	P3_3	C	1	I/O	External data memory access WRn (output)
Port 4	P4_0 to P4_7	B	8	I/O	External memory access A0 to A7 (output)
Port 6	P6_0 to P6_3	D	4	I/O	External interrupt EXINT0 to EXINT3 (input)
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)
Port 9	P9_0, P9_1	G	2	I/O	External interrupt EXINT4, EXINT5 (input)
Port 10	P10_0	D	1	I/O	SIO3 transmit-receive clock SIOCK3 (I/O)
	P10_1	D	1	I/O	SIO3 receive data input SIOI3 (input)
	P10_2	D	1	I/O	SIO3 receive data output SIOO3 (output)
	P10_3	H	1	I/O	SIO4 transmit/receive clock SIOCK4 (I/O)
	P10_4	D	1	I/O	SIO4 transmit data output SIOO4 (output)
	P10_5	D	1	I/O	SIO4 receive data input SIOI4 (input)
Port 12	P12_0 to P12_3	E	4	I	A/D converter analog input AI0 to AI3 (input)
Port 13	P13_0, P13_1	I	2	I	External interrupts EXINT8, EXINT9 (input)
Port 15	P15_0	D	1	I/O	SIO6 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)
Port 20	P20_0 to P20_7	D	8	I/O	NAND Flash Memory Access FD0 to FD7 (I/O)
Port 21	P21_0	D	1	I/O	NAND Flash Memory Access FRDn (output)
	P21_1	D	1	I/O	NAND Flash Memory Access FWDn (output)
	P21_2	D	1	I/O	NAND Flash Memory Access FCLE (output)
	P21_3	D	1	I/O	NAND Flash Memory Access FALE (output)
	P21_4	D	1	I/O	NAND Flash Memory Access FRB (input)

* P3_2/RDn is an output-only port when used in the emulator.

5.2 Hardware Configuration of Each Port

In the ML66525 family, corresponding to each function, there are five categories of ports.

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).

Because type A ports access external program memory as a secondary function, the port status is determined by the status of the EAn pin (that specifies external memory access).

When reset (due to RESn input, BRK instruction execution, watchdog timer overflow or an opcode trap), the pin status will be as follows.

EAn pin status	Port initial status
H	High impedance input port
L	Secondary function I/O port

Figure 5-1 shows the type A configuration.

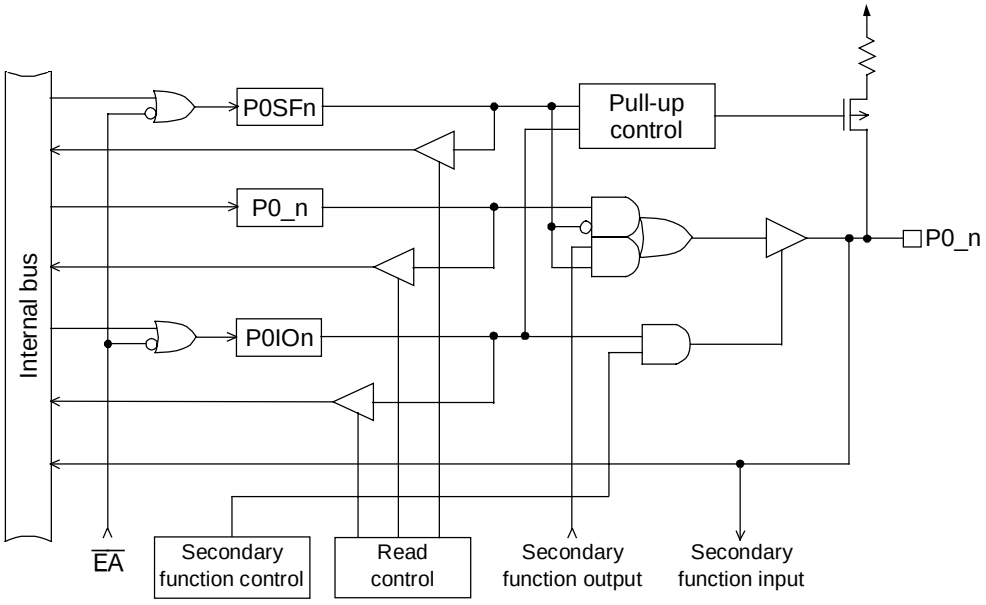


Figure 5-1 Type A Configuration

5.2.2 Type B (P1, P2, P3_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 EAn pin (that specifies external memory access).

When reset (due to RESn input, BRK instruction execution, watchdog timer overflow or an opcode trap), the pin status will be as follows.

EAn pin status	Port initial status
H	High impedance input port
L	Secondary function I/O port

Figure 5-2 shows the type B configuration.

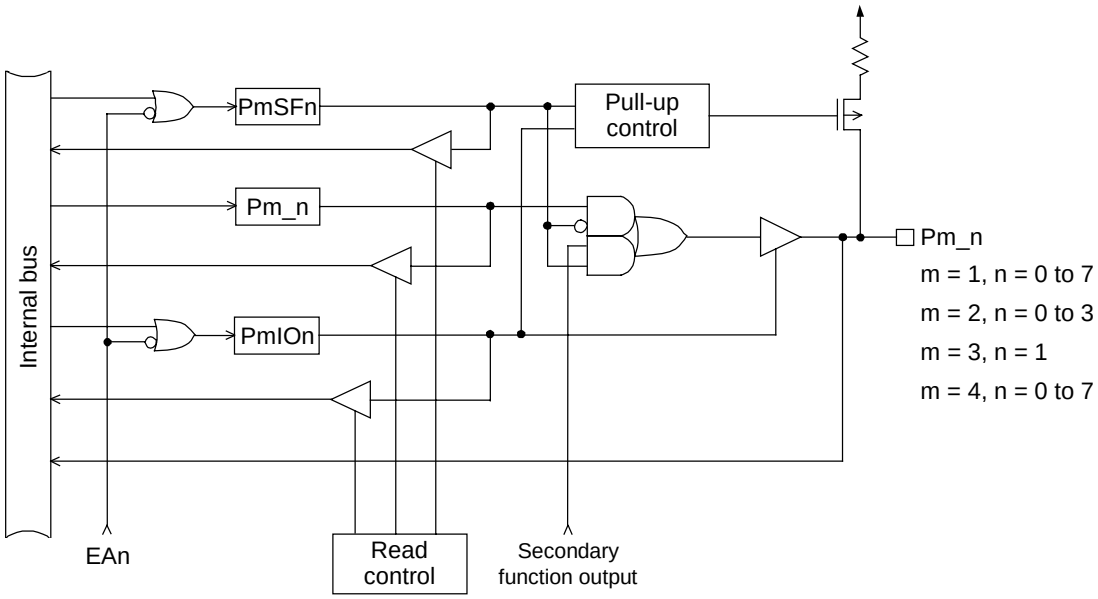


Figure 5-2 Type B Configuration

5.2.3 Type C (P3_3)

The type C port has a secondary function. Depending on the state of the port mode registers (P3IO3) and the port secondary function control registers (P3SF3), the port configuration is switched between input, pulled-up input, output, and secondary function output (external memory access).

When reset (due to RESn input, BRK instruction execution, watchdog timer overflow or an opcode trap), the initial value of P3IO3 and P3SF3 is "0" and the port will be configured as a high impedance input port.

Figure 5-3 shows the type C configuration.

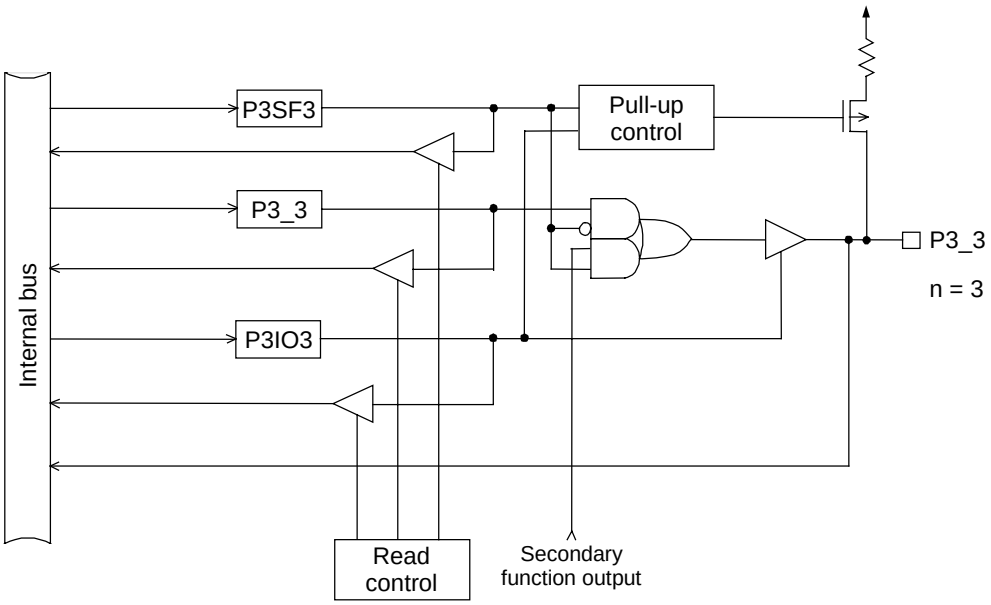


Figure 5-3 Type C Configuration

5.2.4 Type D (P6, P7, P8, P10_0 to P10_2, P10_4, P10_5, P15, P20, P21)

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 RESn 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.

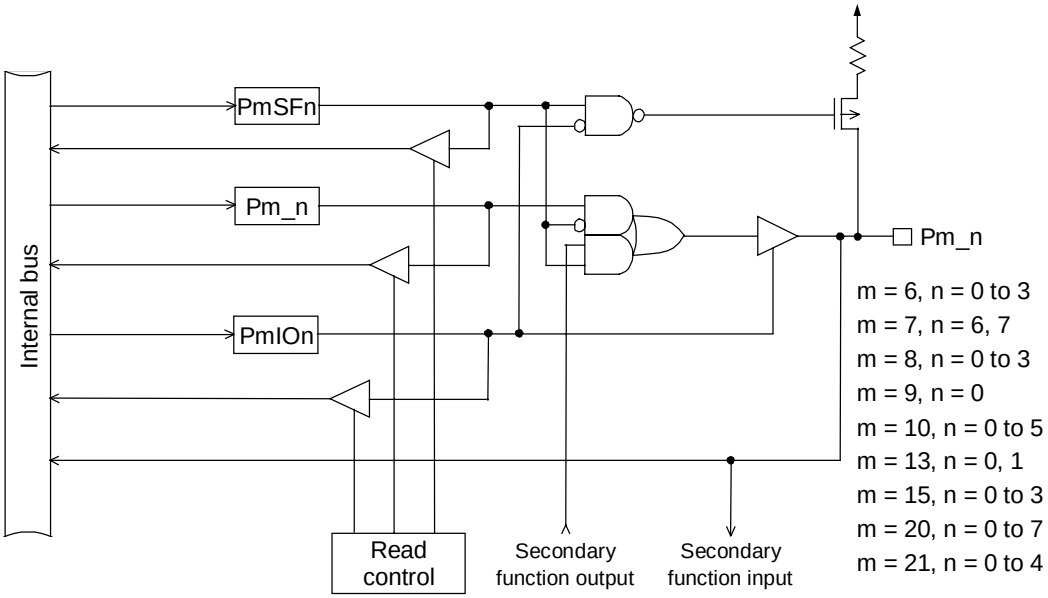


Figure 5-4 Type D Configuration

5.2.5 Type E (P12)

The type E 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 5-5 shows the type E configuration.

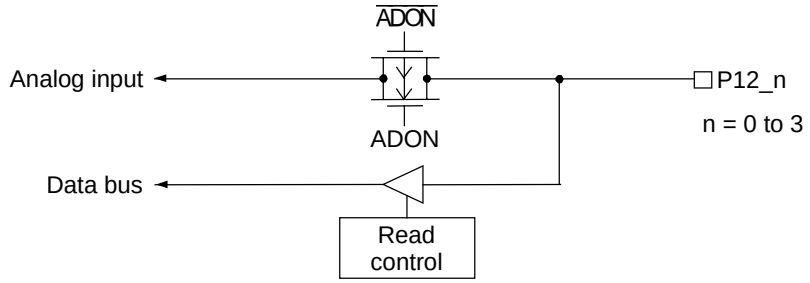


Figure 5-5 Type E Configuration

5.2.6 Type F (P3_2)

The type F port has a secondary function. Depending on the state of the port mode register (P3IO2) and the port secondary function control register (P3SF2), the port configuration is switched between input, pulled-up input, output, and secondary function output (external memory access).

Further, the 80 system or the 68 system can be selected for the secondary function output depending on the state of bit 3 (STBSEL) of the internal control register (P5IO). (See Section 13.4.)

When reset (due to RESn input, BRK instruction execution, watchdog timer overflow or an opcode trap), the initial value of P3IO2 and P3SF2 is "0" and the port will be configured as a high impedance input port. Figure 5-6 shows the type F configuration.

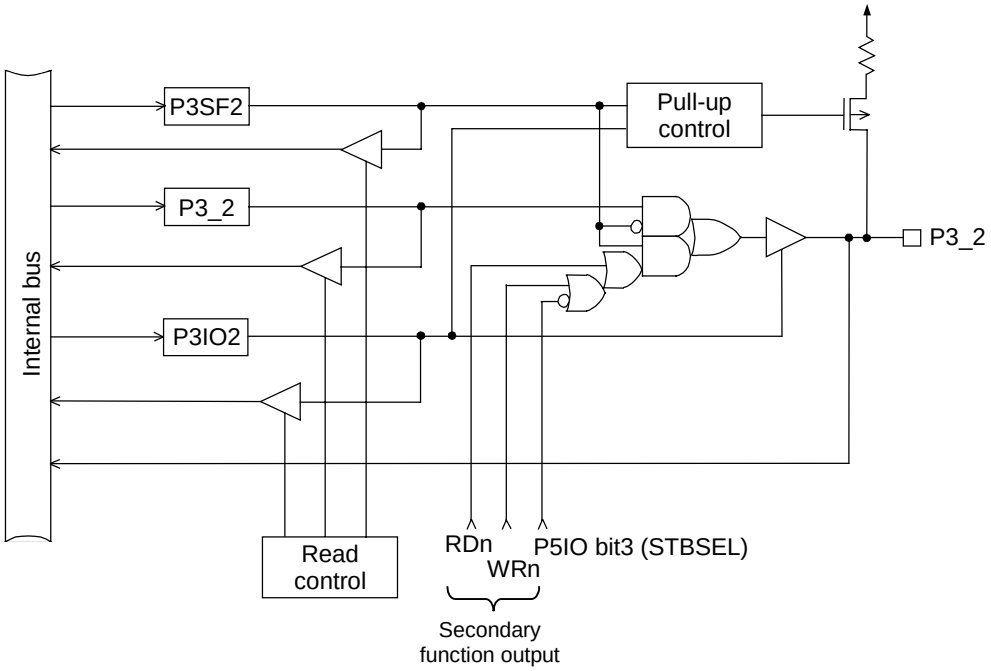


Figure 5-6 Type F Configuration

[Note]

The input state and the pull-up state cannot be set in the emulator.

5.2.7 Type G (P9_0)

The type G port has a secondary function. Depending on the state of the port mode register (P9IO0) and the port secondary function control register (P9SF0), the port configuration is switched between input (primary or secondary function) and output. This port does not have the pull-up function.

Further, since this port has a 5 V input tolerant structure, it is possible to directly input a 5 V signal.

When reset (due to RESn input, BRK instruction execution, watchdog timer overflow or an opcode trap), the initial value of P9IO0 and P9SF0 is "0" and the port will be configured as a high impedance input port. Figure 5-7 shows the type G configuration.

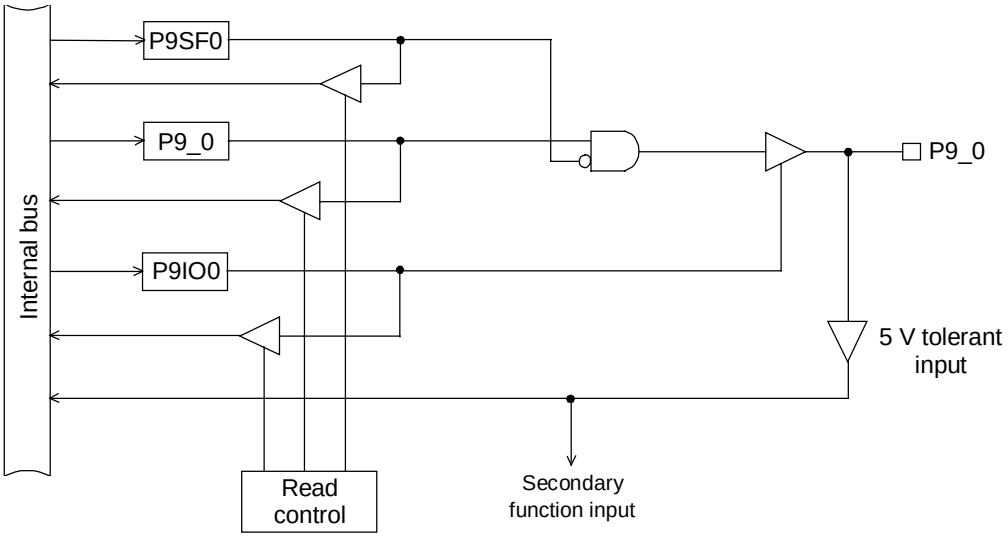


Figure 5-7 Type G Configuration

5.2.8 Type H (P10_3)

The type H port has a secondary function. Depending on the state of the port mode register (P10IO3) and the port secondary function control register (P10SF3), the port configuration is switched between input, pulled-up input, output, and secondary function output. Note that the selection between input and output must be made by setting bit 5 (SCK4IO) of the internal control register (P5IO) apart from the port mode register (P10IO3).

Further, for both input and output (primary or secondary function), it is possible to switch the polarity by bit 4 (SCK4INV) of the internal control register (P5IO). (See Section 13.4.)

When reset (due to RESn input, BRK instruction execution, watchdog timer overflow or an opcode trap), the initial value of P10IO3 and P10SF3 is "0" and the port will be configured as a high impedance input port. Figure 5-8 shows the type H configuration.

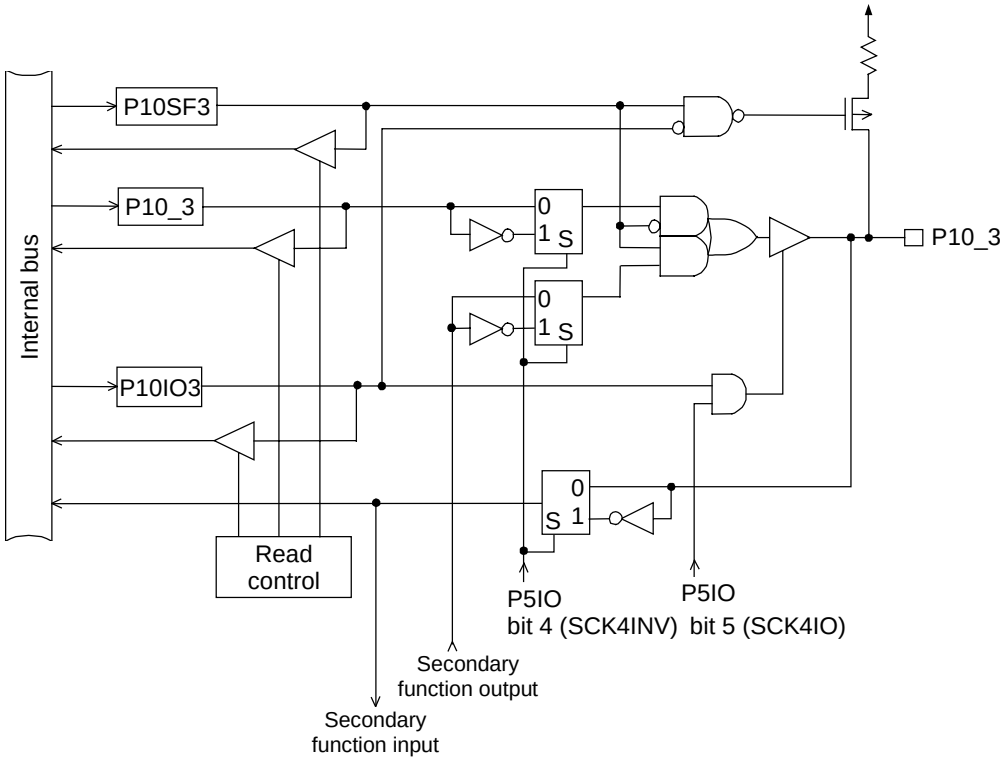


Figure 5-8 Type H Configuration

5.2.9 Type I (P13)

The type I 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).

Figure 5-9 shows the type I configuration.

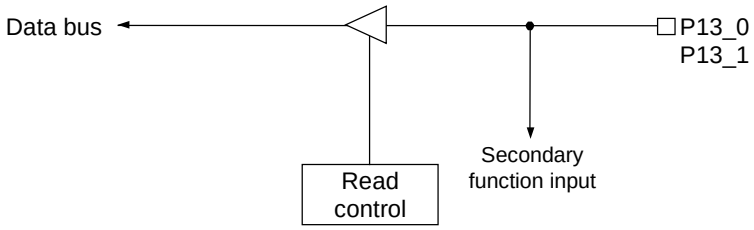


Figure 5-9 Type I Configuration

5.3 Port Registers

There are three types of port control registers in the ML66525 family.

- Port data registers (Pn: n = 0 to 4, 6 to 10, 12, 13, 15, 20, 21)
- Port mode registers (PnIO: n = 0 to 4, 6 to 10, 15, 20, 21)
- Port secondary function control registers (PnSF: n = 0 to 4, 6 to 10, 13, 15, 20, 21)

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	5-16
0019	Port 1 data register	P1	—	R/W	8	00	5-18
001A☆	Port 2 data register	P2	—	R/W	8	00	5-20
001B☆	Port 3 data register	P3	—	R/W	8	00	5-22
001C	Port 4 data register	P4	—	R/W	8	00	5-24
001E☆	Port 6 data register	P6	—	R/W	8	00	5-26
001F☆	Port 7 data register	P7	—	R/W	8	00	5-28
00B8☆	Port 8 data register	P8	—	R/W	8	00	5-30
00B9☆	Port 9 data register	P9	—	R/W	8	00	5-32
00BA☆	Port 10 data register	P10	—	R/W	8	00	5-34
00BC☆	Port 12 data register	P12	—	R	8	Undefined	5-37
00BD☆	Port 13 data register	P13	—	R	8	Undefined	5-38
00BF☆	Port 15 data register	P15	—	R/W	8	00	5-39
1B80	Port 20 data register	P20	—	R/W	8	00	5-41
1B81☆	Port 21 data register	P21	—	R/W	8	00	5-43
0020	Port 0 mode register	P0IO	—	R/W	8	00/FF	5-16
0021	Port 1 mode register	P1IO	—	R/W	8	00/FF	5-18
0022☆	Port 2 mode register	P2IO	—	R/W	8	00/0F	5-20
0023☆	Port 3 mode register	P3IO	—	R/W	8	00/02	5-22
0024	Port 4 mode register	P4IO	—	R/W	8	00/FF	5-24
0026☆	Port 6 mode register	P6IO	—	R/W	8	00	5-26
0027☆	Port 7 mode register	P7IO	—	R/W	8	00	5-28
00C0☆	Port 8 mode register	P8IO	—	R/W	8	00	5-30
00C1☆	Port 9 mode register	P9IO	—	R/W	8	00	5-32
00C2☆	Port 10 mode register	P10IO	—	R/W	8	00	5-34
00C5☆	Port 15 mode register	P15IO	—	R/W	8	00	5-39
1B82	Port 20 mode register	P20IO	—	R/W	8	00	5-41
1B83☆	Port 21 mode register	P21IO	—	R/W	8	00	5-43

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	5-16
0029	Port 1 secondary function control register	P1SF	—	R/W	8	00/FF	5-18
002A☆	Port 2 secondary function control register	P2SF	—	R/W	8	00/0F	5-20
002B☆	Port 3 secondary function control register	P3SF	—	R/W	8	00/02	5-22
002C	Port 4 secondary function control register	P4SF	—	R/W	8	00/FF	5-24
002E☆	Port 6 secondary function control register	P6SF	—	R/W	8	00	5-26
002F☆	Port 7 secondary function control register	P7SF	—	R/W	8	00	5-28
00C8☆	Port 8 secondary function control register	P8SF	—	R/W	8	00	5-30
00C9☆	Port 9 secondary function control register	P9SF	—	R/W	8	00	5-32
00CA☆	Port 10 secondary function control register	P10SF	—	R/W	8	00	5-34
00C7	Port 15 secondary function control register	P15SF	—	R/W	8	00	5-39
1B84	Port 20 secondary function control register	P20SF	—	R/W	8	00	5-41
1B85	Port 21 secondary function control register	P21SF	—	R/W	8	00	5-43

[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 EAn pin (mode registers and secondary control registers for port 0 to port 4). Listings are in the order of EAn = high-level/low-level.
4. For details, refer to Chapter 22, "Special Function Registers (SFRs)".

5.3.1 Port Data Registers (Pn : n = 0 to 4, 6 to 10, 12, 13, 15, 20, 21)

Port data registers (Pn : n = 0 to 4, 6 to 10, 12, 13, 15, 20, 21) store the port output data.

Pn registers are allocated as SFRs and when reset (due to a RESn 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 4, 6 to 10, 15, 20, 21)

Port mode registers (PnIO : n = 0 to 4, 6 to 10, 15, 20, 21) specify whether I/O ports are inputs or outputs.

PnIO registers are allocated as SFRs and when reset (due to a RESn 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 EAn 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 4, 6 to 10, 15, 20, 21)

Port secondary function control registers (PnSF : n = 0 to 4, 6 to 10, 15, 20, 21) specify the secondary function output for ports.

PnSF registers are allocated as SFRs and when reset (due to RESn 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 EAn 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 4, 6 to 10, 15, 20, 21) 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 4, 6 to 10, 15, 20, 21

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). 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 5-6 shows the configuration of the port 0 data register (P0), port 0 mode register (P0IO) and the port 0 secondary function control register (P0SF).

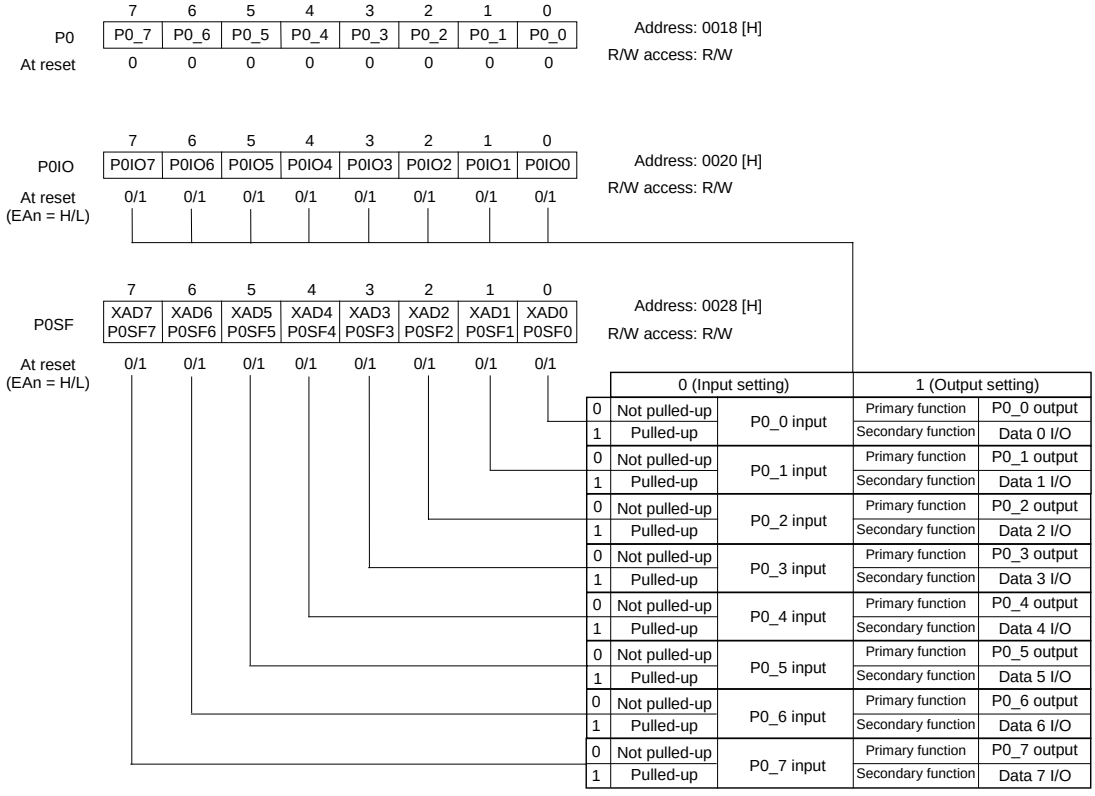


Figure 5-6 P0, P0IO, P0SF Configuration

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 RESn input, BRK instruction execution, watchdog timer overflow, or opcode trap), if the EAn 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 EAn 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).

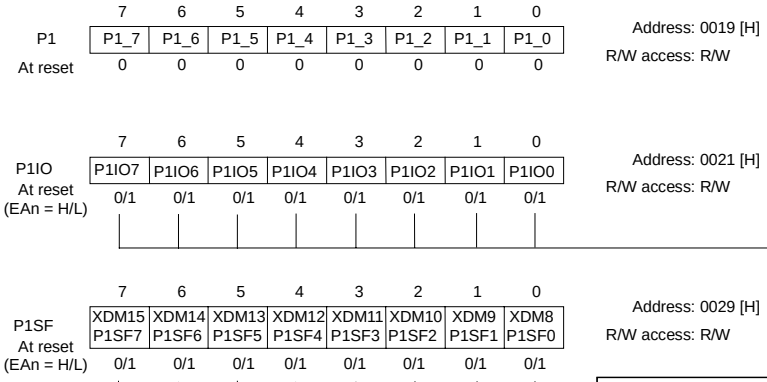
5.5 Port 1 (P1)

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 5-7 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 5-7 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 RESn input, BRK instruction execution, watchdog timer overflow, or opcode trap), if the EAn 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 EAn 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 5-8 shows the configuration of the port 2 data register (P2), port 2 mode register (P2IO) and the port 2 secondary function control register (P2SF).

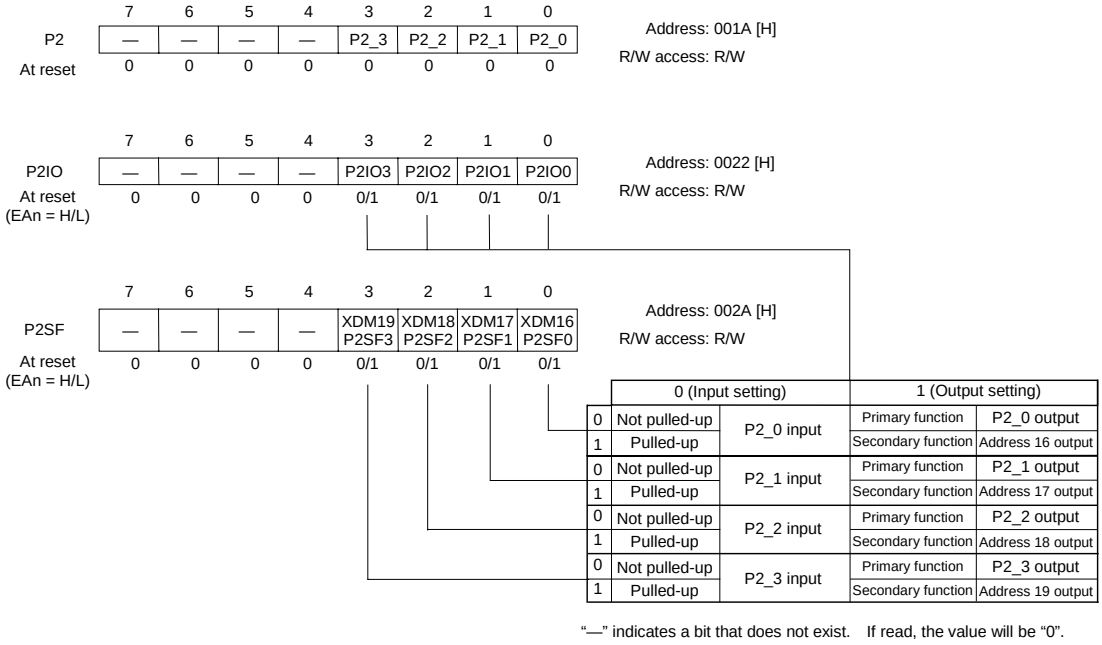


Figure 5-8 P2, P2IO, P2SF Configuration

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 RESn input, BRK instruction execution, watchdog timer overflow, or opcode trap), if the EAn 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 EAn 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 3-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 (PSENn, RDn, and WRn 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.

Bit 2 of port 3 is a read-only port.

Be sure to set bit 2 (PO3IO2) of P3IO to "1" by programming.

Figure 5-9 shows the configuration of the port 3 data register (P3), port 3 mode register (P3IO) and the port 3 secondary function control register (P3SF).

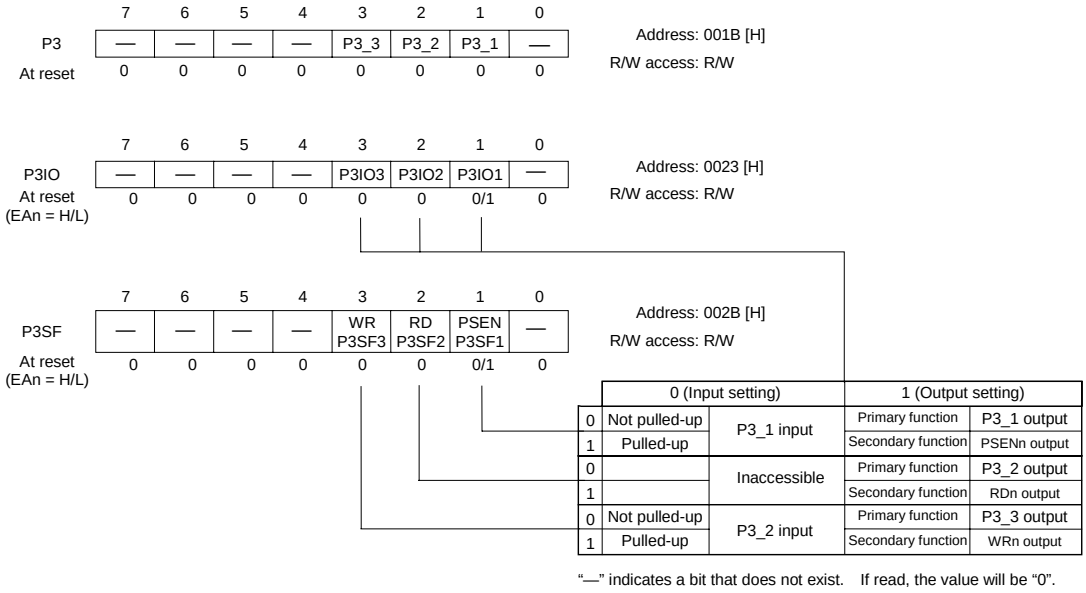


Figure 5-9 P3, P3IO, P3SF Configuration

[Note]

In setting bit 2 of port 3, do not select "Inaccessible".

At reset (RESn signal input, BRK instruction execution, overflow of watchdog timer, opcode trap), "Inaccessible" is set. Therefore, be sure to set bit 2 of P3IO to "1" by programming.

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 RESn input, BRK instruction execution, watchdog timer overflow, or opcode trap), if the EAn 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 EAn pin is at a low level, only P3_1 will be set as a secondary function I/O port (P3IO = 02H, P3SF = 02H) and the contents of P3 will be 00H.

Table 5-8 Read Data

	P3IO	P3SF	Read data
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). 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 5-10 shows the configuration of the port 4 data register (P4), port 4 mode register (P4IO) and the port 4 secondary function control register (P4SF).

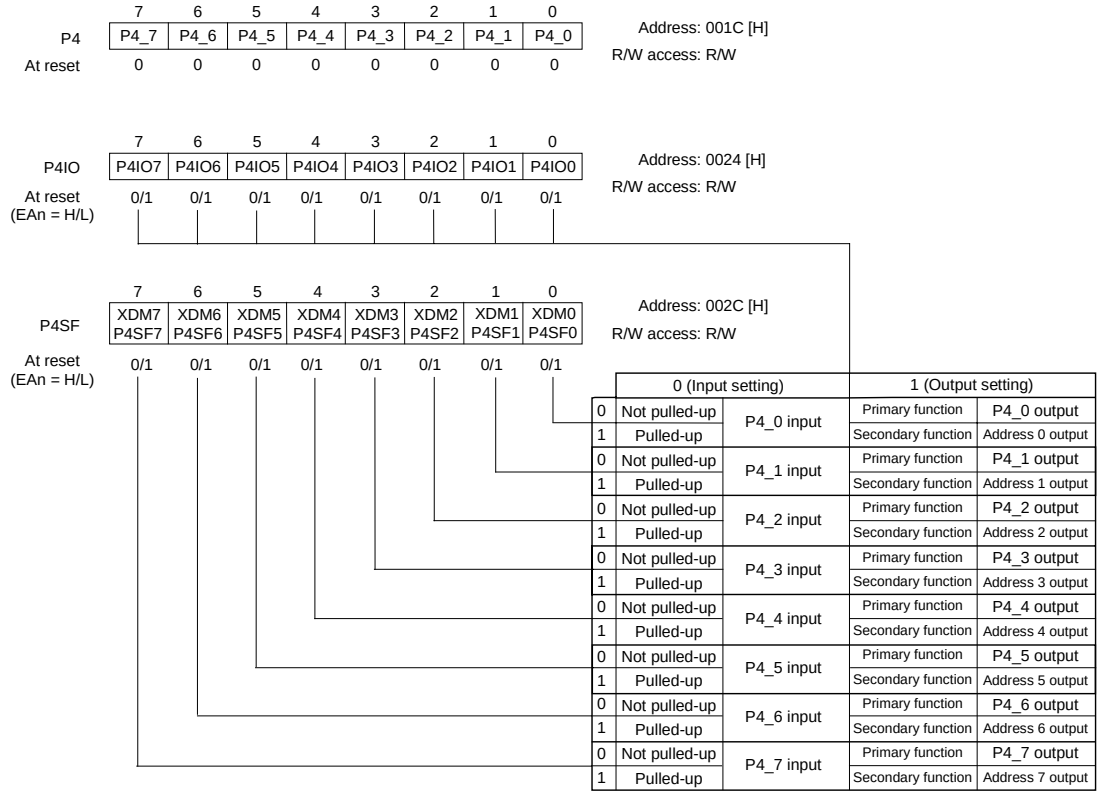


Figure 5-10 P4, P4IO, P4SF Configuration

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 RESn input, BRK instruction execution, watchdog timer overflow, or opcode trap), if the EAn 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 EAn 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 6 (P6)

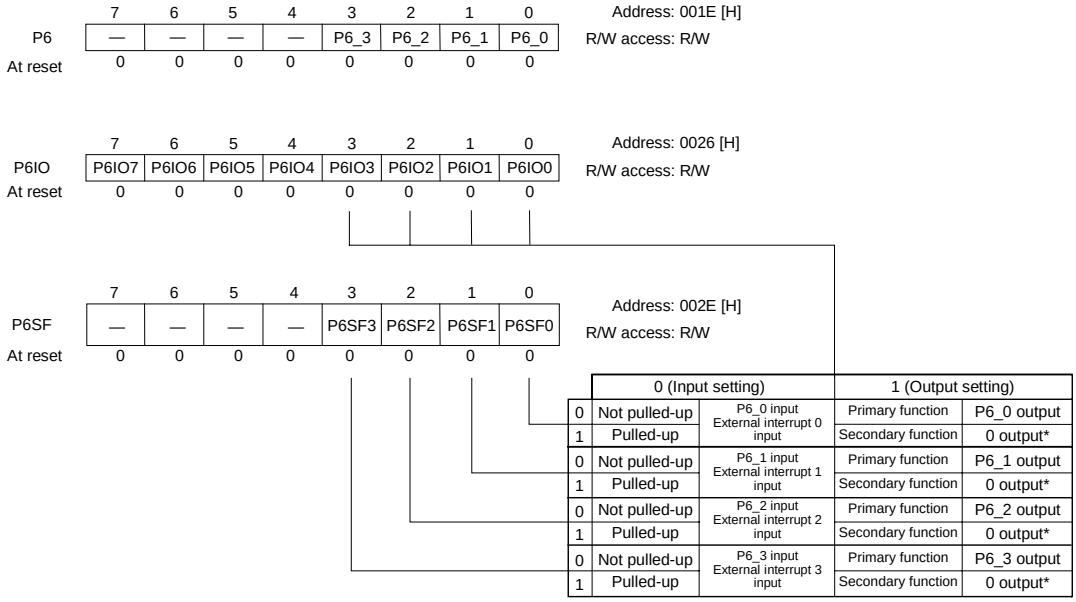
Port 6 is a 4-bit I/O port. Each individual bit can be specified as input or output by the port 6 mode register (P6IO). When output is specified (corresponding bits of P6IO = "1"), the value of the corresponding bits in the port 6 data register (P6) will be output from their appropriate pins.

In addition to its port function, P6 is assigned a secondary function (external interrupt input). 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 port 6 is 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 5-11 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 output*: "0" is output, regardless of the value of the port data register.
"—" indicates a bit that does not exist.

Figure 5-11 P6, P6IO, P6SF Configuration

Table 5-10 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 RESn 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 5-10 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"

[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 5-10. The modified values will be written to P6 (port 6 data register).

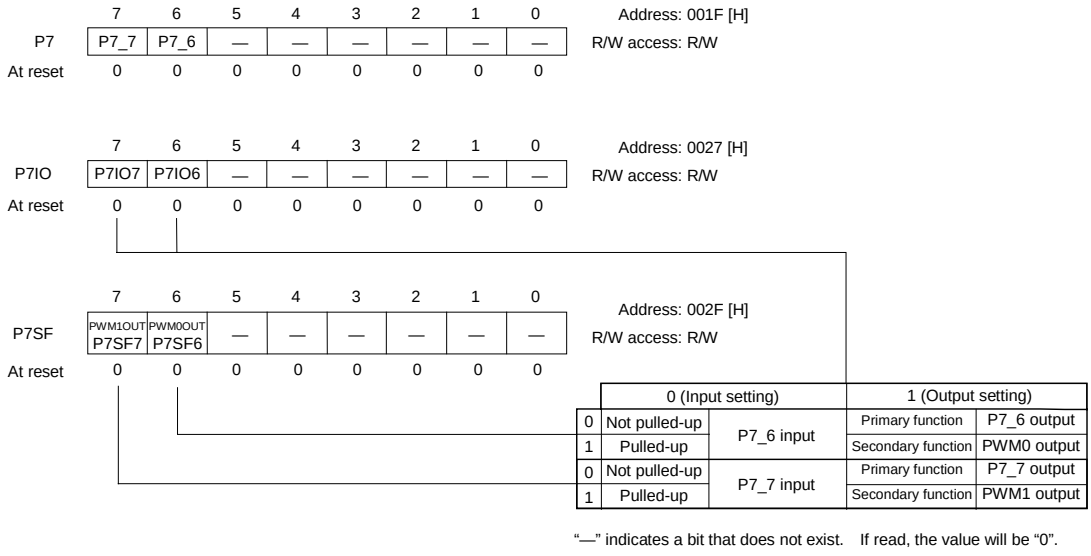
5.10 Port 7 (P7)

Port 7 is a 2-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 SIO0 receive data input). 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 a secondary function input is to be used, reset corresponding bits of the port 7 mode register (P7IO) 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 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 5-12 shows the configuration of the port 7 data register (P7), port 7 mode register (P7IO) and the port 7 secondary function control register (P7SF).



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

Figure 5-12 P7, P7IO, P7SF Configuration

Table 5-11 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 RESn 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 5-11 P7 Read Data

	P7IO	P7SF	Read data
P7_5	0	*	P7_5/TM3EVT pin state
	1	0	Value of bit 5 of P7 (port data register)
	1	1	"0"
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 5-11. The modified values will be written to P7 (port 7 data register).

5.11 Port 8 (P8)

Port 8 is a 4-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 5-13 shows the configuration of the port 8 data register (P8), port 8 mode register (P8IO) and the port 8 secondary function control register (P8SF).

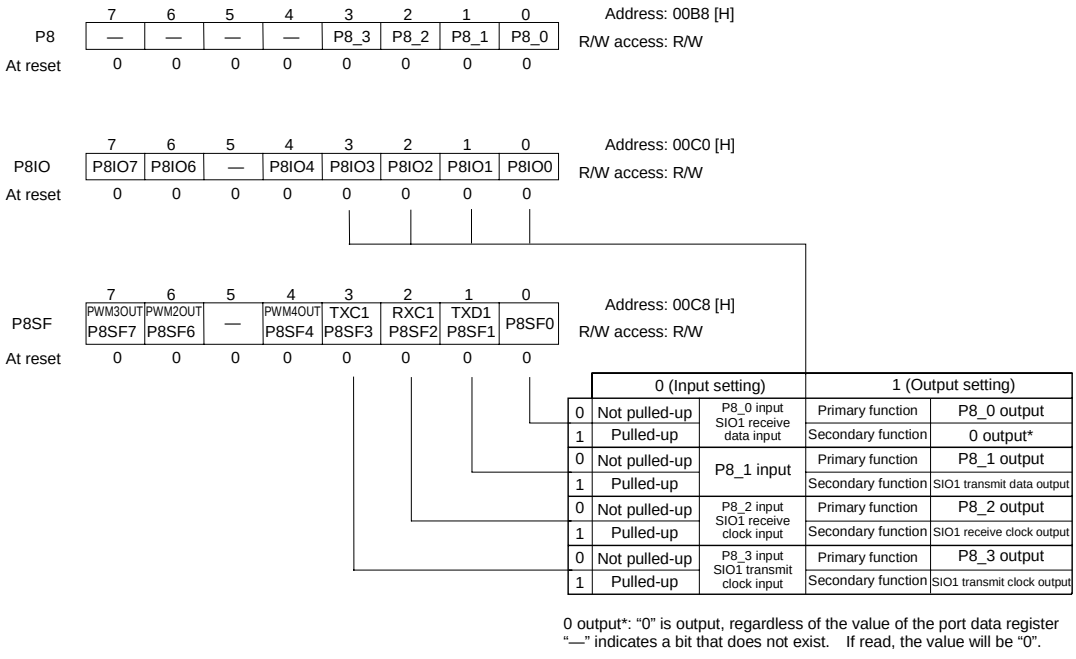


Figure 5-13 P8, P8IO, P8SF Configuration

Table 5-12 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 RESn 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 5-12 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

"*" 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 5-12. The modified values will be written to P8 (port 8 data register).

5.12 Port 9 (P9)

Port 9 is a 1-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 Vbus detection interrupt). 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).

Bit 0 of port 9 cannot 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 5-14 shows the configuration of the port 9 data register (P9), port 9 mode register (P9IO) and the port 9 secondary function control register (P9SF).

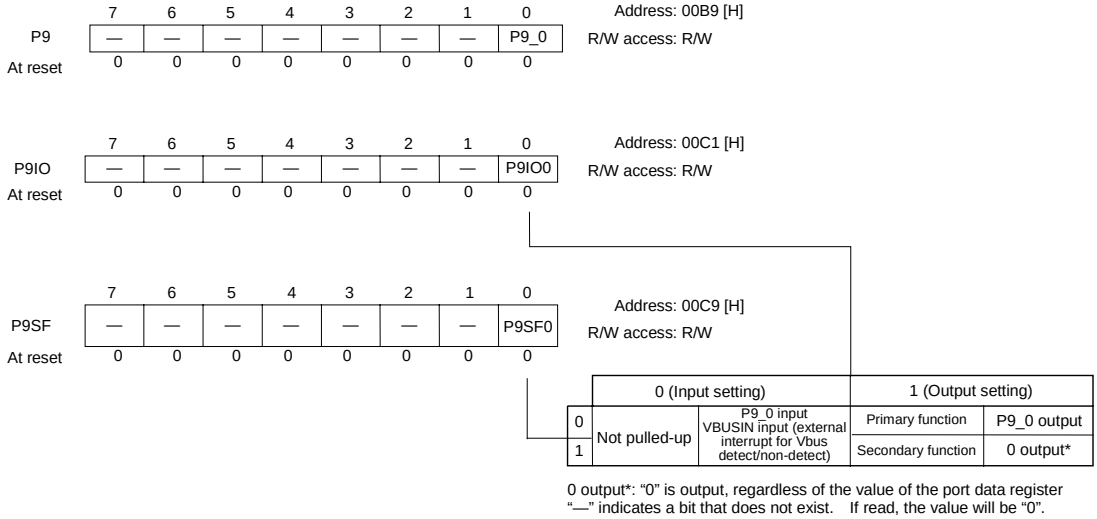


Figure 5-14 P9, P9IO, P9SF Configuration

Table 5-13 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 RESn 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 5-13 P9 Read Data

	P9IO	P9SF	Read data
P9_0	0	*	P9_0/VBUSIN pin state
	1	0	Value of bit 0 of P9 (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 P9, depending on the settings of P9IO and P9SF, values will be read as listed in Table 5-13. The modified values will be written to P9 (port 9 data register).

5.13 Port 10 (P10)

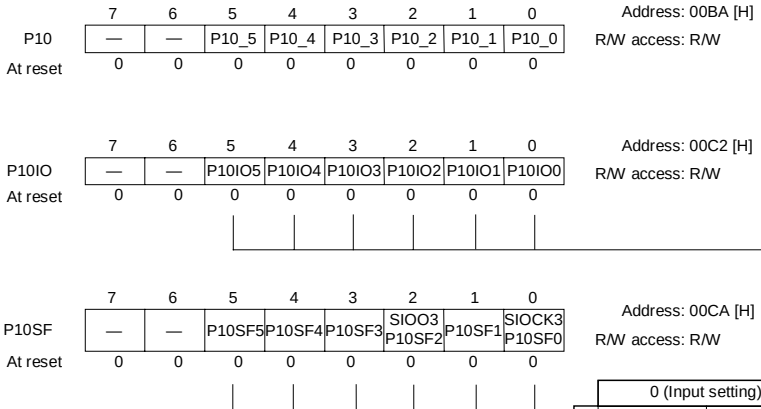
Port 10 is a 6-bit I/O port. Each individual bit can be specified as input or output by the port 10 mode register (P10IO). When output is specified (corresponding bits of P10IO = "1"), the value of the corresponding bits in the port 10 data register (P10) will be output from their appropriate pins.

In addition to its port function, P10 is assigned secondary functions (such as SIO3 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 bits 1, 3, 4, and 5 of port 10 are 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 5-15 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_0 input SIO3 transmit-receive clock input	Primary function	P10_0 output
1	Pulled-up		Secondary function	SIO3 transmit-receive clock output
0	Not pulled-up	P10_1 input SIO3 receive data input	Primary function	P10_1 output
1	Pulled-up		Secondary function	0 output*
0	Not pulled-up	P10_2 input	Primary function	P10_2 output
1	Pulled-up		Secondary function	SIO3 transmit data output
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 receive data output
0	Not pulled-up	P10_5 input SIO4 receive data 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 5-15 P10, P10IO, P10SF Configuration

Table 5-14 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 RESn 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 5-14 P10 Read Data

	P10IO	P10SF	Read data
P10_0	0	*	P10_0/SIOCK3 pin state
	1	0	Value of bit 0 of P10 (port data register)
	1	1	SIOCK3 output data
P10_1	0	*	P10_1/SIOI3 pin state
	1	0	Value of bit 1 of P10 (port data register)
	1	1	"0"
P10_2	0	*	P10_2 pin state
	1	0	Value of bit 2 of P10 (port data register)
	1	1	SIOO3 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 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 5-14. The modified values will be written to P10 (port 10 data register).

However, regarding the P10_3/SIOCK4 pin, the output setting of the pin also requires the setting using bit 5 (SCK4IO) of the internal control register (P5IO). In other words, when using P10_3 as an output (primary or secondary function), it is required to set SCK4IO to "1" in addition to setting P10IO3 to "1".

Further, the polarity of the P10_3/SIOCK4 pin can be selected for both input and output using bit 4 (SCK4INV) of the internal control register (P5IO). The polarity of the P10_3/SIOCK4 pin will be positive when SCK4INV is set to "0", and negative when SCK4INV is set to "1".

When a read instruction is executed for P10_3, the content of the data that is read out is given in Table 5-15 depending on the settings of P10IO3, P10SF3 or SCK4IO, and SCK4INV. The shaded portions are the modes that are not normally set.

Table 5-15 P10_3 Read Data

	P10IO4	P10SF4	SCK4IO	SCK4INV	Read data	I/O	Polarity
P10_3	0	*	0	0	State of P10_3/SIOCK4 pin	Input	Positive
	0	*	0	1	Inverted state of P10_3/SIOCK4 pin	Input	Negative
	0	*	1	0	State of P10_3/SIOCK4 pin	Input	Positive
	0	*	1	1	Inverted state of P10_3/SIOCK4 pin	Input	Negative
	1	0	0	0	State of P10_3/SIOCK4 pin	Input	Positive
	1	0	0	1	Inverted state of P10_3/SIOCK4 pin	Input	Negative
	1	0	1	0	Value of bit 3 of P10_3 (port data register)	Output	Positive
	1	0	1	1	Inverted value of bit 3 of P10_3 (port data register)	Output	Negative
	1	1	0	0	State of P10_3/SIOCK4 pin	Input	Positive
	1	1	0	1	Inverted state of P10_3/SIOCK4 pin	Input	Negative
	1	1	1	0	SIOCK4 output data	Output	Positive
	1	1	1	1	Inverted data of SIOCK4 output	Output	Negative

"*" indicates "0" or "1"

5.14 Port 12 (P12)

Port 12 is a 4-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 5-16 shows the configuration of the port 12 data register (P12). Table 5-16 lists the P12 read data.

	7	6	5	4	3	2	1	0
P12	—	—	—	—	P12_3	P12_2	P12_1	P12_0

Address: 00BC [H]
R/W access: R
"—" indicates a bit that does not exist.

Figure 5-16 P12 Configuration

Table 5-16 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

5.15 Port 13 (P13)

Port 13 is a 2-bit input-only port. Therefore, there is no mode register or secondary function control register.

The pin status can be read by the port 13 data register (P13).

In addition to its port function, a secondary function (external interrupt 8/9) is assigned to P13 (same input as the primary function input).

There are no pulled-up inputs at port 13.

Figure 5-17 shows the configuration of the port 13 data register (P13). Table 5-17 lists the P13 read data.

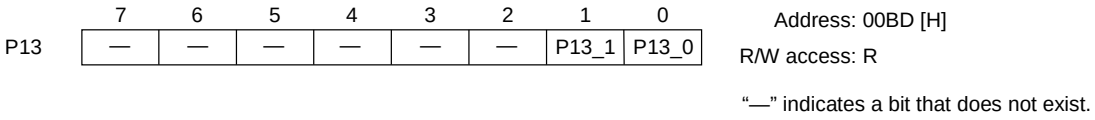


Figure 5-17 P13 Configuration

Table 5-17 P13 Read Data

	Read data
P13_0	P13_0/external interrupt 8 pin
P13_1	P13_1/external interrupt 9 pin

5.16 Port 15 (P15)

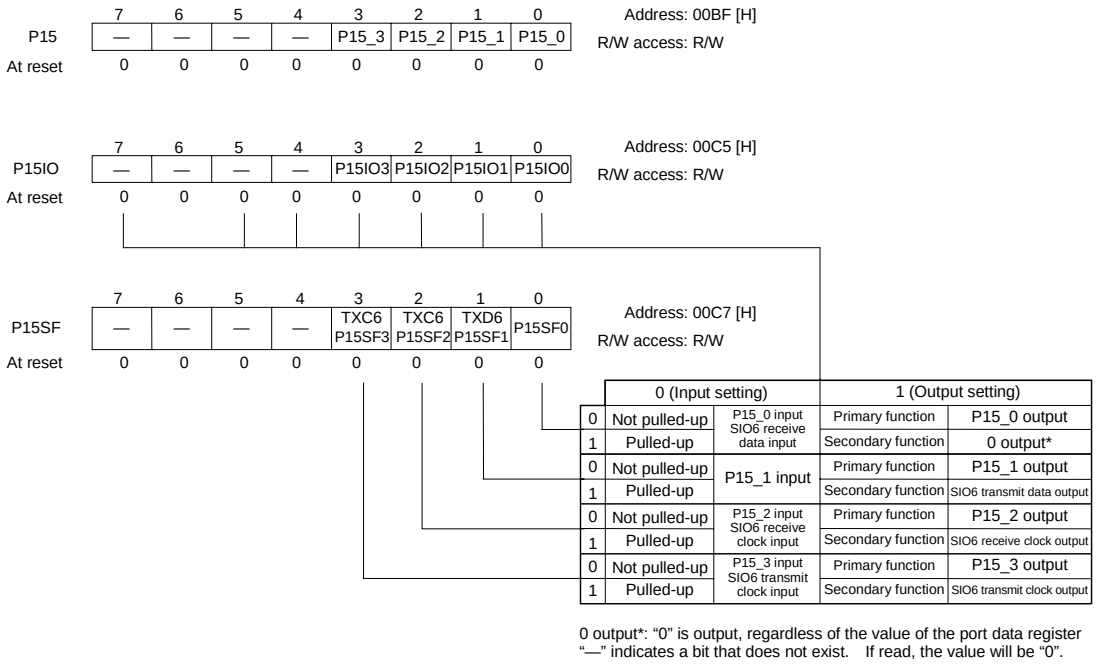
Port 15 is a 4-bit I/O port. Each individual bit can be specified as input or output by the port 15 mode register (P15IO). When output is specified (corresponding bits of P15IO = "1"), the value of the corresponding bits in the port 15 data register (P15) will be output from their appropriate pins.

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

If bit 0 of port 15 is set 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 5-18 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 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 P15, P15IO, P15SF Configuration

Table 5-18 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 RESn 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 5-18 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/RXC1 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 arithmetic, 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 5-18. The modified values will be written to P15 (port 15 data register).

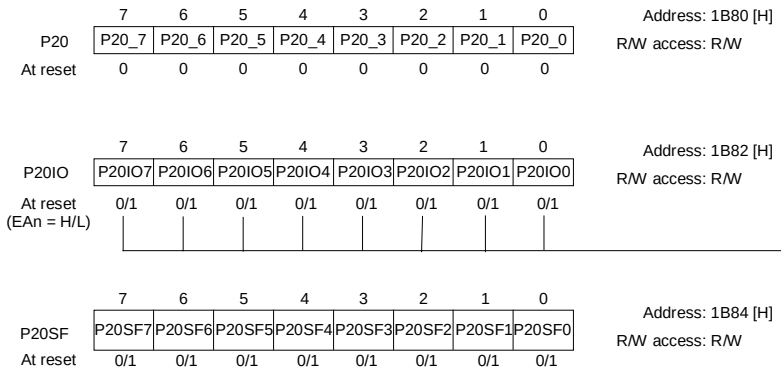
5.17 Port 20 (P20)

Port 20 is an 8-bit I/O port. Each individual bit can be specified as input or output by the port 20 mode register (P20IO). When output is specified (corresponding bits of P20IO = "1"), the value of the corresponding bits in the port 20 data register (P20) will be output from their appropriate pins.

In addition to its port function, P20 is assigned a secondary function (external NAND Flash memory data I/O). If the secondary function is to be used, set the corresponding bits of the port 0 mode register (P20IO) and the port 20 secondary function control register (P20SF) to "1".

If the port is specified as an input (corresponding bits of P20IO = "0") and the port 20 secondary function control register (P20SF) is set to "1", the pin inputs corresponding to those bits will be pulled-up.

Figure 5-19 shows the configuration of the port 20 data register (P20), port 20 mode register (P20IO) and the port 20 secondary function control register (P20SF).



	0 (Input setting)		1 (Output setting)	
0	Not pulled-up	P20_0 input	Primary function	P20_0 output
1	Pulled-up		Secondary function	F data 0 I/O
0	Not pulled-up	P20_1 input	Primary function	P20_1 output
1	Pulled-up		Secondary function	F data 1 I/O
0	Not pulled-up	P20_2 input	Primary function	P20_2 output
1	Pulled-up		Secondary function	F data 2 I/O
0	Not pulled-up	P20_3 input	Primary function	P20_3 output
1	Pulled-up		Secondary function	F data 3 I/O
0	Not pulled-up	P20_4 input	Primary function	P20_4 output
1	Pulled-up		Secondary function	F data 4 I/O
0	Not pulled-up	P20_5 input	Primary function	P20_5 output
1	Pulled-up		Secondary function	F data 5 I/O
0	Not pulled-up	P20_6 input	Primary function	P20_6 output
1	Pulled-up		Secondary function	F data 6 I/O
0	Not pulled-up	P20_7 input	Primary function	P20_7 output
1	Pulled-up		Secondary function	F data 7 I/O

Figure 5-19 P20, P20IO, P20SF Configuration

Table 5-19 lists the data that is read, depending on the settings of P20IO and P20SF, when executing an instruction to read P20.

At reset (due to RESn input, BRK instruction execution, watchdog timer overflow, or opcode trap), if the EAn pin is at a high level, P20 will become a high impedance input port (P20IO = 00H, P20SF = 00H) and the contents of P0 will be 00H. If the EAn pin is at a low level, P20 will be set as a secondary function I/O port (P20IO = FFH, P20SF = FFH) and the contents of P20 will be 00H.

Table 5-19 P20 Read Data

	P20IO	P20SF	Read data
P20_0	0	*	P20_0 pin state
	1	*	Value of bit 0 of P0 (port data register)
P20_1	0	*	P20_1 pin state
	1	*	Value of bit 1 of P0 (port data register)
P20_2	0	*	P20_2 pin state
	1	*	Value of bit 2 of P0 (port data register)
P20_3	0	*	P20_3 pin state
	1	*	Value of bit 3 of P0 (port data register)
P20_4	0	*	P20_4 pin state
	1	*	Value of bit 4 of P0 (port data register)
P20_5	0	*	P20_5 pin state
	1	*	Value of bit 5 of P0 (port data register)
P20_6	0	*	P20_6 pin state
	1	*	Value of bit 6 of P0 (port data register)
P20_7	0	*	P20_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 P20, depending on the settings of P20IO and P20SF, values will be read as listed in Table 5-19. The modified values will be written to P20 (port 20 data register).

5.18 Port 21 (P21)

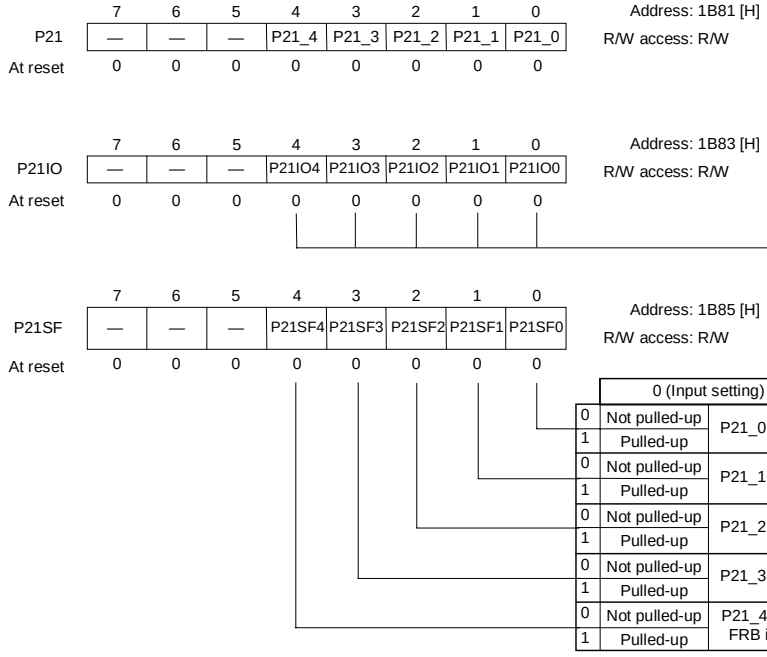
Port 21 is a 5-bit I/O port. Each individual bit can be specified as input or output by the port 21 mode register (P21IO). When output is specified (corresponding bits of P21IO = "1"), the value of the corresponding bits in the port 21 data register (P8) will be output from their appropriate pins.

In addition to its port function, P21 is assigned secondary functions (such as external NAND Flash memory RDn, WRn output). If a secondary function output is to be used, set the corresponding bits of the port 21 mode register (P21IO) and the port 21 secondary function control register (P21SF) to "1". If a secondary function input is to be used, reset corresponding bits of the port 21 mode register (P21IO) 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 P21IO = "0") and the port 21 secondary function control register (P21SF) is set to "1", the pin inputs corresponding to those bits will be pulled-up.

If bit 4 of port 21 is set as a secondary function output (P21IO0 = 1, P21SF0 = 1), the output will be fixed at "0", regardless of the value of the port 21 data register.

Figure 5-20 shows the configuration of the port 21 data register (P21), port 21 mode register (P21IO) and the port 21 secondary function control register (P21SF).



	0 (Input setting)		1 (Output setting)	
0	Not pulled-up	P21_0 input	Primary function	P21_0 output
1	Pulled-up		Secondary function	FRDn output
0	Not pulled-up	P21_1 input	Primary function	P21_1 output
1	Pulled-up		Secondary function	FWRn output
0	Not pulled-up	P21_2 input	Primary function	P21_2 output
1	Pulled-up		Secondary function	SIO1 receive clock output
0	Not pulled-up	P21_3 input	Primary function	P21_3 output
1	Pulled-up		Secondary function	FCLE output
0	Not pulled-up	P21_4 input FRB input	Primary function	P21_4 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 5-20 P21, P21IO, P21SF Configuration

Table 5-20 lists the data that is read, depending on the settings of P21IO and P21SF, when executing an instruction to read P21.

At reset (due to a RESn input, BRK instruction execution, watchdog timer overflow, or opcode trap), P21 will become a high impedance input port (P21IO = 00H, P21SF = 00H) and the contents of P21 will be 00H.

Table 5-20 P21 Read Data

	P21IO	P21SF	Read data
P21_0	0	*	P21_0/RXD1 pin state
	1	*	Value of bit 0 of P21 (port data register)
P21_1	0	*	P21_1 pin state
	1	*	Value of bit 1 of P21 (port data register)
P21_2	0	*	P21_2/RXC1 pin state
	1	*	Value of bit 2 of P21 (port data register)
P21_3	0	*	P21_21/TXC1 pin state
	1	*	Value of bit 3 of P21 (port data register)
P21_4	0	*	P21_4 pin state
	1	0	Value of bit 4 of P21 (port data register)
	1	1	TM4OUT output data

"*" indicates "0" or "1"

[Note]

If arithmetic, SB, RB, XORB or other read-modify-write instructions are executed for P21, depending on the settings of P21IO and P21SF, values will be read as listed in Table 5-20. The modified values will be written to P21 (port 21 data register).

Chapter 6

Clock Oscillation Circuit

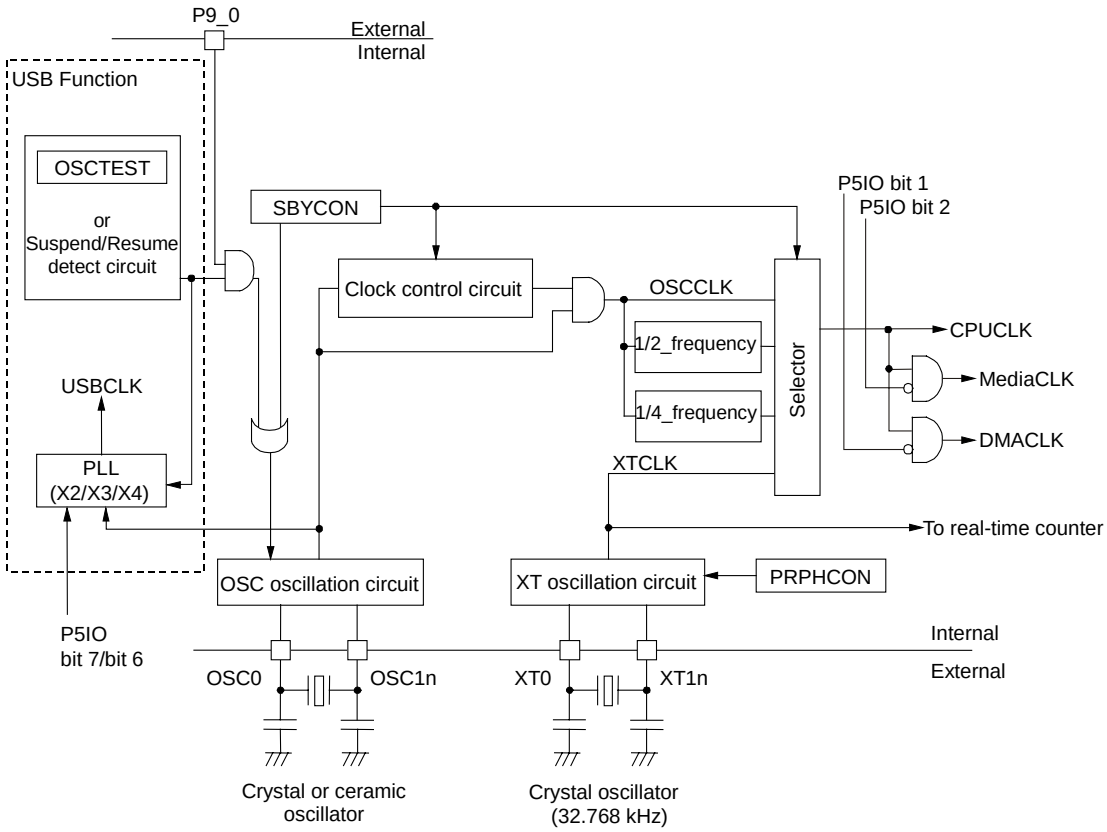
6. Clock Oscillation Circuit

6.1 Overview

The ML66525 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 (subclock). 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
MediaCLK: Clock for Media control function
DMACLK: Clock for internal DMA function
USBCLK: Clock for USB function (48 MHz)
SBYCON: Standby control register
PRPHCON: Peripheral control register
P5IO: Internal control register
OSCTEST: Oscillation test 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	13-2

[Notes]

1. A star (☆) in the address column indicates a missing bit.
2. For details, refer to Chapter 22, "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 OSC1n.

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

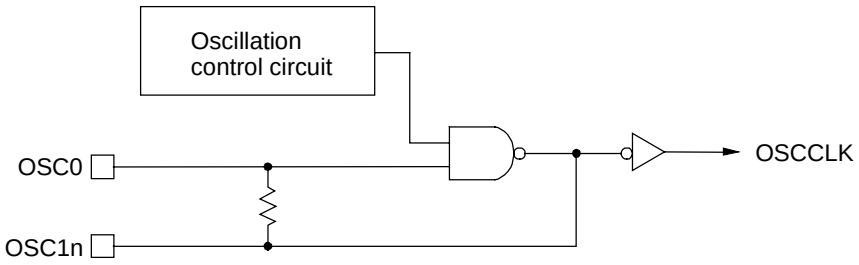


Figure 6-2 OSC Oscillation Circuit Configuration

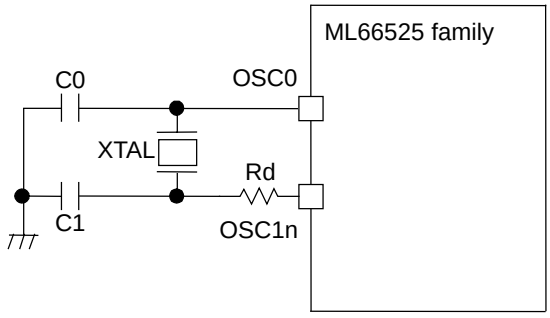


Figure 6-3 OSC Crystal Oscillation Circuit Connection Example

[Notes]

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

Table 6-2 shows examples of circuit constants in the case of using a ceralock of Murata MFG. make.

Table 6-2 Examples of Circuit Constants

Frequency	Product number	Recommended constants			Operating conditions	
		C1 [pF]	C2 [pF]	Rd [Ω]	Power supply voltage range [V]	Temperature range [°C]
12.00M	CSTLA12M0T58-B0	(100)	(100)	0	2.4 to 3.6	–30 to +70
	CSTCV12M0T54J-R0	(22)	(22)	470		
16.00M	CSTLS16M0X55-B0	(30)	(30)	47	2.7 to 3.6	
	CSACV16M0X55J-R0	33	33	47		
24.00M	CSTLS24M0X54-B0	(22)	(22)	47	2.7 to 3.6	
	CSACW24M0X53-R0	22	22	47		
	CSACW24M0X13xxx-R0*	22	22	47		

[Notes]

1. Products with their product numbers beginning with CST have load capacitors C1 and C2 built in.
2. xxx denotes a custom number.
3. "*" indicates an oscillator with high accuracy for USB use.
4. That the operating condition for frequencies of 16 and 24 MHz is rated as 2.7 to 3.6 V means that there is no such constant as can cover the entire range from 2.4 to 3.6 V. When the device is operated at a voltage less than 2.7 V, set the constants by using a printed circuit board for an application product.

If the main clock pulse (OSCCLK) is to be supplied externally, connect it directly to the OSC0 pin input. Leave the OSC1n pin open (unconnected).

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

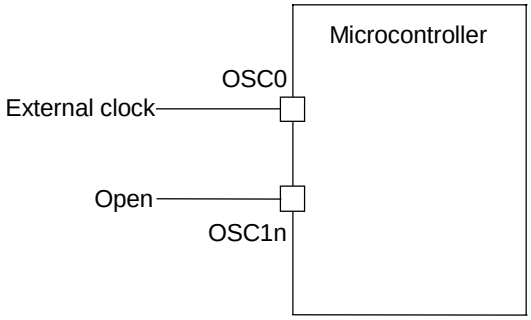


Figure 6-4 Connection Example for External OSC Clock Input

[Note]

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

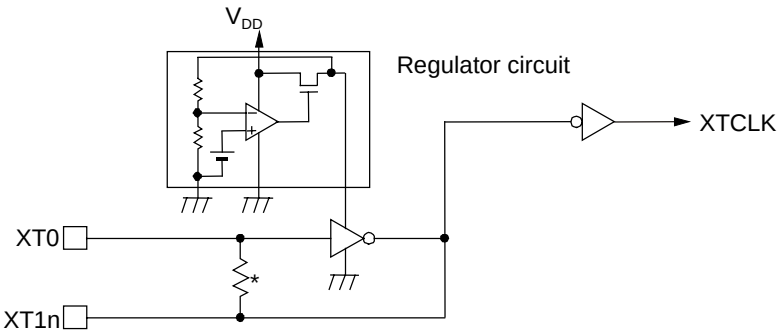
The standby control register (SBYCON) can be set to halt the OSC oscillation circuit. When resuming oscillation of the OSC oscillation circuit from a halted state, the main clock pulse (OSCCLK) will be transmit after waiting for the oscillation stabilization time, the number of clock cycles specified 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 by using a circuit board for an application product, 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 XT1n.

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

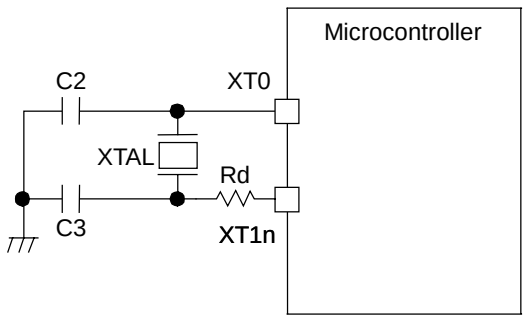


Figure 6-6 XT Crystal Oscillation Circuit Connection Example

[Notes]

1. The values of C2, C3 and Rd 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 XT1n pin open (unconnected).

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

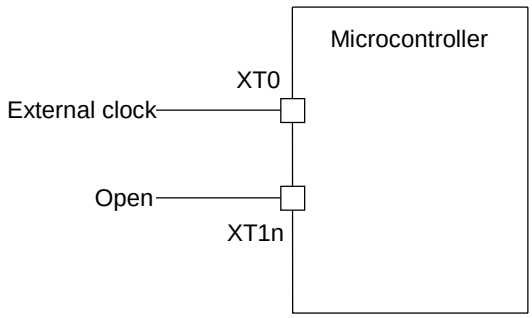


Figure 6-7 Connection Example for External XT Clock Input

[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 subclock (XTCLK) as the CPU operation clock (CPUCLK).

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

6.6 PLL Circuit

The PLL circuit generates the clock (48 MHz) inside the USB macro by frequency multiplication of the output clock from the main clock oscillator circuit.

The multiplication factor should be selected by setting bit 7 and bit 6 (FSEL1, FSEL0) of the internal control register (P5IO) as shown below so that the clock inside the USB macro becomes 48 MHz. (See Section 13.4.)

When the main clock frequency is 24 MHz, (FSEL1, FSEL0) = (0, 0)

When the main clock frequency is 12 MHz, (FSEL1, FSEL0) = (0, 1)

When the main clock frequency is 16 MHz, (FSEL1, FSEL0) = (1, 0)

Further, in order to use the clock (48 MHz) generated by frequency multiplication by the PLL circuit inside the USB macro, it is required to set bit 5 (PLL Enable) of the system control register (SYSCON) in the USB macro to "1". If this setting is not made, the main clock will be used as it is inside the USB macro and the USB functions will not operate correctly. (See Chapter 17.)

In order to power down the PLL circuit, either –

1) set the oscillation test register (OSCTEST) inside the USB macro to PLL power-down, or

2) use the function inside the USB macro of detecting the suspend conditions and automatically powering down the PLL circuit. Further, when the resume conditions are detected, it is possible to automatically return the PLL to normal operation.

Chapter 7

Time Base Counter (TBC)

7. Time Base Counter (TBC)

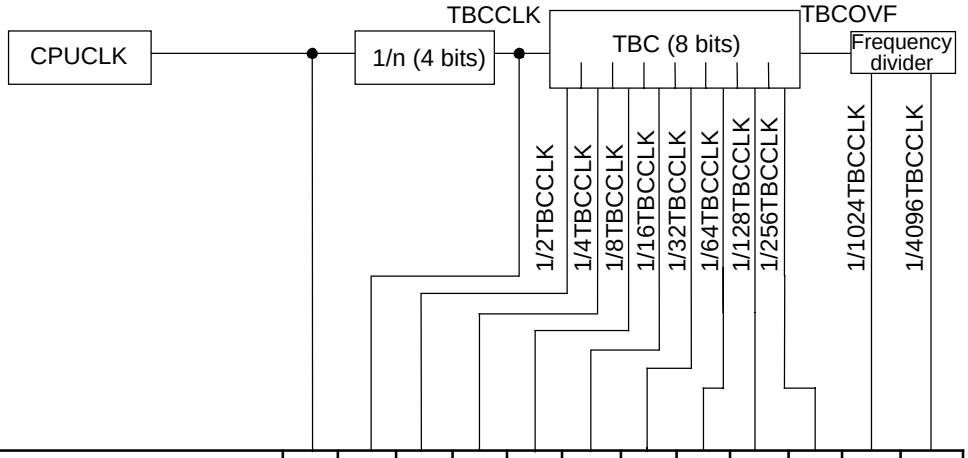
7.1 Overview

The ML66525 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	○		○	○								
SIO3				○		○		○				
SIO4	○		○	○	○							
16-bit timer 0		○	○	○	○	○	○	○				
8-bit timer 3/BRG		○	○	○	○		○			○		
8-bit timer 4/BRG		○	○	○	○		○			○		
8-bit timer 5/BRG		○	○	○	○		○			○		
8-bit timer 6/WDT					○	○	○	○	○	○	○	○
16-bit timer 7		○	○	○	○	○	○	○	○			
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 dividing counter	—		R	16	F0	7-2

[Notes]

1. A star (☆) in the address column indicates a missing bit.
2. For details, refer to Chapter 22, "Special Function Registers (SFRs)".

7.4 1/n Counter

To generate base clocks for internal peripheral modules from the wide ranging CPUCLK frequency, the ML66525 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 (RESn 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.

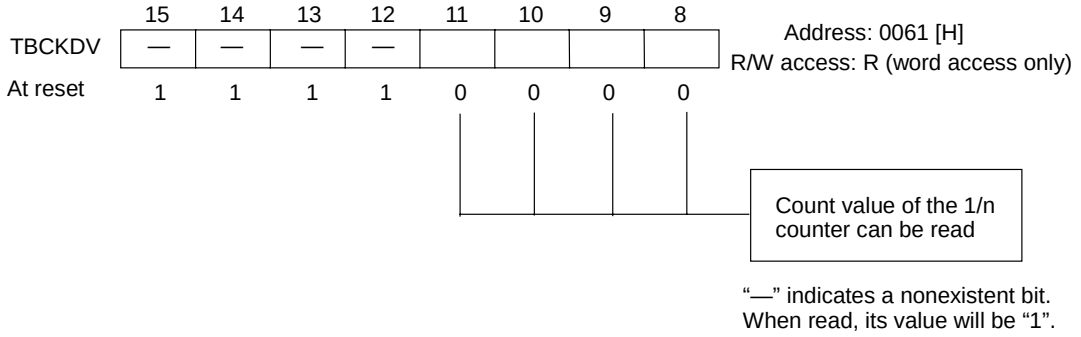


Figure 7-2 Configuration of Upper 8 Bits of TBCKDV

(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 (RESn 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.

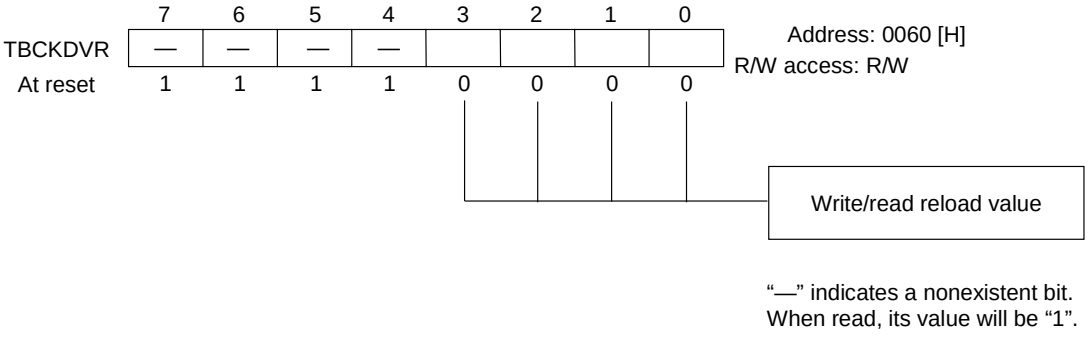


Figure 7-3 TBCKDVR (Lower 8 Bits of TBCKDV) Configuration

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 (RESn 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 (RESn 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 ML66525 family has the following internal general-purpose timers: two 16-bit auto-reload timers (timers 0 and 7), one 8-bit auto-reload timers (timer 9), three 8-bit auto-reload timers that also function as serial communication baud rate generators (timers 3,4 and 5), and an 8-bit auto-reload timer that also functions as a watchdog timer (timer 6).

8.2 General-Purpose 8-Bit/16-Bit Timer Configurations

Table 8-1 lists a summary of the function of each general-purpose timer. Circles (○) 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 3	8	○	—	—	—	○ (SIO6)	—
Timer 4	8	○	—	—	—	○ (SIO1)	—
Timer 5	8	○	—	—	—	○ (SIO3, 4)	—
Timer 6	8	○	—	—	—	—	○
Timer 7	16	○	—	—	—	—	—
Timer 9	8	○	—	—	○	—	—

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	General-purpose 16-bit timer 0 counter	—	TM0C	R/W	16	Undefined	8-4
0063							
0064	General-purpose 16-bit timer 0 register	—	TM0R	R/W	16	Undefined	8-4
0065							
0066☆	General-purpose 16-bit timer 0 control register	TM0CON	—	R/W	16	70	8-4
0070	General-purpose 8-bit timer 3 counter	TM3C	—	R/W	8	Undefined	8-10
0071	General-purpose 8-bit timer 3 register	TM3R	—	R/W	8	Undefined	8-10
0072☆	General-purpose 8-bit timer 3 control register	TM3CON	—	R/W	8	70	8-10
0074	General-purpose 8-bit timer 4 counter	TM4C	—	R/W	8	Undefined	8-16
0075	General-purpose 8-bit timer 4 register	TM4R	—	R/W	8	Undefined	8-16
0076☆	General-purpose 8-bit timer 4 control register	TM4CON	—	R/W	8	70	8-16
0078	General-purpose 8-bit timer 5 counter	TM5C	—	R/W	8	Undefined	8-22
0079	General-purpose 8-bit timer 5 register	TM5R	—	R/W	8	Undefined	8-22
007A☆	General-purpose 8-bit timer 5 control register	TM5CON	—	R/W	8	70	8-22
007C	General-purpose 8-bit timer 6 counter	TM6C	—	R/W	8	Undefined	8-28
007D	General-purpose 8-bit timer 6 register	TM6R	—	R/W	8	Undefined	8-28
007E☆	General-purpose 8-bit timer 6 control register	TM6CON	—	R/W	8	10	8-29
00CC	General-purpose 8-bit timer 9 counter	TM9C	—	R/W	8	Undefined	8-37
00CD	General-purpose 8-bit timer 9 register	TM9R	—	R/W	8	Undefined	8-37
00CE☆	General-purpose 8-bit timer 9 control register	TM9CON	—	R/W	8	70	8-37
00D0	General-purpose 16-bit timer 7 counter	—	TM7C	R/W	16	Undefined	8-43
00D1							
00D2	General-purpose 16-bit timer 7 register	—	TM7R	R/W	16	Undefined	8-43
00D3							
00D4☆	General-purpose 16-bit timer 7 control register	TM7CON	—	R/W	8	70	8-43

[Notes]

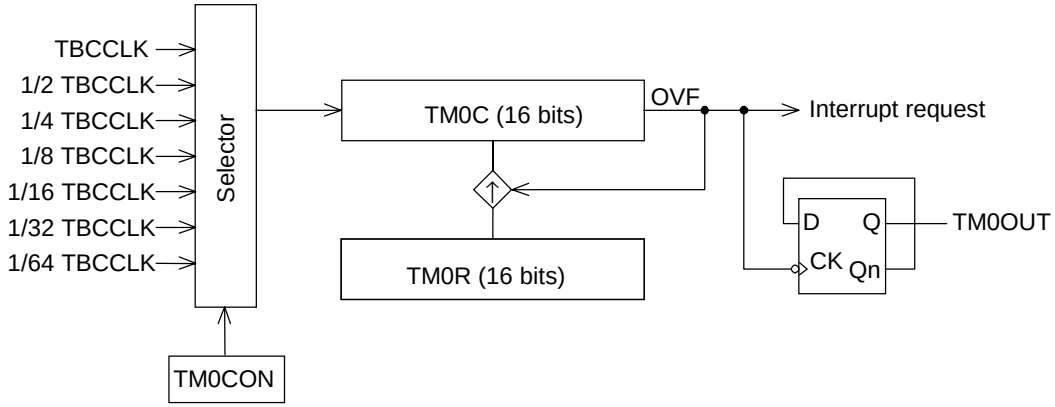
1. Addresses are not consecutive in some places.
2. A star (☆) in the address column indicates a missing bit.
3. For details, refer to Chapter 22, "Special Function Registers (SFRs)".

8.4 Timer 0

Timer 0 is a 16-bit auto-reload timer.

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
TM0OUT: Timer 0 output

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 (RESn 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 (RESn 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 (RESn 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.

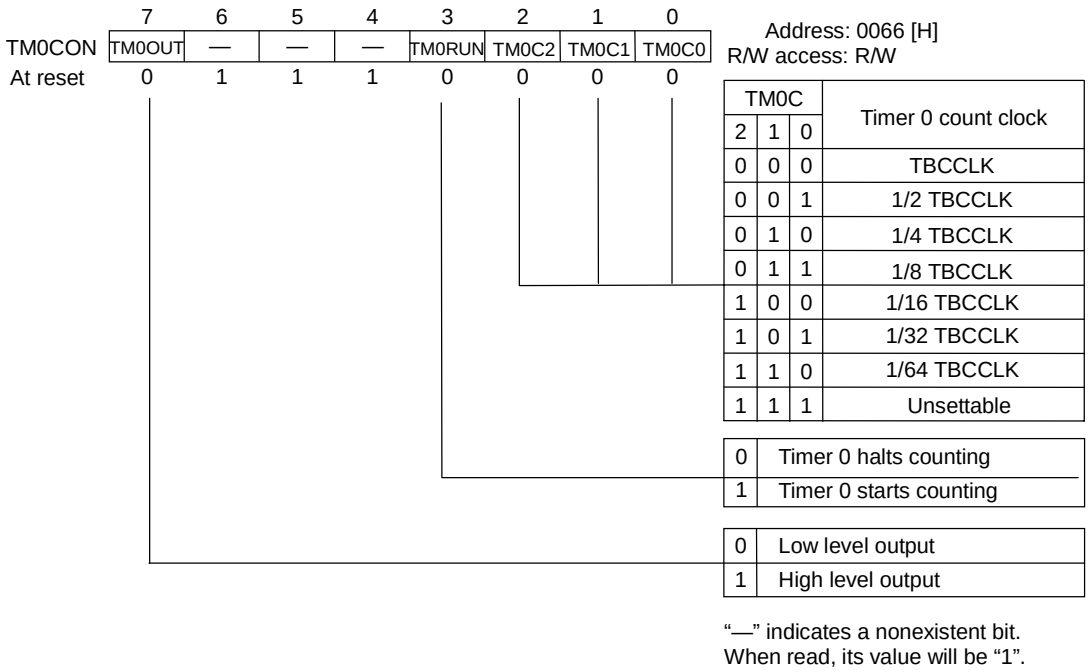


Figure 8-2 TM0CON Configuration

[Note]

Do not select "Unsettable" for timer 0 count clock.

If you select "Unsettable", timer 0 will not operate normally.

8.4.3 Example of Timer 0-related Register Settings

(1) 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).

(2) 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.

(3) 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. 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 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 1/1 and 1/4 TBCCLK).

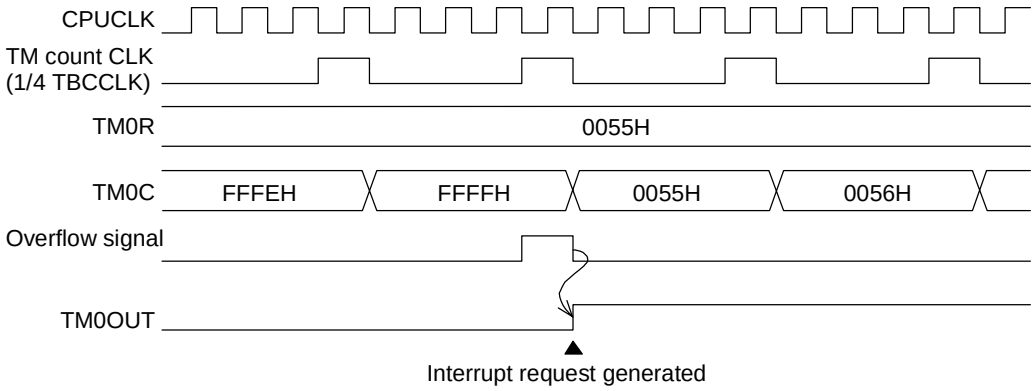


Figure 8-3 Timer 0 Operation

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 of registers that contain interrupt processing flags		IRQ1	IE1	IP2	
	Reference page	15-13	15-18	15-23	

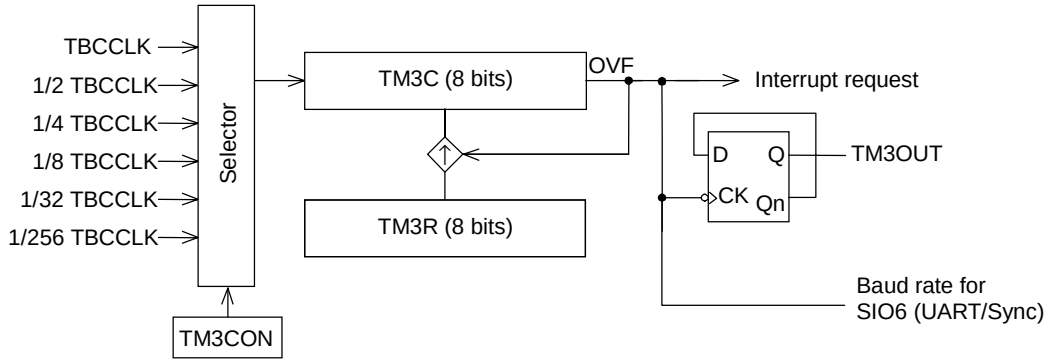
For further details regarding interrupt processing, refer to Chapter 15, "Interrupt Processing Functions".

8.5 Timer 3

Timer 3 is an 8-bit auto-reload timer that has a function for a serial communication baud rate generator for SIO6.

8.5.1 Timer 3 Configuration

Figure 8-4 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
TM3OUT: Timer 3 output

Figure 8-4 Timer 3 Configuration

8.5.2 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 (RESn 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 (RESn 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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), TM3CON becomes 70H.

Figure 8-5 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.

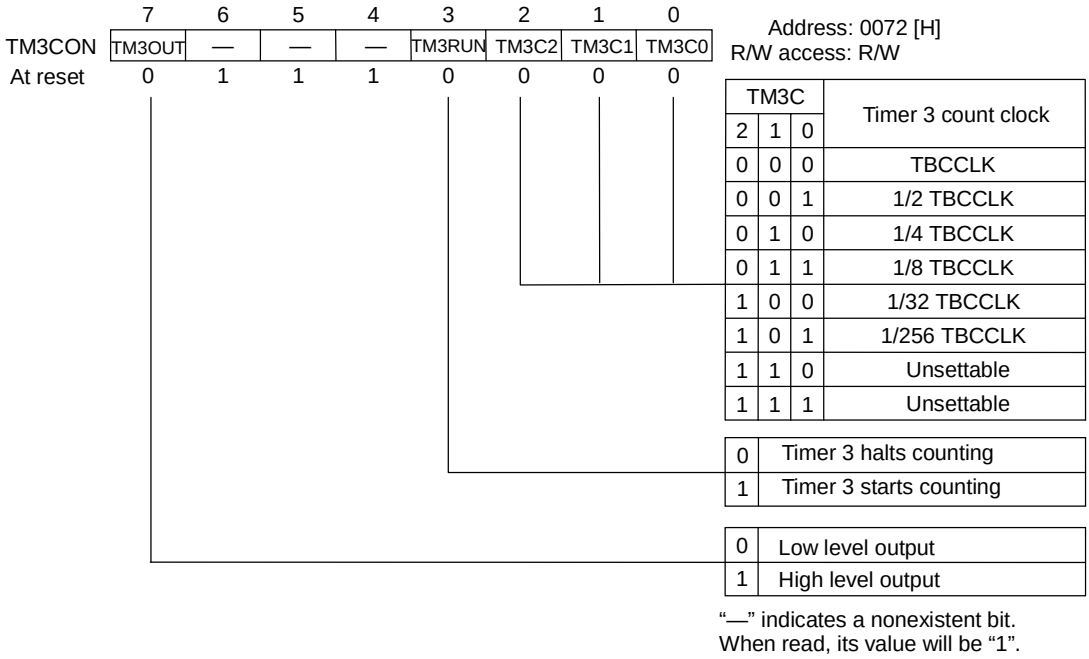


Figure 8-5 TM3CON Configuration

[Note]

Do not select "Unsettable" for timer 3 count clock.

If you select "Unsettable", timer 3 will not operate normally.

8.5.3 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.

8.5.4 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 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 8-6 shows an operation example (for settings of 1/n counter frequency division ratio 1/1 and 1/4 TBCCLK).

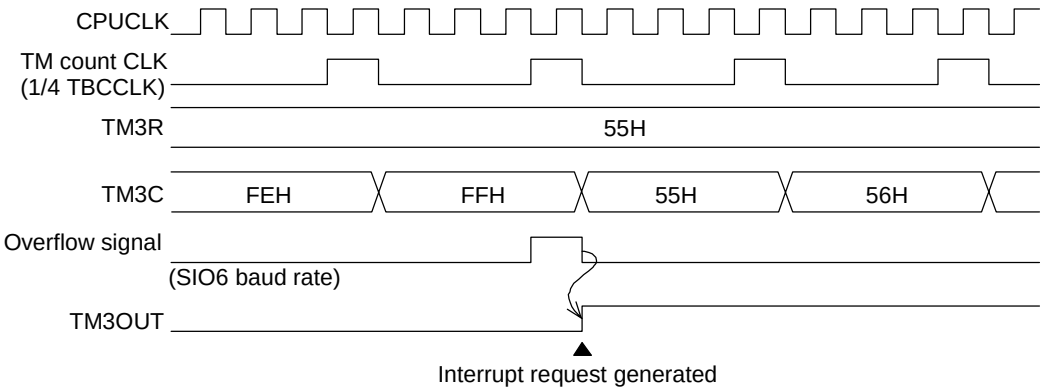


Figure 8-6 Timer 3 Operation Example

8.5.5 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 (ETM0OV) 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 8-4 lists the vector address of the timer 3 interrupt factor and the interrupt processing flags.

Table 8-4 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 of registers that contain interrupt processing flags		IRQ1	IE1	IP3	
	Reference page	15-13	15-18	15-24	

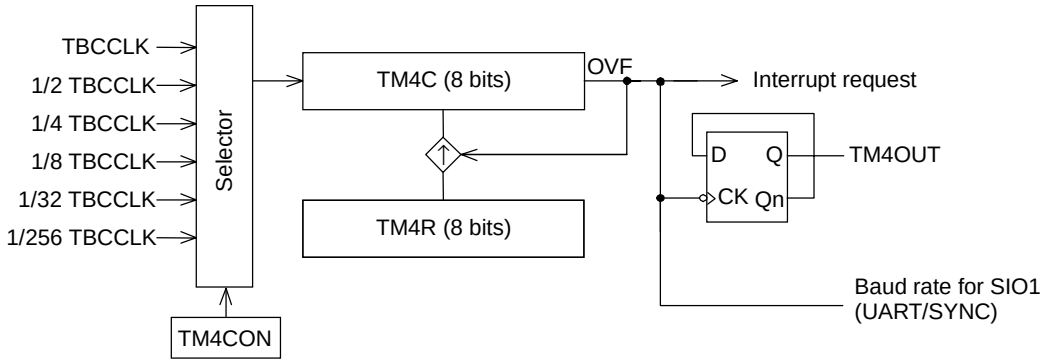
For further details regarding interrupt processing, refer to Chapter 15, "Interrupt Processing Functions".

8.6 Timer 4

Timer 4 is an 8-bit auto-reload timer that has functions for timer output and a serial communication baud rate generator for SIO1.

8.6.1 Timer 4 Configuration

Figure 8-7 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

Figure 8-7 Timer 4 Configuration

8.6.2 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 (RESn 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 (RESn 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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), TM4CON becomes 70H.

Figure 8-8 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.

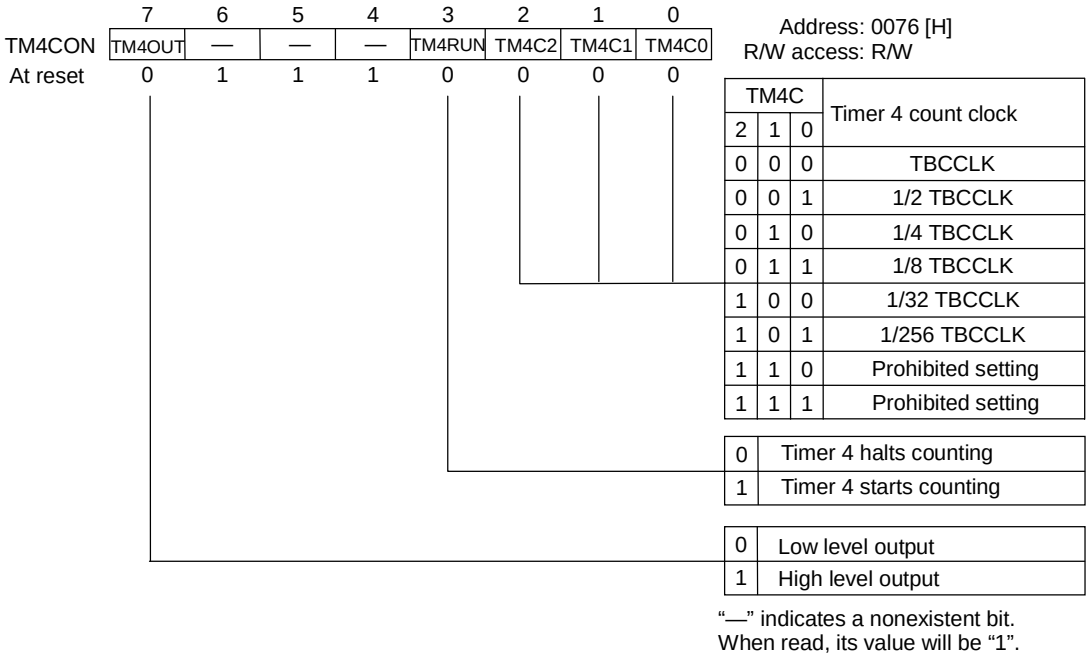


Figure 8-8 TM4CON Configuration

[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.

8.6.3 Example of Timer 4-related Register 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 written simultaneously and automatically to the general-purpose 8-bit timer 4 register (TM4R).

(2) 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.

(3) 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.

8.6.4 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 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 8-9 shows an operation example (for settings of 1/n counter frequency division ratio 1/1 and 1/4 TBCCLK).

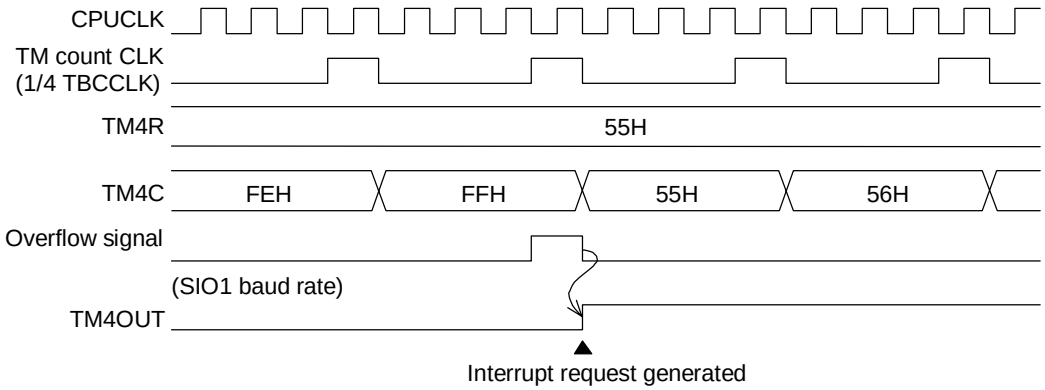


Figure 8-9 Timer 4 Operation Example

8.6.5 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 8-5 lists the vector address of the timer 4 interrupt factor and the interrupt processing flags.

Table 8-5 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 of registers that contain interrupt processing flags		IRQ2	IE2	IP5	
	Reference page	15-14	15-19	15-26	

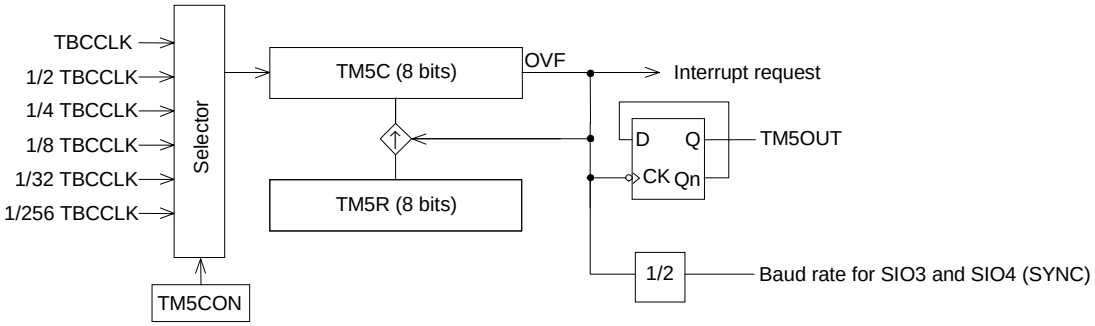
For further details regarding interrupt processing, refer to Chapter 15, "Interrupt Processing Functions".

8.7 Timer 5

Timer 5 is an 8-bit auto-reload timer that has a function for a serial communication baud rate generator for SIO3 and SIO4.

8.7.1 Timer 5 Configuration

Figure 8-10 shows the timer 5 configuration.



TM5C: General-purpose 8-bit timer 0 counter
TM5R: General-purpose 8-bit timer 0 register
TM5CON: General-purpose 8-bit timer 0 control register
TM5OUT: Timer 5 output
1/2: 1/2 dividing circuit

Figure 8-10 Timer 5 Configuration

8.7.2 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 SIO3 and SIO4.

TM5C can be read from and written to by the program.

When reset (RESn 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 (RESn 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, bit 3 (TM5RUN) starts or halts the counting, and bit 7 (TM5OUT) specifies the initial (when activated) timer output level ("H" or "L" level). The contents of bit 7 (TM5OUT) are inverted 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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of TM5CON are 70H.

Figure 8-11 shows the TM5CON configuration.

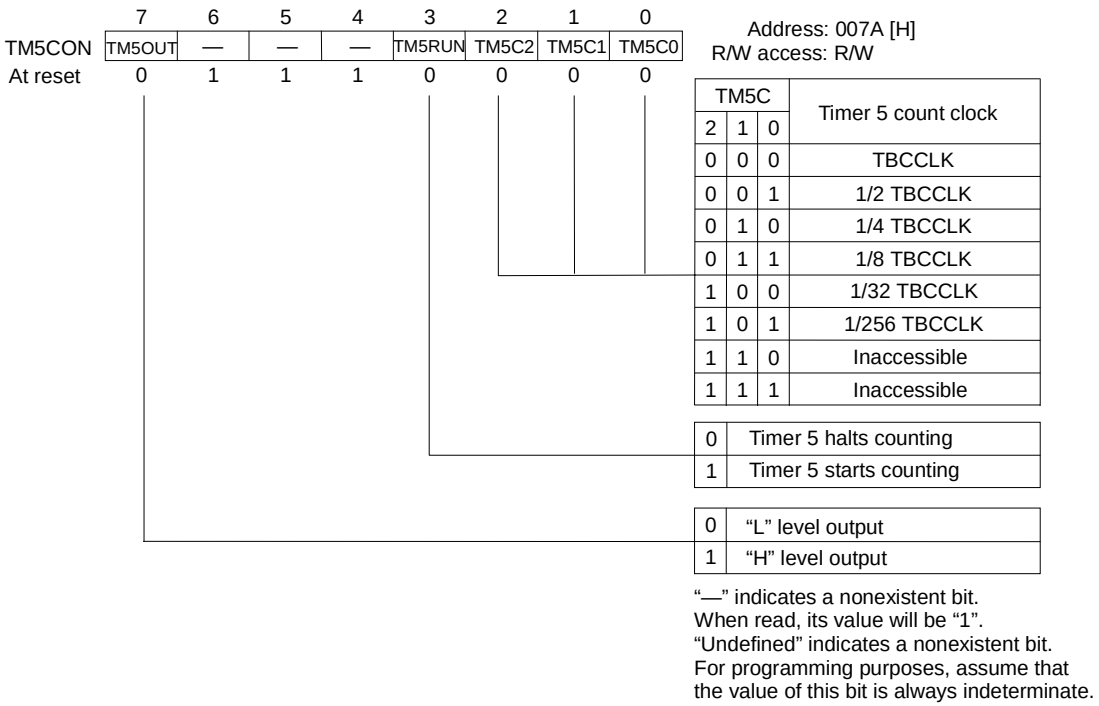


Figure 8-11 TM5CON Configuration

[Note]

Do not select "Inaccessible" for timer 5 count clock.

If you select "Inaccessible", timer 5 will not operate normally.

8.7.3 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, bit 7 (TM5OUT) specifies the initial value. If bit 3 (TM5RUN) is set to "1", timer 5 will begin counting. If reset to "0", timer 5 will halt counting.

8.7.4 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, and TM5OUT is inversely output. Bit 7 (TM5OUT) of TM5CON specifies the initial value of TM5OUT output. This operation is repeated until the TM5RUN bit is reset to "0". Overflow of TM5C can be used as a baud rate generator for SIO3 and SIO4. Figure 8-12 shows an operation example (for settings of 1/n counter frequency division ratio 1/1 and 1/4 TBCCLK).

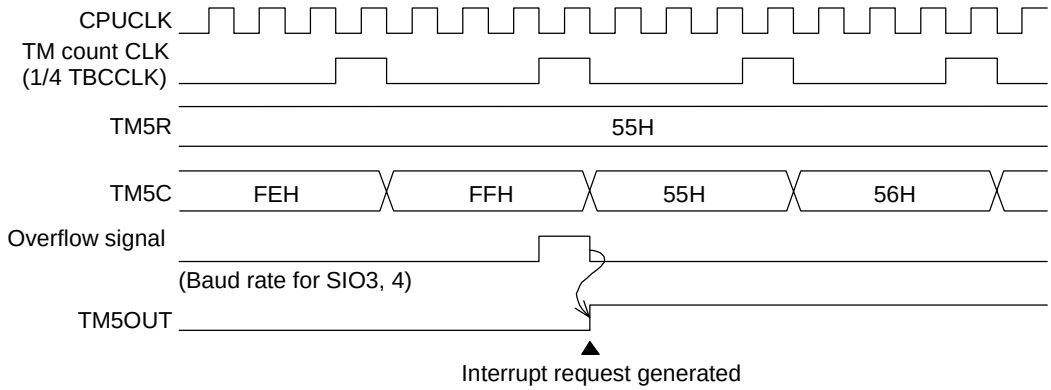


Figure 8-12 Timer 5 Operation Example

8.7.5 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 8-6 lists the vector address of the timer 5 interrupt factor and the interrupt processing flags.

Table 8-6 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 of registers that contain interrupt processing flags		IRQ3	IE3	IP6	
	Reference page	15-15	15-20	15-27	

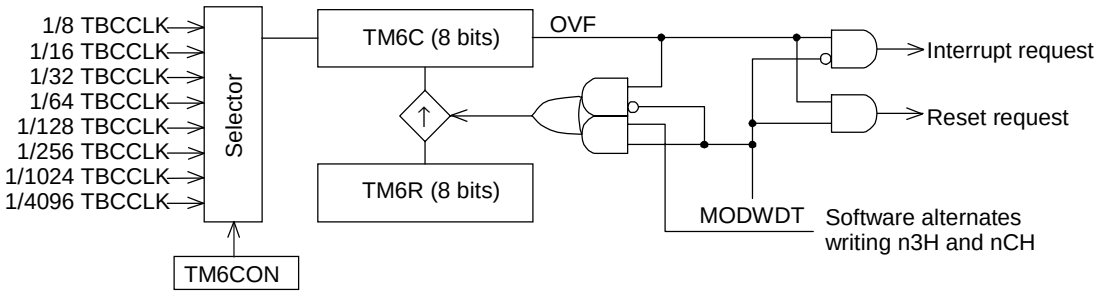
For further details regarding interrupt processing, refer to Chapter 15, "Interrupt Processing Functions".

8.8 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.

8.8.1 Timer 6 Configuration

Figure 8-13 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 8-13 Timer 6 Configuration

8.8.2 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 RESn 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 RESn 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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), TM6CON becomes 10H.

Figure 8-14 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.

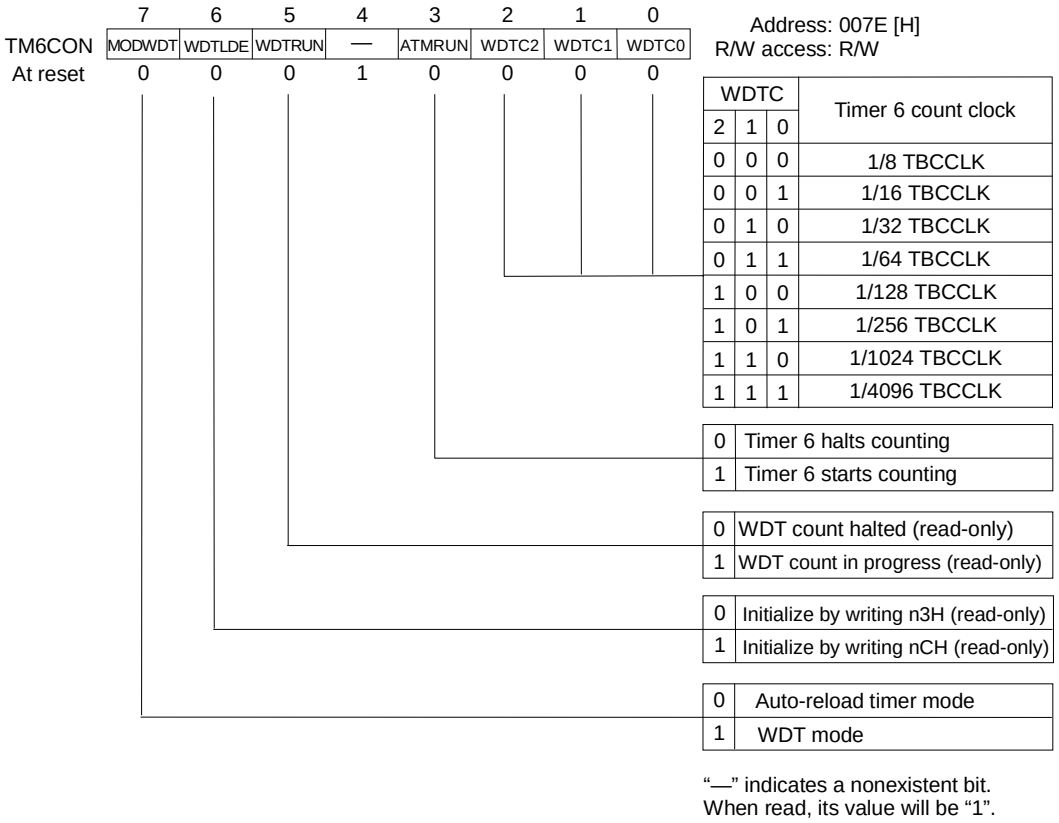


Figure 8-14 TM6CON Configuration

8.8.3 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.)

8.8.4 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 8-15 shows an operation example (for settings of 1/n counter frequency division ratio 1/1 and 1/8 TBCCLK).

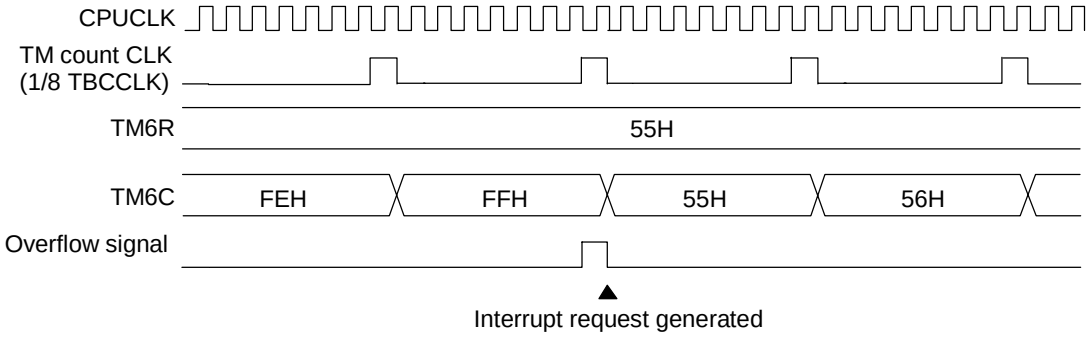


Figure 8-15 Timer 6 Operation (During Auto-Reload Timer Mode)

- Watchdog timer (WDT) mode

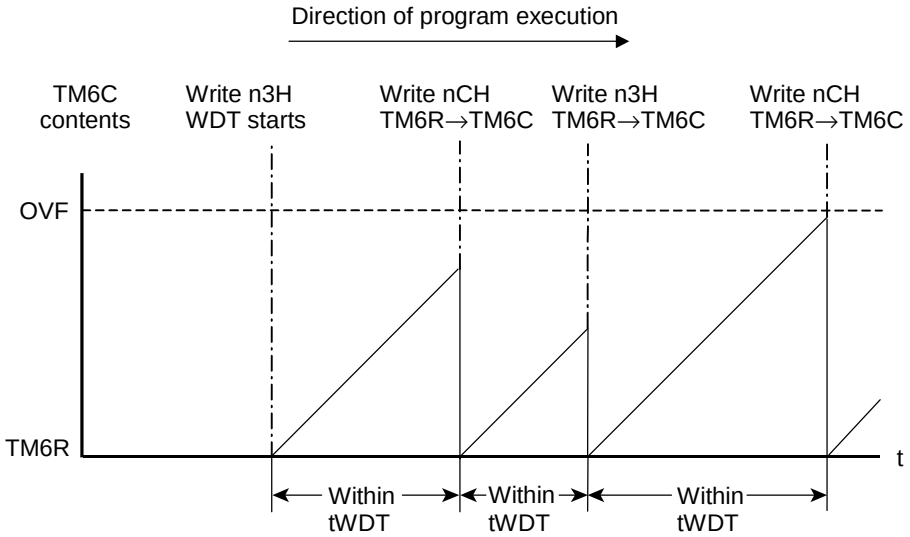
  When the MODWDT bit in TM6CON is set to "1", the mode changes to the WDT mode. Once the WDT mode is set, it is not possible to return to the auto-reload timer mode until the system is reset. In the WDT mode, writing "n3H" to TM6C will cause the WDT count operation to begin. Thereafter, alternately writing "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.

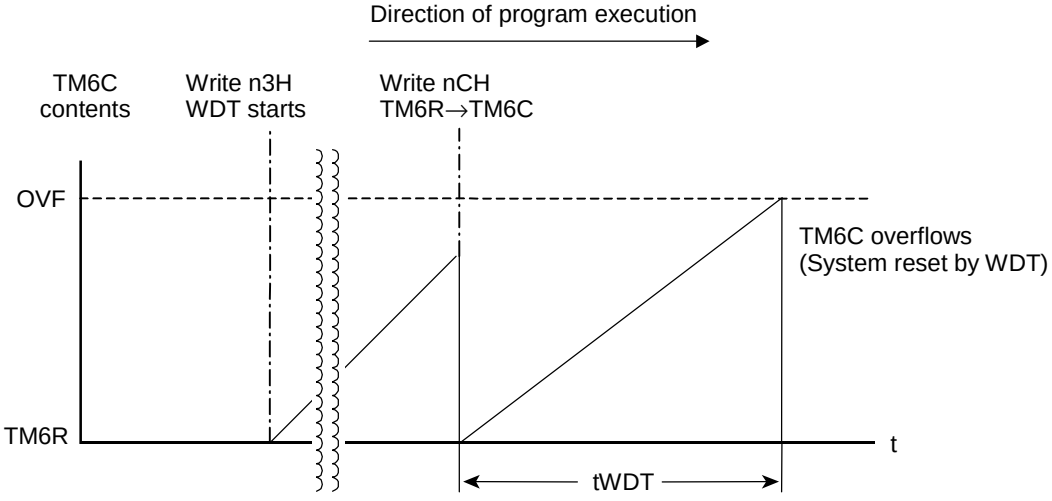
  The time (tWDT) until TM6C overflows can be expressed by the below equation, where f [MHz] is the fundamental clock (CPUCLK), T is the TM6C count clock (divided value of TBCCLK), n is the divisor for the 1/n counter at the TBC front stage, and R is the value of TM6R.

$$t_{WDT} = (1/f) \times T \times n \times (256 - R) \quad [\mu s] \quad (R: 0 \text{ to } 255)$$

  Figure 8-16 shows timing diagrams of an out-of-control program and detection by WDT. Figure 8-17 shows an example of an out-of-control program.



(a) Correct program execution



(b) Program runs out-of-control

Figure 8-16 Timing Diagram of Out-of-Control Program Detection

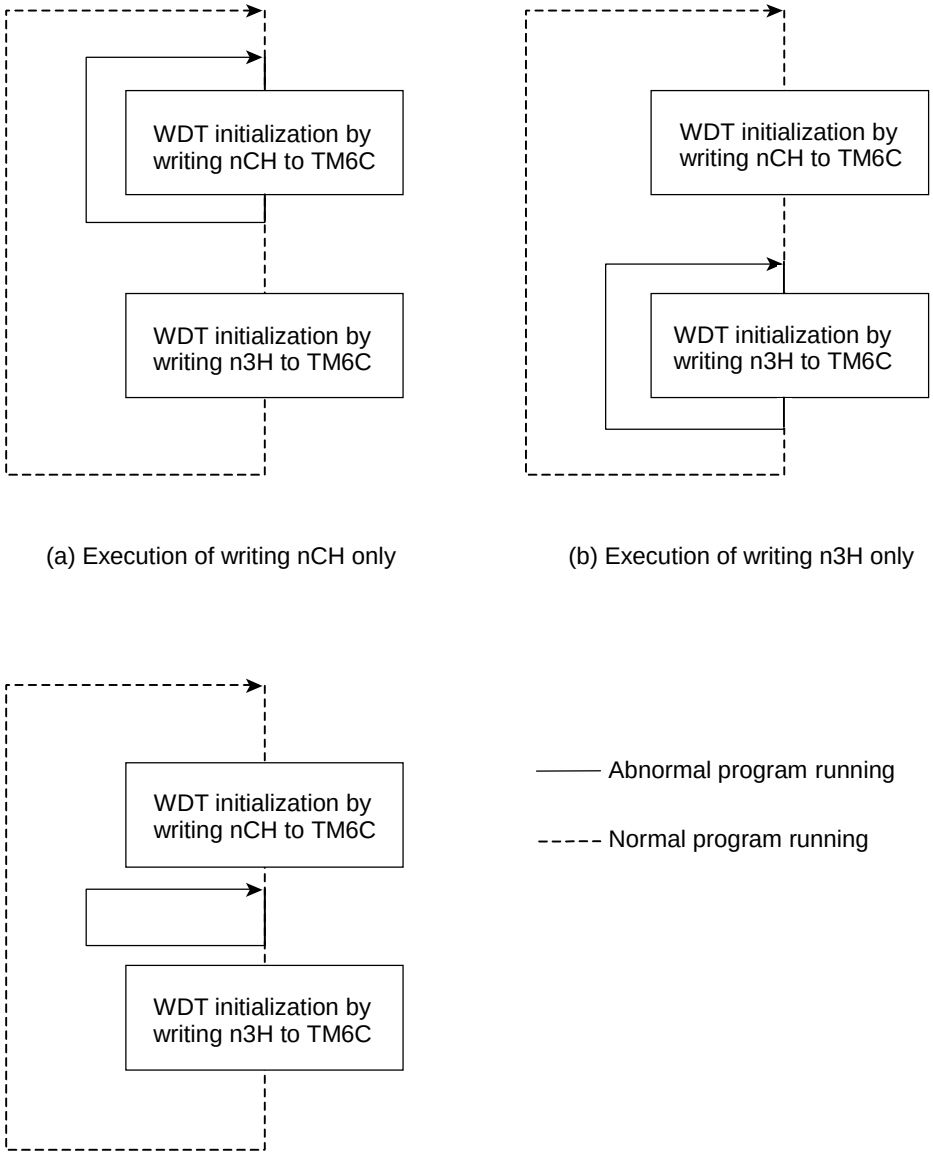


Figure 8-17 Example of Out-of-Control Program Detection

8.8.5 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 8-7 lists the vector address of the timer 6 interrupt factor and the interrupt processing flags.

Table 8-7 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 of registers that contain interrupt processing flags		IRQ3	IE3	IP7	
	Reference page	15-15	15-20	15-28	

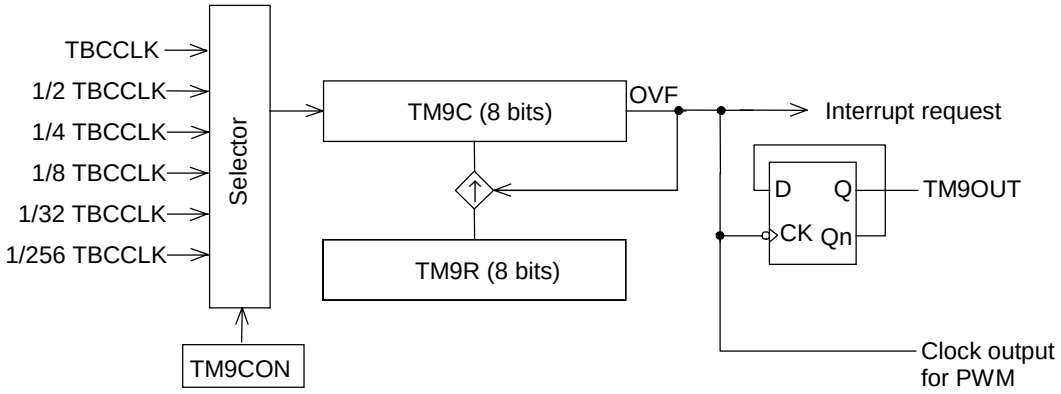
For further details regarding interrupt processing, refer to Chapter 15, "Interrupt Processing Functions".

8.9 Timer 9

Timer 9 is an 8-bit auto-reload timer.

8.9.1 Timer 9 Configuration

Figure 8-18 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
TM9OUT: Timer 9 output

Figure 8-18 Timer 9 Configuration

8.9.2 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 (RESn 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 (RESn 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) starts or halts 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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), TM9CON becomes 70H.

Figure 8-16 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.

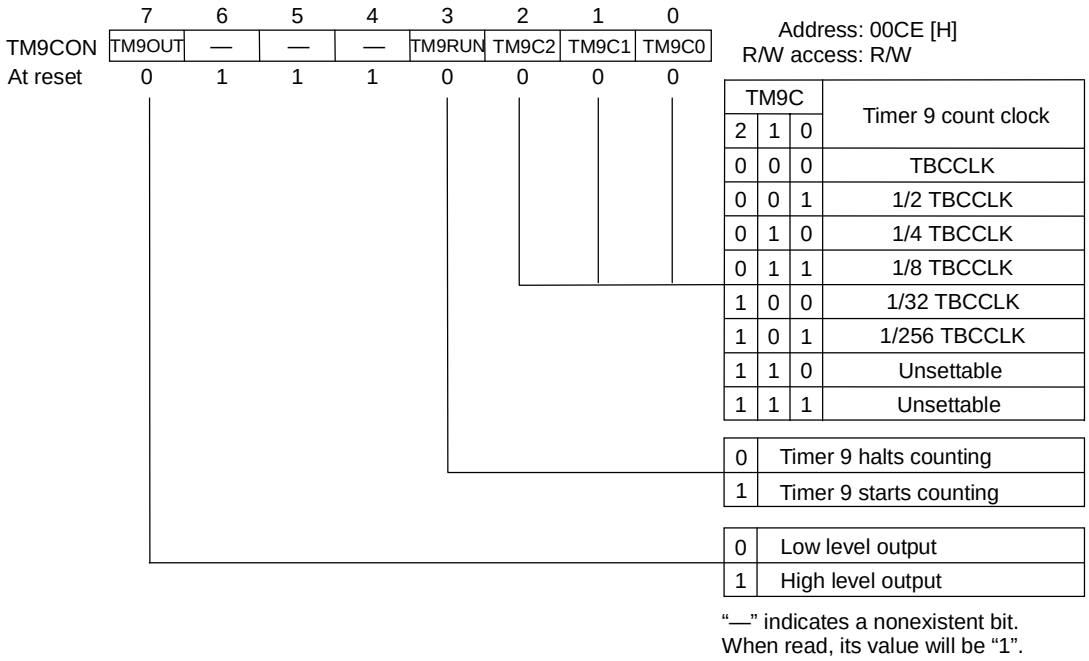


Figure 8-19 TM9CON Configuration

[Note]

Do not select "Unsettable" for timer 9 count clock.

If you select "Unsettable", timer 9 will not operate normally.

8.9.3 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.

8.9.4 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 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 8-20 shows an operation example (for settings of 1/n counter frequency division ratio 1/1 and 1/4 TBCCLK).

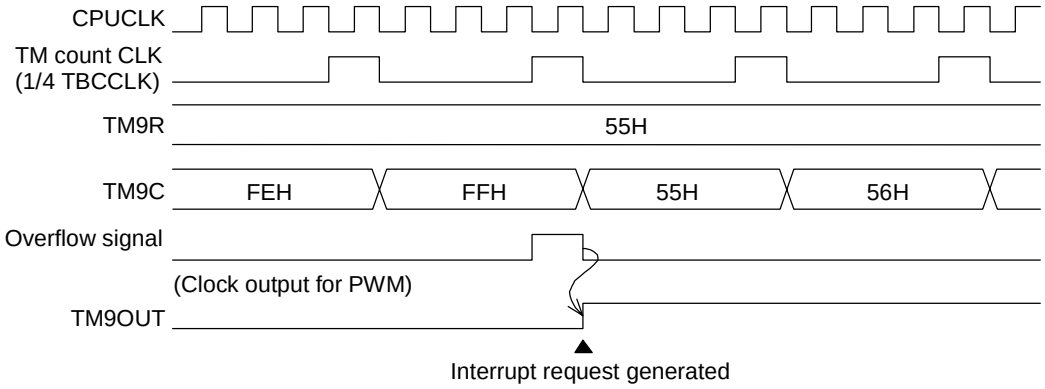


Figure 8-20 Timer 9 Operation Example

8.9.5 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 8-8 lists the vector address of the timer 9 interrupt factor and the interrupt processing flags.

Table 8-8 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 of registers that contain interrupt processing flags		IRQ4	IE4	IP9	
	Reference page	15-16	15-21	15-30	

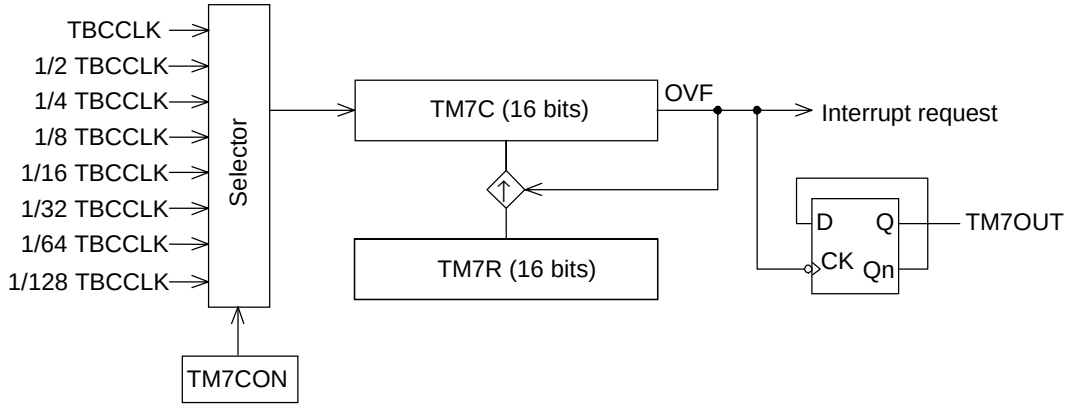
For further details regarding interrupt processing, refer to Chapter 15, "Interrupt Processing Functions".

8.10 Timer 7

Timer 7 is a 16-bit auto-reload timer.

8.10.1 Timer 7 Configuration

Figure 8-21 shows the timer 7 configuration.



TM7C: General-purpose 16-bit timer 7 counter
TM7R: General-purpose 16-bit timer 7 register
TM7CON: General-purpose 16-bit timer 7 control register
TM7OUT: Timer 0 output

Figure 8-21 Timer 7 Configuration

8.10.2 Description of Timer 7 Registers

(1) General-purpose 16-bit timer 7 counter (TM7C)

The general-purpose 16-bit timer 7 counter (TM7C) 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 7 register (TM7R).

TM7C can be read from and written to by the program.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of TM7C are undefined.

[Note]

Writing a timer value to TM7C causes the same value to also be written to the general-purpose 16-bit timer 7 register (TM7R).

(2) General-purpose 16-bit timer 7 register (TM7R)

The general-purpose 16-bit timer 7 register (TM7R) consists of 16 bits. This register stores the value to be reloaded into the general-purpose 16-bit timer 7 counter (TM7C).

TM7R can be read from and written to by the program.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of TM7R are undefined.

(3) General-purpose 16-bit timer 7 control register (TM7CON)

The general-purpose 16-bit timer 7 control register (TM7CON) consists of 5 bits. Bits 0 to 2 (TM7C0 to TM7C2) of TM7CON select the timer 0 count clock, bit 3 (TM7RUN) starts or halts the counting, and bit 7 (TM7OUT) specifies the initial timer output level (High or Low) at start-up. And each time TM7C overflows, the content of bit 7 (TM7OUT) is reversed.

TM7CON 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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), TM7CON becomes 70H.

Figure 8-2 shows the TM7CON configuration.

[Note]

Just before TM7C overflows, if an SB, RB, XORB or other read-modify-write instruction is performed on TM7CON, then TM7OUT may not operate correctly.

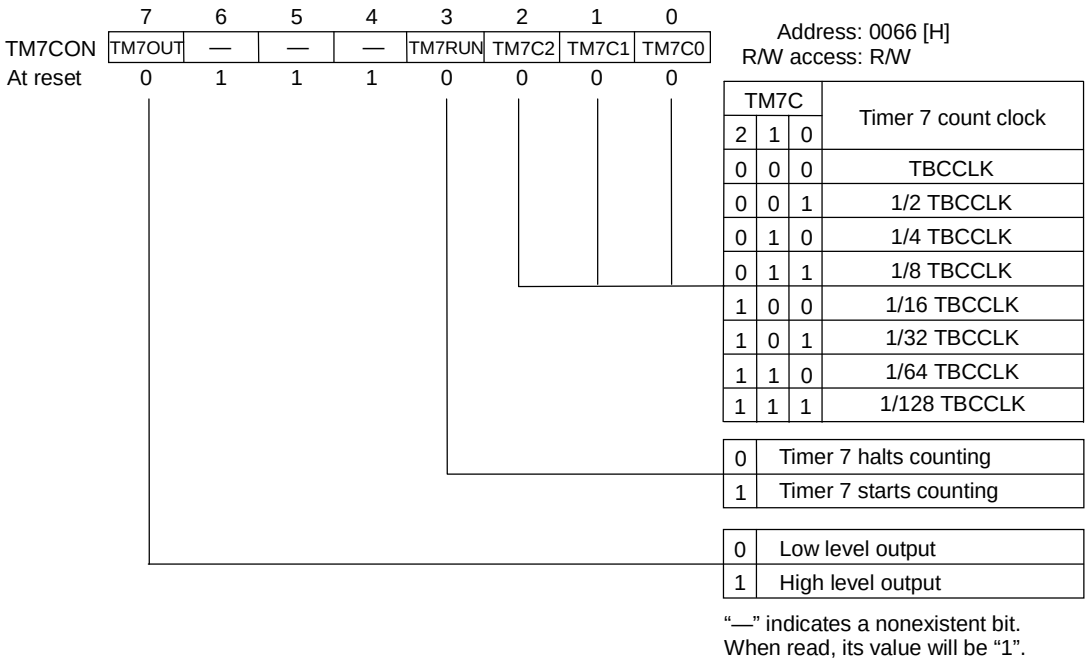


Figure 8-22 TM7CON Configuration

8.10.3 Example of Timer 7-related Register Settings

(1) General-purpose 16-bit timer 7 counter (TM7C)
Set the timer value that will be valid at the start of counting. When writing to TM7C, the same value will also be simultaneously and automatically written to the general-purpose 16-bit timer 7 register (TM7R).

(2) General-purpose 16-bit timer 7 register (TM7R)
This register sets the value to be loaded after general-purpose 16-bit timer 7 counter (TM7C) overflows. If the timer value (TM7C) and the reload value (TM7R) are identical, this register will automatically be set just by setting TM7C. If the values are different or are to be modified, this register must be set explicitly.

(3) General-purpose 16-bit timer 7 control register (TM7CON)
Bits 0 to 2 (TM7C0 to TM7C2) of this register set the count clock for timer 7. If TM7OUT (timer output) is to be used, specify the initial value with bit 7 (TM0OUT). If bit 3 (TM7RUN) is set to "1", timer 0 will begin counting. If reset to "0", timer 7 will halt counting.

8.10.4 Timer 7 Operation

When the TM7RUN bit is set to "1", timer 0 will begin counting upward, running on the count clock selected by TM7CON. When TM7C overflows, an interrupt request is generated, the contents of TM7R are loaded into TM7C and the TM7OUT output is inverted. The initial value of the TM7OUT is specified by bit 7 (TM7OUT) of TM7CON. This operation is repeated until the TM7RUN bit is reset to "0". Figure 8-3 shows an operation example (for settings of 1/n counter frequency division ratio 1/1 and 1/4 TBCCLK).

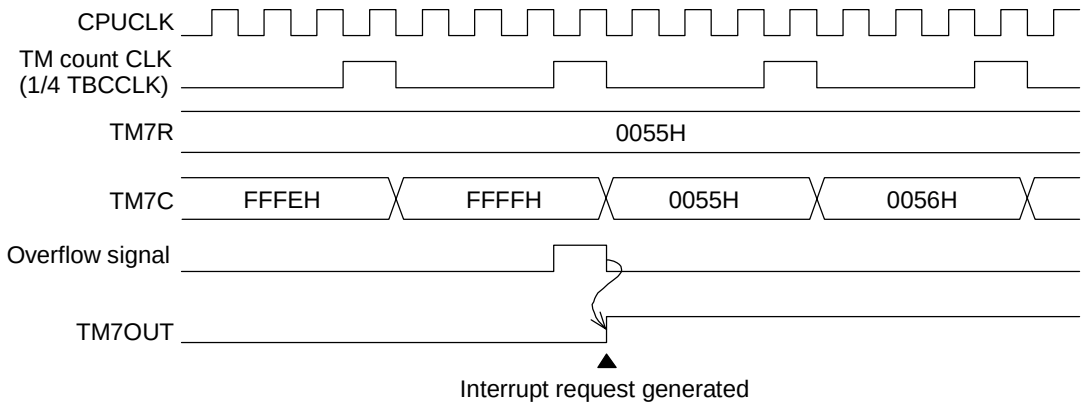


Figure 8-23 Timer 7 Operation

8.10.5 Timer 7 Interrupt

When a timer 7 interrupt factor occurs, the interrupt request flag (QTM7OV) is set to "1". The interrupt request flag (QTM7OV) is located in interrupt request register 1 (IRQ1).

Interrupts can be enabled or disabled by the interrupt enable flag (ETM7OV). The interrupt enable flag (ETM7OV) is located in interrupt enable register 1 (IE1).

Three levels of priority can be set with the interrupt priority setting flags (P0TM7OV and P1TM7OV). The interrupt priority setting flags are located in interrupt priority control register 2 (IP2).

Table 8-9 lists the vector address of the timer 7 interrupt factor and the interrupt processing flags.

Table 8-9 Timer 7 Vector Address and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	0
Overflow of timer 7	0032	QTM7OV	ETM7OV	P1TM7OV	P0TM7OV
Symbols of registers that contain interrupt processing flags		IRQ2	IE2	IP5	
	Reference page	15-14	15-19	15-26	

For further details regarding interrupt processing, refer to Chapter 15, "Interrupt Processing Functions".

Chapter 9

Real-Time Counter (RTC)

9. Real-Time Counter (RTC)

9.1 Overview

The ML66525 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 and HALT modes). The STOP and HALT modes can be released by the real-time counter interrupt.

9.2 Real-Time Counter Configuration

Figure 9-1 shows the real-time counter configuration.

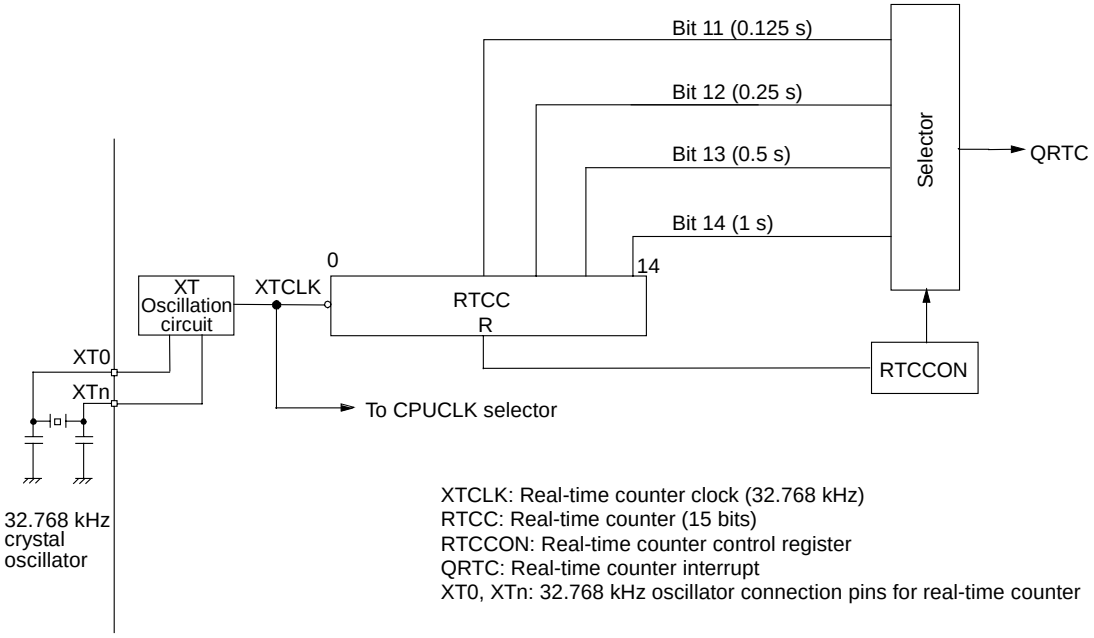


Figure 9-1 Real-Time Counter Configuration

9.3 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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), RTCCON becomes F0H.

Figure 9-2 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.

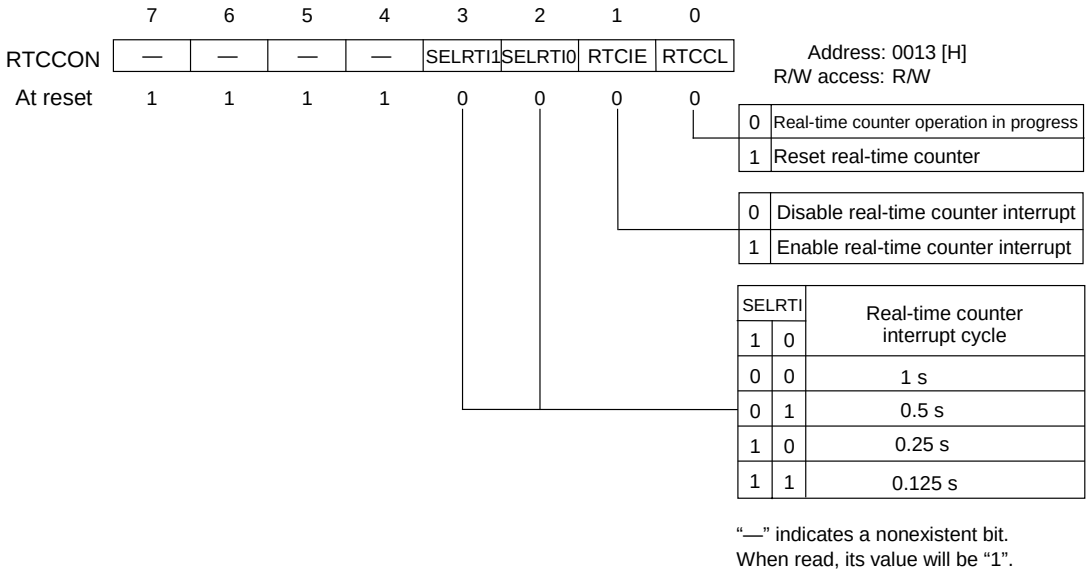


Figure 9-2 RTCCON Configuration

9.4 Example of Real-Time Counter Register Settings

(1) Peripheral Control Register (PRPHCON)

If the real-time counter 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.

9.5 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 (RESn 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".

9.6 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 9-1 lists the vector address of the real-time counter interrupt factor and the interrupt processing flags.

Table 9-1 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 of registers that contain interrupt processing flags		IRQ3	IE3	IP7	
	Reference page	15-15	15-20	15-28	

For further details regarding interrupt processing, refer to Chapter 15, "Interrupt Processing Functions".

Chapter 10

PWM Function

10. PWM Function

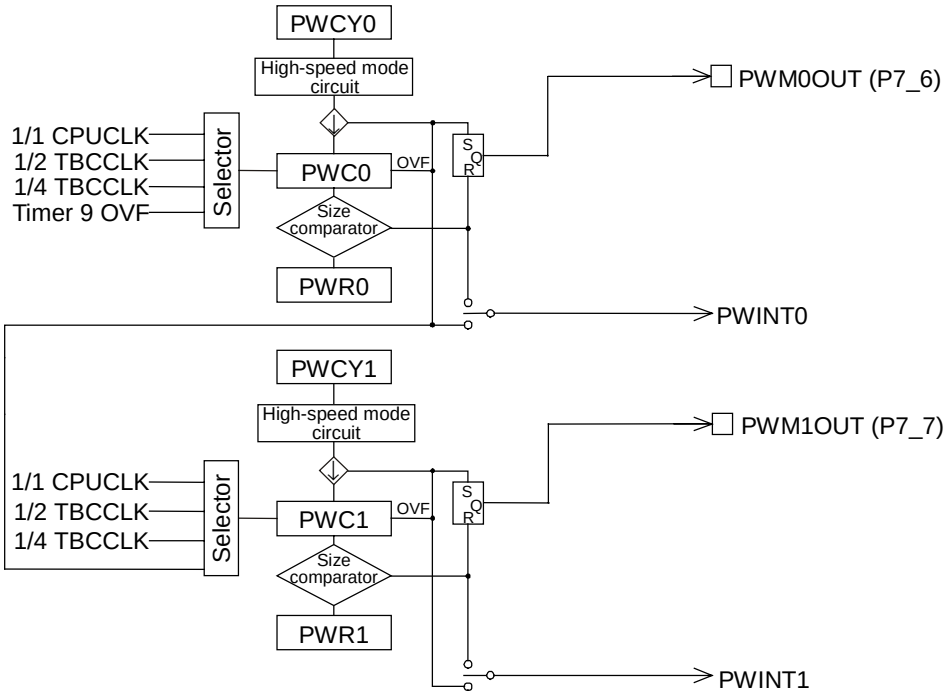
10.1 Overview

The ML66525 family contains 2 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 1 channel 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.

10.2 PWM Configuration

The ML66525 family has two sets of 8-bit PWMs (8-bit PWM0 and 8-bit PWM1). These can be cascaded and used as 16-bit PWM (16-bit mode).

Figure 10-1 shows the PWM configuration.



PWCY0, PWCY1: PWM cycle register (8 bits)
PWC0, PWC1: PWM counter (8 bits)
PWR0, PWR1: PWM register (8 bits)
PWM0OUT, PWM1OUT: PWM output pin
PWMINT0, PWMINT1: Interrupt request

Figure 10-1 PWM Configuration

10.3 PWM Register

Table 10-1 lists a summary of SFRs for PWM control.

Table 10-1 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	10-4
0091	PWM register 1	PWR1				00	
0094	PWM cycle register 0	PWCY0	PWCY	R/W	8/16	00	10-3
0095	PWM cycle register 1	PWCY1				00	
0096	PWM counter 0	PWC0	PWC	R/W	8/16	00	10-3
0097	PWM counter 1	PWC1				00	
0098	PWM control register 0	PWCON0	—	R/W	8	00	10-4
0099☆	PWM control register 1	PWCON1	—	R/W	8	FE	10-6

[Notes]

1. Addresses are not always consecutive.
2. A star (☆) in the address column indicates a missing bit.
3. For details, refer to Chapter 22, "Special Function Registers (SFRs)".

10.3.1 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 (RESn 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 (RESn 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. 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.

(3) PWM registers (PWR0 and PWR1)

The PWM registers (PWR0 and PWR1) are 8-bit registers that set the duty value. The duty value setting for PWR0 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 and PWR1 can be read from and written to by the program. PWR0 and PWR1 can also be accessed as the 16-bit PWR01. During a 16-bit access, PWR1 is the upper 8 bits and PWR0 is the lower 8 bits of PWR01.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), PWR0 and PWR1 become 00H.

[Note]

During the 16-bit mode, the duty value set by PWR01 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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), PWCON0 becomes 00H.

Figure 10-2 shows the PWCON0 configuration.

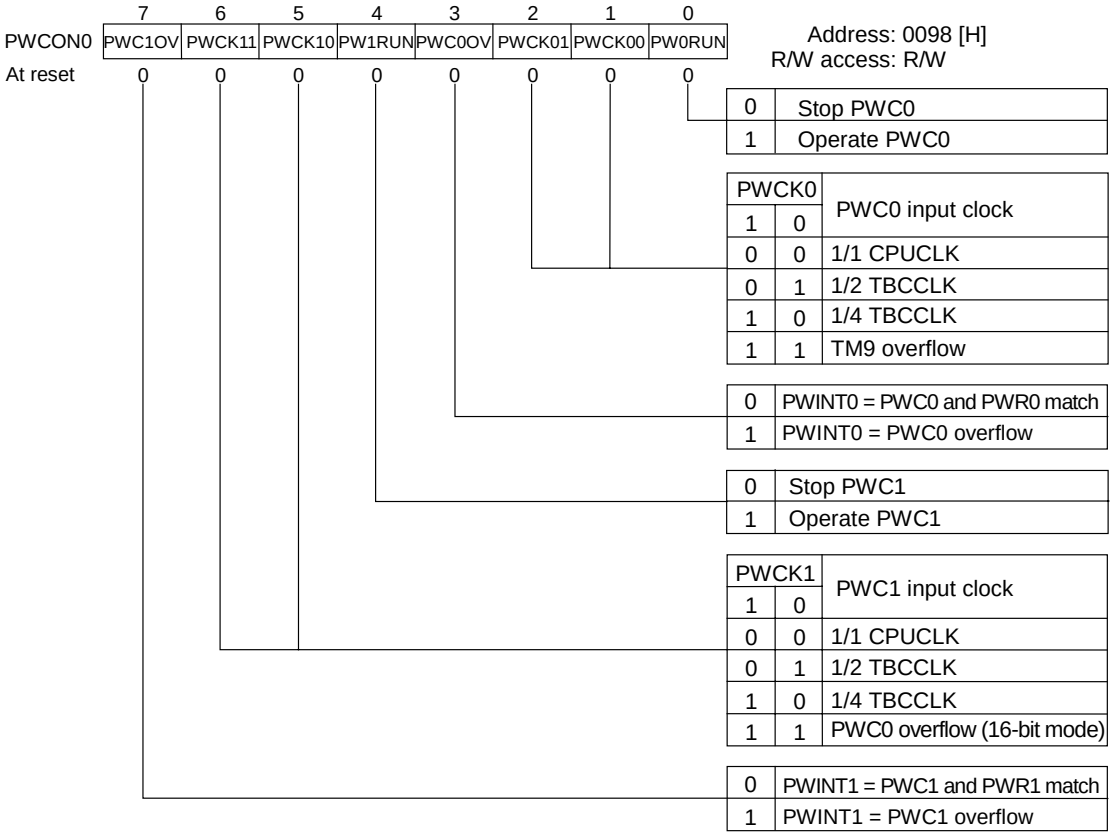


Figure 10-2 PWCON0 Configuration

(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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), PWCON1 becomes FEH.

Figure 10-3 shows the PWCON1 configuration.

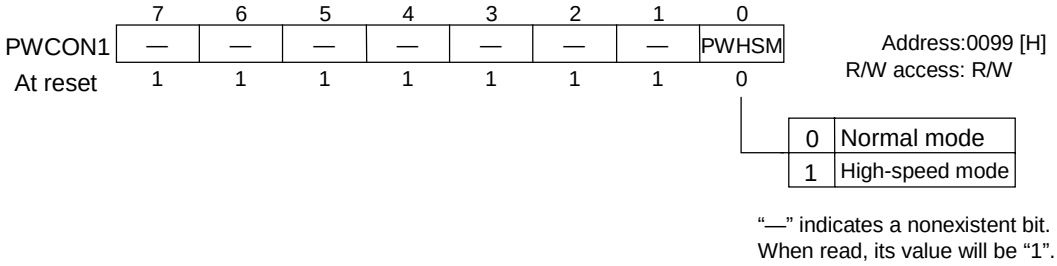


Figure 10-3 PWCON1 Configuration

[Note]

High-speed mode is only valid when 16-bit PWM is used.

10.3.2 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 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. When the PWM function does not operate, this port is fixed at "1".

(3) 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.

(4) PWM cycle registers (PWCY0, PWCY1)
If PWM0OUT is to be used, set the PWM cycle in PWCY0. If PWM1OUT is to be used, set the PWM cycle in PWCY1.

(5) PWM registers (PWR0 and PWR1)
If PWMnOUT are to be used, set the desired output duty value in PWRn (where n = 0 and 1). Set a value for PWR0 that is larger than the value of PWCY0. Set a value for PWR1 that is larger than the value of PWCY1.

(6) PWM control register 0 (PWMCON0)

If PWM0OUT is 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 is 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 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. When the PWM function does not operate, this port is fixed at "1".

(3) 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.

(4) PWM cycle register (PWCY)

Set the PWM cycle in PWCY.

(5) PWM registers (PWR01)

If PWM1OUT is to be used, set the desired output duty value in PWR01. Set a value for PWR01 that is larger than the value of PWCY.

(6) PWM control register 0 (PWMCON0)

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).

(7) PWM control register 1 (PWMCON1)

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)

10.4 PWM Operation

10.4.1 PWM Operation During 8-Bit Mode

During the 8-bit mode, PWM output can use the two output pins of PWM0OUT and PWM1OUT.

The output from the PWMnOUT pin (when n = 0 and 1) goes to the High level the moment the corresponding pin (P7_6 and P7_7) is configured as a secondary function output.

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". 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", a High level is output to the PWMnOut pin.

[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 10-4 shows an example of PWM output operation.

10.4.2 PWM Operation During 16-Bit Mode

During the 16-bit mode, PWM output can use the one output pin of PWM1OUT.

The output from the PWM1OUT pin goes to the High level the moment P7_7 is configured as a secondary function output.

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". PWC continues to count upward. When its value matches the contents of PWR01, a PWINT1 interrupt request is generated, the output flip-flop is reset to "0", and a Low level is output from the PWM1OUT pin. If PWC overflows, the output flip-flop is set to "1", and the PWM1OUT 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. When the RUN bit is reset to "0", a High level is output to the PWM1OUT pin.

However, even in the 16-bit mode, 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 is 0000H, the duty output is 1/65536. Increasing the value of PWR01 increases the output duty (High level). If the value of PWR01 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 10-4 shows an example of PWM output operation. Figure 10-5 shows an example of the timing at which PWM output changes.

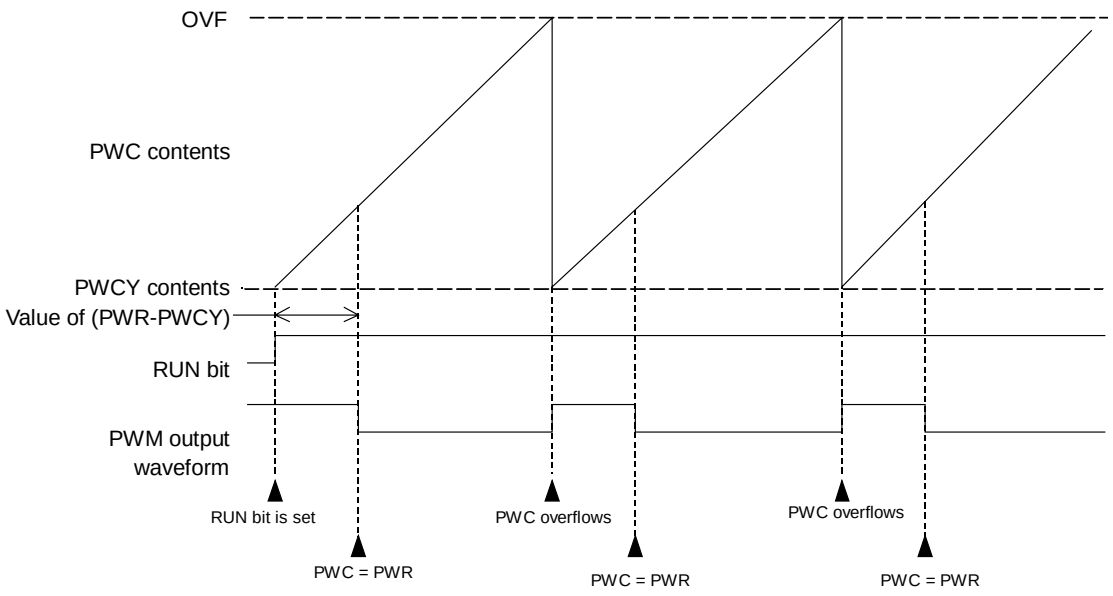


Figure 10-4 Example of PWM Output Operation

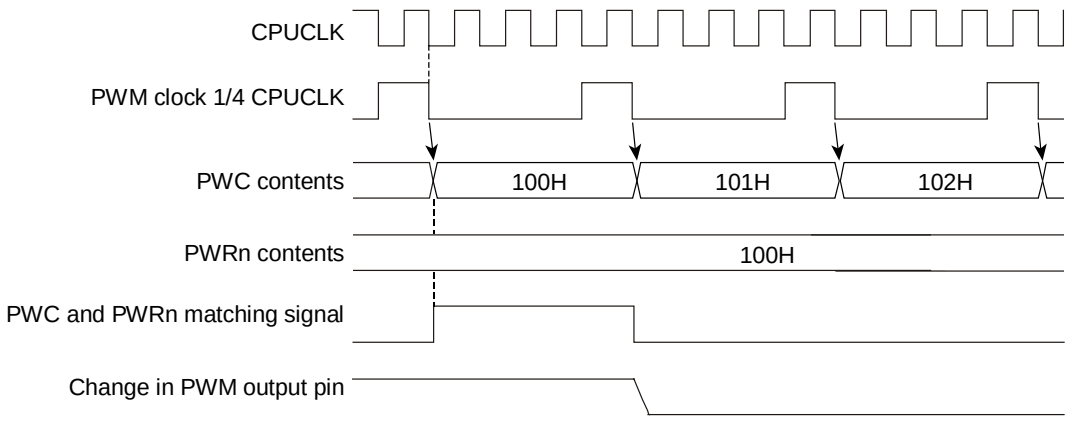


Figure 10-5 Example of PWM Output Change Timing

10.4.3 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 10-6, 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.

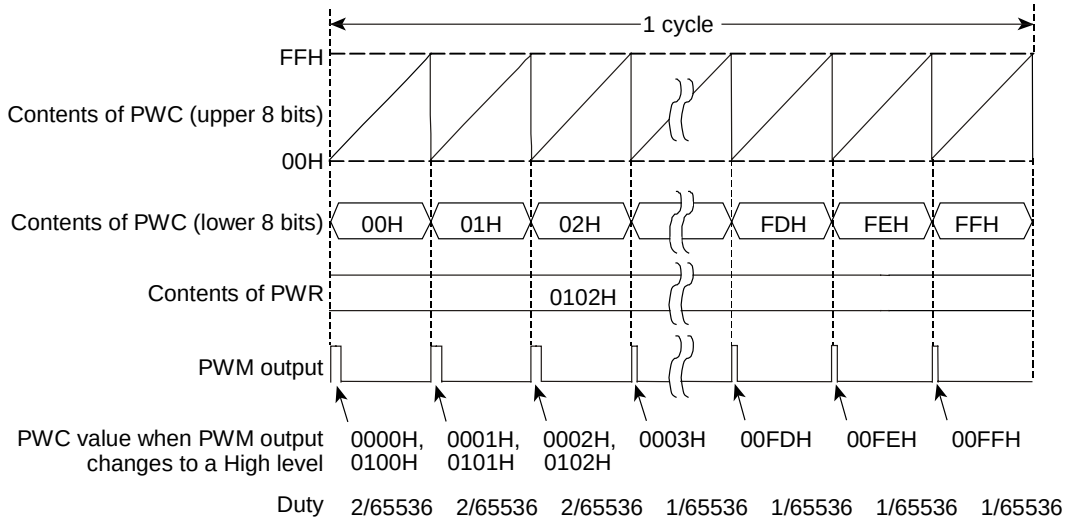


Figure 10-6 PWM Output Waveform During High-Speed Mode

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 366 Hz (for a main clock of 24 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 93.75 kHz (for a main clock of 24 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 10-7 shows an example of PWM output during the high-speed mode when PWCY is 0000H (longest cycle).

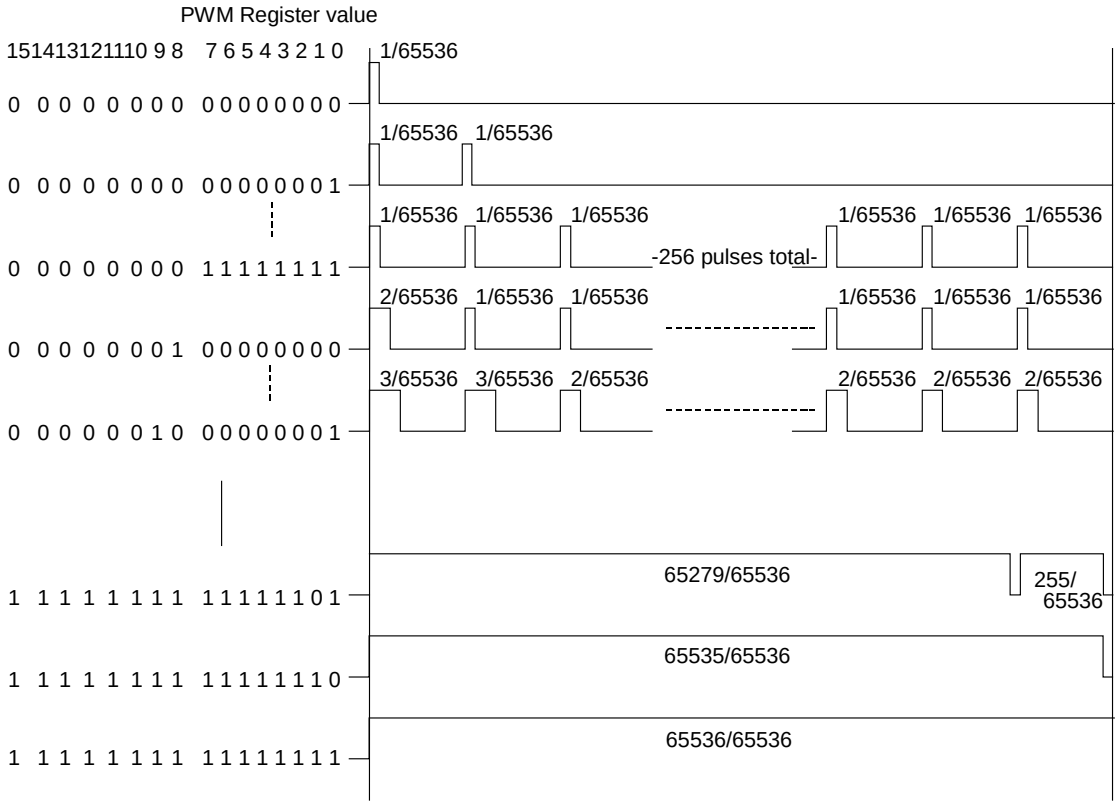


Figure 10-7 Example of PWM Output During High-Speed Mode

10.5 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 10-2 lists the vector address of each PWM interrupt factor and the interrupt processing flags.

Table 10-2 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					
Symbols of registers that contain interrupt processing flags		IRQ4	IE4	IP8	
	Reference page	15-16	15-21	15-29	

For further details regarding interrupt processing, refer to Chapter 15, "Interrupt Processing Functions".

Chapter 11

Serial Port Functions

11. Serial Port Functions

11.1 Overview

The ML66525 family contains four built-in serial port channels: UART/Synchronous receiver transmitter serial ports SIO1 and SIO6, universal synchronous receiver transmitter serial port SIO3, and synchronous receiver transmitter serial port with 32-byte FIFO SIO4.

11.2 Serial Port Configuration

Figure 11-1 shows the configuration of the serial ports.

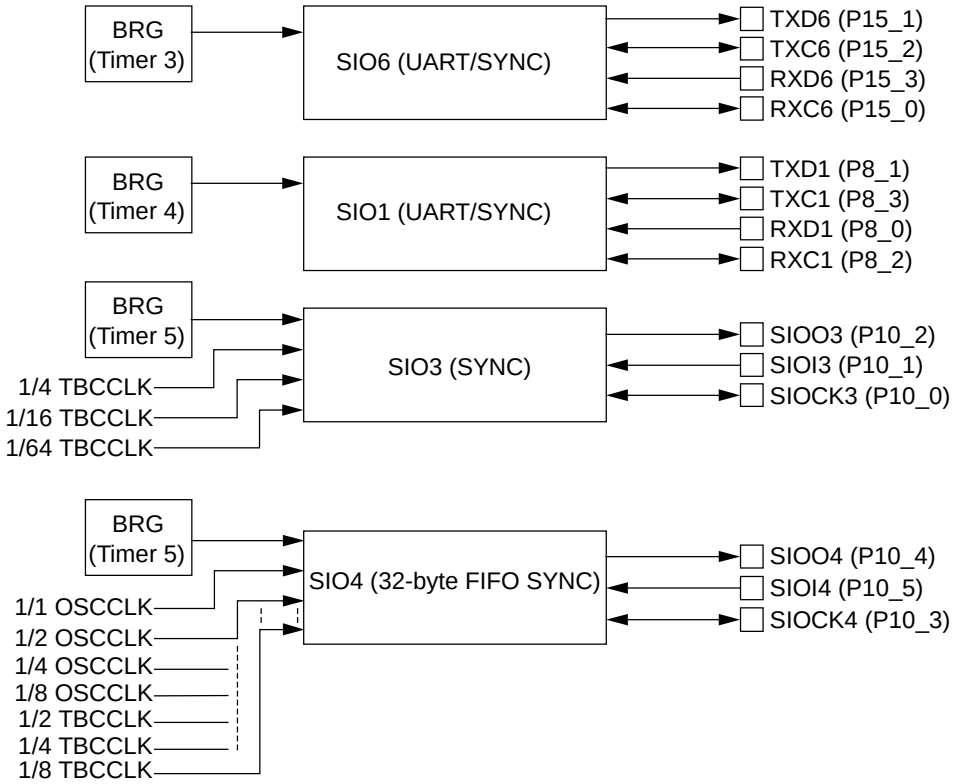


Figure 11-1 Serial Port Configuration

11.3 Serial Port Registers

Table 11-1 lists a summary of SFRs for control of the serial port functions.

Table 11-1 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	11-16
0085	SIO1 receive control register	SR1CON	—	R/W	8	00	11-18
0086	SIO1 transmit-receive buffer register	S1BUF	—	R/W	8	Undefined	11-22
0087 ☆	SIO1 status register	S1STAT	—	R/W	8	00	11-20
008A ☆	SIO3 control register	SIO3CON	—	R/W	8	80	11-41
008B	SIO3 register	SIO3R	—	R/W	8	Undefined	11-43
008C ☆	SIO4 control register	SIO4CON	—	R/W	8	80	11-48
008D	FIFO control register	FIFOCON	—	R/W	8	F1	11-50
008E	SIO4 serial input FIFO data register	SIN4	—	R	8	Undefined	11-52
008F	SIO4 serial output FIFO data register	SOUT4	—	W	8	Undefined	11-52
00F4 ☆	SIO6 transmit control register	ST6CON	—	R/W	8	04	11-4
00F5	SIO6 receive control register	SR6CON	—	R/W	8	00	11-6
00F6	SIO6 transmit-receive buffer register	S6BUF	—	R/W	8	Undefined	11-10
00F7 ☆	SIO6 status register	S6STAT	—	R/W	8	00	11-8

[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 22, "Special Function Registers (SFRs)".

11.4 SIO6

The SIO6 has a UART mode and a synchronous mode. Timer 3 is used as a baud rate generator exclusively for SIO6.

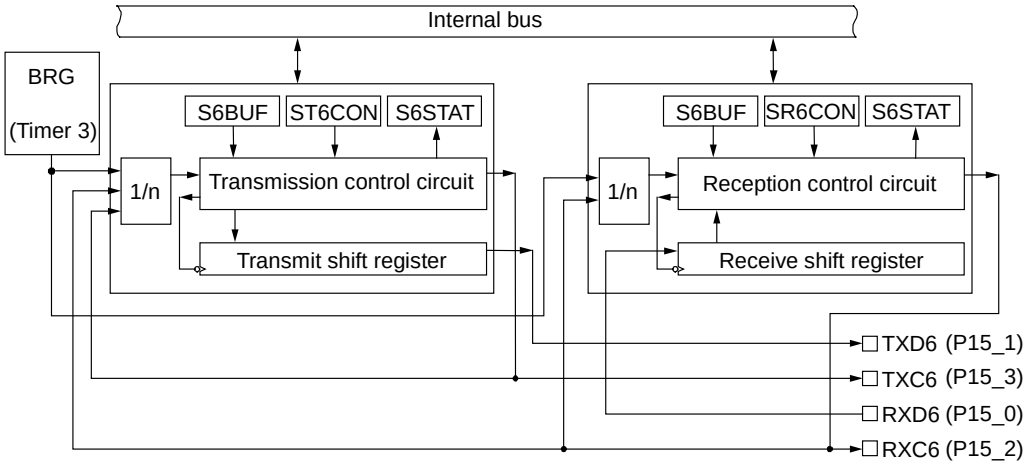
Table 11-2 lists specifications of SIO6.

Table 11-2 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 = 24 MHz)	1.5 Mbps	6 Mbps
Other	LSB first An external clock can be used for the UART baud rate	LSB first Master mode/slave mode

11.4.1 SIO6 Configuration

Figure 11-2 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 11-2 SIO6 Configuration

11.4.2 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 (RESn 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 11-3 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.

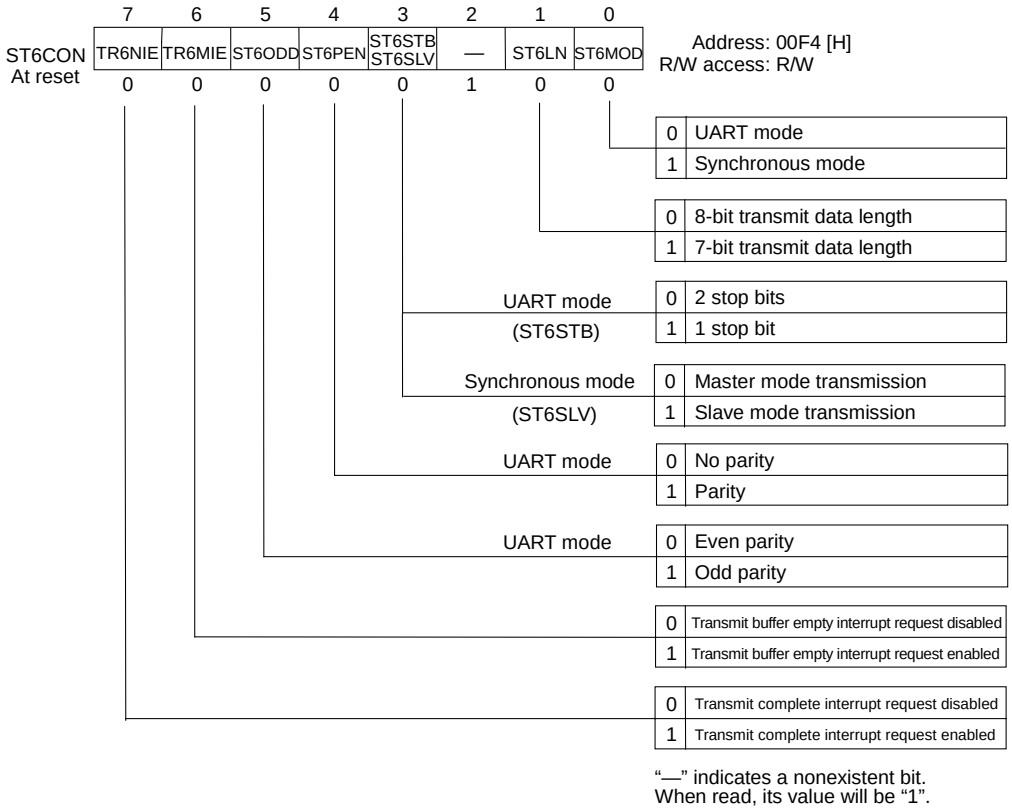


Figure 11-3 ST6CON Configuration

(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 (RESn 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 11-4 shows the SR6CON configuration.

[Description of each bit]

- SR6MOD (bit 0)
  ST6MOD 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.

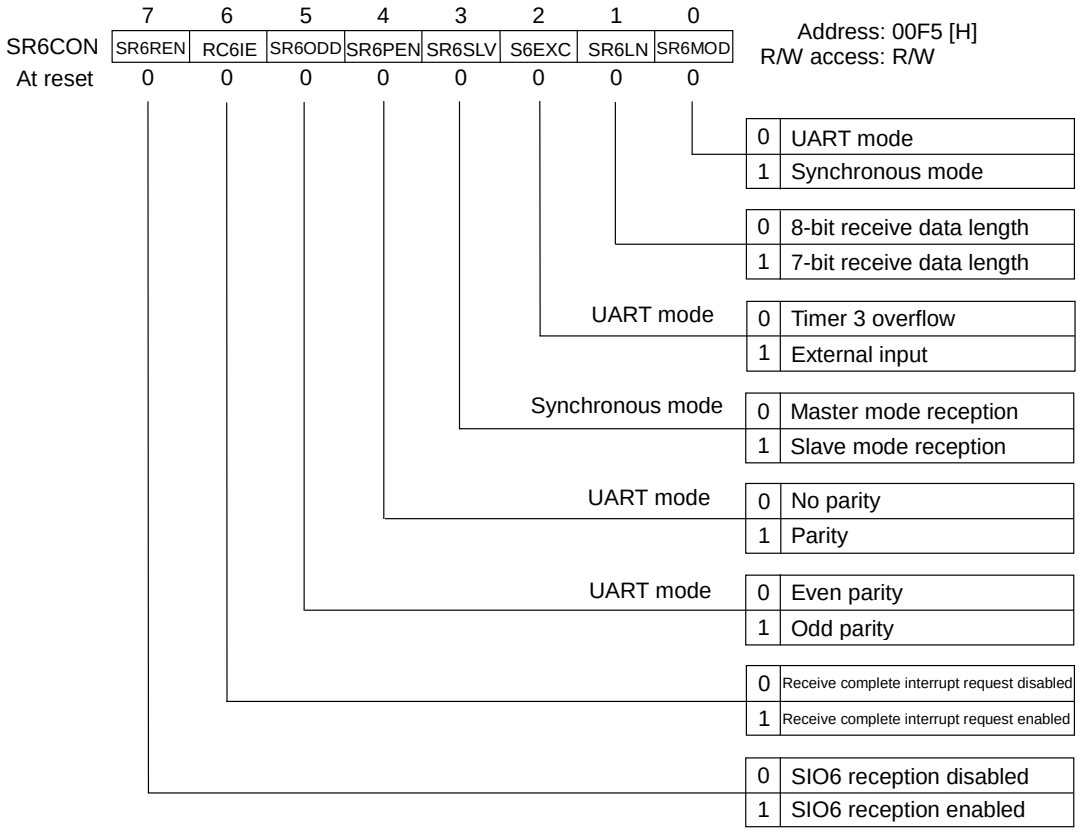


Figure 11-4 SR0CON Configuration

(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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), S6STAT becomes 00H.

Figure 11-5 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.

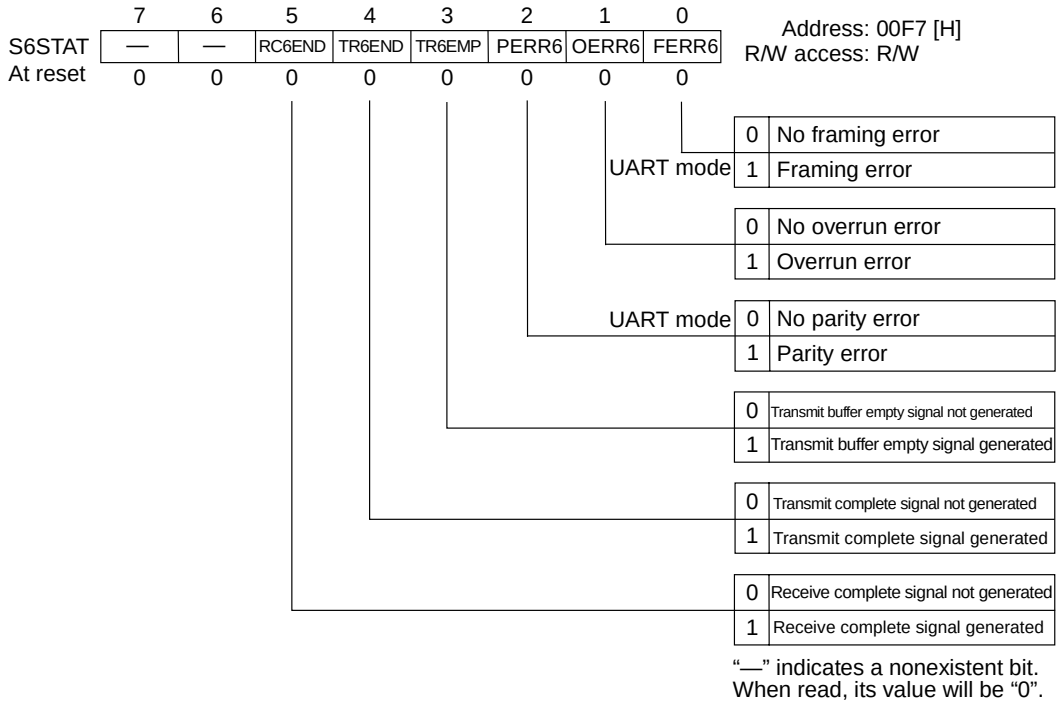


Figure 11-5 S6STAT Configuration

(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 (RESn 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 11-3 lists SIO6 transmit-receive frame lengths.

Table 11-3 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

11.4.3 Example of SIO6-related Register Settings

11.4.3.1 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.

11.4.3.2 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.

11.4.3.3 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

11.4.4 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 11-4 lists the vector address of the SIO6 interrupt factors and the interrupt processing flags.

Table 11-4 SIO6 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>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 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>15-15</td><td>15-20</td><td colspan="2">15-27</td></tr>
</table>

For further details regarding interrupt processing, refer to Chapter 15, "Interrupt Processing Functions".

11.5 SIO1

The SIO1 has a UART mode and a synchronous mode. Timer 4 is used as a baud rate generator exclusively for SIO1.

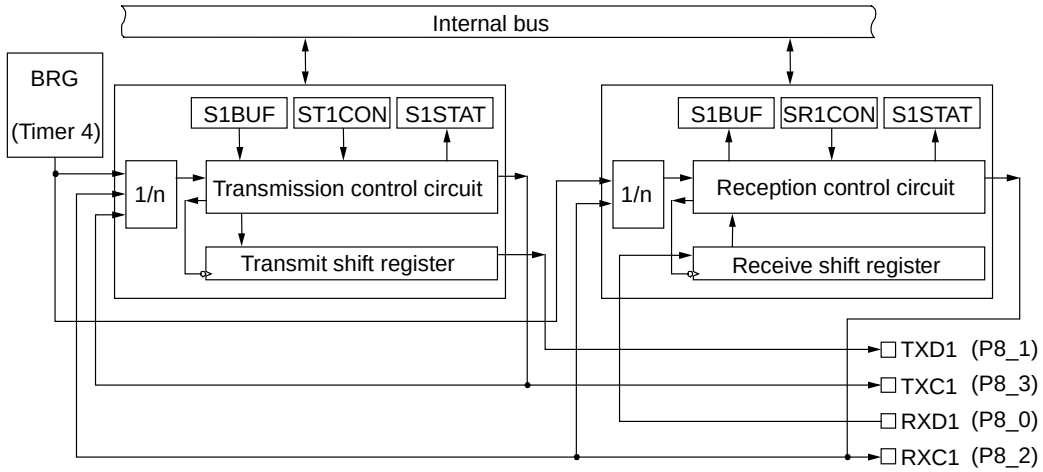
Table 11-5 lists specifications of SIO1.

Table 11-5 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 = 24 MHz)	1.5 Mbps	6 Mbps
Other	LSB first An external clock can be used for the UART baud rate	LSB first Master mode/slave mode

11.5.1 SIO1 Configuration

Figure 11-6 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 11-6 SIO1 Configuration

11.5.2 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 (RESn 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 11-7 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.

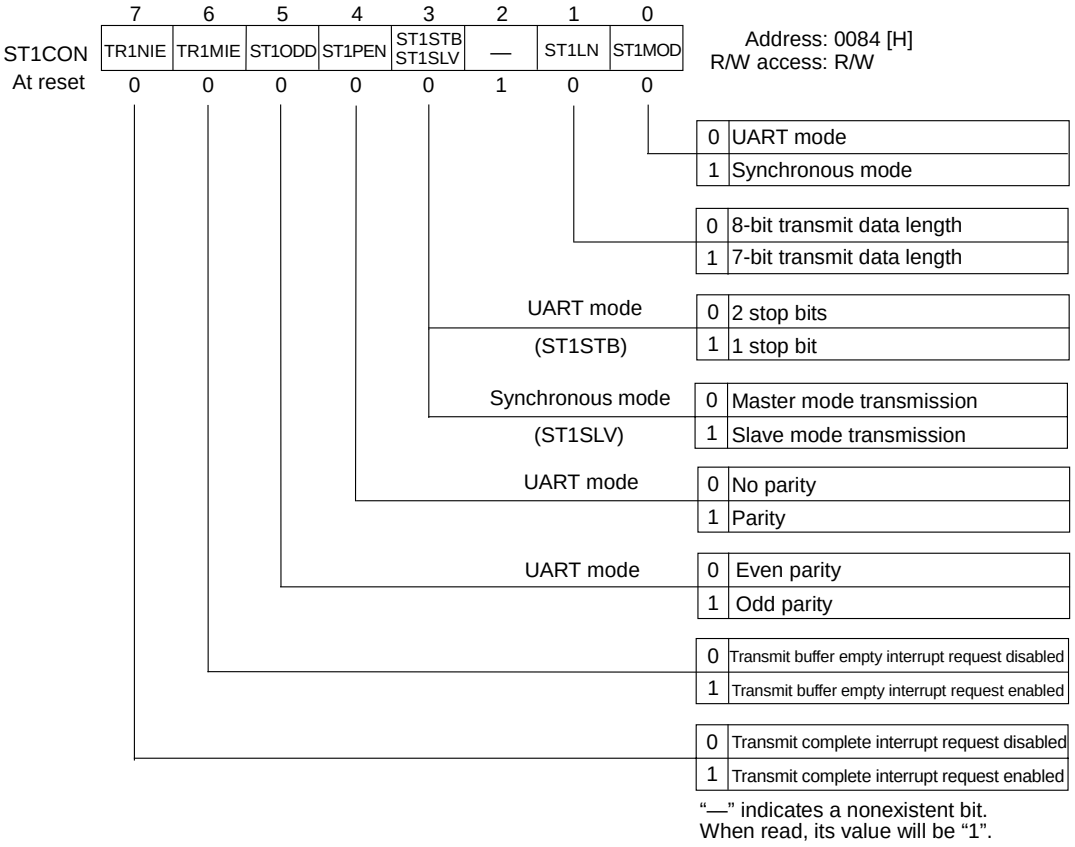


Figure 11-7 ST1CON Configuration

(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 (RESn 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 11-8 shows the SR1CON configuration.

[Description of each bit]

- SR1MOD (bit 0)
  ST1MOD specifies the 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.

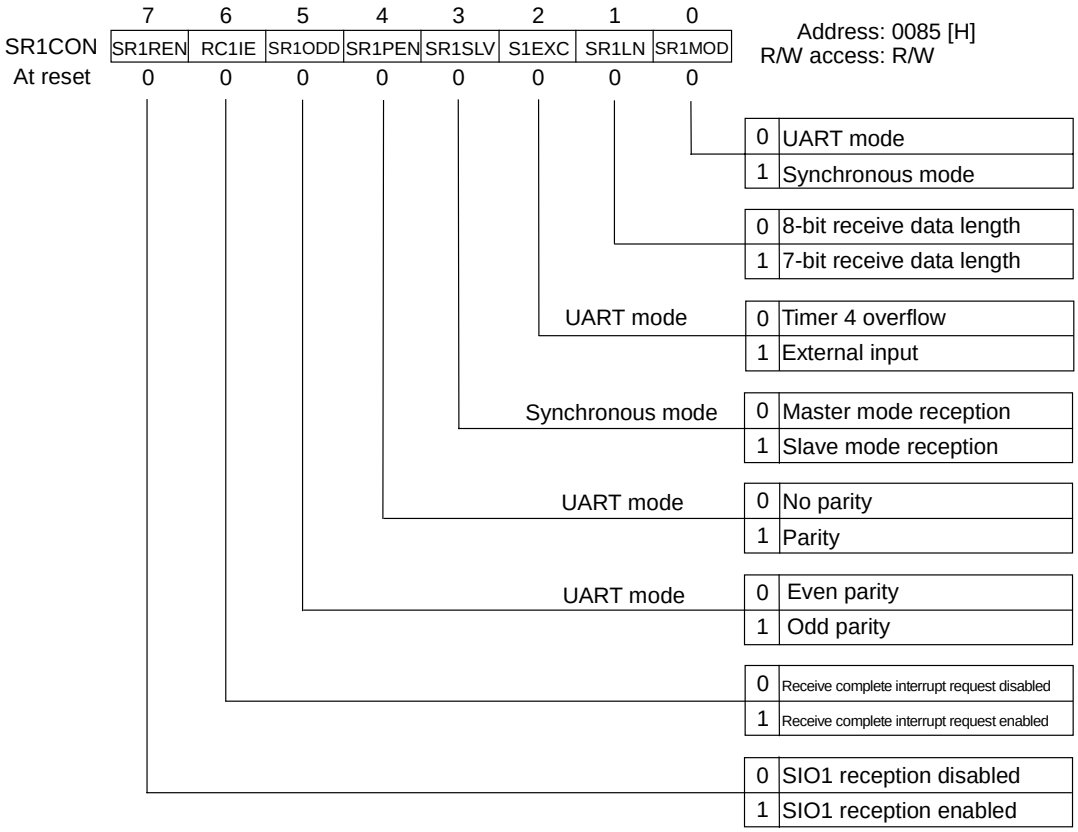


Figure 11-8 SR1CON Configuration

(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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), S1STAT becomes 00H.

Figure 11-9 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.

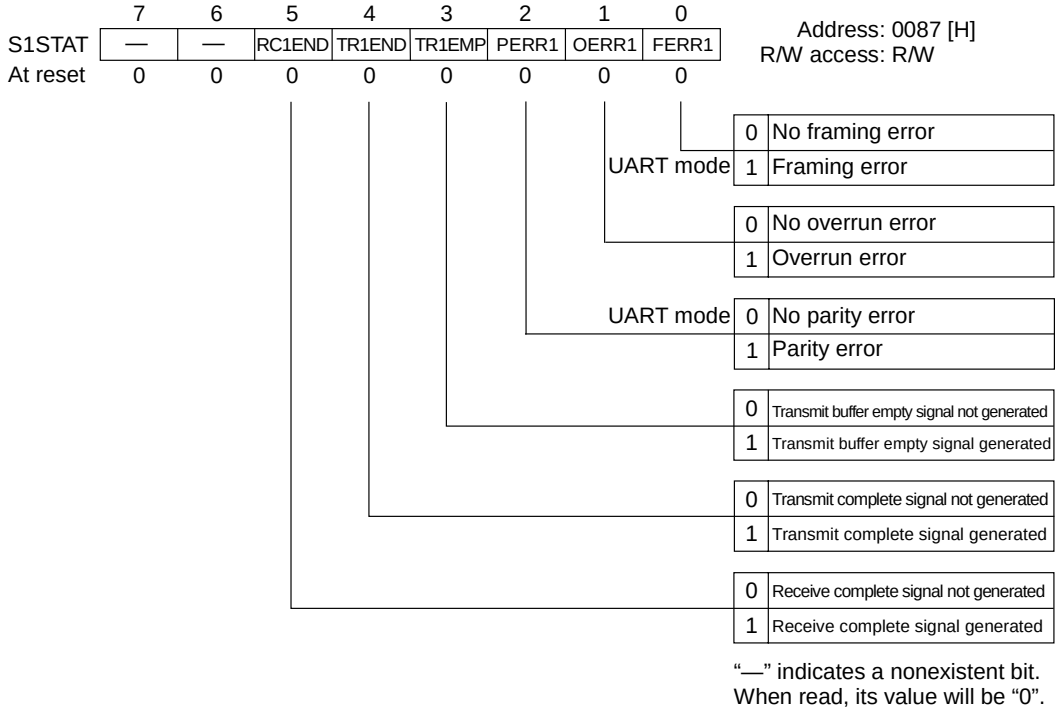


Figure 11-9 S1STAT Configuration

(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 (RESn 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 11-6 lists SIO1 transmit-receive frame lengths.

Table 11-6 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

11.5.3 Example of SIO1-related Register Settings

11.5.3.1 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 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.

11.5.3.2 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.

11.5.3.3 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

11.5.4 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 11-7 lists the vector address of the SIO1 interrupt factors and the interrupt processing flags.

Table 11-7 SIO1 Vector Address and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	0
SIO1 transmit buffer empty signal is generated	0038	QSIO1	ESIO1	P1SIO1	P0SIO1
SIO1 transmit complete signal is generated					
SIO1 receive complete signal is generated					
Symbols of registers that contain interrupt processing flags		IRQ2	IE2	IP5	
	Reference page	15-14	15-19	15-26	

For further details regarding interrupt processing, refer to Chapter 15, "Interrupt Processing Functions".

11.6 SIO6, SIO1 Operation

11.6.1 Transmit Operation

- UART mode

  Figure 11-10 shows the timing diagram of operation during UART transmission.

  The clock pulse from the baud rate generator (timer 3 for SIO6 and timer 4 for SIO1) 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 (RXC1, 6) for SIO1, 6. The externally input clock is processed as shown in figure 11-11, and is input to the 1/n dividing counter (1/16 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 SIO6 and SIO1 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 transmitted, if the next data to be transmit has been written to SnBUF, the transmit complete signal will not be generated.

  Figure 11-14 shows the timing diagram of operation during continuous transmission.

- Synchronous mode

[Master mode]

Figure 11-12 shows the timing diagram of operation during master mode transmission.

The clock pulse from the baud rate generator (timer 3 for SIO6 and timer 4 for SIO1) 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, S1BUF") acts as a trigger to start transmission.

One CPU clock after the write signal is generated, transmit data in S1BUF 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 S1BUF, 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 SIO6 and SIO1 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 transmitted, if the next data to be transmit has been written to SnBUF, the transmit complete signal will not be generated.

Figure 11-14 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 11-13 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 (TXC1). 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, S1BUF") 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 ST1CON 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 S1BUF, 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 SIO6 and SIO1 have S1BUF 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 transmitted, if the next data to be transmit has been written to SnBUF, the transmit complete signal will not be generated.

Figure 11-14 shows the timing diagram of operation during continuous transmission.

[Notes]

1. 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.
2. If a transmission error occurs that causes an external input clock to halt halfway through the transmission, it is possible to clear the SnBUF data by first setting bit 3 (STnSLV) of STnCON to "0" (master mode), then transmitting 8 bits of data. The transmission by an external input clock can be resumed by setting bit 3 (STnSLV) of STnCON to "1" after transmitting the 8 bits of data.

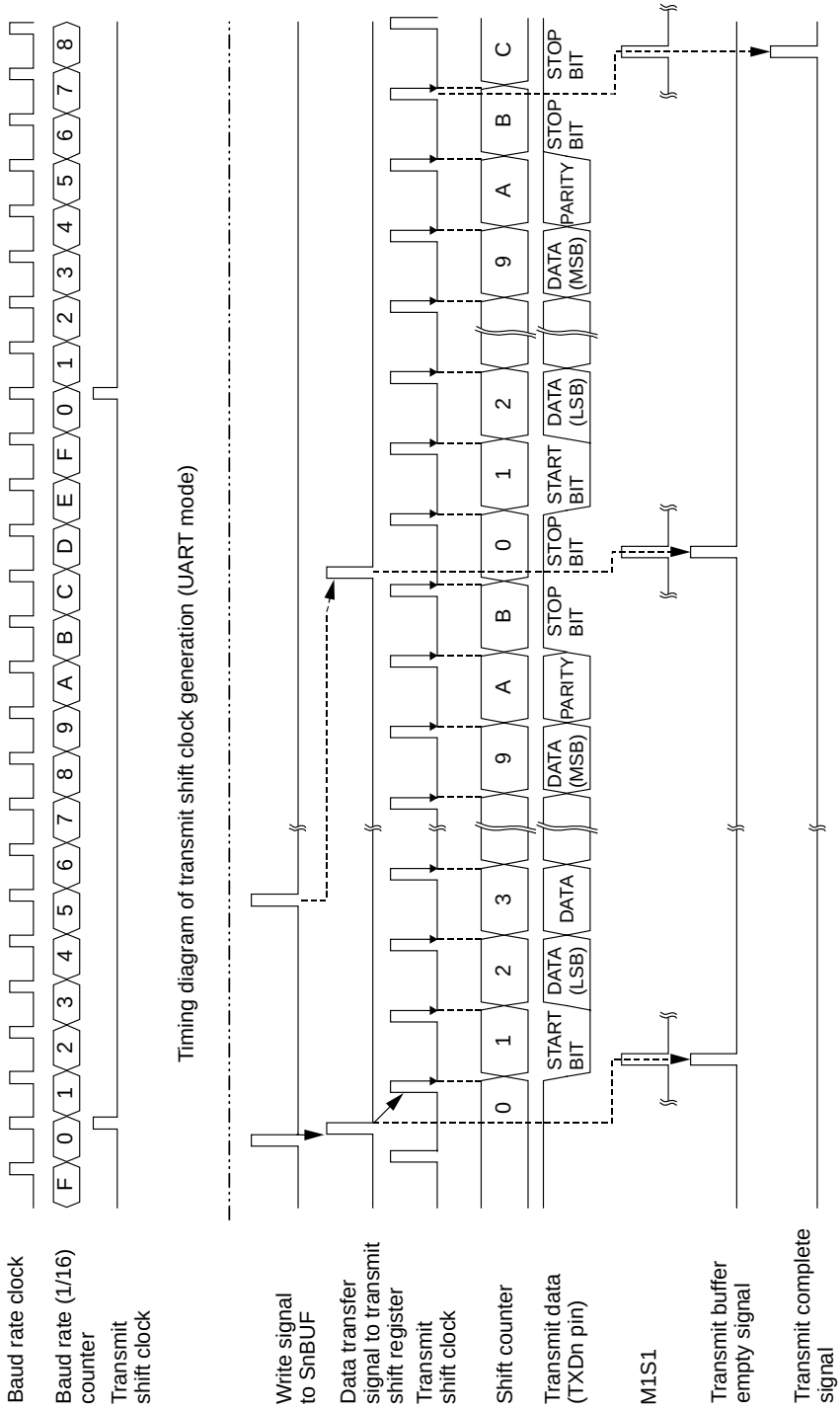


Figure 11-10 Transmission Timing Diagram (UART Mode)

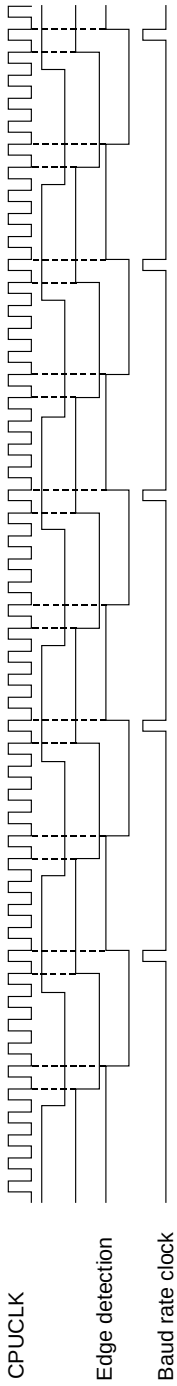


Figure 11-11 Timing Diagram of Baud Rate Clock Generation by External Clock (UART Mode, Transmission and Reception)

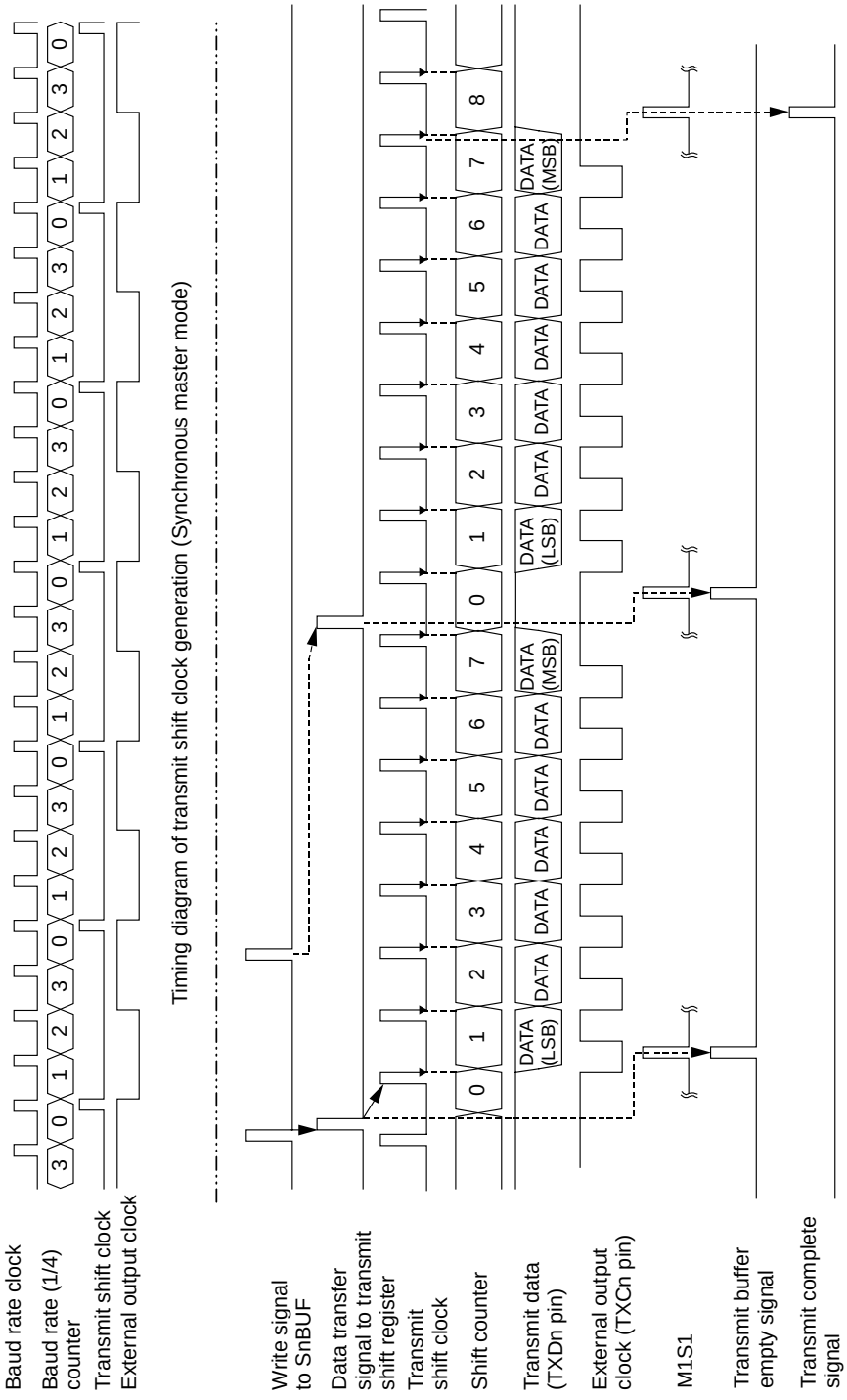


Figure 11-12 Transmission Timing Diagram (Synchronous Master Mode)

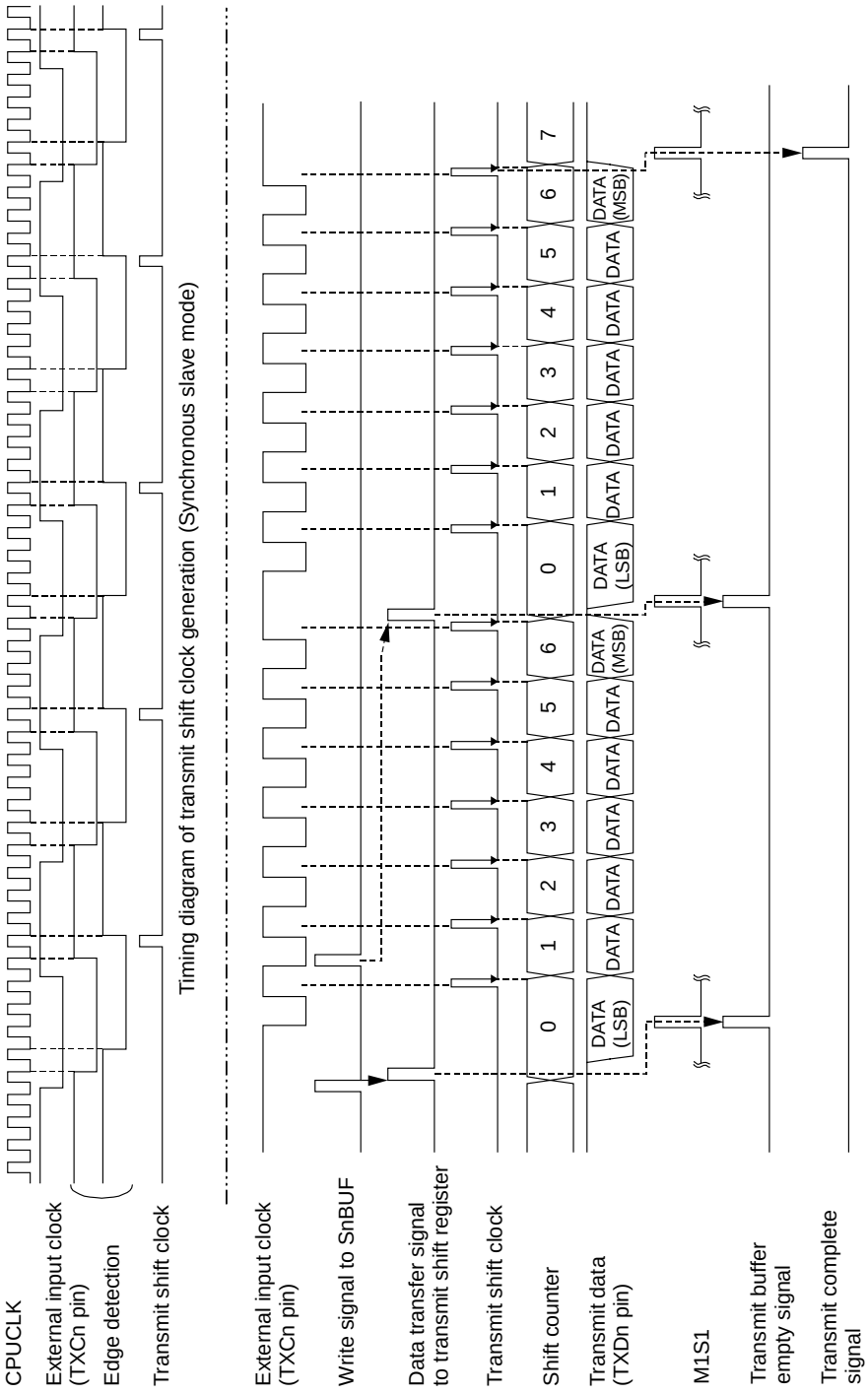


Figure 11-13 Transmission Timing Diagram (Synchronous Slave Mode)

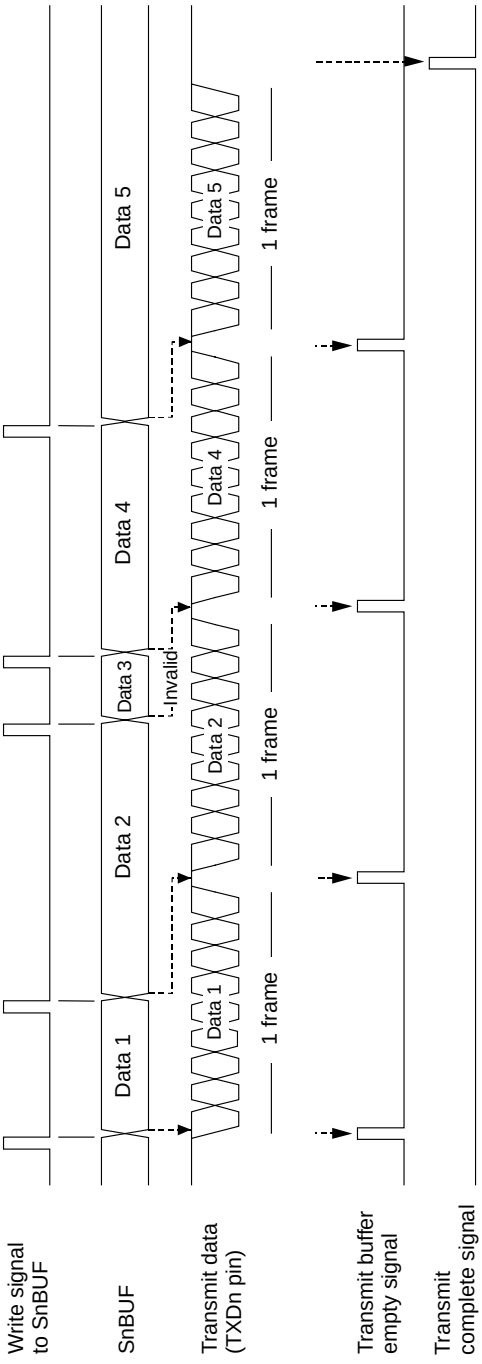


Figure 11-14 Transmission Timing Diagram (During Continuous Transmission)

11.6.2 Receive Operation

- UART mode

  Figure 11-15 shows the timing diagram of operation during UART reception.

  The clock pulse from the baud rate generator (timer 3 for SIO6 and timer 4 for SIO1) 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 input pin (RXCn) for SIOn. The externally input clock is processed as shown in figure 11-11, and is input to the 1/n (1/16 dividing) dividing counter as the baud rate clock.

  The 1/16 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 11-15) become the sampling clock for the receive data input pin (RXDn). The 10th pulse (value 9 of the baud rate (1/16) counter in figure 11-15) 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 11-16 shows the timing diagram of operation during master mode reception.

The clock pulse from the baud rate generator (timer 3 for SIO6 and timer 4 for SIO1) 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 11-16) 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 11-16) 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 11-17 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 (RXC1)). 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 (RXC1).

While the external input clock is at a Low level, the value of the receive data input pin (RXD1) 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.

[Note]
If a receive error occurs that causes an external input clock to halt halfway through the reception, it is possible to clear the SnBUF data by setting bit 7 (SRnREN) of SRnCON to "0" (SIOn reception disabled). The reception by an external input clock can be resumed by setting bit 7 (SRnREN) of SRnCON to "1" (SIOn reception enabled).

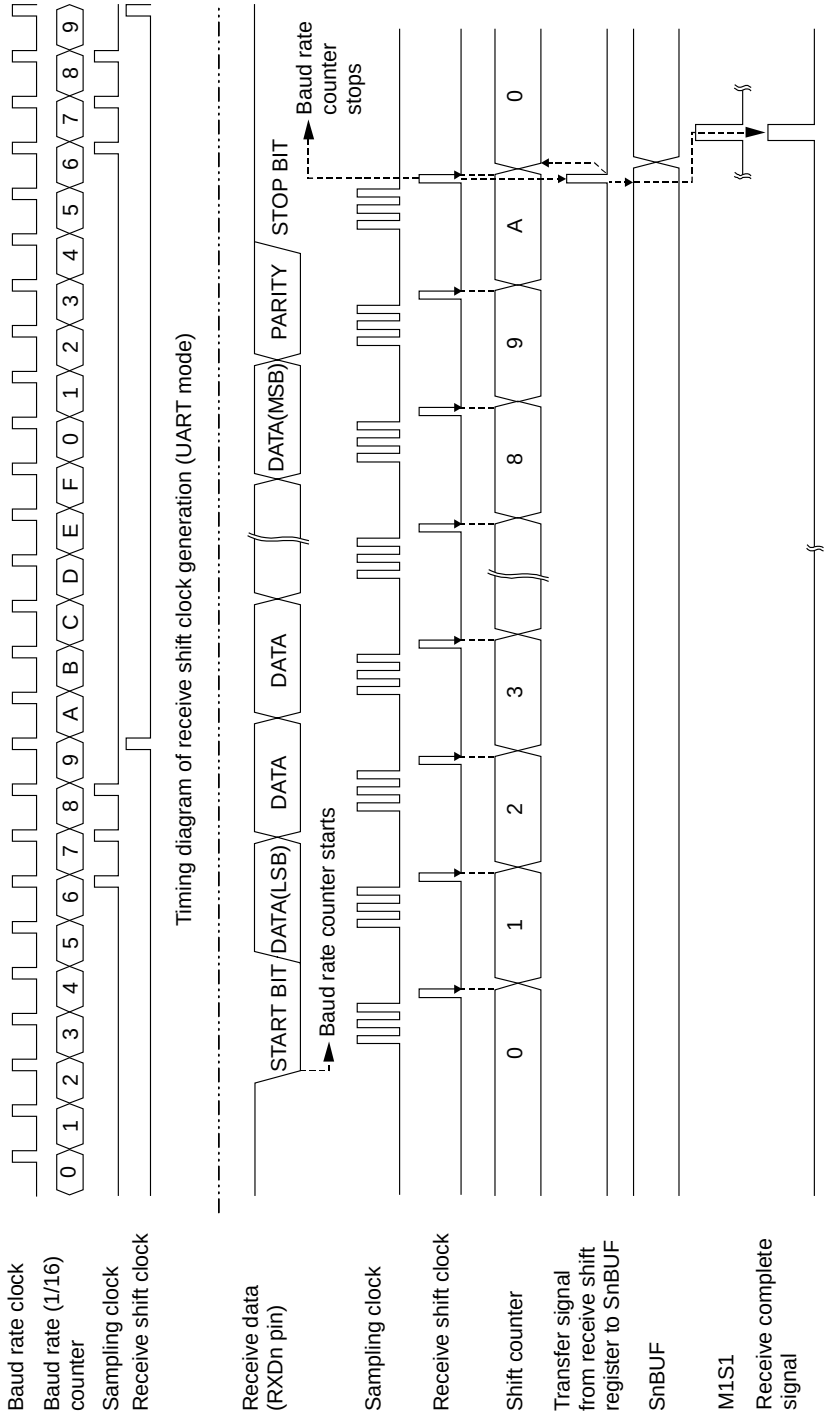


Figure 11-15 Reception Timing Diagram (UART Mode)

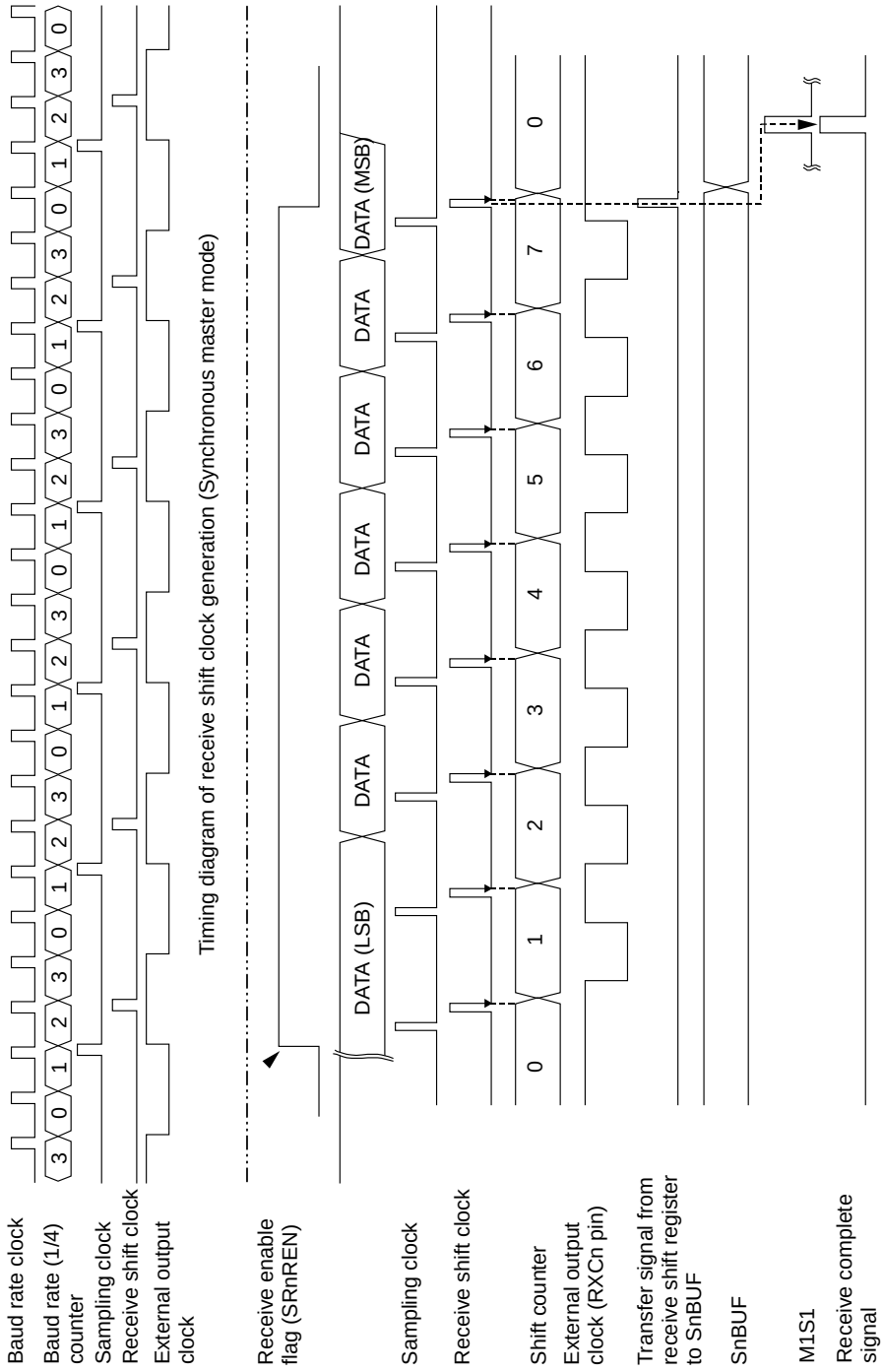


Figure 11-16 Reception Timing Diagram (Synchronous Master Mode)

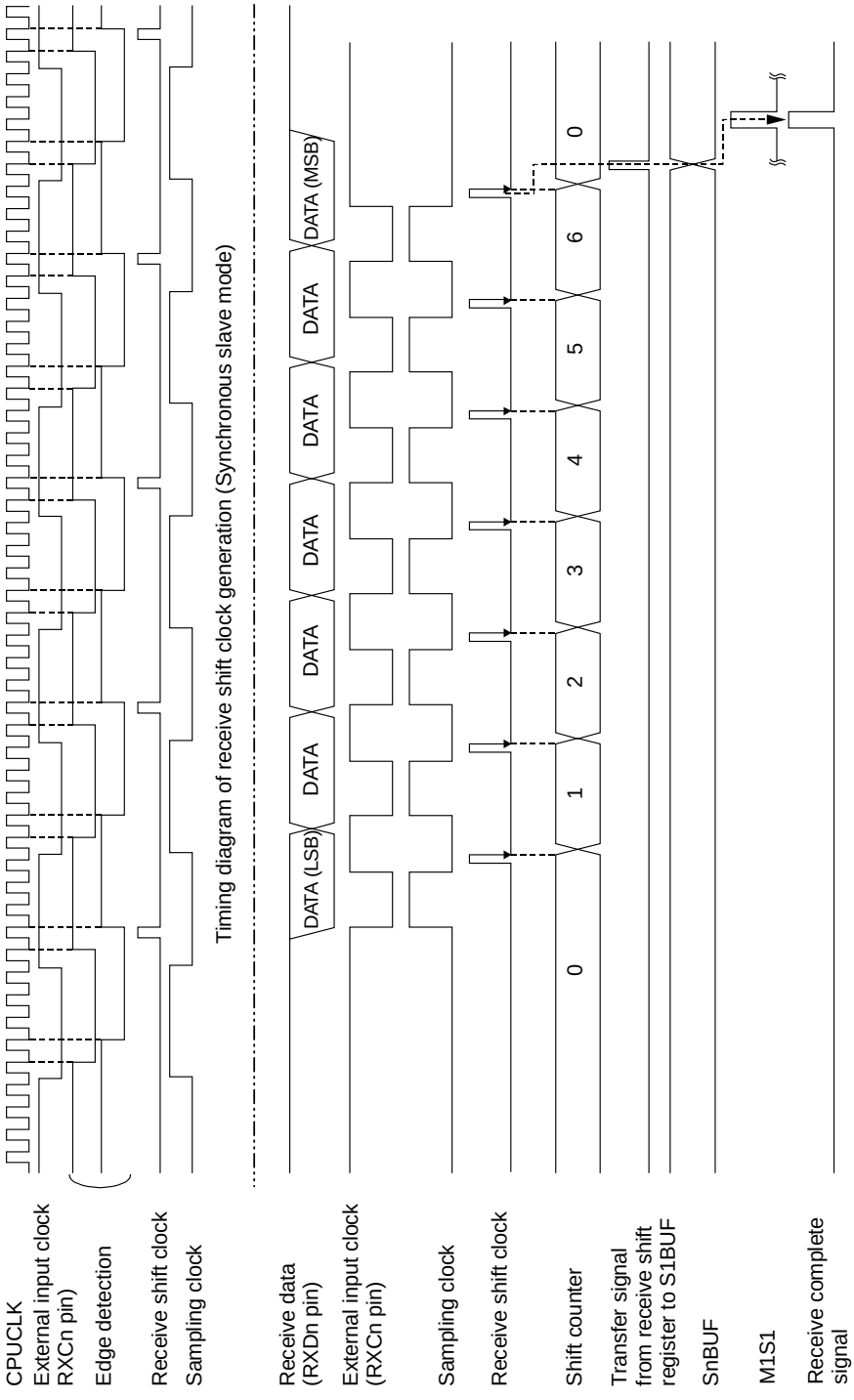


Figure 11-17 Reception Timing Diagram (Synchronous Slave Mode)

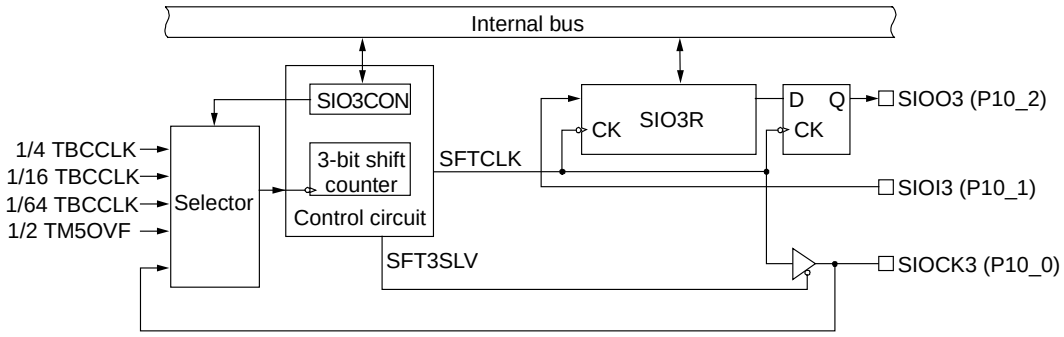
11.7 SIO3

SIO3 is an 8-bit serial port used for clocked synchronous communication. Synchronized to the clock (internal or external clock) specified by the SIO3 control register (SIO3CON), SIO3 simultaneously transmits and receives 8 bits of data. When transmission and reception are complete, a transmit-receive interrupt is requested. Operation of a master mode that can generate the shift clock internally and supply it externally, and a slave mode that can receive an externally supplied shift clock are possible.

With the slave mode, the external clock input enables 8 bits of data to be transmitted and received, even when the CPU is in the STOP mode. At such a time, the CPU's STOP mode may be released by a transmit-receive complete interrupt request.

11.7.1 SIO3 Configuration

Figure 11-18 shows the SIO3 configuration.



SIO3CON: SIO3 control register
SIO3R: SIO3 register (8 bits)
SFTCLK: shift clock
SFT3SLV: SIO3 master/slave mode switch
SIOO3: SIO3 transmit data output pin (P10_2)
SIOI3: SIO3 receive data input pin (P10_1)
SIOCK3: SIO3 transmit-receive clock I/O pin (P10_0)

Figure 11-18 SIO3 Configuration

11.7.2 Description of SIO3 Registers

(1) SIO3 control register (SIO3CON)

The SIO3 control register (SIO3CON) is an 8-bit register that controls SIO3 operation.

SIO3CON can be read from and written to by the program. However, bits 3 through 6 are read-only and write operations are invalid.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), SIO3CON becomes 80H, the SIO3 operating mode changes to the master mode and the shift clock becomes 1/4 TBCCLK.

[Note]

If SIO3CON is to be modified, make those changes after transmission and reception are complete. If SIO3CON is modified before transmission and reception are completed, the current transmission and reception will not be executed correctly.

Figure 11-19 shows the SIO3CON configuration.

[Description of each bit]

- SFT3CK0, SFT3CK1 (bits 0 and 1)
  During the master mode, SFT3CK0 and SFT3CK1 specify the shift clock. In the slave mode, these bits are invalid.
- SFT3SLV (bit 2)
  SFT3SLV specifies master or slave operation of SIO3. In the master mode, the shift clock is output from the SIOCK3 pin. In the slave mode, the shift clock is input to the SIOCK3 pin.
- S3BUSY (bit 3)
  When SIO3 data transmission and reception begins (data write to SIO3R), S3BUSY is automatically set to "1". When transmission and reception are completed, it is automatically reset to "0".
- SFT3CT0 to SFT3CT2 (bits 4 though 6)
  SFT3CT0 to SFT3CT2 read the value of the 3-bit counter during SIO3 data transmission and reception. This can be used to determine which bit is currently being transmitted or received. SFT3CT0 to SFT3CT2 are read-only. Write operations are invalid.

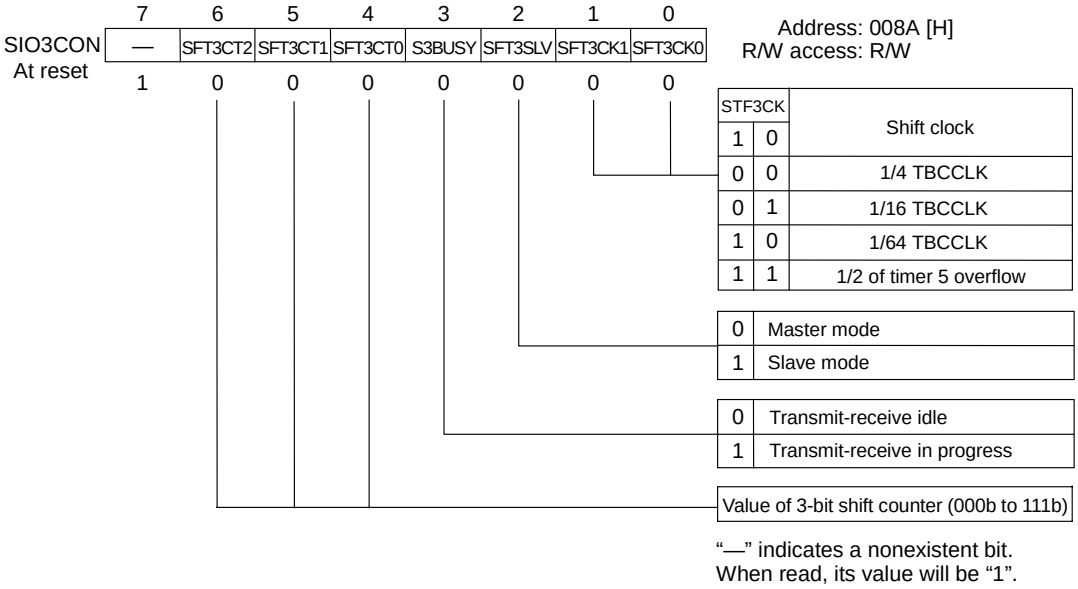


Figure 11-19 SIO3CON Configuration

(2) SIO3 register (SIO3R)

The SIO3 register (SIO3R) is an 8-bit shift register that performs the shift operations during transmission and reception.

Writing 8-bit data to SIO3R starts transmission and reception of SIO3.

SIO3R can be read from and written to by the program. Read the received data before transmission and reception of the next data begins.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of SIO3R is undefined.

[Note]

- In the slave mode, write to SIO3R while the SIOCK3 pin is at a High level.
- If new data is written to SIO3R during transmission and reception, the data currently being transmitted and received will be destroyed.
- Because writing to SIO3R triggers transmission and reception, write dummy data if only reception is to be performed.

11.7.3 Example of SIO3-related Register Settings

- Transmit-receive settings

(1) Port 10 mode register (P10IO)

If the master mode is to be used, set bit 0 (P10IO0) to "1" to configure that port as an output. If the slave mode is to be used, reset bit 0 (P10IO0) to "0" to configure that port as an input. Reset bit 1 (P10IO1) to "0" to configure that port as an input. Set bit 2 (P10IO2) to "1" to configure that port as an output.

(2) Port 10 secondary function control register (P10SF)

If the master mode is to be used, set bit 0 (P10SF0) to "1" to configure that port as a secondary function output. If the slave mode is to be used, specify with bit 0 (P10SF0) whether that port is to be pulled-up. Specify with bit 1 (P10SF1) whether the SIOI3 pin is to be pulled-up. Set bit 2 (P10SF2) to "1" to configure that port as a secondary function output.

(3) SIO3 control register (SIO3CON)

Specify the SIO3 shift clock with bits 0 and 1 (SFT3CK0, SFT3CK1). Select master mode or slave mode with bit 2 (SFT3SLV).

(4) SIO3 register (SIO3R)

In the case of transmission, write the data to be transmit. For reception, write dummy data. Transmission and reception will begin after data is written to SIO3R.

- Baud rate generator (Timer 5) settings

  If 1/2 of timer 5 overflow has been selected for use as the baud rate clock, implement the following settings.

(1) General-purpose 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 control register (TM5CON)

Bits 0 to 2 (TM5C0 to TM5C2) of this register specify the count clock for timer 5. If bit 3 (TM5RUN) is set to "1", timer 5 will begin counting. If reset to "0", timer 5 will halt counting.

[Equation to calculate baud rate]

$B = f_{(TM5)} \times 1/(256 - D) \times 1/2$

B : baud rate [bps]
$f_{(TM5)}$: timer 5 input clock frequency [Hz]
D : reload value (0 to 255)

11.7.4 SIO3 Operation

Transmission and reception are started by writing 8-bit data to the SIO3 register (SIO3R). After the transmission and reception of 8 bits of data are complete, a transmit-receive complete interrupt request is generated and the operation is complete.

In the master mode, the clock selected by bits 0 and 1 (SFT3CK0, SFT3CK1) of SIO3CON is the shift clock. The shift clock is output from the SIOCK3 pin.

In the slave mode, the shift clock is input to the SIOCK3 pin. If 8 or more consecutive clock pulses are input, the clock beginning with the 9th pulse will be ignored.

In both the master and slave modes, synchronized with the falling edge of the shift clock, SIO3 outputs shift-out data from the SIOO3 pin. Synchronized with the rising edge of the shift clock, shift-in data is input to the SIOI3 pin.

It is assumed that external devices change the shift-in data at the falling edge of the clock and fetch the shift-out data at the rising edge of the clock.

When transmission and reception are started (data is written to SIO3R), the BUSY flag of SIO3CON is set to "1". When 8 bits of data have been transmitted and received, the BUSY flag is automatically cleared to "0". When transmission and reception are completed, an interrupt is generated in synchronization with the first state of the next instruction (M1S1).

Figure 11-20 shows the timing of SIO3 operation.

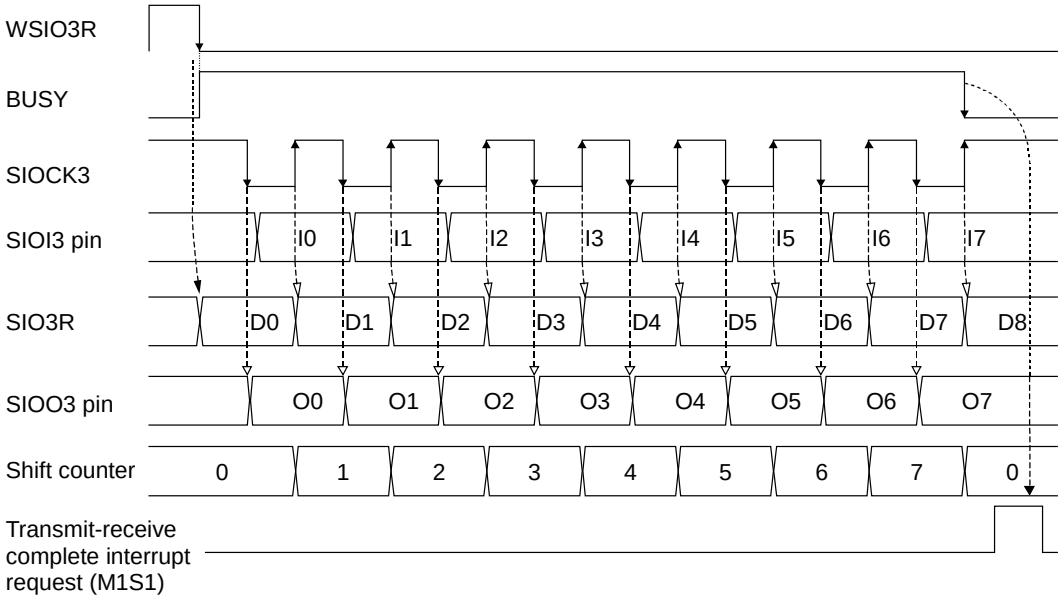


Figure 11-20 SIO3 Timing Diagram

Table 12-8 lists the output pin state immediately after being set as a secondary function output at reset and the state during the interval between two transmit-receive operations (interval beginning at the completion of an 8-bit data transmission and reception until the next data is written to SIO3R).

Table 11-8 Output Pins

Pin name	Value at reset	Interval between two transmit-receive operations
SIOCK3	"H"	"H"
SIOO3	Undefined	Last data of the previous transmission and reception

11.7.5 SIO3 Interrupt

When the SIO3 interrupt factor occurs, the interrupt request flag (QSIO3) is set to "1". The interrupt request flag (QSIO3) is located in interrupt request register 3 (IRQ3).

Interrupts can be enabled or disabled by the interrupt enable flag (ESIO3). The interrupt enable flag (ESIO3) is located in interrupt enable register 3 (IE3).

Three levels of priority can be set with the interrupt priority setting flags (P0SIO3 and P1SIO3). The interrupt priority setting flags (P0SIO3 and P1SIO3) are located in interrupt priority control register 6 (IP6).

Table 11-9 lists the vector address of the SIO3 interrupt factor and the interrupt processing flags.

Table 11-9 SIO3 Vector Address and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	0
SIO3 transmit-receive complete signal is generated	003E	QSIO3	ESIO3	P1SIO3	P0SIO3
Symbols of registers that contain interrupt processing flags		IRQ3	IE3	IP6	
	Reference page	15-15	15-20	15-27	

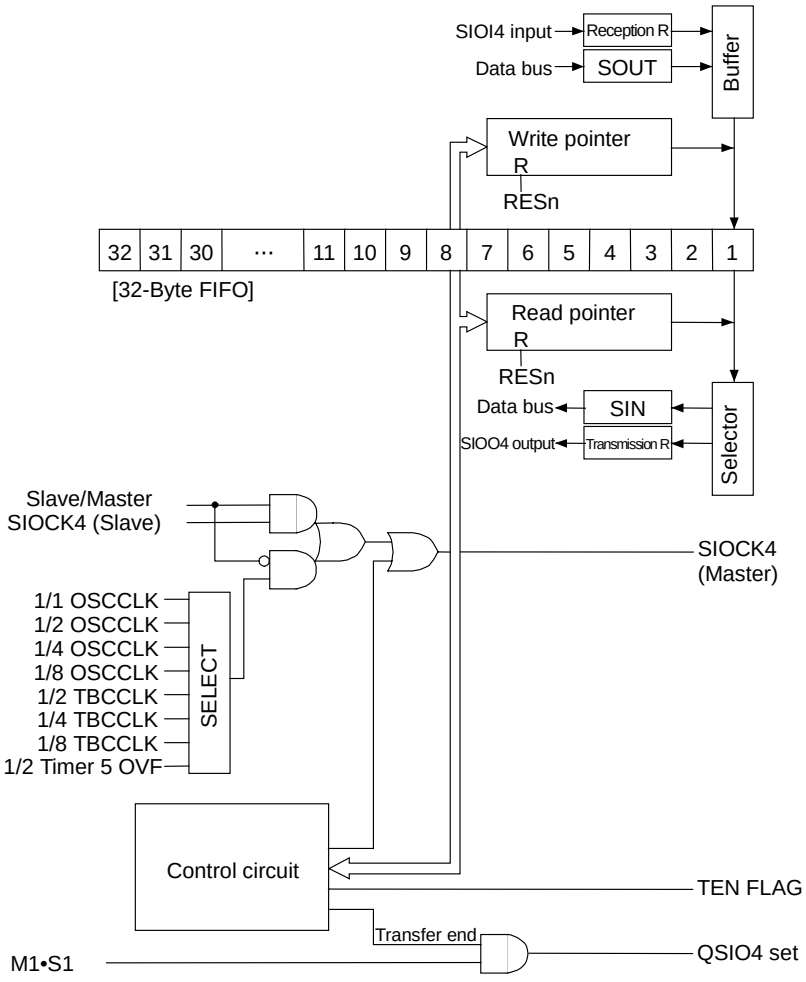
For further details regarding interrupt processing, refer to Chapter 15, "Interrupt Processing Functions".

11.8 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. The maximum communication speed is 2.5 Mbps. When transmission and reception are complete, a transmit-receive interrupt is requested.

11.8.1 SIO4 Configuration

Figure 11-18 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 11-21 SIO4 Configuration

11.8.2 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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), SIO4CON becomes 80H.

Figure 11-22 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.

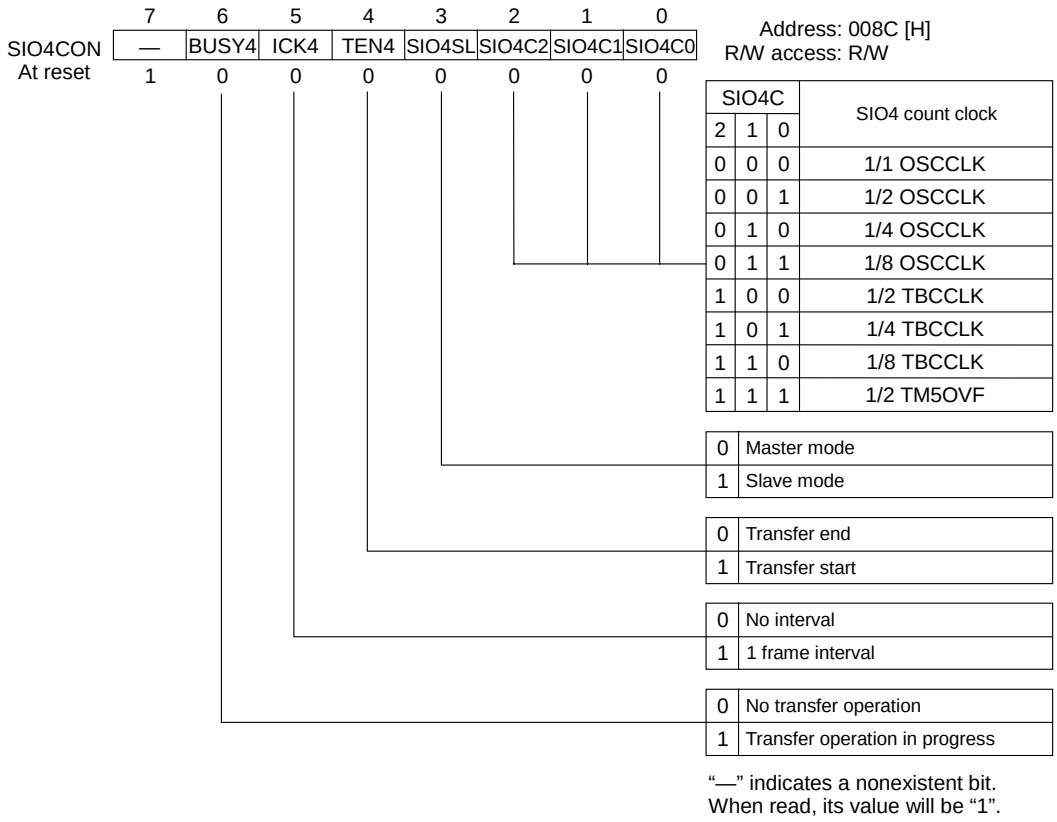


Figure 11-22 SIO4CON Configuration

[Note]

The maximum communication speed is 2.5 Mbps. Adjust the setting value of the SIO4 count clock.

(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, 2, and 4 to 7 are invalid.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of FIFOCON becomes F1H.

Figure 11-23 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".

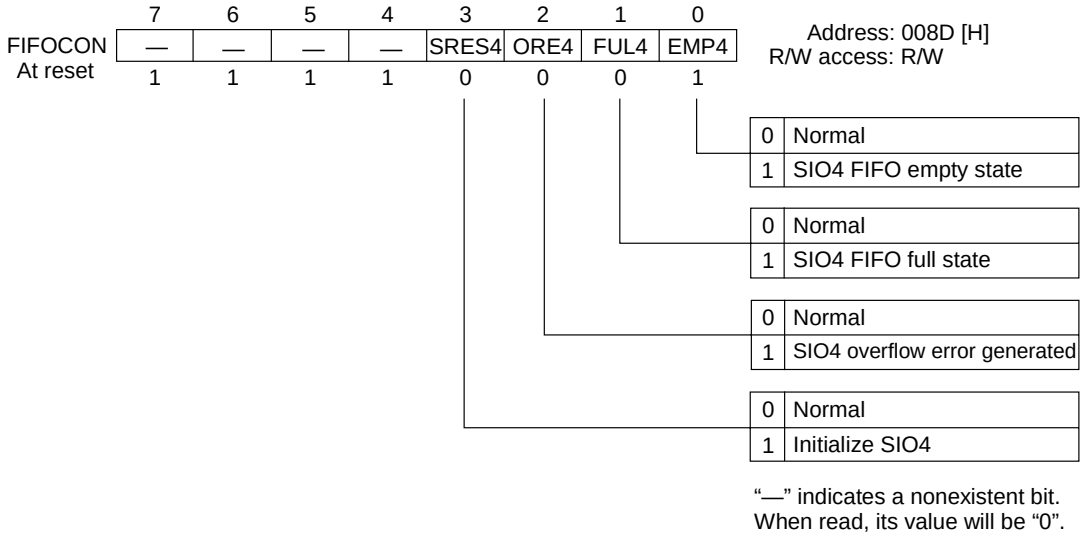


Figure 11-23 FIFOCON Configuration

(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 (RESn 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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the contents of SOUT 4 are undefined.

(5) Internal Control Register (P5IO)

	7	6	5	4	3	2	1	0
P5IO	FSEL1	FSEL0	SCK4IO	SCK4INV	STBSEL	MECKEN	UDCKEN	USBRST
at reset	0	0	0	0	0	0	0	0

Address: 0025 [H]
R/W access: R/W

0	P10_3/SIOCK4 : positive polarity
1	P10_3/SIOCK4：negative polarity

0	P10_3/SIOCK4 : input
1	P10_3/SIOCK4 : output

Refer to Section 13.4 for details.

11.8.3 Example of SIO4-related Register Settings

- Master mode settings

(1) Port 10 mode register (P10IO)

If SIOCK4 (clock output) is to be used, set both bit 3 (P10IO3) and bit 5 (SCK4IO) of the internal control register (P5IO) 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 both bit 3 (P10IO3) and bit 5 (SCK4IO) of the internal control register 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".

11.8.4 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 11-10 lists the vector address of the SIO4 interrupt factor and the interrupt processing flags.

Table 11-10 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 of registers that contain interrupt processing flags		IRQ3	IE3	IP6	
	Reference page	15-15	15-20	15-27	

For further details regarding interrupt processing, refer to Chapter 15, "Interrupt Processing Functions".

11.8.5 SIO4 Operation

SIO4 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 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 11-24 shows the timing of SIO4 operation during continuous transmission.

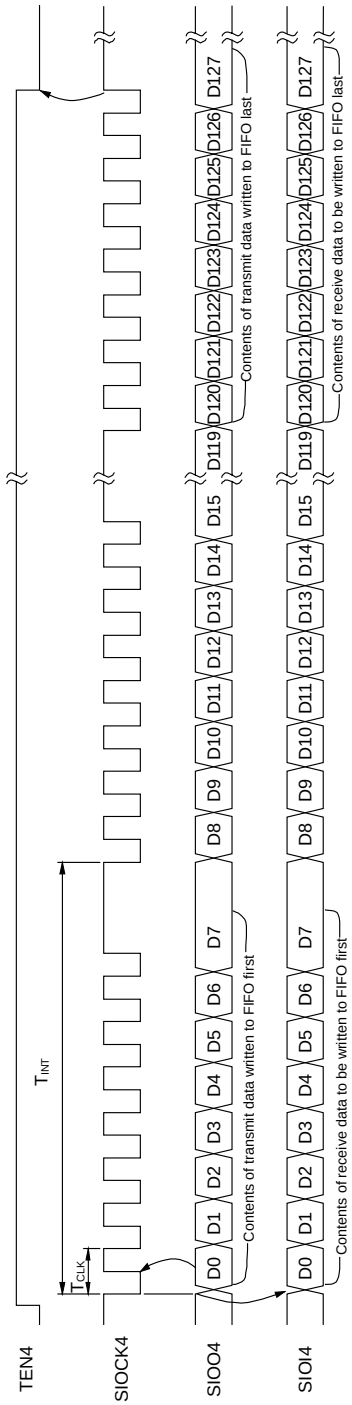


Figure 11-24 SIO4 Timing Diagram (During Continuous Transmission)

Chapter 12

A/D Converter Functions

12. A/D Converter Functions

12.1 Overview

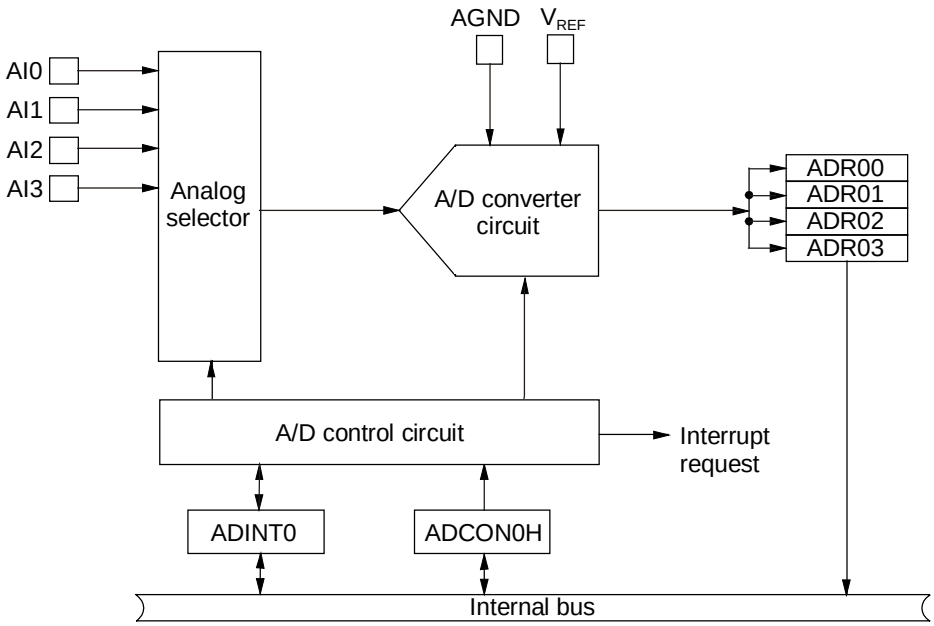
The ML66525 family has an internal 4-channel A/D converter with 10-bit resolution.

The A/D converter can operate 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.

12.2 A/D Converter Configuration

Figure 12-1 shows the A/D converter configuration.



AI0 to AI3: analog input pins (P12_0 to P12_3)

ADR00 to ADR03: 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 12-1 A/D Converter Configuration

12.3 A/D Converter Registers

Table 12-1 lists a summary of SFRs for control of the A/D converter.

Table 12-1 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
009D☆	A/D control register 0H	ADCON0H	—	R/W	8	80	12-3
009E☆	A/D interrupt control register 0	ADINT0	—	R/W	8	F0	12-5
00A0	A/D result register 00	—	ADR00	R	16	Undefined	12-6
00A1		—					
00A2	A/D result register 01	—	ADR01	R	16	Undefined	12-6
00A3		—					
00A4	A/D result register 02	—	ADR02	R	16	Undefined	12-6
00A5		—					
00A6	A/D result register 03	—	ADR03	R	16	Undefined	12-6
00A7		—					

[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 ADR03. If written to, the contents of all the registers from ADR00 through ADR03 may be overwritten.
4. For details, refer to Chapter 22, "Special Function Registers (SFRs)".

12.3.1 Description of A/D Converter Registers

(1) A/D control register 0H (ADCON0H)

ADCON0H is a 5-bit register that mainly controls the select mode of the A/D converter.

ADCON0H can be read from and written to by the program. If bits 2 and 3 are to be written to, a value of "0" must be written; if bit 7 is to be written to, a value of "1" must be written. If read, bits 2 and 3 are always "0" and bit 7 is always "1".

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), ADCON0H becomes 80H.

Figure 12-2 shows the ADCON0H configuration.

[Description of each bit]

- ADSTM00 and ADSTM01 (bits 0 and 1)
  ADSTM00 and ADSTM01 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, ADTM01, ADTM02 (bits 5, 6 and 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 12.5, "Notes Regarding Usage of A/D Converter".
  During A/D conversion, changes to the number of clocks will be ignored.

	7	6	5	4	3	2	1	0
ADCON0H	ADTM02	ADTM01	ADTM00	STS0	"0"	"0"	ADSTM01	ADSTM00
At reset	0	0	0	0	0	0	0	0

Address: 009D [H]
R/W access: R/W

ADSTM0		A/D select channel
1	0	
0	0	ch0
0	1	ch1
1	0	ch2
1	1	ch3

0	Stop select mode A/D conversion
1	Operate select mode A/D conversion

ADTM0			Number of A/D conversion clocks for 1 channel (when f = 24 MHz)
2	1	0	
0	0	0	1536 CPUCLK (64 μs)
0	0	1	1024 CPUCLK (42.7 μs)
0	1	0	768 CPUCLK (32 μs)
0	1	1	512 CPUCLK (21.3 μs)
1	0	0	384 CPUCLK (16 μs)
1	0	1	256 CPUCLK (10.7 μs)
1	1	0	192 CPUCLK (8 μs)
1	1	1	128 CPUCLK (5.3 μs)

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

Figure 12-2 ADCON0H Configuration

(2) A/D interrupt control register (ADINT0)

ADINT0 is a 2-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. When writing to bits 0 and 2, be sure to write "0" to these bits. If read, a value of "0" will always be obtained for bits 0 and 2 and a value of "1" will always be obtained for bits 4 through 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), ADINT0 becomes F0H.

Figure 12-3 shows the ADINT0 configuration.

[Description of each bit]

- 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.
- ADSTIE0 (bit 3)
  ADSTIE0 enables or disables interrupt requests when A/D conversion is completed in the select mode.

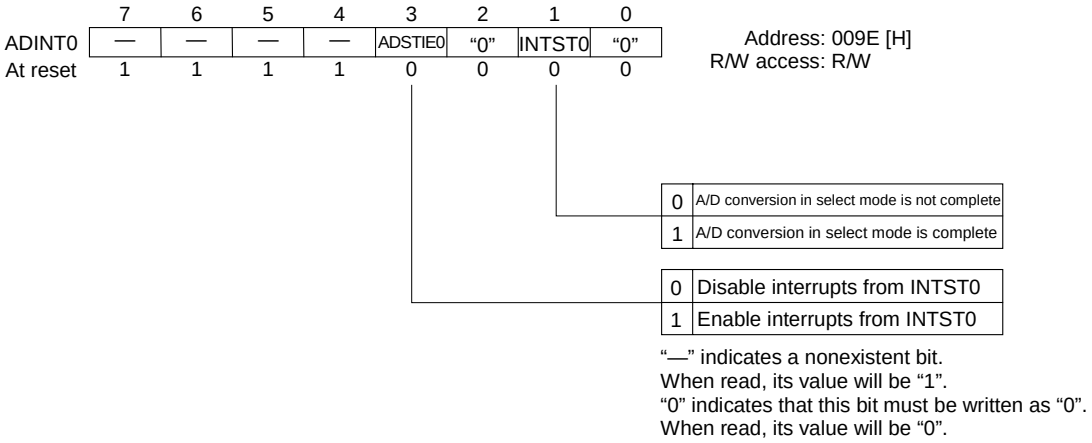


Figure 12-3 ADINT0 Configuration

(3) A/D result registers (ADR00 to ADR03)

A/D result registers (ADR00 to ADR03) consist of 10 bits and store the A/D conversion results.

A/D result registers (ADR00 to ADR03) can only be read in word access operations by the program.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of ADR00 to ADR03 is undefined.

Figure 12-4 shows the configuration of the A/D result registers (ADR00 to ADR03).

R/W access: R (word access only)

	Address [H]	7	6	5	4	3	2	1	0	
ADR03	00A7	—	—	—	—	—	—	bit9	bit8	bit9 : MSB
	00A6	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	bit0 : LSB
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 12-4 A/D Result Registers (ADR00 to ADR07) Configuration

[Note]

Do not write to the A/D result registers (ADR00 to ADR03). If written to, all the registers from ADR00 to ADR03 may be overwritten.

12.3.2 Example of A/D Converter-related Register Settings

(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, 6 and 7 (ADTM00, ADTM01, 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.

12.4 A/D Converter Operation

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.

12.5 Notes Regarding Usage of A/D Converter

12.5.1 Considerations When Setting the Conversion Time

Figure 12-5 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, ADTM01 and ADTM02 of the A/D control register 0H (ADCON0H).

Table 12-2 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.

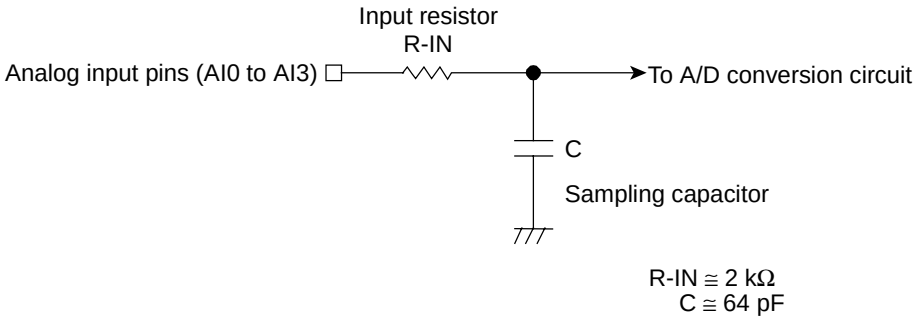


Figure 12-5 Equivalent Circuit of Analog Input Section

Table 12-2 Clock Allocation in A/D Conversion Processes

ADTM0			Number of clocks for A/D conversion of 1 channel	Number of clocks required by each process		
2	1	0		Sampling	A/D conversion	Other
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
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

Unit: 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 and ADTM02 settings

3) Actual precision of the A/D converter (comparator, RC 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".

12.5.2 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 ML66525 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 to 47 µF or larger between V_{REF} and AGND, the stability of V_{REF} can be maintained by connecting a 0.01 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 12-6 shows an example of noise-suppression measures.

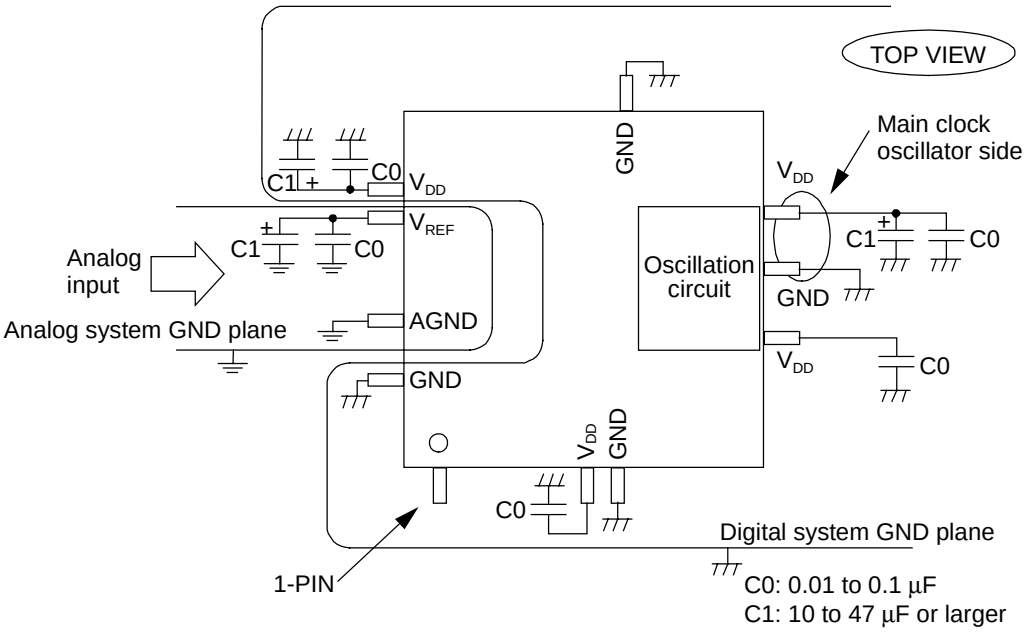


Figure 12-6 Example of Noise-Suppression Measures

12.6 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 12-3 lists the vector address of the A/D converter interrupt factors and the interrupt processing flags.

Table 12-3 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 the select mode is complete	0044	QAD	EAD	P1AD	P0AD
Symbols of registers that contain interrupt processing flags		IRQ3	IE3	IP7	
	Reference page	15-15	15-20	15-28	

For further details regarding interrupt processing, refer to Chapter 15, "Interrupt Processing Functions".

Chapter 13

Peripheral Functions

13. Peripheral Functions

13.1 Overview

The ML66525 family has the following functions to service peripheral ICs: an external XTCLK input control function. This function can be specified with the peripheral control register (PRPHCON).

In addition, the ML66525 has an internal control register (P5IO) to control the internal USB controller, internal DMA controller, and internal Media controller.

The ML66525 family has V_{DD}_IO, V_{DD}_CORE, and VBUS as the power supply pins, that is, it has three separate power supplies.

13.2 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.

13.3 Peripheral Control Register (PRPHCON)

The peripheral control register (PRPHCON) consists of 1 bit.

If bit 4 (EXTXT) of PRPHCON is set to "1", an external clock can be input to the XT oscillation circuit.

PRPHCON can be read from and written to by the program. However, write operations are invalid for bits 2, 3 and 7. When writing to bits 0, 1, 5, and 6, be sure to write "0" to these bits. If read, a value of "0" will be always obtained for these bits and a value of "1" will always be obtained for bits 2, 3 and 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), PRPHCON becomes 8CH.

Figure 13-1 shows the PRPHCON configuration.

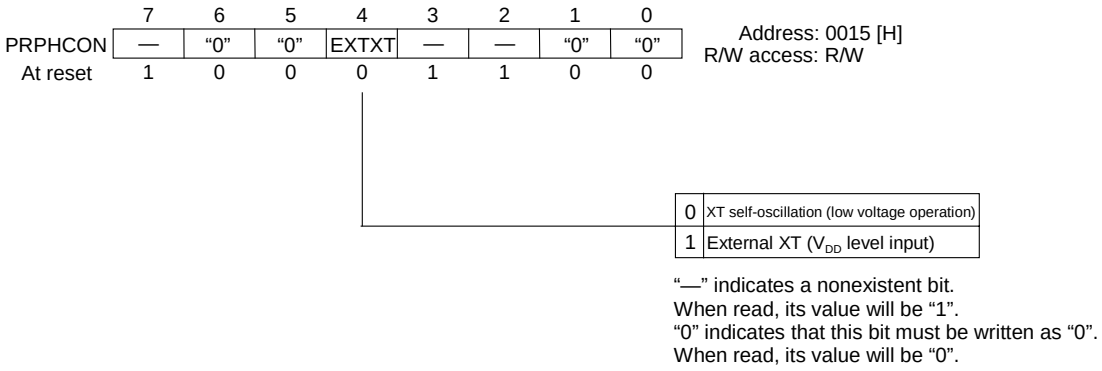


Figure 13-1 PRPHCON Configuration

13.4 Internal Control Register (P5IO)

The internal control register (P5IO) consists of 8 bits.

P5IO can be read or written by software.

At reset (RESn signal input, BRK command execution, overflow of watchdog timer, operation code trap), P5IO is 00H.

Figure 13-2 shows the P5IO configuration.

[Description of each bit]

- FSEL0, FSEL1 (bits 6 and 7)
  Multiplication select flags for PLL in the USB macro. (0, 0) = x2, (0, 1) = x4, (1,0) = x3, (1, 1) = prohibited
- SCK4IO (bit 5)
  I/O switching flag for P10_3/SIOCK4 pin. I/O switching by P10IO3 is also required.
  "0" = input, "1" = output
- SCK4INV (bit 4)
  Polarity select flag for P10_3/SIOCK4 pin. "0" = positive polarity, "1" = negative polarity
- STBSEL (bit 3)
  80 system/68 system select flag for secondary function output of P3_2/RDn. "0" = 80 system, "1" = 68 system
- MECKEN (bit 2)
  Clock supply select signal for Media control block.
  "0" = clock supply, "1" = clock stop
- UDCKEN (bit 1)
  Clock supply select signal for USB DMA controller.
  "0" = clock supply, "1" = clock stop
- USBRST (bit 0)
  Hardware reset control flag for the USB macro.
  "0" = USB reset, "1" = USB reset release

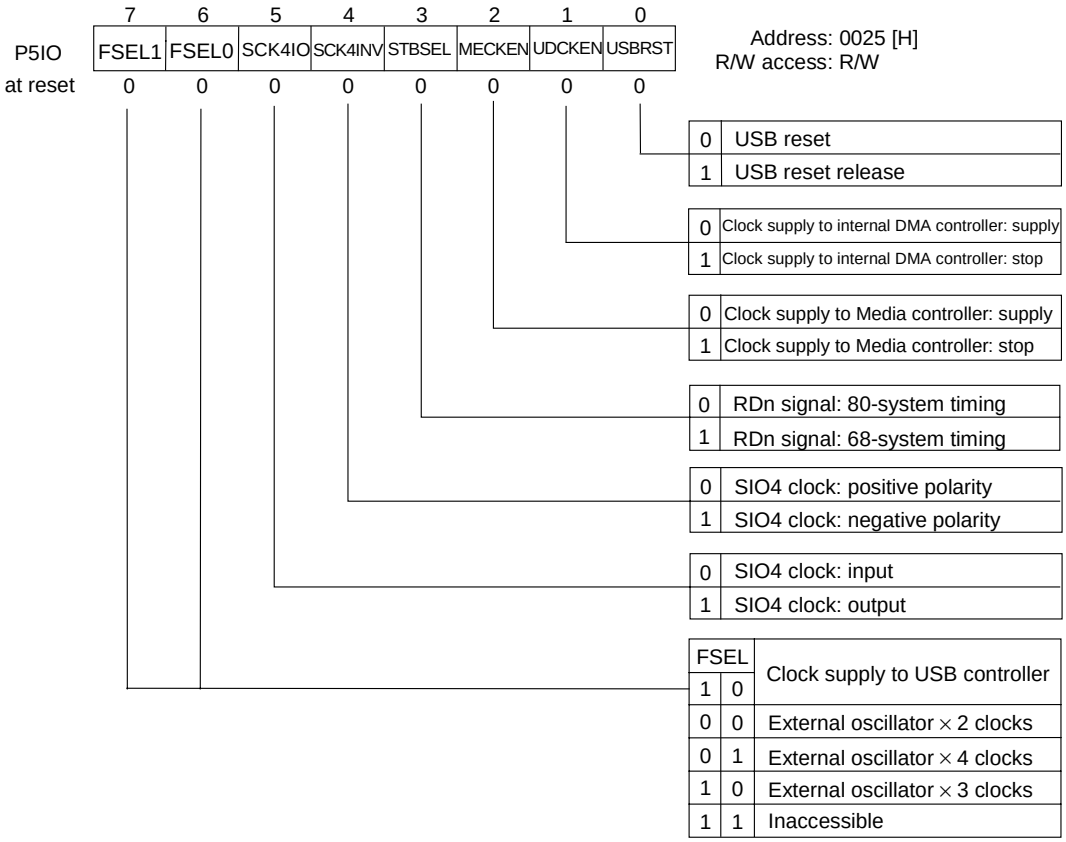


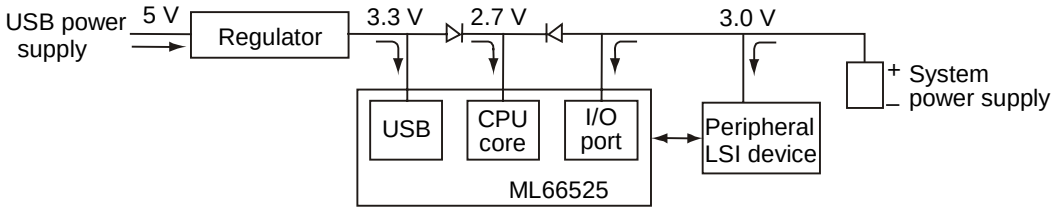
Figure 13-2 P5IO Configuration

13.5 Three Separate Power Supplies

With the use of individual power supplies for V_{DD}_IO, V_{DD}_CORE, and VBUS, the ML66525 family devices can operate under low supply current.

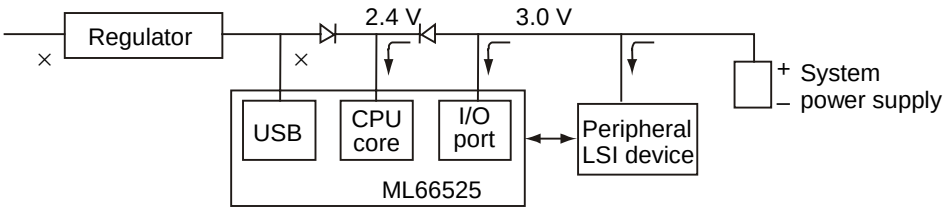
(1) When connected to the USB

When the device is connected to the USB, the USB power supply (5 V) is used and the power to the system is only supplied to the I/O port section, thereby decreasing the supply current.



(2) When not connected to the USB

When the device is not connected to the USB, the power is supplied from the system power supply to the CPU core on a low voltage, thereby decreasing the supply current.



Chapter 14

External Interrupt Functions

14. External Interrupt Functions

14.1 Overview

The ML66525 family is equipped with 8 external interrupt inputs that can be classified into 2 categories. One category is maskable interrupts, of which there are 7 (EXINT0 to EXINT3, EXINT8, EXINT9). The other category is non-maskable interrupts, and there is 1 (NMI).

EXINT0 to EXINT3, EXINT8, and EXINT9 are assigned as secondary functions of ports P6_0 to P6_3, P13_0, and P13_1. EXINT4 is assigned to P9_0/VBUSIN. If EXINT0 to EXINT4 are to be used, configure the corresponding ports as inputs of secondary functions.

NMI has its own dedicated pin.

As for internal interrupt causes, EXINT5 is assigned as the USB controller interrupt, EXINT6 as the internal DMA controller interrupt, and EXINT7 as the internal Media controller interrupt.

14.2 External Interrupt Registers

Table 14-1 lists a summary of SFRs for the control of external interrupts.

Table 14-1 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	14-2
0059	External Interrupt Control Register 1	EXI1CON	—	R/W	8	00	14-3
005A☆	External Interrupt Control Register 2	EXI2CON	—	R/W	8	0C/4C	14-4
005B☆	External Interrupt Control Register 8	EX8ICON	—	R/W	8	00	14-5

[Notes]

1. A star (☆) in the address column indicates a missing bit.
2. For details, refer to Chapter 22, "Special Function Registers (SFRs)".

14.2.1 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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), EXI0CON becomes 00H.

Figure 14-1 shows the configuration of EXI0CON.

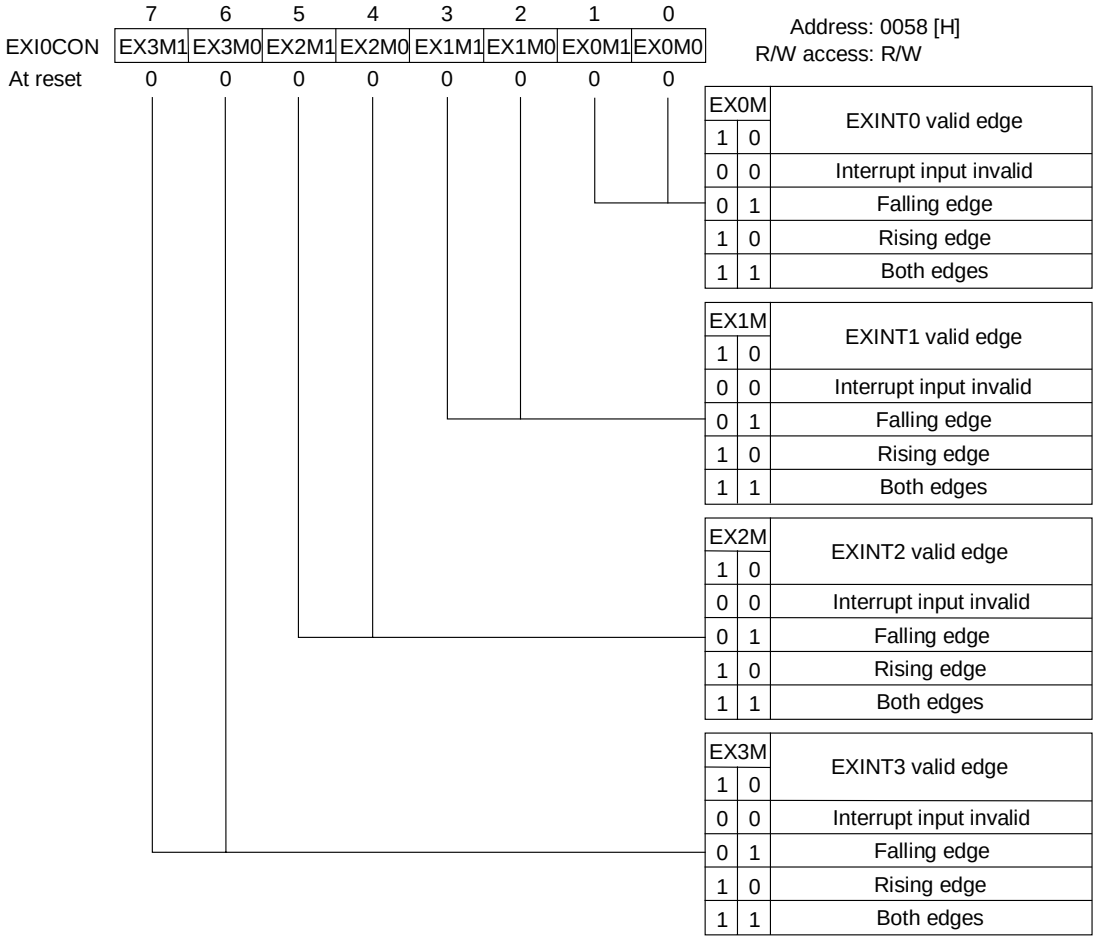


Figure 14-1 EXI0CON Configuration

(2) External interrupt control register 1 (EXI1CON)

The external interrupt control register 1 (EXI1CON) consists of 8 bits and sets Vbus detection interrupt (EXINT4), internal USB controller interrupt (EXINT5), internal DMA interrupt (EXINT6), and internal Media controller interrupt (EXINT7).

For each 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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), EXI1CON becomes 00H.

Figure 14-2 shows the configuration of EXI1CON.

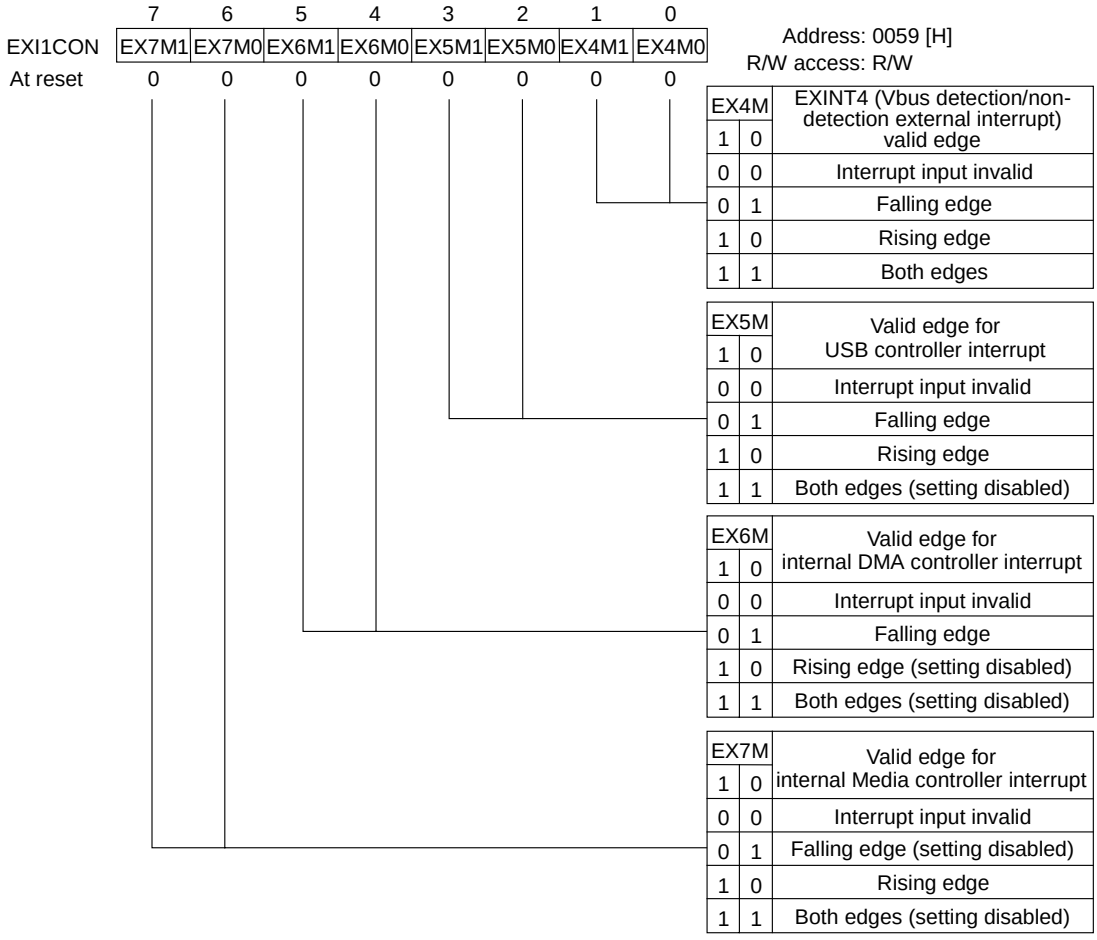


Figure 14-2 EXI1CON Configuration

[Notes]
1. For the valid edge for USB controller interrupt, specify "falling edge" when bit 4 of the POLSEL register (address 1A30H) is "0", and specify "rising edge" when bit 4 of the POLSEL register is "1".
2. Specify "falling edge" for the valid edge for internal DMA controller interrupt.
3. Specify "rising edge" for the valid edge for Media controller interrupt.

(3) External interrupt control register 2 (EXI2CON)

The external interrupt control register 2 (EXI2CON) consists of 6 bits. The external interrupts EXINT8 and EXINT9 are set by bits 0 and 1 (EX8M0 and EX8M1). 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 an NMI routine.

When reset (RESn 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 14-3 shows the configuration of EXI2CON.

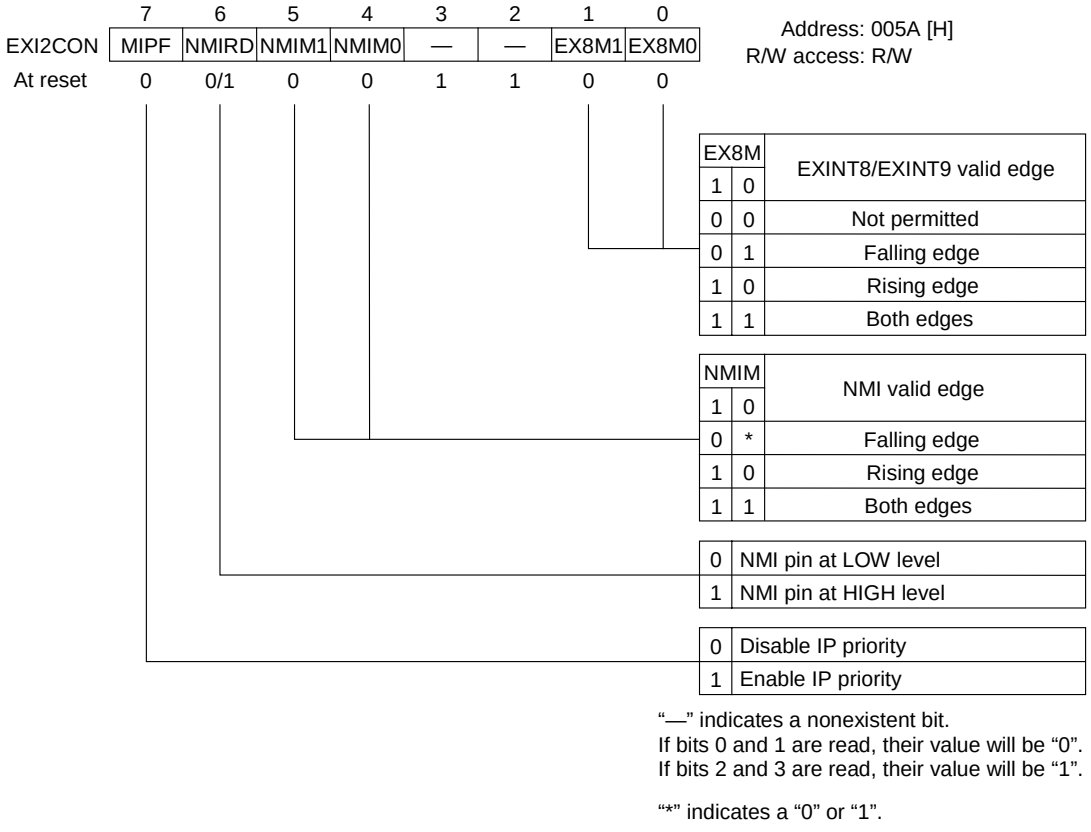


Figure 14-3 EXI2CON Configuration

(4) External interrupt control register 8 (EX8ICON)

The external interrupt control register 8 consists of 2 bits.

Bit 0 (EXI8S) of EX8ICON enables or disables external interrupt 8.

Bit 1 (EXI9S) of EX8ICON enables or disables external interrupt 9.

EX8ICON can be read from or written to by software.

When writing to bits 2 to 7, be sure to write "0" to these bits. If read, a value of "0" will be always obtained for bit 2 through bit 7.

At reset (RESn signal input, BRK instruction execution, overflow of watchdog timer, opcode trap), EX8ICON is 00H.

Figure 14-4 shows the configuration of EX8ICON.

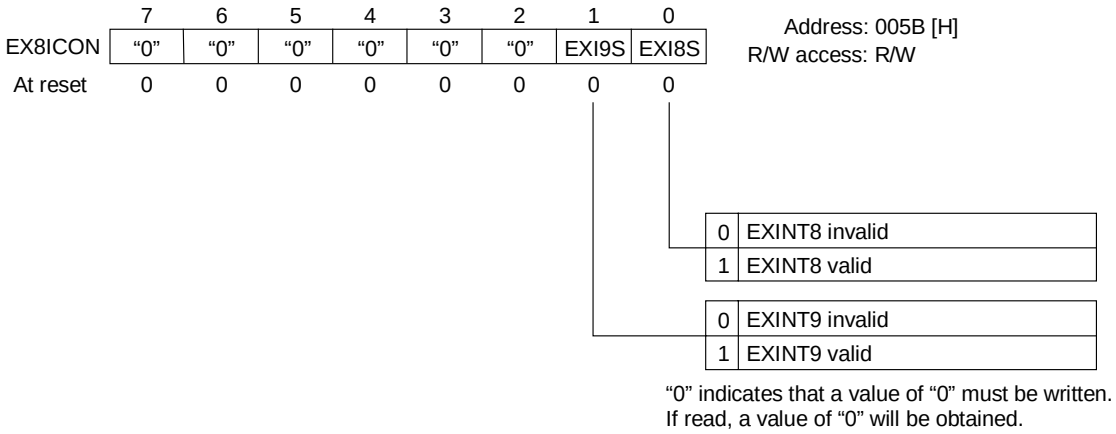


Figure 14-4 EX8ICON Configuration

[Note]

If both EXINT8 and EXINT9 are enabled and then an interrupt has occurred, it is necessary to determine which interrupt has occurred by checking the port state. If it is found that a pulse-like signal caused the interrupt at EXINT8 or EXINT9 or both, EXINT8 and EXINT9 cannot be enabled concurrently to be used as external interrupts.

14.2.2 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 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).

(3) 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).

(4) External interrupt control register 8 (EX8ICON)
If EXINT8 is to be used, set bit 0 (EXI8S) to "1".
If EXINT9 is to be used, set bit 1 (EXI9S) to "1".

(5) External interrupt control register 1 (EXI1CON)
If EXINT8 and EXINT9 are to be used, specify the valid edge with bits 0 and 1 (EX8M0, EX8M1).

(6) 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".

14.3 EXINT0 to EXINT9 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 3 (IRQ0 to IRQ3).

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 3 (IE0 to IE3).

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, 4 and 7 (IP0, IP2, IP4 and IP7).

Table 14-2 lists the vector addresses for each pin input of EXINT0 to 9 and the interrupt processing flags.

*n (n = 1 to 12) in the table indicates the register in which each flag is allocated.

Table 14-2 EXINT0 to EXINT9 Vector Addresses and Interrupt Processing Flags

Interrupt factor	Vector address [H]	Interrupt request	Interrupt enable	Priority level	
				1	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 (Vbus detect interrupt)	002A	QINT4 *3	EINT4 *6	P1INT4	P0INT4 *9
EXINT5 (internal USB controller interrupt)	002C	QINT5	EINT5	P1INT5	P0INT5
EXINT6 (Internal DMA controller interrupt)	002E	QINT6	EINT6	P1INT6	P0INT6
EXINT7 (Internal Media controller interrupt)	0030	QINT7	EINT7	P1INT7	P0INT7
EXINT8, 9 (External interrupts 8, 9)	0046	QINT8 *10	EINT8 *11	P1INT8	P0INT8 *12
Symbols of registers that contain interrupt processing flags		IRQ0 *1 IRQ1 *2 IRQ2 *3 IRQ3 *10	IE0 *4 IE1 *5 IE2 *6 IE3 *11	IP0 *7 IP2 *8 IP4 *9 IP7 *12	
	Reference page	15-12 15-13 15-14 15-15	15-17 15-18 15-19 15-20	15-22 15-23 15-25 15-28	

For further details regarding interrupt processing, refer to Chapter 15, "Interrupt Processing Functions".

Chapter 15

Interrupt Processing Functions

15. Interrupt Processing Functions

15.1 Overview

The ML66525 family has 31 types of interrupts (7 external and 24 internal). These are assigned to 23 vectors. One of the external interrupts is a non-maskable interrupt. Three levels of priority can be set for maskable interrupts.

Table 15-1 lists interrupts and their corresponding vector addresses.

Table 15-1 Interrupts and Their Corresponding Vector Addresses

Vector address [H]	Interrupt
0008	NMI pin input (non-maskable interrupt)
000A	EXINT0 pin input (external interrupt 0)
001A	Timer 0 overflow
001C	EXTINT1 pin input (external interrupt 1)
001E	EXTINT2 pin input (external interrupt 2)
0020	EXTINT3 pin input (external interrupt 3)
0026	Timer 3 overflow
002A	EXTINT4 pin input (external interrupt 4, Vbus detection use only)
002C	Internal USB controller interrupt
002E	Internal DMA controller interrupt
0030	Internal Media controller interrupt
0032	Timer 7 overfolw
0036	Timer 4 overflow
0038	SIO1 transmit buffer empty, transmit completion, receive completion
003A	Timer 5 overflow
003E	SIO3 transmit/receive completion SIO1 transmit buffer empty, transmit completion, receive completion
0040	SIO4 transfer completion
0042	Timer 6 overflow
0044	A/D conversion, select mode cancelled
0046	EXINT8/EXINT9 pin inputs (external interrupts 8,9)
0048	Real-time counter output (cycle: 0.125 to 1 second)
006A	PWC0 overflow, PWC0 and PWR0 match
006C	PWC1 overflow, PWC1 and PWR1 match
0072	Timer 9 overflow

15.2 Interrupt Function Registers

Table 15-2 lists a summary of SFRs for interrupt processing.

Table 15-2 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	2-17
0005		PSWH				00	
005A☆	External interrupt control register 2	EXI2CON	—	R/W	8	0C/4C	14-4
0030☆	Interrupt request register 0	IRQ0	—	R/W	8	00	15-12
0031☆	Interrupt request register 1	IRQ1	—	R/W	8	00	15-13
0032☆	Interrupt request register 2	IRQ2	—	R/W	8	00	15-14
0033☆	Interrupt request register 3	IRQ3	—	R/W	8	00	15-15
005C☆	Interrupt request register 4	IRQ4	—	R/W	8	E0	15-16
0034☆	Interrupt enable register 0	IE0	—	R/W	8	00	15-17
0035☆	Interrupt enable register 1	IE1	—	R/W	8	00	15-18
0036☆	Interrupt enable register 2	IE2	—	R/W	8	00	15-19
0037☆	Interrupt enable register 3	IE3	—	R/W	8	00	15-20
005D☆	Interrupt enable register 4	IE4	—	R/W	8	E0	15-21
0038☆	Interrupt priority control register 0	IP0	—	R/W	8	00	15-22
003A	Interrupt priority control register 2	IP2	—	R/W	8	00	15-23
003B☆	Interrupt priority control register 3	IP3	—	R/W	8	00	15-24
003C☆	Interrupt priority control register 4	IP4	—	R/W	8	00	15-25
003D☆	Interrupt priority control register 5	IP5	—	R/W	8	00	15-26
003E☆	Interrupt priority control register 6	IP6	—	R/W	8	00	15-27
003F☆	Interrupt priority control register 7	IP7	—	R/W	8	00	15-28
005E☆	Interrupt priority control register 8	IP8	—	R/W	8	00	15-29
005F☆	Interrupt priority control register 9	IP9	—	R/W	8	FC	15-30

[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 22, "Special Function Registers (SFRs)".

15.3 Description of Interrupt Processing

15.3.1 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 (RESn 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 14-4

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 an RTI instruction at the end of the NMI routine.

When an 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.

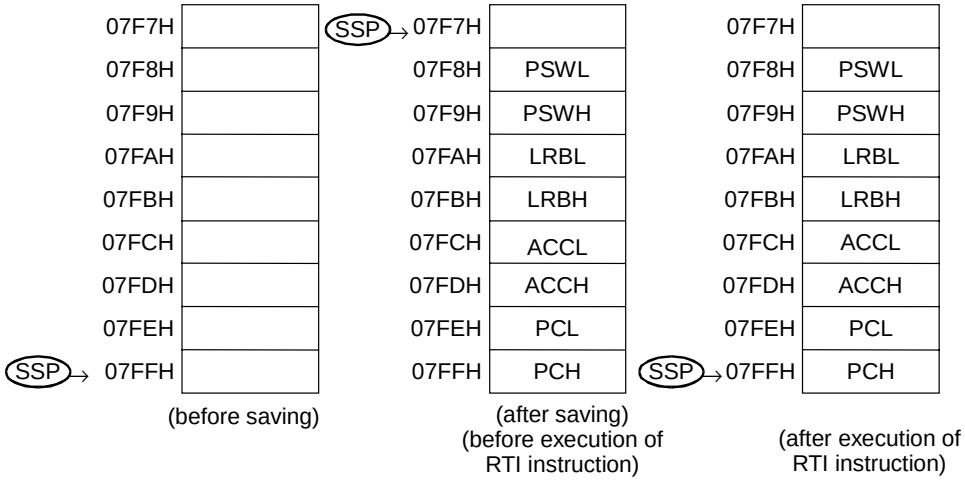
- Restore the program status word (PSW)
- Restore the local register base (LRB)
- Restore the accumulator (ACC)
- Restore the program counter (PC)
- Enable maskable interrupts
- Enable multiple interrupts by the non-maskable interrupt

Figure 15-1 shows examples of saving and restoring the PC, ACC, LRB and PSW.

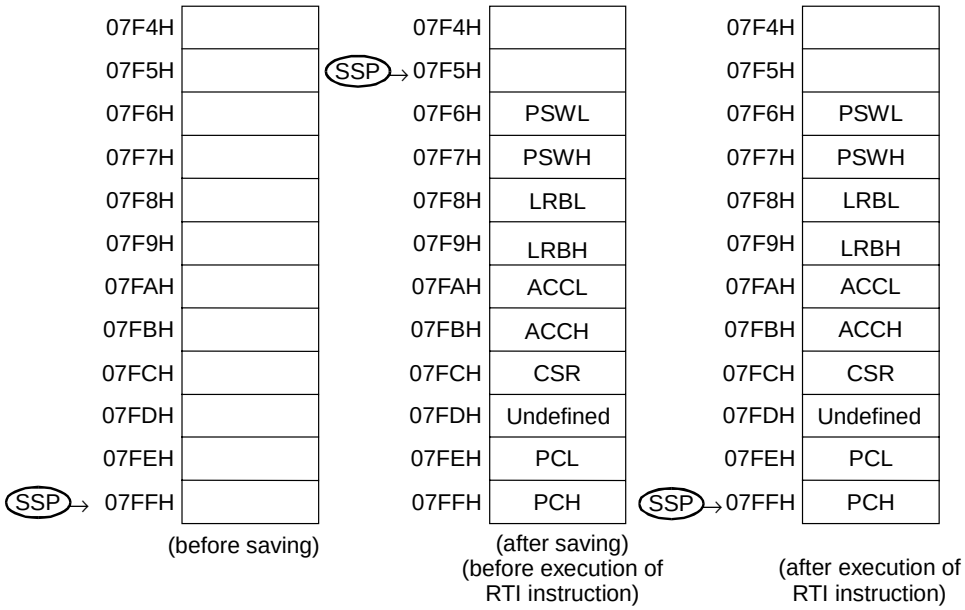
[Note]

If the program memory space has been expanded to 1 MB, 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 64 KB program memory space)



• Interrupt processing example (for a program memory space greater than 64 KB)



SSP: System Stack Pointer

Figure 15-1 Examples of Saving and Restoring the PC, ACC, LRB and PSW

15.3.2 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 2-17

(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 14-4

(5) Interrupt priority control registers (IP0, IP2 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 15.3.3, "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 15-2 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.

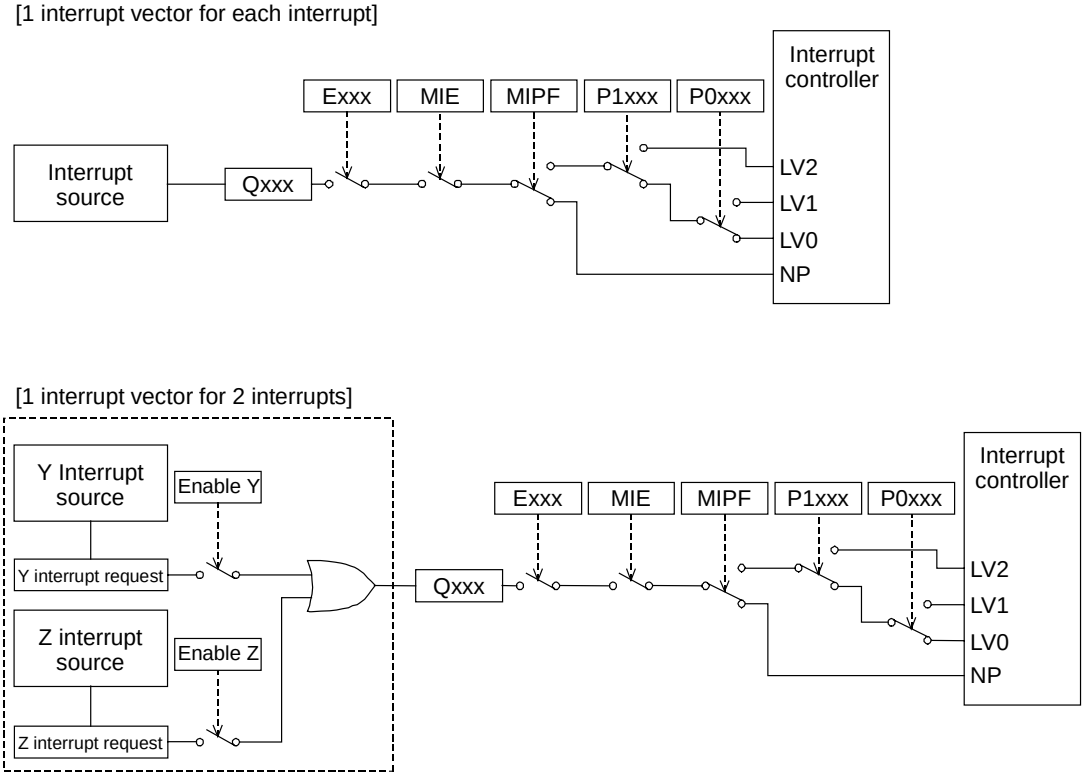


Figure 15-2 Maskable Interrupt Control Block Diagram

Table 15-3 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 15-3) 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 15-3 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	Timer 0 overflow	001A	QTM0OV	ETM0OV	P1TM0OV	P0TM0OV
3	EXINT1 pin input (external interrupt 1)	001C	QINT1	EINT1	P1INT1	P0INT1
4	EXINT2 pin input (external interrupt 2)	001E	QINT2	EINT2	P1INT2	P0INT2
5	EXINT3 pin input (external interrupt 3)	0020	QINT3	EINT3	P1INT3	P0INT3
6	Timer 3 overflow	0026	QTM3OV	ETM3OV	P1TM3OV	P0TM3OV
7	EXINT4 input (Vbus detect interrupt)	002A	QINT4	EINT4	P1INT4	P0INT4
8	EXINT5 input (internal USB controller interrupt)	002C	QINT5	EINT5	P1INT5	P0INT5
9	EXINT6 input (internal DMA controller interrupt)	002E	QINT6	EINT6	P1INT6	P0INT6
10	EXINT7 input (internal Media controller interrupt)	0030	QINT7	EINT7	P1INT7	P0INT7
11	Timer overflow	0032	QTM7OV	ETM7OV	P1TM7OV	P0TM7OV
12	Timer 4 overflow	0036	QTM4OV	ETM4OV	P1TM4OV	P0TM4OV
13	SIO1 transmit buffer empty, transmit complete, receive complete	0038	QSIO1	ESIO1	P1SIO1	P0SIO1
14	Timer 5 overflow	003A	QTM5OV	ETM5OV	P1TM5OV	P0TM5OV
15	SIO3 transmit-receive complete	003E	QSIO3	ESIO3	P1SIO3	P0SIO3
	SIO6 transmit buffer empty, transmit complete, receive complete	003E	QSIO6	ESIO6	P1SIO6	P0SIO6
16	SIO4 transfer complete	0040	QINT8	EINT8	P1INT8	P0INT8
17	Timer 6 overflow	0042	QTM6OV	ETM6OV	P1TM6OV	P0TM6OV
18	One cycle of A/D conversion scan channels complete, A/D conversion select mode complete	0044	QAD	EAD	P1AD	P0AD
19	EXINT8/EXINT9 pin input (external interrupts 8, 9)	0046	QINT8	EINT8	P1INT8	P0INT8
20	Real-time counter output (interval: 0.125 to 1 s)	0048	QRTC	ERTC	P1RTC	P0RTC
21	PWC0 overflow, match of PWC0 and PWR0	006A	QPWM0	EPWM0	P1PWM0	P0PWM0
22	PWC1 overflow, match of PWC1 and PWR1	006C	QPWM1	EPWM1	P1PWM1	P0PWM1
23	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 an RTI instruction at the end of the maskable interrupt routine.

When an 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 15-1 shows examples of saving and storing the PC, ACC, LRB and PSW.

[Note]

If the program memory space has been expanded to 1 MB, 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.

15.3.3 Priority Control of Maskable Interrupts

The ML66525 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 15-3.) If several interrupts are awaiting processing, the interrupt vector with the lowest address will be processed first. (See Table 15-3.)

(2) Multiple interrupt control

During execution of an interrupt routine, other maskable interrupts may be enabled. This is known as "multiple interrupt control". Multiple interrupt control is carried out by enabling multiple interrupt (MIE = "1") within the maskable interrupt routine when a maskable interrupt occurs.

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 15-4.)

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 15-3.)

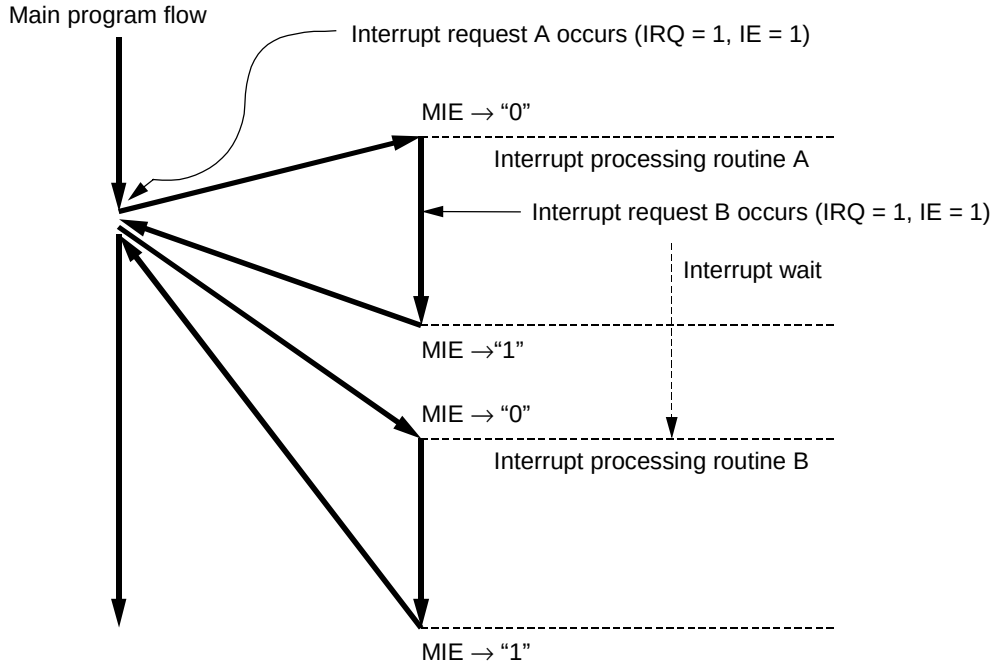


Figure 15-3 Fundamental Interrupt Control

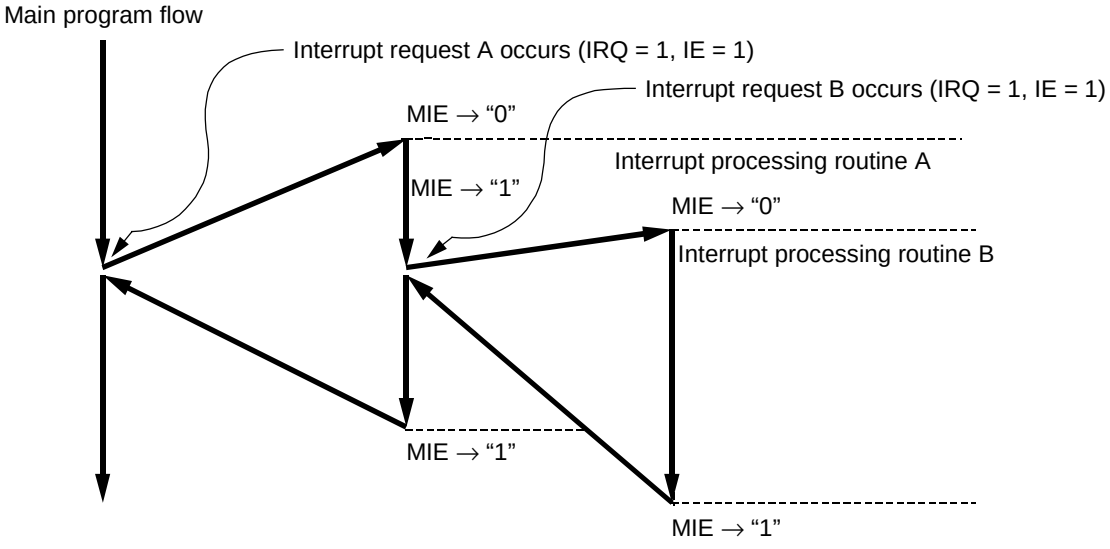


Figure 15-4 Multiple Interrupt Control

15.4 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.

15.4.1 Interrupt Request Registers (IRQ0 to IRQ4)

(1) Interrupt request register 0 (IRQ0)

Interrupt request register 0 (IRQ0) consists of 1 bit. The bit is set to "1" by external interrupt 0 (bit 0).

IRQ0 can be read or written by the program. However, if writing to bits 1 through 7, always write those bits as "0". If read, a value of "0" will always be obtained for bits 1 through 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IRQ0 becomes 00H.

Figure 15-5 shows the configuration of IRQ0.

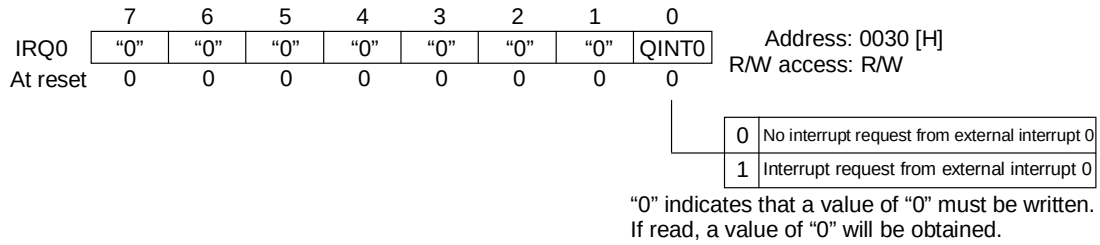


Figure 15-5 IRQ0 Configuration

(2) Interrupt request register 1 (IRQ1)

Interrupt request register 1 (IRQ1) consists of 5 bits. Bits are set to "1" corresponding to overflow of timer 0 (bit 0), external interrupts 1 to 3 (bits 1 to 3), overflow of timer 3 (bit 6), and SIO0 transmit buffer empty/transmit complete/receive complete (bit 7).

IRQ1 can be read or written by the program. However, if writing to bits 4, 5 and 7, always write those bits as "0". If read, a value of "0" will always be obtained for bits 4, 5 and 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IRQ1 becomes 00H.

Figure 15-6 shows the configuration of IRQ1.

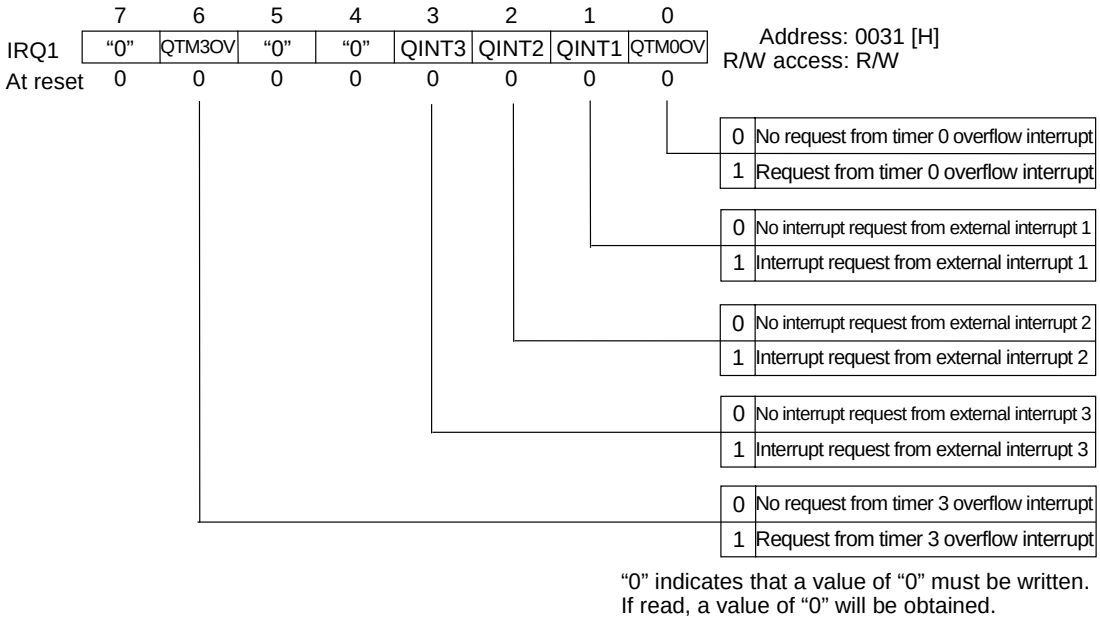


Figure 15-6 IRQ1 Configuration

(3) Interrupt request register 2 (IRQ2)

Interrupt request register 2 (IRQ2) consists of 7 bits. Bits are set to "1" corresponding to Vbus detect interrupt (EXINT4, bit 0), internal USB controller interrupt (EXINT5, bit 1), internal DMA controller interrupt (EXINT6, bit 2), internal Media controller interrupt (EXINT7, bit 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 bit 5, always write this bit as "0". If read, a value of "0" will always be obtained for bit 5.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IRQ2 becomes 00H.

Figure 15-7 shows the configuration of IRQ2.

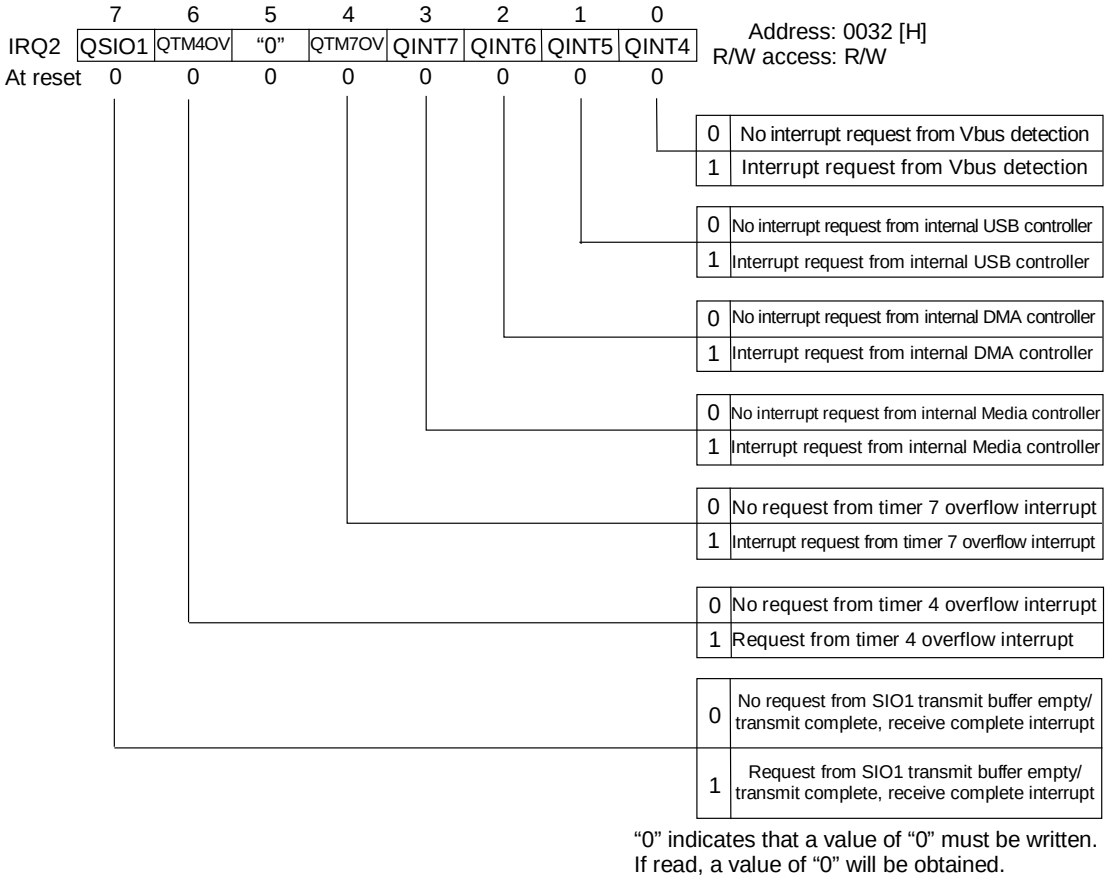


Figure 15-7 IRQ2 Configuration

(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), SIO3/SIO6 transmit-receive completion (bit 2), SIO4 transmit-receive completion (bit 3), overflow of timer 6 (bit 4), A/D conversion complete/select mode complete (bit 5), external interrupt 8/9 (bit 6) and real-time counter output (bit 7).

IRQ3 can be read or written by the program. However, if writing to bit 1, always write this bit as "0". If read, a value of "0" will always be obtained for bit 1.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IRQ3 becomes 00H.

Figure 15-8 shows the configuration of IRQ3.

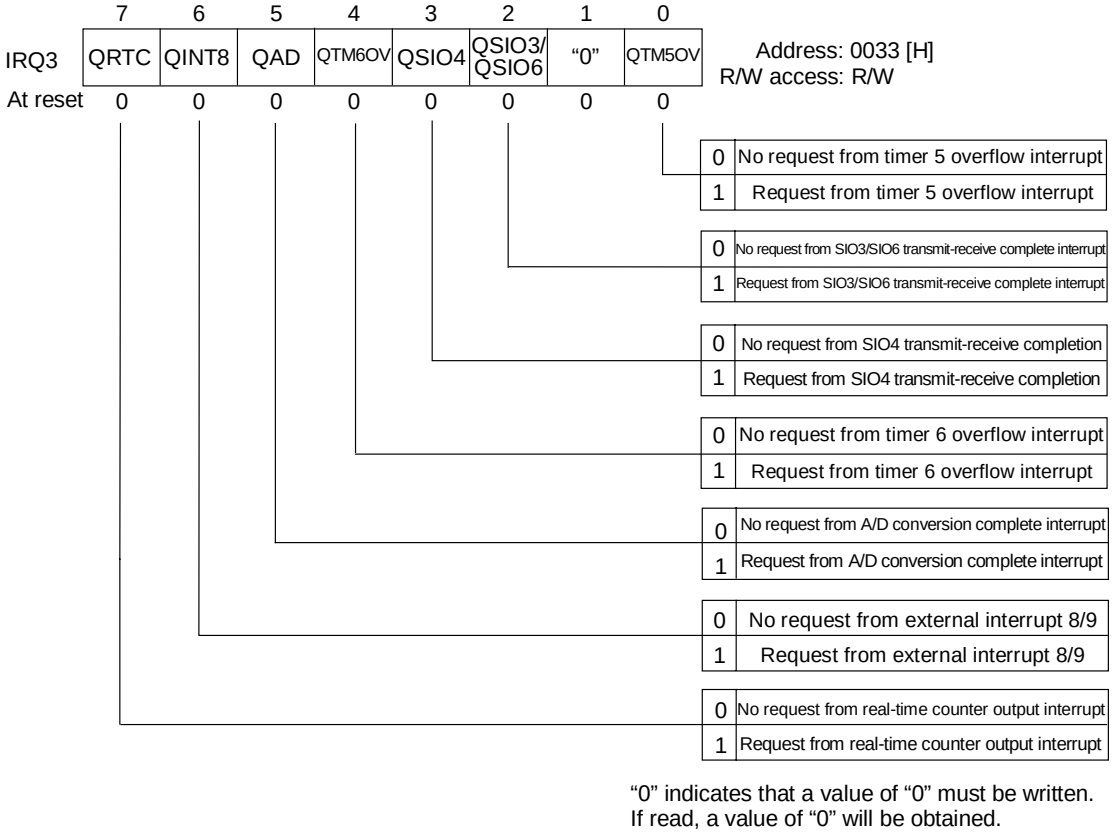


Figure 15-8 IRQ3 Configuration

(5) Interrupt request register 4 (IRQ4)

Interrupt request register 4 (IRQ4) consists of 3 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), 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.

If writing to bits 2 and 3, always write these bits as "0".

When read, a value of "0" will always be obtained for bits 2 and 3.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IRQ4 becomes E0H.

Figure 15-9 shows the configuration of IRQ4.

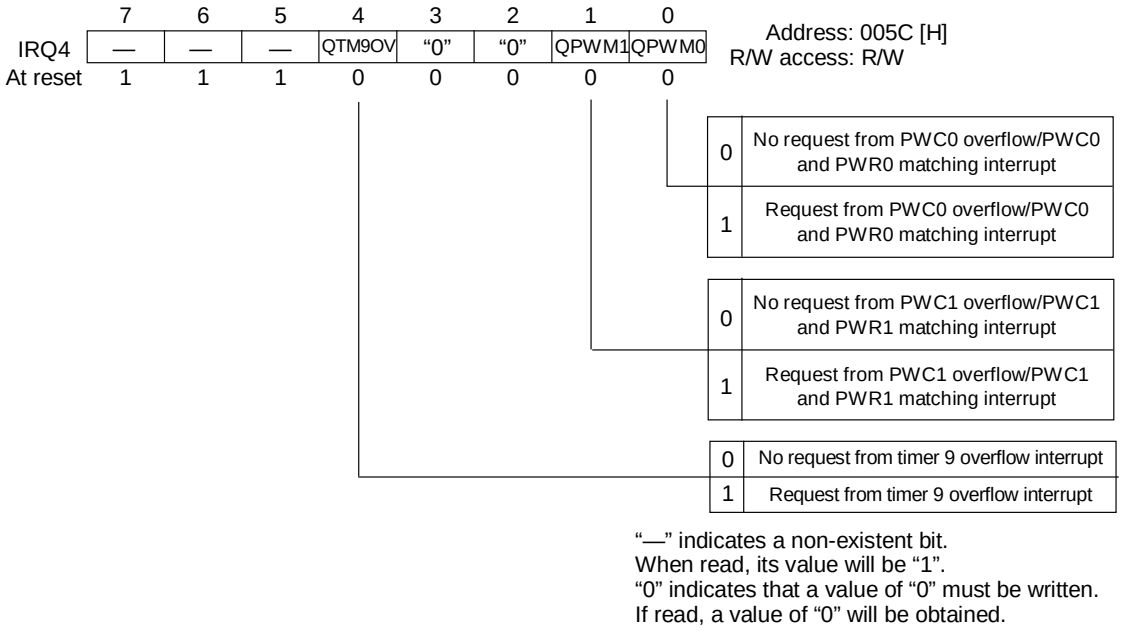


Figure 15-9 IRQ4 Configuration

15.4.2 Interrupt Enable Registers (IE0 to IE4)

(1) Interrupt enable register 0 (IE0)

Interrupt enable register 0 (IE0) consists of 1 bit. The generation of interrupts is enabled by setting the bit for external interrupt 0 (bit 0) to "1".

IE0 can be read or written by the program. However, if writing to bits 2 through 7, always write those bits as "0". If read, a value of "0" will always be obtained for bits 2 through 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IE0 becomes 00H.

Figure 15-10 shows the configuration of IE0.

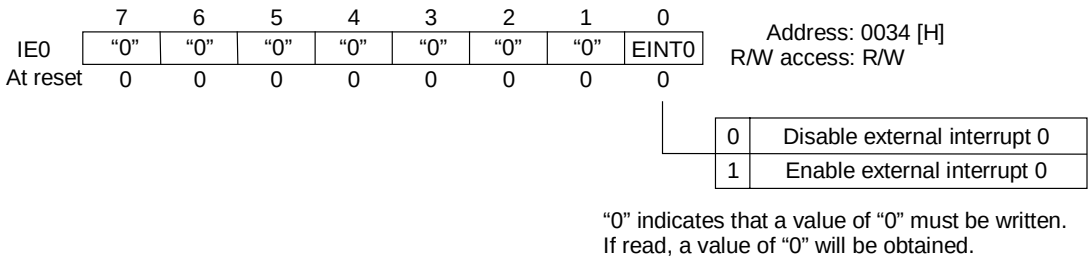


Figure 15-10 IE0 Configuration

(2) Interrupt enable register 1 (IE1)

Interrupt enable register 1 (IE1) consists of 5 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), overflow of timer 3 (bit 6).

IE1 can be read or written by the program. However, if writing to bits 4, 5 and 7, always write those bits as "0". If read, a value of "0" will always be obtained for bits 4, 5 and 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IE1 becomes 00H.

Figure 15-11 shows the configuration of IE1.

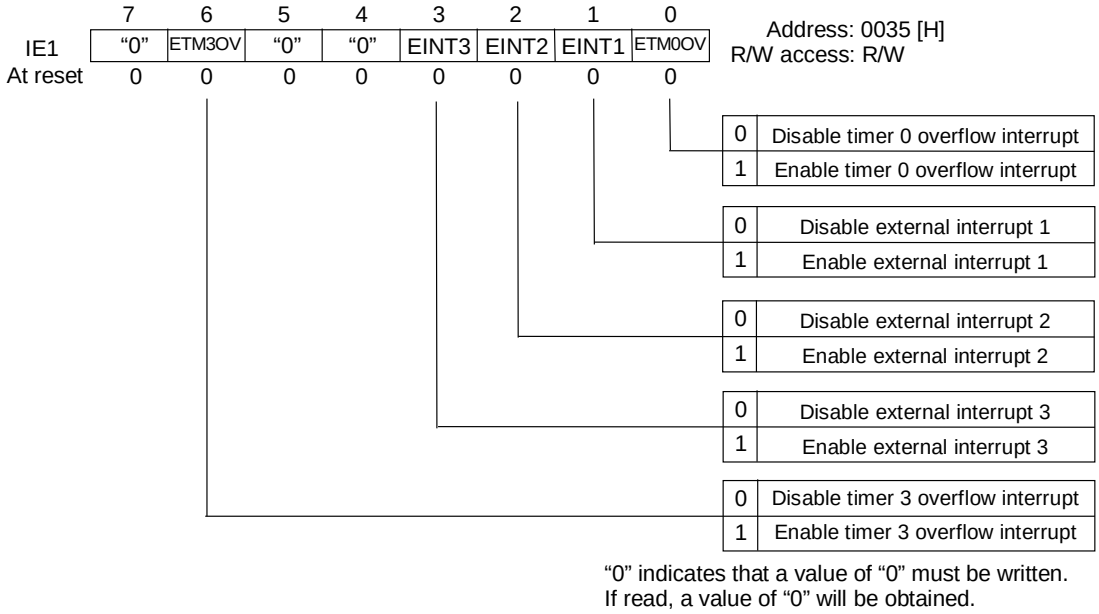


Figure 15-11 IE1 Configuration

(3) Interrupt enable register 2 (IE2)

Interrupt enable register 2 (IE2) consists of 7 bits. The generation of interrupts is enabled by setting bits to "1" corresponding to Vbus detect interrupt (EXINT4, bit 0), internal USB controller interrupt (EXINT5, bit 1), internal DMA controller interrupt (EXINT6, bit 2), and internal Media controller interrupt (EXINT7, bit 3), overflow of timer 7 (bit 4), 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 bit 5, always write those bits as "0". If read, a value of "0" will always be obtained for bit 5.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IE2 becomes 00H.

Figure 15-12 shows the configuration of IE2.

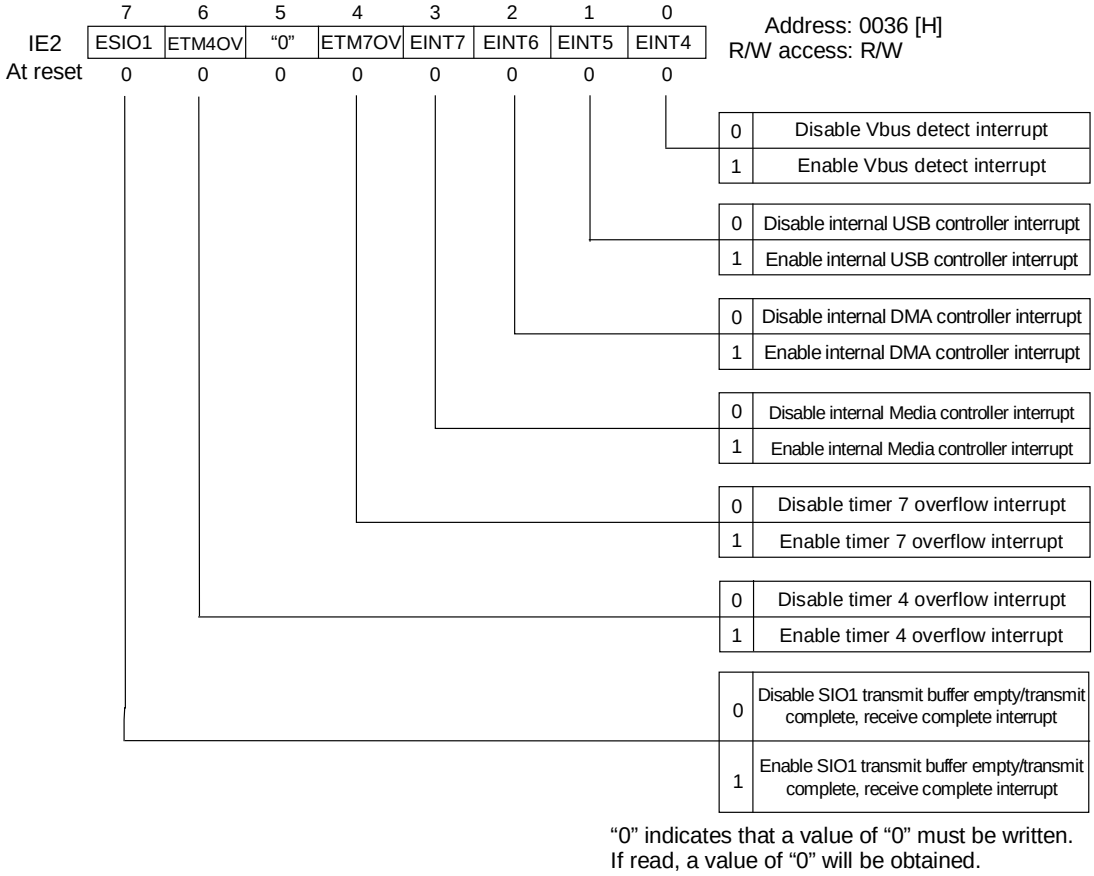


Figure 15-12 IE2 Configuration

(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), SIO3/SIO6 transmit-receive completion (bit 2), overflow of timer 6 (bit 4), A/D conversion complete (bit 5), external interrupt 8/9 (bit 6), and real-time counter output (bit 7).

IE3 can be read or written by the program. However, if writing to bit 1, always write this bit as "0". If read, a value of "0" will always be obtained for bit 1.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IE3 becomes 00H.

Figure 15-13 shows the configuration of IE3.

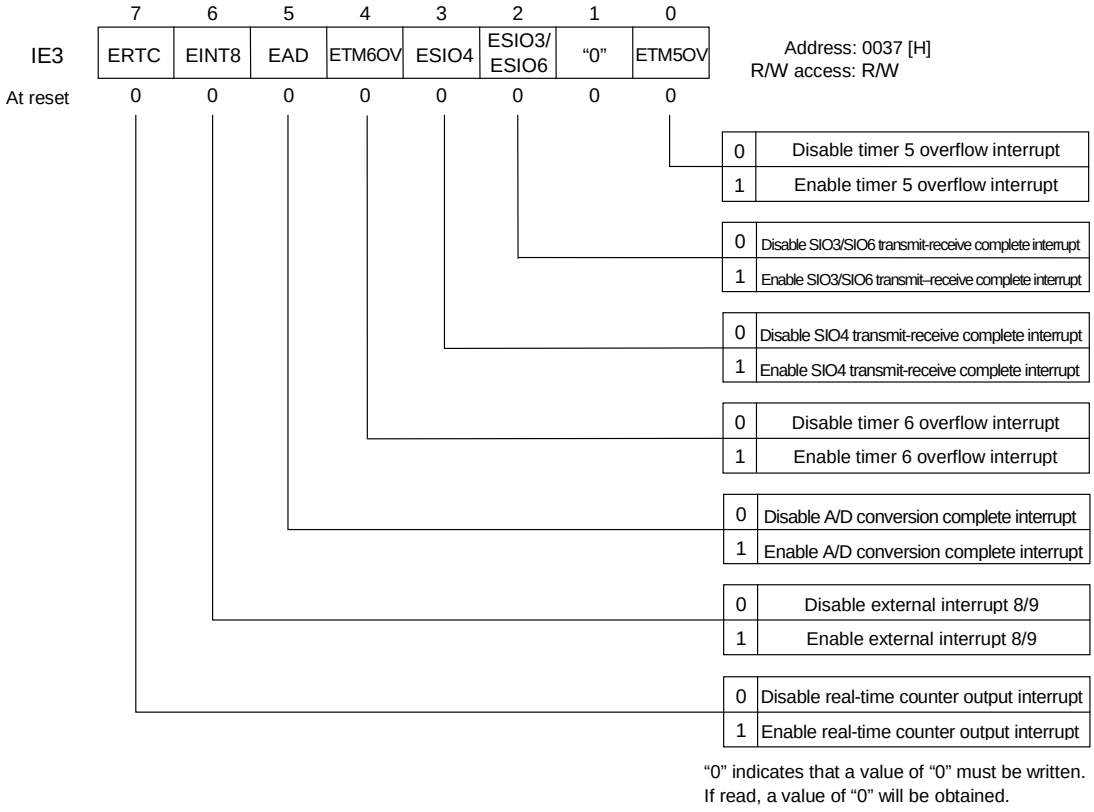


Figure 15-13 IE3 Configuration

(5) Interrupt enable register 4 (IE4)

Interrupt enable register 4 (IE4) consists of 3 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), 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.

If writing to bits 2 and 3, always write these bits as "0".

If read, a value of "0" will always be obtained for bits 2 and 3.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IE4 becomes E0H.

Figure 15-14 shows the configuration of IE4.

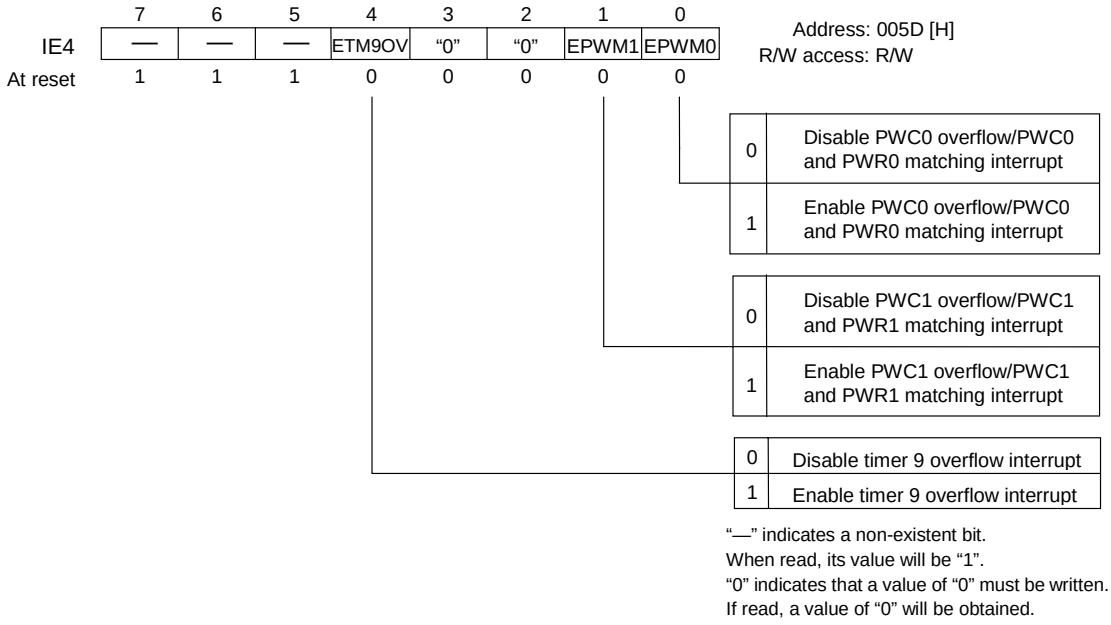


Figure 15-14 IE4 Configuration

15.4.3 Interrupt Priority Control Registers (IP0, IP2 to IP9)

(1) Interrupt priority control register 0 (IP0)

Interrupt priority control register 0 (IP0) consists of 2 bits and specifies the interrupt priority for external interrupt 0 (bits 0 and 1).

IP0 can be read or written by the program. However, if writing to bits 2 through 7, always write those bits as "0". If read, a value of "0" will always be obtained for bits 2 through 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP0 becomes 00H.

Figure 15-15 shows the configuration of IP0.

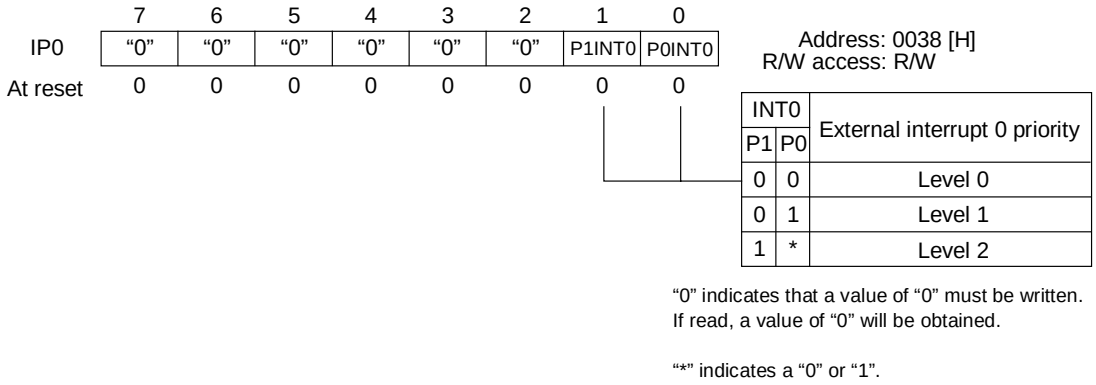


Figure 15-15 IP0 Configuration

(2) 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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP2 becomes 00H.

Figure 16-17 shows the configuration of IP2.

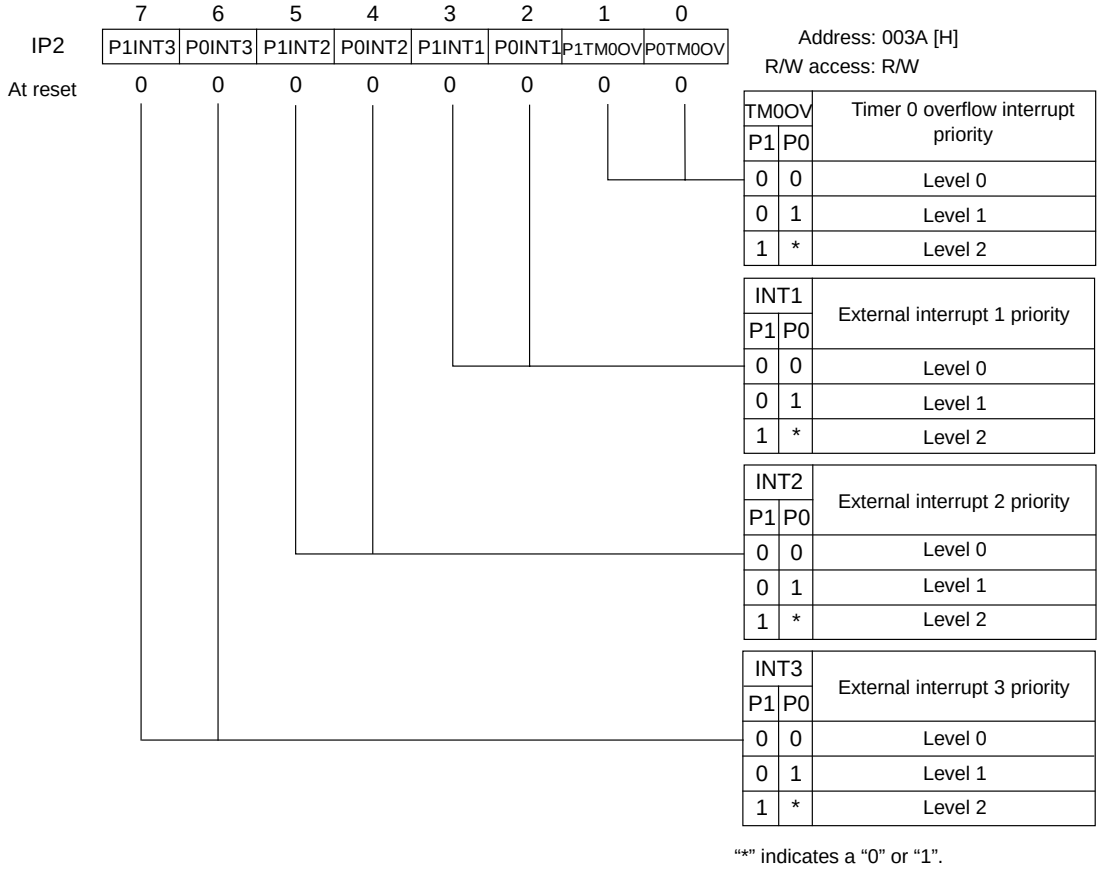


Figure 15-16 IP2 Configuration

(3) Interrupt priority control register 3 (IP3)

Interrupt priority control register 3 (IP3) consists of 2 bits and specifies interrupt priority for overflow of timer 3 (bits 4 and 5).

IP3 can be read or written by the program. However, if writing to bits 0 through 3, 6 and 7, always write those bits as "0". If read, a value of "0" will always be obtained for bits 0 through 3, 6 and 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP3 becomes 00H.

Figure 15-17 shows the configuration of IP3.

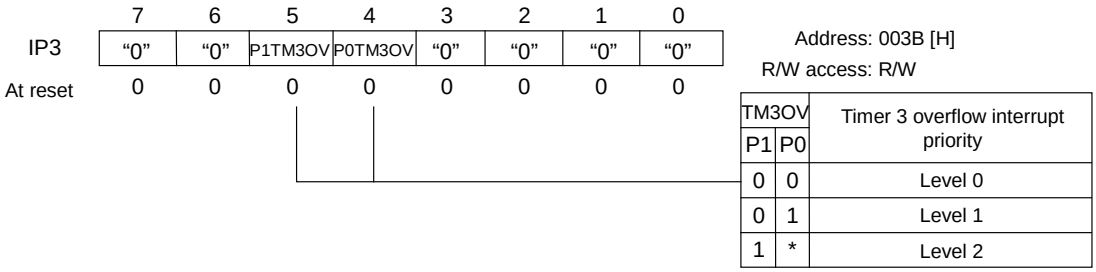


Figure 15-17 IP3 Configuration

(4) Interrupt priority control register 4 (IP4)

Interrupt priority control register 4 (IP4) consists of 8 bits and specifies interrupt priority for Vbus detect interrupt (bits 0 and 1), internal USB controller interrupt (bits 2 and 3), internal DMA controller interrupt (bits 4 and 5), and internal Media controller interrupt (bits 6 and 7).

IP4 can be read or written by the program.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP4 becomes 00H.

Figure 15-18 shows the configuration of IP4.

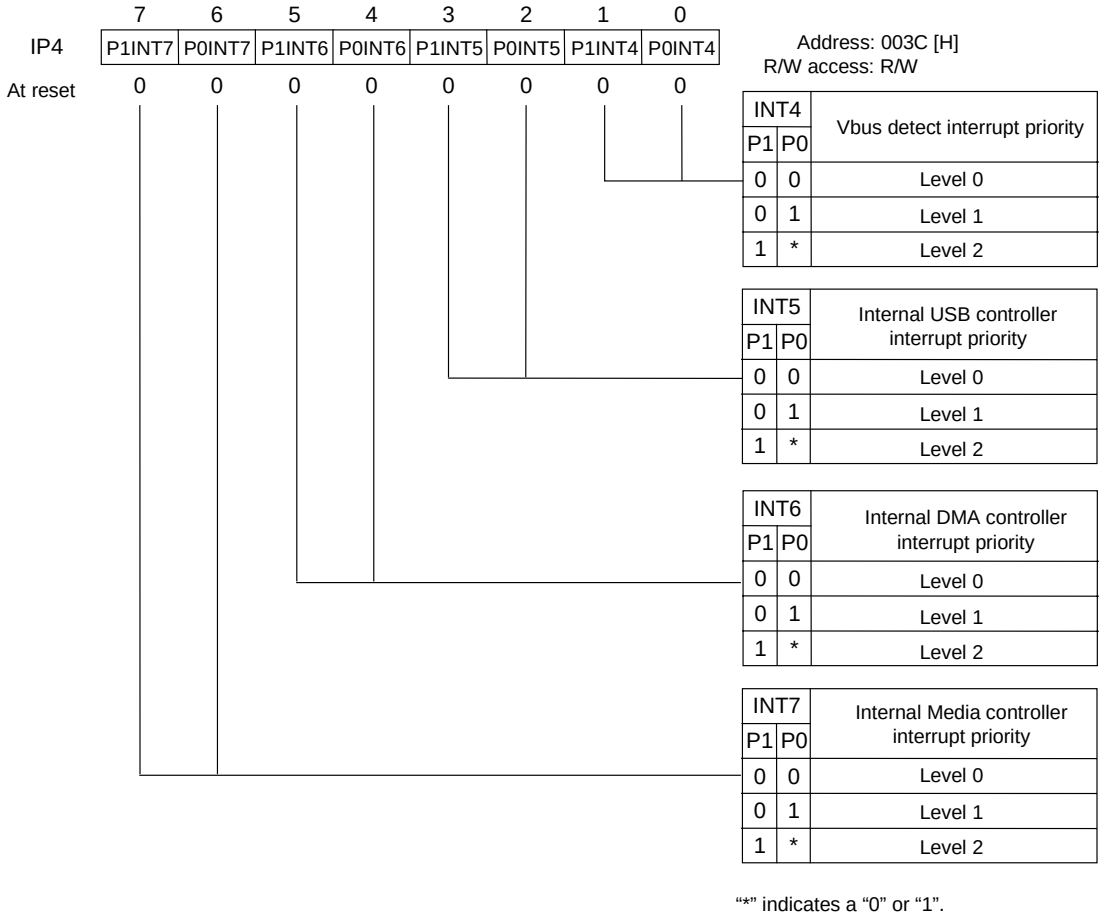


Figure 15-18 IP4 Configuration

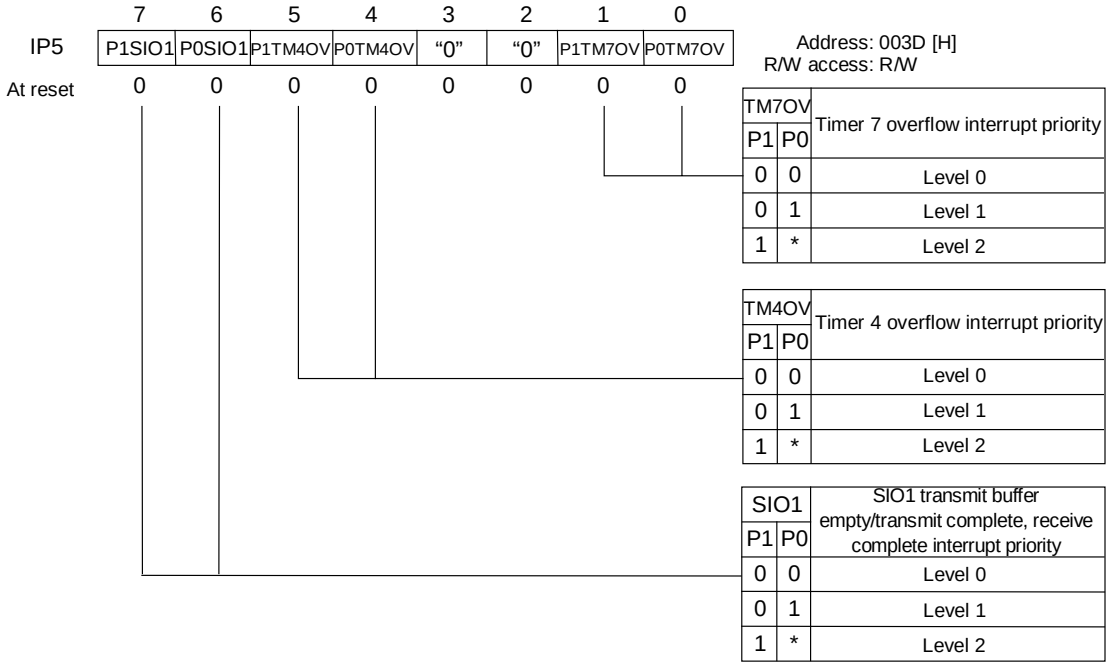
(5) Interrupt priority control register 5 (IP5)

Interrupt priority control register 5 (IP5) consists of 6 bits and specifies interrupt priority for overflow of timer 7 (bits 0 and 1), 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 2 and 3, always write those bits as "0". If read, a value of "0" will always be obtained for bits 2 and 3.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP5 becomes 00H.

Figure 15-19 shows the configuration of IP5.



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

"*" indicates a "0" or "1".

Figure 15-19 IP5 Configuration

(6) Interrupt priority control register 6 (IP6)

Interrupt priority control register 6 (IP6) consists of 6 bits and specifies interrupt priority for overflow of timer 5 (bits 0 and 1) and SIO3/SIO6 transmit-receive completion (bits 4 and 5) SIO4 transmit-receive complete (bits 6 and 7).

IP6 can be read or written by the program. However, if writing to bits 2 and 3, always write those bits as "0". If read, a value of "0" will always be obtained for bits 2 and 3.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP6 becomes 00H.

Figure 15-20 shows the configuration of IP6.

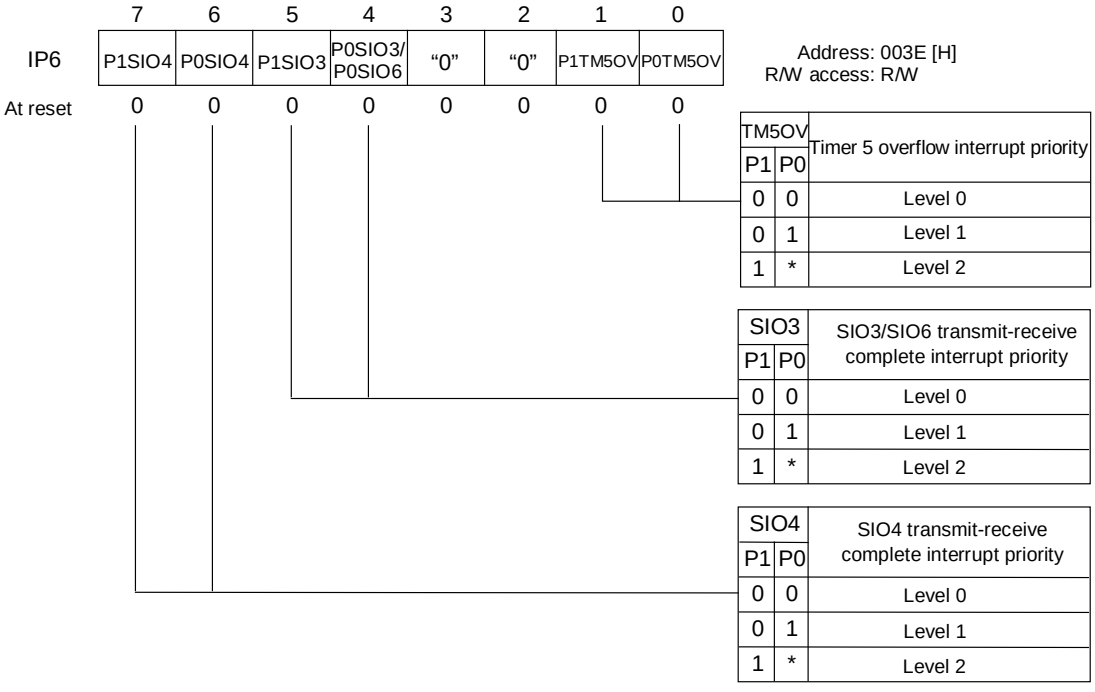


Figure 15-20 IP6 Configuration

(7) Interrupt priority control register 7 (IP7)

Interrupt priority control register 7 (IP7) consists of 8 bits and specifies interrupt priority for overflow of timer 6 (bits 0 and 1), A/D conversion complete (bits 2 and 3), external interrupt 8/9 (bits 4 and 5) and real-time counter output (bits 6 and 7).

IP7 can be read or written by the program.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP7 becomes 00H.

Figure 15-21 shows the configuration of IP7.

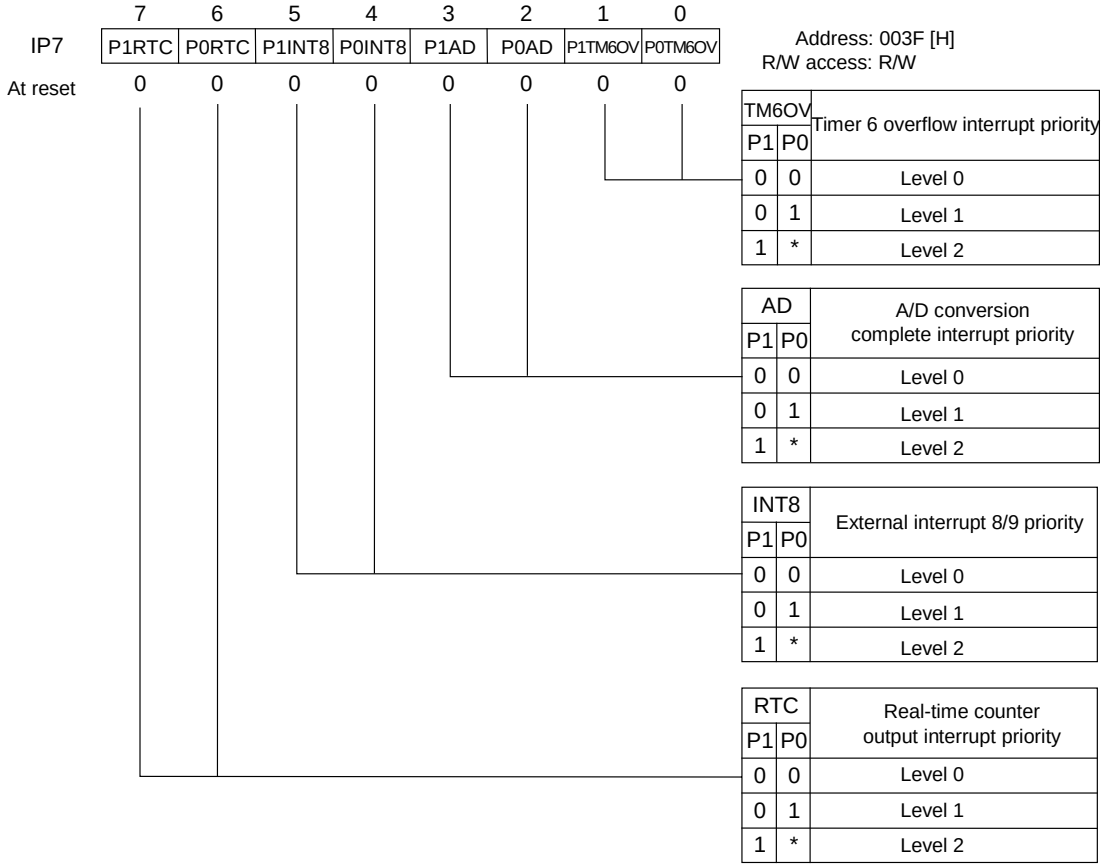


Figure 15-21 IP7 Configuration

(8) Interrupt priority control register 8 (IP8)

Interrupt priority control register 8 (IP8) consists of 4 bits and specifies interrupt priority for overflow of PWC0/matching of PWC0 and PWR0 (bits 0 and 1) and overflow of PWC1/matching of PWC1 and PWR1 (bits 2 and 3).

IP8 can be read or written by the program.

If writing to bits 4 through 7, always write these bits as "0".

If read, a value of "0" will always be obtained for bits 4 through 7 .

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP8 becomes 00H.

Figure 15-22 shows the configuration of IP8.

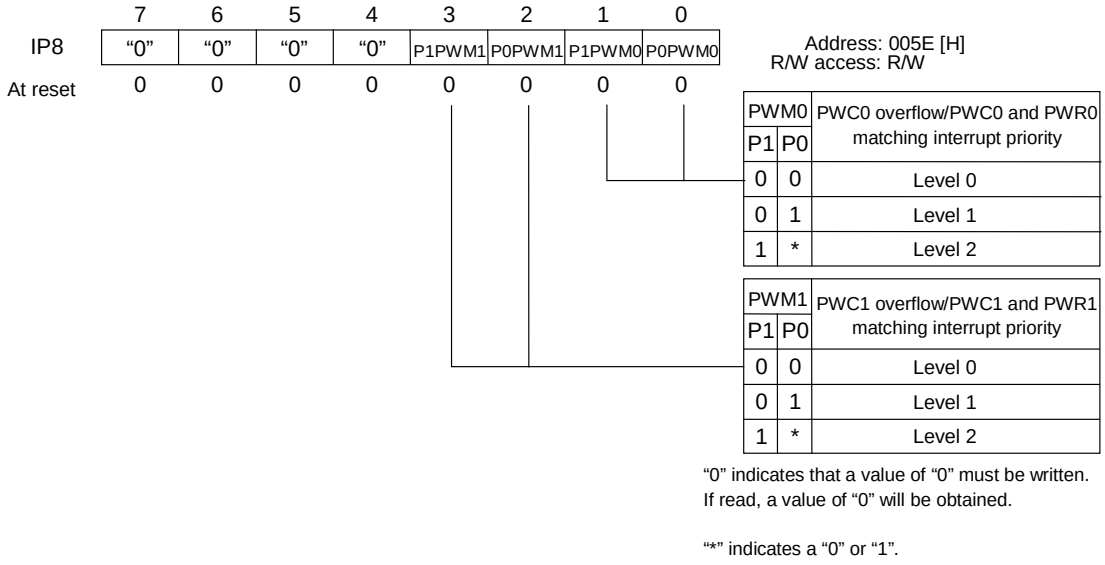


Figure 15-22 IP8 Configuration

(9) 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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), IP9 becomes FCH.

Figure 15-23 shows the configuration of IP9.

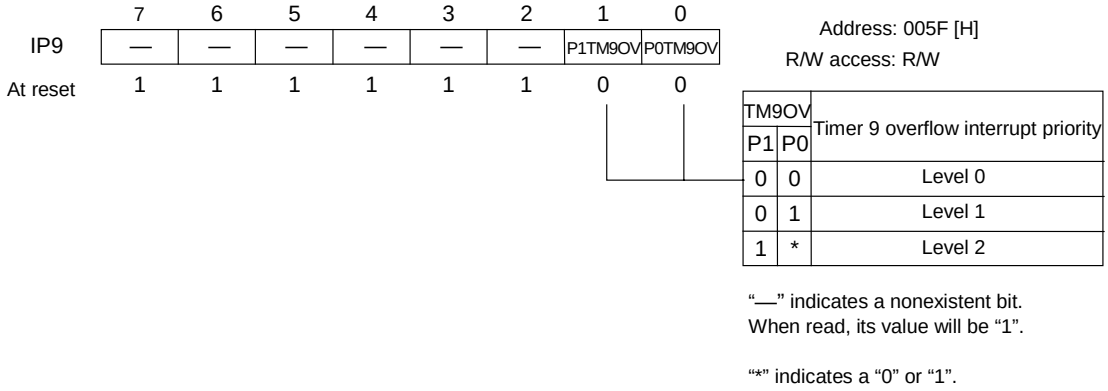


Figure 15-23 IP9 Configuration

Chapter 16

Bus Port Functions

16 Bus Port Functions

16.1 Overview

The ML66525 family can externally expand program memory (usually ROM) up to a maximum of 1 MB and data memory (usually RAM) up to a maximum of 1 MB.

Bus ports (A0 to A19, D0 to D7) and control signals (PSENn, RDn, WRn) are used to access the external program memory and external data memory.

Bus ports are assigned as the secondary functions of port 0 (P0), port 1 (P1), port 2 (P2) and port 4 (P4). The 20 address (A0 to A19) lines and 8 data (D0 to D7) lines of the bus are separate. Unnecessary upper addresses can be reset as normal I/O ports.

PSENn (P3_1) is used as a strobe signal to read the external program memory. RDn (P3_2) and WRn (P3_3) are used as read and write strobes for external data memory.

16.2 Port Operation

16.2.1 Port Operation When Accessing Program Memory

When accessing internal program memory (addresses 0H to 1FFFFH with the EAn pin at a high level), P0, P1, P2, P3_1 and P4 operate as I/O ports.

When accessing external program memory (the EAn pin at a low level or addresses 20000H to FFFFFH with the EAn 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 PSENn output port.

If the EAn 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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap). If the EAn pin is at a high level, before external program memory is accessed, it is necessary to switch to bus port functions by setting secondary function control registers and mode registers.

Of the ports that are automatically set as bus port functions when the EAn 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 16-1 lists the operation of P0, P1, P2, P3_1 and P4 during a program memory access.

Table 16-1 P0, P1, P2, P3_1 and P4 Operation During Program Memory Access

Memory to be accessed	Address	P0 operation	P1, P2, P4 operation	P3_1 operation
Internal program	When EAn = H, 0H to 1FFFFH	I/O port		
External program	When EAn = 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, PSENn output
	When EAn = L, 0H to FFFFFH	Program data input	Address output	PSENn output

[Note]
When P0, P1, P2 and P4 are set as secondary function outputs, each of these ports enters a pulled-up state while external program memory is not accessed.

16.2.2 Port Operation When Accessing Data Memory

When accessing internal data memory (addresses 0H to 1FFFH), P0, P1, P2, P3_2, P3_3 and P4 operate as I/O ports.

When accessing external data memory (addresses 2000H 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 RDn and WRn output pins.

If the EAn 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 (RESn 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 RDn and WRn 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 EAn 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 16-2 lists the operation of P0, P1, P2, P3_2, P3_3 and P4 during a data memory access.

Table 16-2 P0, P1, P2, P3_2, P3_3 and P4 Operation During Data Memory Access

Data to be accessed	Address	P0 operation	P1, P2, P4 operation	P3_2, P3_3 operation
Internal data	0H to 1FFFH	I/O port		
External data	2000H to FFFFFH	After set as secondary function output*, data I/O	After set as secondary function output*, address output	After set as secondary function output*, RDn and WRn output

* If the EAn pin is at a low level, P0, P1, P2 and P4 are automatically set as secondary function outputs when reset.

[Note]
When P0, P1, P2 and P4 are set as secondary function outputs, each of these ports enters a pulled-up state while external data memory is not accessed.

16.3 External Memory Access

16.3.1 External Program Memory Access

A program memory space of 1 MB maximum (00000H to 0FFFFFH) can be accessed with the 16-bit program counter (PC) and 4-bit code segment register (CSR). When the EAn pin is set to a high level, program addresses from 20000H to FFFFFH access external program memory. When the EAn pin is set to a low level, program addresses from 00000H to FFFFFH access external program memory.

If the EAn pin is set to a high level and external program memory is to be used from 20000H to FFFFFH, then P0, P1, P2 and P4 must be set as secondary function outputs. In addition, P3_1 (PSENn output) must also be set as a secondary function output.

Figure 16-1 shows an example connection of external program memory (ROM).

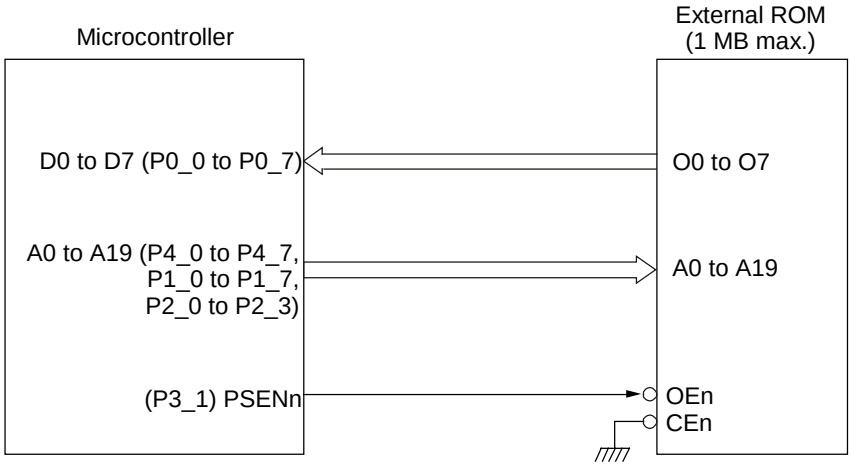


Figure 16-1 External ROM Connection Example

16.3.2 External Data Memory Access

A data memory space of 1 MB 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 1FFFH access internal data memory. Data addresses from 2000H 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 (WRn and RDn outputs).

If the EAn pin is at a low level, P0, P1, P2 and P4 automatically become secondary function outputs.

If necessary, place external pull-up resistors to connect to the WRn and RDn pins.

Figure 16-2 shows an example connection of external data memory (RAM).

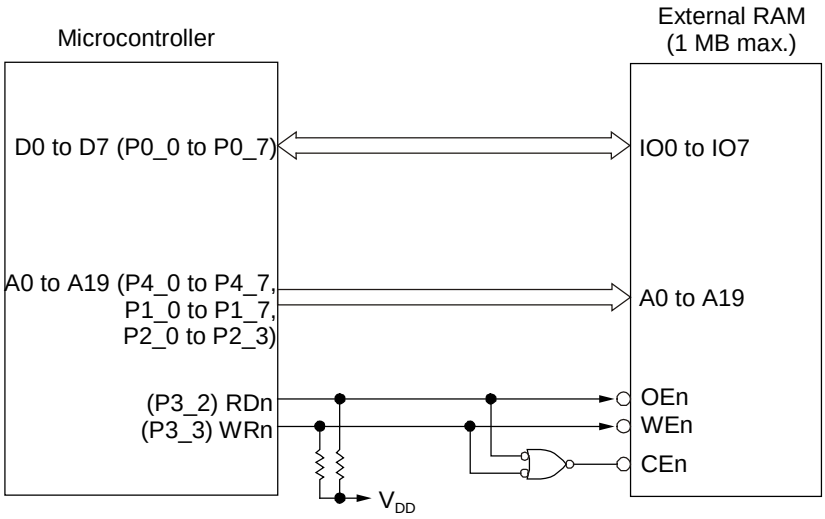


Figure 16-2 External RAM Connection Example

16.4 External Memory Access Timing

16.4.1 External Program Memory Access Timing

Figures 16-3 and 16-4 show the timing for accessing external program memory.

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 21, "Electrical Characteristics".

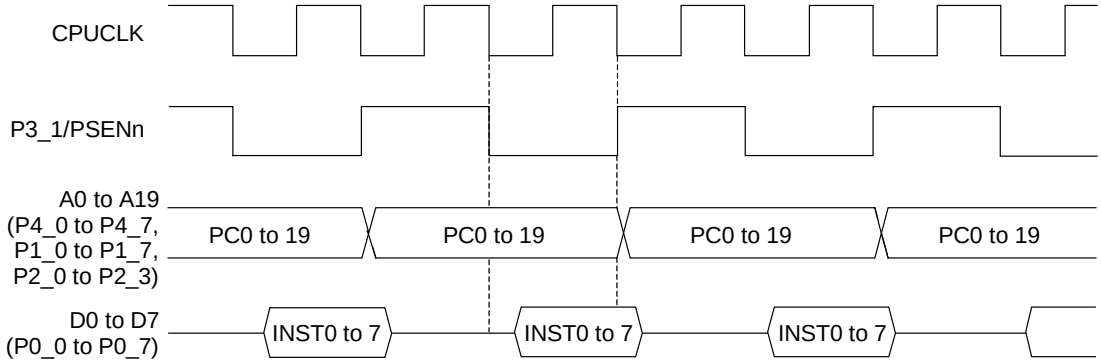


Figure 16-3 External Program Memory Access Timing (No Wait Cycles)

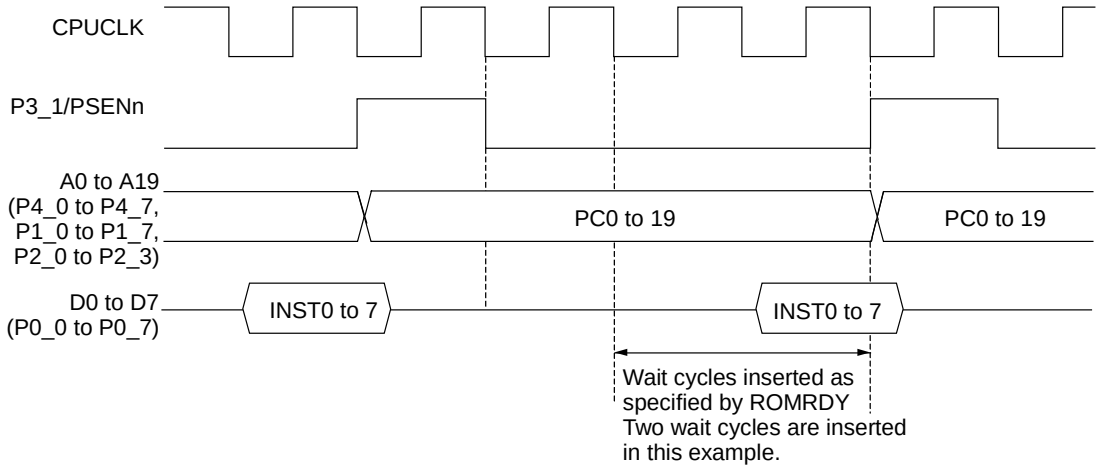


Figure 16-4 External Program Memory Access Timing (2 Wait Cycles)

16.4.2 External Data Memory Access Timing

Figures 16-5 and 16-6 show the timing for accessing data program memory.

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 21, "Electrical Characteristics".

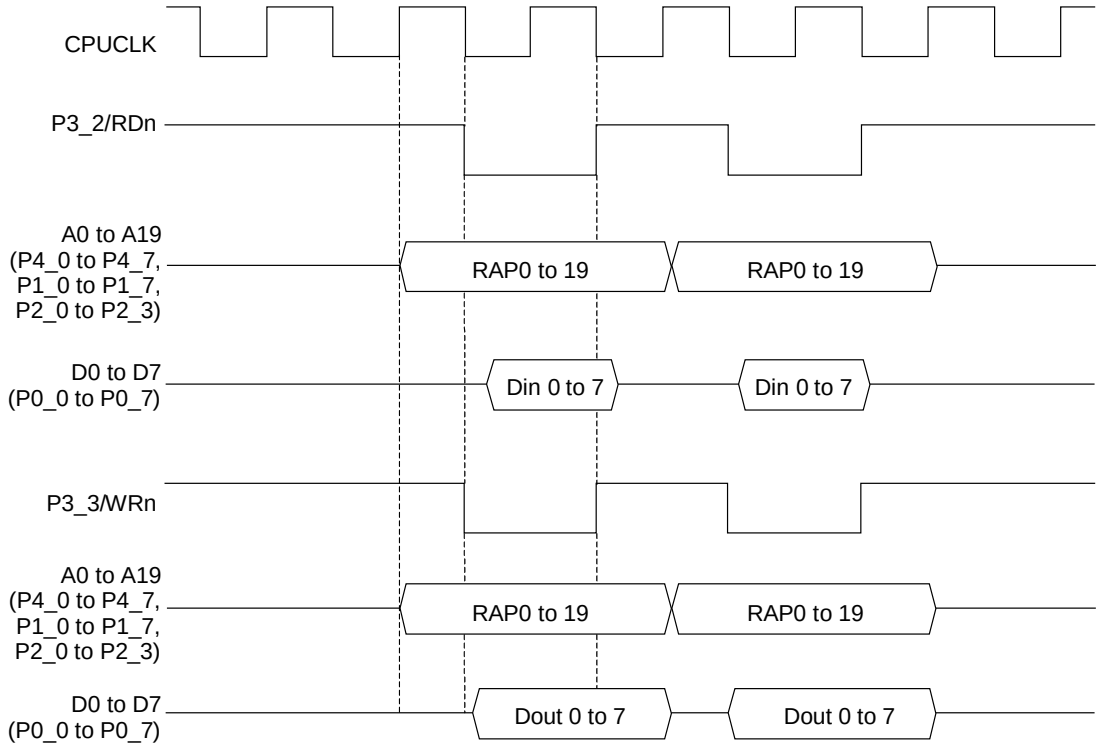


Figure 16-5 External Data Memory Access Timing (Word Access: No Wait Cycles)

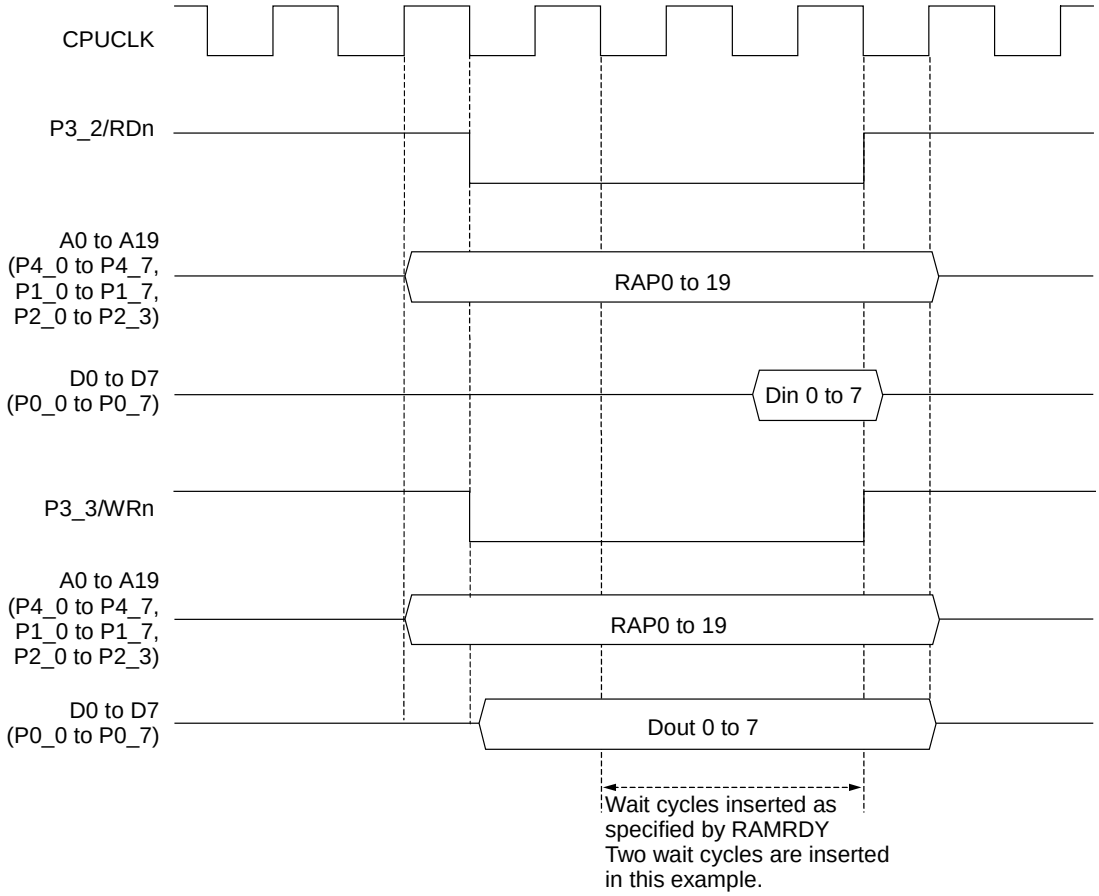


Figure 16-6 External Data Memory Access Timing (Byte Access: 2 Wait Cycles)

16.4.3 On the P3_2/RDn Pin

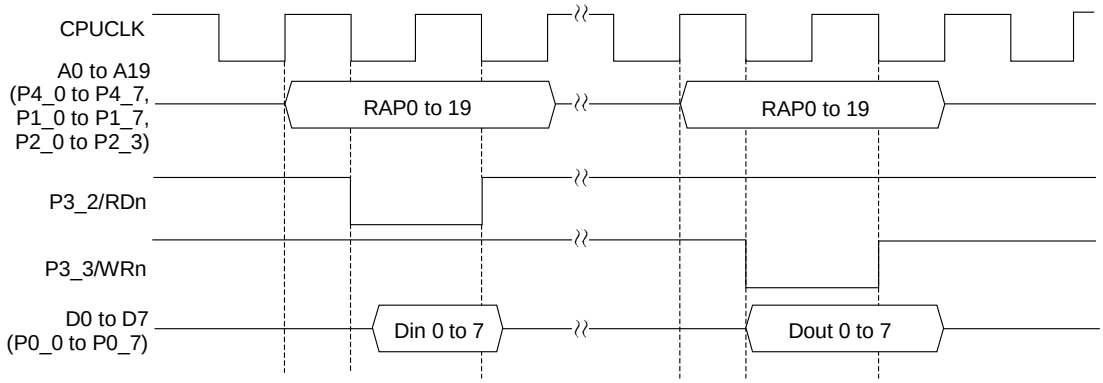
The P3_2/RDn pin can be set to be either 80-system-compatible or 68-system-compatible using bit 3 (STBSEL) of the internal control register (P5IO).

This pin will be 80-system-compatible when STBSEL is "0" and 80-system-compatible when STBSEL is "1".

In the case of both 80- and 68-system-compatible modes, it is required to set both the P3_2/RDn and P3_2/WRn pins to secondary function output.

1) 80 system compatible (when STBSEL is "0")

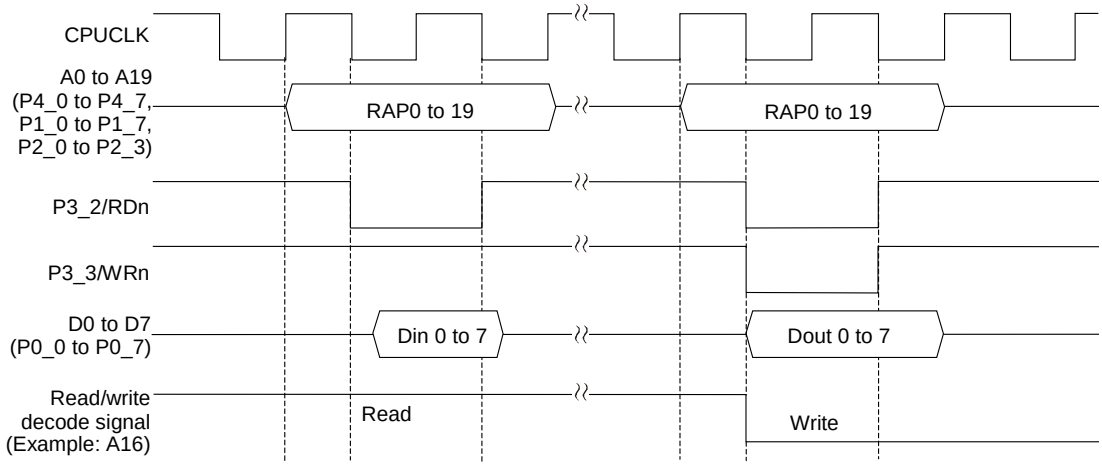
The read strobe signal (RDn) and the write strobe signal (WRn) are output individually from separate pins.



2) 68 system compatible (when STBSEL is "1")

The read strobe signal (RDn) and the write strobe signal (WRn) are both output from P3_2.

The judgment of the state of whether the signal is a read strobe or a write strobe should be made by a signal obtained by decoding the address.



[Note]

In this case, even the P3_3/WRn signal, which is useless in the 68-system-compatible mode, will also be output.

16.5 Notes Regarding Usage of Bus Port Function

16.5.1 Dummy Read Strobe Output

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 RDn signal (dummy RD) will be output before the actual access (WRn 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 16-3 and 16-4 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 16-3 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 16-4 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

16.5.2 External Bus Access Timing

The ML66525 family employs a high-speed separate address and data bus type for external bussing.

If the ML66525 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 WRn signal, are connected using the bus port function.

Refer to the AC characteristics (Section 21, 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_{IL}/V_{IH} = 0.8 V/2.0 V
- Ambient temperature Ta = –30 to +70°C
- Power supply voltage V_{DD} = 2.4 to 3.6 V

In order to set up the address value before the WRn signal is made to its logically active state under the above conditions, the following operating frequency should be provided:

- f = 12.5 MHz or less for V_{DD} = 2.4 to 3.6 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 ML66525 family is to be applied.

The contents described above should be considered to connect the peripheral devices, such as general-purpose SRAMs.

Chapter 17

USB Control Function

17. USB Control Function

17.1 Overview

Universal Serial Bus (USB) controller includes a USB serial interface engine, USB transceiver, FIFOs, control/status registers, application interface circuit, and oscillation circuit.

USB controller supports four types of data transfer such as control transfer, bulk transfer, interrupt transfer and isochronous transfer, and also supports six endpoints.

17.2 Features

- USB1.1 compliant
- Supports full-speed (12 Mbps).
- Supports four types of transfer: control transfer: bulk transfer, interrupt transfer, and isochronous transfer.
- Endpoints: 6 endpoints
  Control EP 1
  Bulk/interrupt EP 3
  Isochronous/bulk/interrupt EP 2
- Built-in FIFO for data storage
- A two-layer configuration of FIFO for each of EP1, EP2, EP4, and EP5
- DMA transfer is possible (EP1, EP2, EP4, and EP5).
- Supports bus-powered device.
  The suspend condition is automatically detected and the low-power mode is activated.
  Normal operation is automatically restarted when the resume condition is detected.
- Built-in USB transceiver circuit

17.3 Functional Descriptions

17.3.1 USB Interface

The following functions are bases for a USB protocol. Therefore, the application can process a lot of its own functions.

- Bit synchronization
- Encoding and decoding NRZI signals.
- Generating and detecting Sync bytes.
- Bit stuffing
- Generating and checking CRCs (CRC5, CRC16).
- Encoding and decoding PID (packet identifier).
  1. Decoding token.
  2. Encoding and decoding handshake.
- Generating and detecting SOP.
- Enpacket (packing) and depacket (unpacking)
- Comparing device addresses.
- Storing 8-byte setup data from a host into the setup register.
- Transmitting data in transmit FIFO.
- Storing receive data into receive FIFO of the corresponding endpoint.

17.3.2 USB Transfer Modes

Supports four kinds of transfer modes such as control transfer mode, interrupt transfer mode, bulk transfer mode, and isochronous transfer mode, which are specified by USB Standards.

(a) The control transfer mode is used to receive and respond to configurations and commands from a host, and to exchange status information between the host and peripherals.

(b) The bulk transfer mode is used to transfer a lot of data in the limited service period when the band width of USB bus becomes sufficient.

(c) The interrupt transfer mode is used to transfer a small amount of data with less frequency in the limited service period.

(d) The isochronous transfer mode is used to continuously transfer audio data, moving pictures data and other data.

17.3.3 Endpoints and FIFOs

Depending on the register setting, 5EP mode or 6EP mode can be selected. 5EP mode has five end points and 6EP mode six. Although the transfer mode that can be used by EP0 is fixed, it is possible to select either the bulk transfer mode or the interrupt transfer mode for the end points EP1, EP2, and EP3, and one of the modes of isochronous, bulk, or interrupt transfer can be selected for EP4 and EP5. In addition, it is possible to selectively set the direction of data transfer for EP1 to EP5.

End Point	FIFO Capacity (bytes)		Transfer mode	Remarks
	5EP mode	6EP mode		
EP0	Reception 32 Transmission 32		Control transfer	
EP1	64x2		Bulk/interrupt transfer (IN/OUT)	DMA Data Transfer Possible
EP2	64x2		Bulk/interrupt transfer (IN/OUT)	DMA Data Transfer Possible
EP3	32		Bulk/interrupt transfer (IN/OUT)	
EP4	128x2	64x2	Isochronous/bulk/ interrupt transfer (IN/OUT)	DMA Data Transfer Possible
EP5	—	64x2	Isochronous/bulk/ interrupt transfer (IN/OUT)	DMA Data Transfer Possible

Note 1: EP3 permits rate feedback data sequence toggling.

Note 2: EP1, EP2, and EP3 are all mutually independent, and can be assigned for bulk transfer or interrupt transfer individually. It is possible to set the maximum packet size up to 64 bytes (32 bytes for EP3) during both bulk transfer and interrupt transfer.

Note 3: It is possible to set EP4 and EP5 to one of the modes of isochronous transfer, bulk transfer, and interrupt transfer. The maximum packet size can be up to 64 bytes when these end points are set to bulk transfer.

Note 4: The choice between EP5 mode and EP6 mode is specified with the D2 bit of the SYSCON register.

Note 5: When EP5 and EP6 are used for isochronous transfer, the maximum packet size for EP4 is 128 bytes in 5EP mode. EP5 cannot be used in 5EP mode.

17.3.4 Operation of Control Transfer

(a) Setup stage

The setup token and 8 bytes of setup data are transmitted from the host. USB controller decodes the setup token, and automatically stores the 8 bytes of setup data in the setup register. When this is completed normally, USB controller returns ACK to the host. The 8-byte setup data is the standard request code defined in Section 9.3 of the USB Standards, or a code of the requests unique to each device class, etc. The request is decoded on CPU side.

(b) Data stage

If the request specified by the 8-byte setup data is also accompanied by transfer of parameter data from the host to the device, the transfer is a control write transfer, and the OUT token and the data packet are transmitted from the host. When these are received normally, USB controller stores the parameter data in the EP0 receive FIFO and returns ACK to the host.

If the request is accompanied by transfer of parameter data from the device to the host, the transfer is a control read transfer, and when the host sends the IN token, USB controller sends the parameter data that was already stored beforehand in the EP0 transmit FIFO by CPU. When the host receives this normally, it returns an ACK to USB controller. On the other hand, in the case of requests that do not contain any parameter data that needs to be transmitted or received, this data stage will not be present and the processing proceeds directly to the status stage from the setup stage.

(c) Status stage

The status stage is a stage intended for reporting the status of the result of executing a request from the device to the host. During a control write transfer or a control transfer without data, the IN token is sent by the host, and USB controller returns a response to it. During a control read transfer, the OUT token and data of zero length are sent by the host, and USB controller returns a response to it.

During the above control transfers, CPU need only read from or write to the 8-byte setup registers mapped at 00h to 07h, the EP0 transmit FIFO mapped at 70h, and the EP0 receive FIFO mapped at 78h according to the interrupt cause, and all other operations will be carried out automatically by USB controller.

17.3.5 Data Packet Transmission and Reception Procedure During Bulk Transfer and Interrupt Transfer Modes

The transfer of data is the major function in all types of transfer modes other than the control transfer mode. When carrying out transfer of data packets between USB controller and the host, the following packet communication is carried out via the USB bus for the data transfer of each packet.

(a) Token packet transfer (IN token or OUT token) from the host to USB controller.

(b) Data packet transfer in the desired direction (from the host to the device or from the device to the host).

(c) Transfer of handshake packet in a direction opposite to that of the data packet.

When packet transfer is completed normally, an ACK packet is returned in step (c) and the operation proceeds to the next packet transfer.

USB controller requests CPU to transmit or receive a packet of data. The interrupt cause will be "packet ready". The transmit packet ready interrupt is one that requests that the packet of data to be transmitted be written in the transmit FIFO, and the receive packet ready interrupt is one that requests CPU to read out the data that has been received and stored in the receive FIFO.

The above procedures of transferring one packet of data are explained below for transmission and reception separately.

1) During transmission

CPU writes one packet of data that has to be transmitted in the transmit FIFO of the corresponding EP in USB controller, and sets to "1" the transmit packet ready bit of the corresponding EP status register of USB controller. When the host transmits the IN token packet to USB controller specifying the communication method, etc., USB controller transmits to the host the data packet stored in the above transmit FIFO. When the host receives one data packet normally, it returns the ACK packet to USB controller. Consequently, USB controller resets the transmit packet ready bit, thereby completing the transfer of one data packet over the USB bus. When the transmit packet ready bit is reset, USB controller gives a request to CPU in terms of a transmit packet ready interrupt thereby prompting CPU to write the next packet of data to be transmitted.

2) During reception

The host sends to USB controller an OUT token followed by a data packet. USB controller stores the received data packet in the receive FIFO of the corresponding EP. When it is confirmed that all the data packets have been accumulated and that there is no error, USB controller returns an ACK packet to the host. At the same time, the receive packet ready bit of the corresponding EP status register will also be set to "1" and a request is sent to CPU in terms of an interrupt. Upon receiving this interrupt, CPU reads out the received data from USB controller and resets the receive packet ready bit.

17.3.6 Data Packet Transmission and Reception Procedure During Isochronous Transfer Mode

Transfer of data is the major function in the isochronous transfermode. When carrying out isochronous transfer between USB controller and the host, the following packet communications are carried out via the USB bus for the data transfer of each packet.

(a) Token packet transfer (IN token or OUT token) from the host to USB controller.

(b) Data packet transfer in the desired direction (from the host to the device or from the device to the host).

In the isochronous transfer mode, there is no handshaking that reports whether or not the packet transfer was done normally.

The interrupt cause is SOF. Upon receiving this interrupt, CPU writes the packet data into the transmit FIFO of the EP set for transmission (ISO IN) in the isochronous transfer mode, or reads out data from the receive FIFO of the EP set for reception (ISO OUT) in the isochronous transfer mode.

The above procedures of transferring one packet of data are explained below for transmission and reception separately.

1) During transmission

The EP for ISO IN has a two-layer FIFO configuration. One FIFO is used for storing the packet data that is written in by the MCU via the local bus. The other FIFO is used for transmitting the stored data to the USB bus when an IN token is received. The roles of the two FIFOs are interchanged when an SOF packet is received.

Upon receiving an SOF interrupt, CPU writes the data to be transmitted during the next frame into the corresponding transmit FIFO of the EP of USB controller. When the host transmits an IN token packet, USB controller transmits to the host the packet data written in the transmit FIFO during one frame before the current frame.

2) During reception

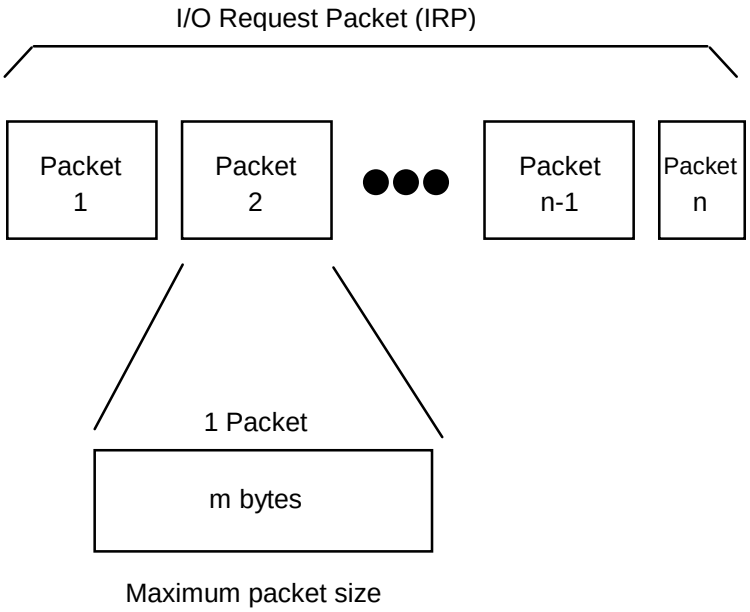
The EP for ISO OUT has a two-layer FIFO configuration. One FIFO is used for storing the packet data that is output to the local bus when the MCU reads the received packet data. The other FIFO is used for storing the packet data received from the USB bus. The roles of the two FIFOs are interchanged when an SOF packet is received.

Upon receiving an SOF interrupt, CPU reads out the data that has been received during the previous frame from the corresponding receive FIFO of the EP of USB controller. When the host transmits an OUT token and a data packet to USB controller, USB controller stores that received data packet in the receive FIFO, and that data packet is read out by CPU during the next frame.

17.3.7 Packets and Packet Sizes

USB controller packs the transmit data into packets and unpacks (restores to the original form) the received data. The packed data that is recognized by the software client is a set of data consisting of one or more packets, and this is called an I/O request (IRP).

Among the several packets in an IRP, all the packets other than the last packet are transferred with the maximum packet size. Only the last packet can be transferred as a "short packet", that is, a packet whose size is less than the maximum packet size.



USB controller has payload registers corresponding to each end point, and it is possible to set the maximum packet size for each end point in these registers. The maximum packet size should be within the capacity of the corresponding FIFO, and can be set as follows:

(1) EP0 Receive packet size can be 32 bytes or less;
(2) EP0 Transmit packet size can be 32 bytes or less;
(3) EP1 Transmit/receive packet size can be 64 bytes or less;
(4) EP2 Transmit/receive packet size can be 64 bytes or less;
(5) EP3 Transmit/receive packet size can be 32 bytes or less;
(6) EP4 Transmit/receive packet size can be 64 bytes or less;
(7) EP5 Transmit/receive packet size can be 64 bytes or less;

On the USB bus, the separation between successive packets is distinguished by appending a special signal condition called EOP (End of Packet) at the end of each packet. The appending of EOP during transmission and the detection and removal of EOP during reception are carried out by USB controller automatically.

(1) At the time of transmission, the packet is deemed to have ended when CPU has completed writing the required number of bytes of data in the transmit FIFO and has then asserted the transmit packet ready bit. (The actual addition of EOP is executed at the time of

transmitting the data over the USB bus after waiting for the IN token from the host.) The packet will be a short packet if the transmit packet ready bit is asserted after writing data with less number of bytes than the maximum packet size. In particular, by asserting the transmit packet ready bit without writing any data, it is possible to form a null packet whose data length is zero.

(2) At the time of reception, when an EOP is detected in the received data string, USB controller2 recognizes it as the end of the received packet and asserts the receive packet ready bit. The number of bytes in the received packet is counted automatically by the receive byte count register (Note 1) corresponding to that end point.

Note 1: Receive byte count register address: 58h to 5Dh and 74h to 75h.

17.3.8 Interrupts

The interrupt causes are the following:

(a) Setup ready for the 8-byte setup data
(b) EP0 receive packet ready
(c) EP0 transmit packet ready
(d) EP1 transmit/receive packet ready
(e) EP2 transmit/receive packet ready
(f) EP3 transmit/receive packet ready
(g) EP4 transmit/receive packet ready
(h) EP5 transmit/receive packet ready
(i) SOF
(j) USB Bus reset assert
(k) USB Bus reset de-assert
(l) Suspend
(m) Awake

The CPU can identify the contents of the interrupt by reading out the interrupt status register 1 (INTSTAT1) and the interrupt status register 2 (INTSTAT2). These interrupts can also be masked dynamically by making individual settings in the interrupt enable register 1 (INTENBL1) and the interrupt enable register 2 (INTENBL2).

The causes of the interrupts, their setting and resetting conditions, and the responses to them are described below.

The functions of the setup ready bit and the packet ready bit can, in some situations, be different from those described here because of some special automatic operations done by USB controller. Please see the descriptions of the registers EP0STAT to EP5STAT for more details of such functions.

(1) Setup ready interrupt

Operation	Source of operation	Description (conditions, responses, etc.)
Setup ready interrupt generation	USB controller	The setup ready bit (D2 of EP0STAT) is asserted when the 8-byte setup control data is received normally and has been stored in the set of setup registers. An interrupt is generated at this time if D1 of INTENBL1 has been asserted. → The firmware can now read the set of setup registers.
End of setup ready interrupt	CPU (firmware)	After making the firmware read the 8-byte setup data, write a "1" in bit D2 of EP0 status register (EP0STAT). This causes the interrupt to be de-asserted. The interrupt will not be de-asserted If a new 8-byte setup data is received during this period. In this case, discard the setup data that was being read at that time and read the new 8-byte setup data.

(2) EP0 Receive packet ready interrupt

This is used mainly during the reception of a data packet in a control write transfer.

Operation	Source of operation	Description (conditions, responses, etc.)
EP0 Receive packet ready interrupt generation	USB controller	The EP0 receive packet ready bit (D0 of EP0STAT) is asserted during a control write transfer when the processing has changed from the setup stage to the data stage, and USB controller has detected EOP of the data packet and has stored the data without error in the EP0 receive FIFO. The end of a packet is recognized when an EOP has arrived in the cases of both full packets and short packets. An interrupt is generated at this time, if the EP0 receive packet ready interrupt enable bit (D6 of INTENBL1) has been asserted. (EOP: End of packet)
End of EP0 receive packet ready interrupt	CPU (firmware)	In the case of EP0 reception, after the number of bytes of the EP0 receive FIFO data indicated by the EP0 receive byte count register (EP0RXCNT) has been read, write a "1" to the EP0 receive packet ready bit (bit D0 of EP0STAT). (This status is reset when a "1" is written in this bit.)

Note: A short packet is a packet with a number of bytes less than the maximum packet size.

(3) EP0 Transmit packet ready interrupt

This is used mainly during the transmission of a data packet in a control read transfer.

Operation	Source of operation	Description (conditions, responses, etc.)
EP0 Transmit packet ready interrupt generation	USB controller	The EP0 transmit packet ready bit (D1 of EP0STAT) is de-asserted during a control read transfer when the processing has changed from the setup stage to the data stage, and it is possible to write the transmit data to the FIFO. At this time, an interrupt is generated if the EP0 transmit packet ready interrupt enable bit (bit D7 of INTENBL1) has been asserted. For the second and subsequent packets, in addition to this condition, before the interrupt is generated, it is necessary for an ACK response to come from the host for the packet that has just been sent.
End of EP0 transmit packet ready interrupt	CPU (firmware)	In the case of EP0 transmission, after the one packet of the EP0 transmit data has been written in EP0TXFIFO, write a "1" into the EP0 transmit packet ready bit (bit D1 of EP0STAT). This puts USB controller in a state in which it can transmit the data (that is, it can transmit the data packet when an IN token arrives), and the internal interrupt signal is de-asserted at the same time. Even when the number of bytes in the write data is less than the maximum packet size, it is possible to transmit the data by writing a "1" into the transmit packet ready status bit. This makes it possible to transmit a short packet.

(4) Receive packet ready interrupts (EP1, EP2, EP3, EP4 bulk, EP5 bulk)

These interrupts are generated when the respective EP has received an appropriate data packet from the USB bus and CPU can read that data.

Operation	Source of operation	Description (conditions, responses, etc.)
Receive packet ready interrupt generation	USB controller	The receive packet ready bit (D0) of the corresponding EP status register (EPnSTAT) is asserted during data reception when the EOP of the data packet has been received and the data has been stored without error in the corresponding FIFO. The end of a packet is recognized when an EOP has arrived in the cases of both full packets and short packets. An interrupt is generated at this time, if the corresponding receive packet ready interrupt enable bit has been asserted. (EOP: End of packet)
End of receive packet ready interrupt	CPU (firmware)	After the number of bytes in the receive FIFO data (EPnFIFO) indicated by the corresponding receive byte count register (EPnRXCNT) has been read, write a "1" into the receive packet ready bit D0 of the corresponding EP status register (EPnSTAT). (This status is reset when a "1" is written in this bit.)

(5) Transmit packet ready interrupts (EP1, EP2, EP3, EP4 bulk, EP5 bulk)

These interrupts are generated when it is possible for CPU to write the data packet to be sent to the USB bus from the corresponding EP.

Operation	Source of operation	Description (conditions, responses, etc.)
Transmit packet ready interrupt generation	USB controller	(1) In the case of bulk transfer and interrupt transfer When the respective EP has been set for transmission, the transmit packet ready bit of the corresponding EP (bit D1 of EPnSTAT) is de-asserted when it is possible to write the transmit data into the FIFO. At this time, an interrupt is generated if the corresponding EP transmit packet ready interrupt enable bit of INTENBL1 has been asserted. For the second and subsequent packets, in addition to this condition, before the interrupt is generated, it is necessary for an ACK response to come from the host for the packet that has just been sent.
End of transmit packet ready interrupt	CPU (firmware)	(1) In the case of bulk transfer and interrupt transfer After the one packet of the corresponding EP transmit data has been written in EPnTXFIFO, write a "1" into the corresponding transmit packet ready bit (bit D1 of EPnSTAT). This puts USB controller in a state in which it can transmit the data and the internal interrupt signal is de-asserted at the same time. Even when the number of bytes in the write data is less than the maximum packet size, it is possible to end the pocket transmission by writing a "1" into the transmit packet ready status bit.

(6) SOF Interrupt

Operation	Source of operation	Description (conditions, responses, etc.)
SOF Interrupt generation	USB controller	When an SOF packet is detected on the USB bus.
End of SOF interrupt	CPU (firmware)	When a "1" is written in the corresponding bit of the interrupt status register 2 (INTSTAT2).

(7) USB Bus reset assert interrupt

Operation	Source of operation	Description (conditions, responses, etc.)
USB Bus reset assert interrupt generation	USB controller	The ML66525 automatically detects the condition when the SE0 state continues for 2.5 µs or longer at the D+ and D– pins. → Carry this out by firmware processing for bus reset.
End of USB bus reset assert interrupt	CPU (firmware)	When a "1" is written in the corresponding bit of the interrupt status register 2 (INTSTAT2).

(8) USB Bus reset de-assert interrupt

Operation	Source of operation	Description (conditions, responses, etc.)
USB Bus reset de-assert interrupt generation	USB controller	When there is a recovery to the J state from the SE0 state of 2.5 µs or longer at the D+ and D– pins. → Carry this out by firmware processing for bus reset release.
End of USB bus reset de-assert	CPU (firmware)	When a "1" is written in the corresponding bit of the interrupt status register 2 (INTSTAT2).

(9) Suspend state interrupt

Operation	Source of operation	Description (conditions, responses, etc.)
Suspend state interrupt generation	USB controller	When the idle condition persists for 3 ms or more at the D+ and D– pins. → The internal oscillations in USB controller are stopped automatically when the idle condition continues for an additional 2 ms after this interrupt has been generated. The firmware can take steps to put the device in the power save mode.
End of suspend state interrupt	CPU (firmware)	When a "1" is written in the corresponding bit of the interrupt status register 2 (INTSTAT2).

(10) Awake interrupt

Operation	Source of operation	Description (conditions, responses, etc.)
Awake interrupt generation	USB controller	When the Resume signal (the SE0 state of about 1344 ns immediately after the K state) is detected at the D+ and D– pins.
End of awake interrupt	CPU (firmware)	When a "1" is written in the corresponding bit of the interrupt status register 2 (INTSTAT2).

17.3.9 Internal DMA (Direct Memory Access)

It is possible to carry out 8-bit wide internal DMA transfer for the bulk transfer of EP1, EP2, EP4, and EP5, and for the isochronous transfer of EP4 and EP5.

It is possible to carry out internal DMA transfers over two channels, Channel 0 and Channel 1. Both demand transfer and single transfer are supported. The settings of the internal DMA transfer mode and parameters are done using the DMA control register and the DMA interval register described later in this manual.

In the demand transfer mode, the DREQ signal is asserted when the reading or writing of a data packet becomes possible. The DREQ signal is de-asserted when the transfer of all the data of the receive packets is completed by the internal DMA controller. Therefore, other devices cannot access the local bus during DMA transfer.

On the other hand, in the single transfer mode, the DREQ signal is de-asserted at the end of transfer of the number of bytes of one transfer, and the other devices can access the local bus during this period.

17.3.10 Power-down

When the USB controller detects a suspended state on the USB bus, it automatically stops the PLL and the USB control function enters the power-down state. At this time, if OSCS (bit 3 of SBYCON) is "1" and the CPU is either in the STOP mode or operating under the subclock, the OSC oscillation circuit stops (see Chapter 3). When the resume signal is detected on the USB bus, the OSC oscillation circuit is activated even if the CPU is in the STOP mode or operating under the subclock, and the PLL circuit is enabled at the same time, so that the power-down state of the USB control function is released.

Writing a "1" and a "0" to the OSCTEST register bit 2 and bit 1 respectively forces the USB control function to enter a power-down state. Power saving can be made real when the USB bus is unconnected (USB not used).

17.3.11 Operation of 2-Layer Structure FIFO During Bulk Transfer

The FIFOs of EP1 and EP2 have a 64 bytes x 2-layer structure. Also, when EP4 is assigned for bulk transfer, its FIFO also has a 64-bytes x 2-layer structure. As a consequence, these FIFOs can temporarily store a maximum of 128 bytes of bulk transfer data.

(1) 2-Layer reception operation ("O" indicates the assert condition and "x" indicates de-assert condition)

	In the case of 1→2→3→4→5a→6 In the case of 1→2→3→4→5b→6	Layer A 64 bytes	Layer B 64 bytes	Layer A PKT RDY	Layer B PKT RDY	EPn receive PKT RDY
1	Start storing data in layer A of reception	■		x	x	x
2	Data of one packet has been stored.	■		○	x	○
3	Start reception and storing of data in layer B.	■		○	x	○
4	Local MCU starts reading layer A.	■		○	x	○
5a	When the storing of packet in layer B is completed following the completion of reading layer A.	■	■	○	○	○
5b	When the reading of packet in layer A is completed following the completion of storing data in layer B.		■	x	x	x
6	From 5a: Layer A has become empty. From 5b: Layer B has become full.		■	x	○	○
7	Starting reading layer B also.		■	x	○	○

- When one packet of receive data is stored in layer A of the FIFO and EOP is received, USB controller asserts the EPn packet ready bit and also the internal interrupt signal. This makes it possible for CPU to read the receive data.
- Subsequently, data can be received from the host, and USB controller switches the FIFO for storing to layer B.
- When one packet of data described above has been read from layer A of the FIFO, make CPU reset the EPn receive packet ready bit (by writing a "1" into bit D0 of EPnSTAT).
- At the time the EPn receive packet ready bit is reset, if the reception of layer B has not been completed, USB controller resets the EPn receive packet ready bit.
- However, if the reception of layer B has been completed at the time the EPn receive packet ready bit is reset, USB controller rejects the request from CPU to reset the EPn receive packet ready bit, and continues to maintain the assert state of the EPn receive packet ready bit. (See Note below.)

[Note]

The interrupt signal is de-asserted once at the time the EPn receive ready bit is reset. Then a switch will be made to layer B status in a maximum of 6 cycles (1 cycle = 1 CPUCLK) and, if there is a receive packet in layer B, the EPn receive interrupt will be generated again.

If layer B FIFO is to be read more than once in succession, read it by interrupt detection after the EPn receive packet ready bit is reset or read it after confirming that the EPn interrupt status bit of the interrupt status register 1 (INTSTAT1) is "1".

(2) 2-Layer transmission operation ("O" indicates the assert condition and "x" indicates de-assert condition)

	In the case of 1→2→3→4→5a→6 In the case of 1→2→3→4→5b→6	Layer A 64 bytes	Layer B 64 bytes	Layer A PKT RDY	Layer B PKT RDY	EPn transmit PKT RDY
1	Layer A and layer B are both empty.			x	x	x
2	The local MCU starts writing into layer A.	■		x	x	x
3	Writing of one packet is completed.	■		○	x	x
4	The data of layer A is transmitted and the next packet is written in layer B.	■	■	○	x	x
5a	When layer A has become empty after the writing in layer B is completed.	■	■	○	○	○
5b	When the writing in layer B has been completed after layer A has become empty.		■	x	x	x
6	From 5a: Layer A has become empty. From 5b: Layer B has become full.		■	x	○	x
7	Transmission of layer B is also started.		■	x	○	x

- If the EPn transmit packet ready interrupt enable bit of INTENBL1 has been asserted, the transmit FIFO is empty, and EPn transmit packet ready bit is de-asserted, the EPn transmit packet ready interrupt is asserted. This makes it possible to write the transmit data into the EPn transmit FIFO.
- When the data of one packet is written in layer A FIFO, make CPU set the transmit packet ready bit (D1 of EPnSTAT) to "1". By setting the transmit packet ready bit to "1", it becomes possible to transmit data to the host. At this time, since layer B is still empty, the assert state of the interrupt is maintained (see Note below), thereby indicating that the next packet data can be written. In this case, although bit D1 of EPnSTAT remains in the "0" condition, USB controller recognizes that transmission is possible from layer A and starts transmission when an IN token is received from the host.
- It is possible for CPU to write the next packet of transmit data in the layer B FIFO while the data in layer A is being transmitted over the USB bus.
- When the writing of the data to be transmitted in layer B has been completed, CPU sets the transmit packet ready bit to "1". At this time, if the transmission of layer A data has not been completed (that is, the ACK message is received from the host and the transmit packet ready bit is reset), the interrupt is de-asserted. (The local MCU cannot yet write the subsequent packet.)
- If the layer A becomes empty before layer B goes into the transmit enable condition and transmission is carried out normally, the ACK response is received from the host. The CPU can write data into layer A FIFO after writing into layer B FIFO.
- The transmission of data in layer A is continued from the state 5a, and when layer A becomes empty and the transmission is completed normally, the ACK response is received from the host, thereby prompting CPU to write data into layer A.

[Note]
If the transmit packet ready bit is set to "1", the interrupt signal is de-asserted once, then it is asserted again in a maximum 6 cycles (1 cycle = 1 CPUCLK).

17.3.12 Error Processing and Retry Operation

1) Error processing during transmission

   When an error such as a CRC error is detected in the data transmitted by USB controller, the host will not send the ACK packet, and hence USB controller does not reset the transmit packet ready status, but waits while retaining the current packet of data. The current packet of data is transmitted again when the next IN token is received from the host.

2) Error processing during reception

   When an error is detected in the data received over the USB bus, USB controller does not assert the interrupt signal to CPU and will also not send any message to the host (leading to a timeout condition).

   When the timeout condition is generated, the host recognizes that an error has occurred, and can take measures such as re-transmitting the data, etc. In addition, since no interrupt request is generated, CPU will not read the erroneous data.

17.3.13 D+ Pull-up Control

Immediately after the power is switched on, the device may not be able to receive the token sent from the host because the preparations at the firmware level have not been completed. In this case, until the initialization at the firmware level is completed, the D+ line of the USB connector (the D– line in the case of a low speed device) is turned off, and the 1.5 kΩ pull-up resistor is turned on after the initialization has been completed. To do this, a switch circuit that can be turned on or off by CPU is provided in series with the 1.5 kΩ resistor.

In USB controller, although it is possible to provide such switch circuits externally, it is also possible to use the internal switch circuit in the ADSEL = 0 state as shown in the following figure. The internal switch can be switched on or off uniquely by the value of the bit D3 of the SYSCON register.

17.4 Registers of USB Control Functions

Table 17-1 shows a list of the special function registers (SFRs) for the USB control functions.

Table 18-1 List of SFRs for the USB Control Functions (1/2)

Category	Address [H]	Register name	Symbol (byte)	Symbol (word)	R/W	8/16 Operation	Initial value [H]	Reference page
FIFO	1A70	EP0 transmit FIFO	EP0TXFIFO	—	W	8	00	17-22
	1A78	EP0 receive FIFO	EP0RXFIFO	—	R	8	Undefined	17-22
	1A79	EP1 transmit/receive FIFO	EP1FIFO	—	R/W	8	Undefined	17-23
	1A7A	EP2 transmit/receive FIFO	EP2FIFO	—	R/W	8	Undefined	17-23
	1A7B	EP3 transmit/receive FIFO	EP3FIFO	—	R/W	8	Undefined	17-24
	1A7C	EP4 transmit/receive FIFO	EP4FIFO	—	R/W	8	Undefined	17-24
	1A7D	EP5 transmit/receive FIFO	EP5FIFO	—	R/W	8	Undefined	17-25
Common	1A00	bRequest type setup register	bmRequestType	—	R	8	00	17-26
	1A01	bRequest setup register	bRequest	—	R	8	00	17-26
	1A02	wValueLSB setup register	wValueLSB	—	R	8	00	17-27
	1A03	wValueMSB setup register	wValueMSB	—	R	8	00	17-27
	1A04	wIndexLSB setup register	wIndexLSB	—	R	8	00	17-28
	1A05	wIndexMSB setup register	wIndexMSB	—	R	8	00	17-28
	1A06	wLengthLSB setup register	wLengthLSB	—	R	8	00	17-29
	1A07	wLengthMSB setup register	wLengthMSB	—	R	8	00	17-29
	1A20	Device address register	DVCADR	—	R/W	8	00	17-32
	1A21	Interrupt status register 1	INTSTAT1	—	R	8	00	17-33
	1A22	Interrupt status register 2	INTSTAT2	—	R/W	8	00	17-34
	1A24	Interrupt enable register 1	INTENBL1	—	R/W	8	01	17-35
	1A25	Interrupt enable register 2	INTENBL2	—	R/W	8	00	17-36
	1A2D	Frame number LSB register	FRAMELSB	—	R	8	00	17-37
	1A2E	Frame number MSB register	FRAMEMSB	—	R	8	00	17-37
	1A2F	System control register	SYSCON	—	R/W	8	00	17-38
	1A30	Polarity selection register	POLSEL	—	R/W	8	00	17-39
DMA	1A10	DMA0 control register	DMA0CON	—	R/W	8	00	17-30
	1A11	DMA0 interval register	DMA0INTVL	—	R/W	8	00	17-31
	1A12	DMA1 control register	DMA1CON	—	R/W	8	00	17-30
	1A13	DMA1 interval register	DMA1INTVL	—	R/W	8	00	17-31

Table 18-1 List of SFRs for the USB Control Functions (2/2)

Category	Address [H]	Register name	Symbol (byte)	Symbol (word)	R/W	8/16 Operation	Initial value [H]	Reference page
EP Support	1A40	EP0 configuration register	EP0CONF	—	R/W	8	00	17-40
	1A41	EP1 configuration register	EP1CONF	—	R/W	8	00	17-41
	1A42	EP2 configuration register	EP2CONF	—	R/W	8	00	17-41
	1A43	EP3 configuration register	EP3CONF	—	R/W	8	00	17-41
	1A44	EP4 configuration register	EP4CONF	—	R/W	8	00	17-42
	1A45	EP5 configuration register	EP5CONF	—	R/W	8	00	17-42
	1A48	EP0 control register	EP0CONT	—	R/W	8	Undefined	17-43
	1A49	EP1 control register	EP1CONT	—	R/W	8	00	17-44
	1A4A	EP2 control register	EP2CONT	—	R/W	8	00	17-44
	1A4B	EP3 control register	EP3CONT	—	R/W	8	00	17-44
	1A4C	EP4 control Register	EP4CONT	—	R/W	8	00	17-44
	1A4D	EP5 control register	EP5CONT	—	R/W	8	00	17-44
	1A50	EP0 payload register	EP0PLD	—	R	8	00	17-45
	1A51	EP1 payload register	EP1PLD	—	R/W	8	00	17-45
	1A52	EP2 payload register	EP2PLD	—	R/W	8	00	17-45
	1A53	EP3 payload register	EP3PLD	—	R/W	8	00	17-46
	1A54	EP4 payload register	EP4PLD	—	R/W	8	00	17-47
	1A55	EP5 payload register	EP5PLD	—	R/W	8	00	17-47
	1A58	EP0 receive byte count register	EP0RXCNT	—	R	8	00	17-48
	1A59	EP1 receive byte count register	EP1RXCNT	—	R	8	00	17-48
	1A5A	EP2 receive byte count register	EP2RXCNT	—	R	8	00	17-48
	1A5B	EP3 receive byte count register	EP3RXCNT	—	R	8	00	17-49
	1A5C	EP4 receive byte count register	EP4RXCNT	—	R	8	00	17-49
	1A5D	EP5 receive byte count register	EP5RXCNT	—	R	8	00	17-49
	1A60	EP0 status register	EP0STAT	—	R	8	00	17-50
	1A61	EP1 status register	EP1STAT	—	R	8	00	17-53
	1A62	EP2 status register	EP2STAT	—	R	8	00	17-53
	1A63	EP3 status register	EP3STAT	—	R	8	00	17-54
	1A64	EP4 status register	EP4STAT	—	R	8	00	17-53
	1A65	EP5 status register	EP5STAT	—	R	8	00	17-53
Option	1A39	Packet error register 1	PKTERR	—	R	8	00	17-55
	1A3A	OSC test	OSCTEST	—	R/W	8	00	17-56

17.4.1 Description of Registers of USB Control Functions

(1) EP0 transmit FIFO (EP0TXFIFO)

Address	1A70
Access type	W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	x	x	x	x	x	x	x	x
After a bus reset	x	x	x	x	x	x	x	x
Definition	EP0 Transmit data							

The EP0 transmit data can be written in by writing to the address 1A70h.

The transmit data to the host in the data stage during a control read transfer is stored in EP0TXFIFO. When an EP0 transmit packet ready interrupt request, writes the transmit data to the address 1A70h.

It is possible to write the packet data successively by writing continuously.

The EP0TXFIFO is cleared under the following conditions.

1. When an ACK signal is received from the host for the data transmission from EP0
2. When a setup packet is received

(2) EP0 receive FIFO (EP0RXFIFO)

Address	1A78
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	x	x	x	x	x	x	X	x
After a bus reset	x	x	x	x	x	x	X	x
Definition	EP0 Receive data							

The receive data from the host computer in the data sate during a control Write transfer is stored in EP0RXFIFO.

The EP0 receive data can be read out through reading the address 1A78h when an EP0 receive packet ready interrupt request. It is possible to read successively the data in the packet by reading continuously.

The EP0RXFIFO is cleared under the following conditions:

1. When the local MPU resets the EP0 receive packet ready bit.
2. When a setup packet is received.
3. When a "0" is written in the stall bit.

(3) EP1 FIFO (EP1FIFO)

Address	1A79
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	x	x	x	x	x	x	x	x
After a bus reset	x	x	x	x	x	x	x	x
Definition	EP1 Transmit data or EP1 receive data							

It is possible to specify the direction of transfer of EP1 by setting the EP1 configuration register EP1CONF. The FIFO address of EP1 is the same in both the transmit direction and the receive direction.

When EP1CONF (D7) = 0, EP1 is in the receive direction and EP1FIFO is in the read-only state.

When EP1CONF (D7) = 1, EP1 is in the transmit direction and EP1FIFO is in the write-only state.

When set for transmission, all bytes of EP1FIFO can be cleared by clearing EP1FIFO (writing a "1" into EP1CONT (D2)).

(4) EP2 FIFO (EP2FIFO)

Address	1A7A
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	x	x	x	x	x	x	x	x
After a bus reset	x	x	x	x	x	x	x	x
Definition	EP2 Transmit data or EP2 receive data							

It is possible to specify the direction of transfer of EP2 by setting the EP2 configuration register EP2CONF. The FIFO address of EP2 is the same in both the transmit direction and the receive direction.

When EP2CONF (D7) = 0, EP2 is in the receive direction and EP2FIFO is in the read-only state.

When EP2CONF (D7) = 1, EP2 is in the transmit direction and EP2FIFO is in the write-only state.

When set for transmission, all bytes of EP2FIFO can be cleared by clearing EP2FIFO (writing a "1" into EP2CONT (D2)).

(5) EP3 FIFO (EP3FIFO)

Address	1A7B
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	x	x	x	x	x	x	x	x
After a bus reset	x	x	x	x	x	x	x	x
Definition	EP3 Transmit data or EP3 receive data							

It is possible to specify the direction of transfer of EP3 by setting the EP3 configuration register EP3CONF. The FIFO address of EP3 is the same in both the transmit direction and the receive direction.

When EP3CONF (D7) = 0, EP3 is in the receive direction and EP3FIFO is in the read-only state.

When EP3CONF (D7) = 1, EP3 is in the transmit direction and EP3FIFO is in the write-only state.

When set for transmission, all bytes of EP3FIFO can be cleared by clearing EP3FIFO (writing a "1" into EP3CONT (D2)).

(6) EP4 FIFO (EP4FIFO)

Address	1A7C
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	x	x	x	x	x	x	x	x
After a bus reset	x	x	x	x	x	x	x	x
Definition	EP4 Transmit data or EP4 receive data							

It is possible to specify the direction of transfer of EP4 by setting the EP4 configuration register EP4CONF. The FIFO address of EP4 is the same in both the transmit direction and the receive direction.

When EP4CONF (D7) = 0, EP4 is in the receive direction and EP4FIFO is in the read-only state.

When EP4CONF (D7) = 1, EP4 is in the transmit direction and EP4FIFO is in the write-only state.

When set for transmission, all bytes of EP4FIFO can be cleared by clearing EP4FIFO (writing a "1" into EP4CONT (D2)).

(7) EP5 FIFO (EP5FIFO)

Address	1A7D
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	x	x	x	x	x	x	x	x
After a bus reset	x	x	x	x	x	x	x	x
Definition	EP5 Transmit data or EP5 receive data							

In USB controller, by making a setting of the system control register, it is possible to select either the 5EP mode with the number of EPs being 5 or the 6EP mode with the number of EPs being 6. In the 5EP mode, only EP0 to EP4 are present and EP5 will not be present. In the EP6 mode, all end points EP0 to EP5 will be valid.

It is possible to specify the direction of transfer of EP5 by setting the EP5 configuration register EP5CONF. The FIFO address of EP5 is the same in both the transmit direction and the receive direction.

When EP5CONF (D7) = 0, EP5 is in the receive direction and EP5FIFO is in the read-only state.

When EP5CONF (D7) = 1, EP5 is in the transmit direction and EP5FIFO is in the write-only state.

When set for transmission, all bytes of EP5FIFO can be cleared by clearing EP5FIFO (writing a "1" into EP5CONT (D2)).

(8) bmRequestType setup register (bmRequestType)

Address	1A00
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition		Type		Receiving side definitions				

0 = Device
1 = Interface
2 = End point
3 = Others
4 to 31 = Reserved

0 = Standard
1 = Class
2 = Vendor
3 = Reserved

Data transfer direction

0 = From the host to the device
1 = From the device to the host

During the setup stage of control transfer based on a request from the host, the 8-byte setup data transmitted by the host is automatically received and is stored in the 8 registers including this register. The formats of these data items are defined in Section 9.3 of the USB Standards.

(9) bRequest setup register (bRequest)

Address	1A01
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	x	x	x	x	x	x	x	x
After a bus reset	x	x	x	x	x	x	x	x
Definition	Request code							

During the setup stage of control transfer based on a request from the host, the 8-byte setup data transmitted by the host is automatically received and the second byte is stored in this register. The content of the request code is defined in Section 9.3 of the USB Standards and in the related standards.

(10) wValueLSB setup register (wValueLSB)

Address	1A02
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	wValueLSB							

During the setup stage of control transfer based on a request from the host, the 8-byte setup data transmitted by the host is automatically received and the third byte is stored in this register. This is the lower-order byte of the two-byte data.

(11) wValueMSB setup register (wValueMSB)

Address	1A03
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	wValueMSB							

During the setup stage of control transfer based on a request from the host, the 8-byte setup data transmitted by the host is automatically received and the fourth byte is stored in this register. This is the higher-order byte of the two-byte data.

(12) wIndexLSB setup register (wIndexLSB)

Address	1A04
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	wIndexLSB							

During the setup stage of control transfer based on a request from the host, the 8-byte setup data transmitted by the host is automatically received and the fifth byte is stored in this register. This is the lower-order byte of the two-byte data.

(13) wIndexMSB setup register (wIndexMSB)

Address	1A05
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	wIndexMSB							

During the setup stage of control transfer based on a request from the host, the 8-byte setup data transmitted by the host is automatically received and the sixth byte is stored in this register. This is the higher-order byte of the two-byte data.

(14) wLengthLSB setup register (wLengthLSB)

Address	1A06
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	wLengthLS							

During the setup stage of control transfer based on a request from the host, the 8-byte setup data transmitted by the host is automatically received and the seventh byte is stored in this register. This is the lower-order byte of the two-byte data.

(15) wLengthMSB setup register (wLengthMSB)

Address	1A07
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	wLengthMSB							

During the setup stage of control transfer based on a request from the host, the 8-byte setup data transmitted by the host is automatically received and the eighth byte is stored in this register. This is the higher-order byte of the two-byte data.

(16) DMA0, 1 control registers (DMA0CON/DMA1CON)

Address	1A10, 1A12
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	The previous value is retained							
Definition					0		0	

DMA Enable
0 = DMA Disabled
1 = DMA Enabled

DMA Byte count
0 = The number of bytes is not inserted
1 = The data of the number of bytes is inserted in the leading byte of the transfer data.

DMA Transfer mode
0 = Single transfer mode
1 = Demand transfer mode

EP Specification
Specifies the target EP for the DMA transfer
0 = EP1, 1 = EP2, 2 = EP4, 3 = EP5 (Note 1)

DMA Interrupting (Note 2)
0 = Normal operation
1 = The DREQ is de-asserted

Note 1: When the EP specifications for the DMA channels 0 and 1 both have the same values, DREQ0, DREQ1 and DACK0, DACK1 will respectively be equivalent.

Note 2: The settings of all bits other than bit D7, that is, of bits D0 to D6 should be completed at the time of initialization (at the latest, before a token packet for EP1 to Ep5 arrives), and should not be altered thereafter. Write a "1" to D7 in order to temporarily stop DMA transfer in the middle. When the transfer is restarted by writing a "0" to D7, it is possible to restart the transfer from the byte (or word) next to the one at which the transfer was interrupted.

Note: When writing data to bits D1 and D3, always write a "0".

(17) DMA0 interval register (DMA0INTVL)

Address	1A11
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	The previous value is retained							
Definition	Interval time							

This specifies the interval in the single DMA0 transfer mode, that is, the time duration after the end of DMA transfer of the previous byte (or the previous word) until DREQ is asserted again. The time for 1 bit is 84 ns (12 MHz, period of one-cycle).

Interval time = (DREQ enable time) + 84xn (ns)

(18) DMA1 interval register (DMA1INTVL)

Address	1A13
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	The previous value is retained							
Definition	Interval time							

This specifies the interval in the single DMA1 transfer mode, that is, the time duration after the end of DMA transfer of the previous byte (or the previous word) until DREQ is asserted again. The time for 1 bit is 84 ns (12 MHz, period of one-cycle).

Interval time = (DREQ enable time) + 84xn (ns)

(19) Device address register (DVCADR)

Address	1A20
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	0	Device address (R/W)						

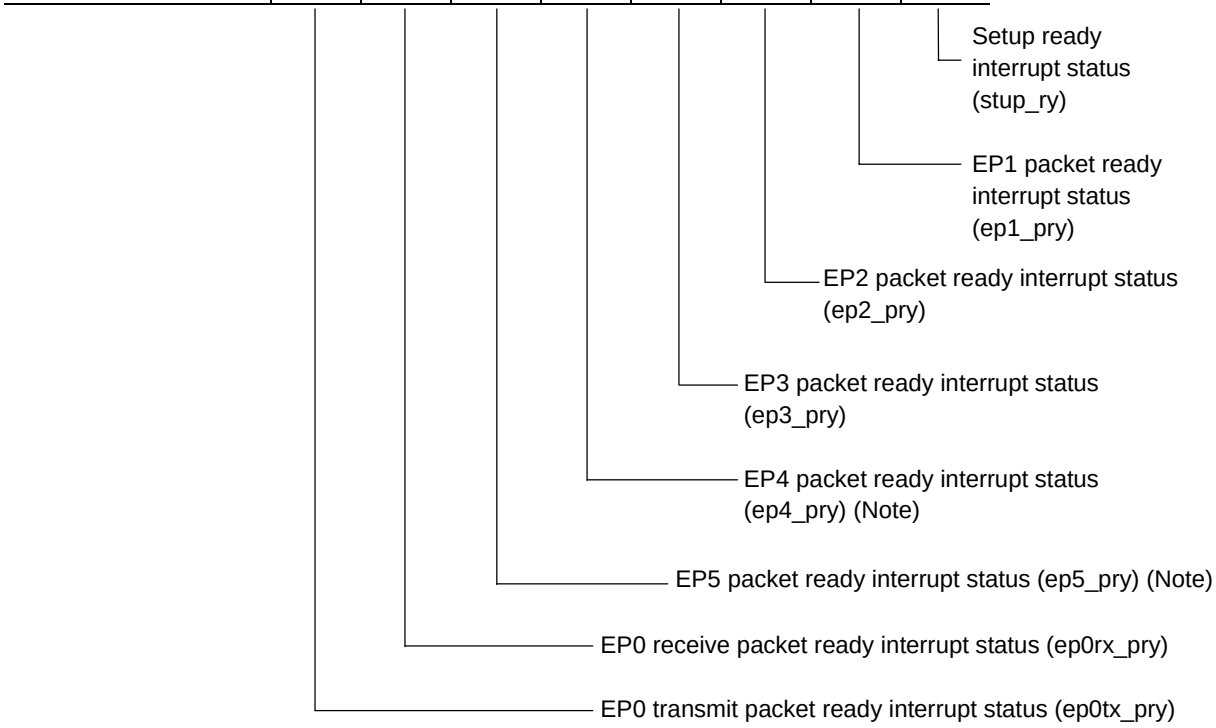
The device address given by a SET_ADDRESS request from the host is written in this register. Thereafter, USB controller judges the specified address in the token from the host, and this device will process only the token packets sent to this device address.

Bit D7 is fixed at "0", and even if a "1" is written, it will be ignored.

(20) Interrupt status register 1 (INTSTAT1)

Address	1A21
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	X	x	x	x	x	x	x	0
Definition								

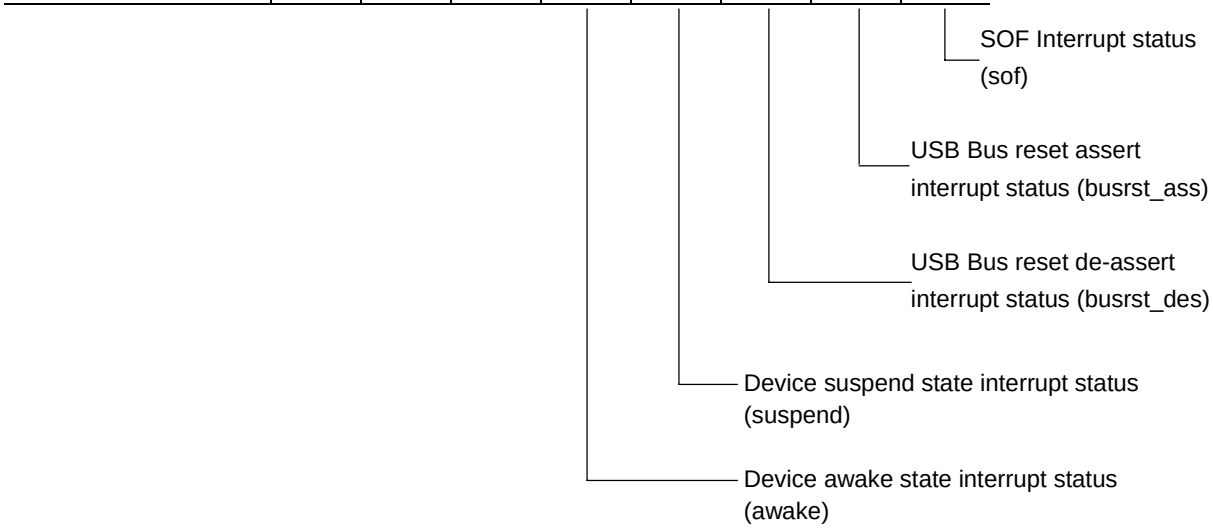


Note: When isochronous transfer has been set in the EP4 or EP5 configuration register, the EP4 or EP5 packet ready interrupt status will always be fixed at "0".

(21) Interrupt status register 2 (INTSTAT2)

Address	1A22
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	0	0	0					



The status bit becomes "1" when the corresponding interrupt is generated.

The status is cleared when a "1" is written in that status bit itself.

(22) Interrupt enable register 1 (INTENBL1)

Address	1A24
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	1
After a bus reset	The previous value is retained							
Definition								

- D0: Setup ready interrupt enable (stup_ry)
- D1: EP1 packet ready interrupt enable (ep1_pry)
- D2: EP2 packet ready interrupt enable (ep2_pry)
- D3: EP3 packet ready interrupt enable (ep3_pry)
- D4: EP4 packet ready interrupt enable (ep4_pry)
- D5: EP5 packet ready interrupt enable (ep5_pry)
- D6: EP0 receive packet ready interrupt enable (ep0rx_pry)
- D7: EP0 transmit packet ready interrupt enable (ep0tx_pry)

(23) Interrupt enable register 2 (INTENBL2)

Address	1A25
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	The previous value is retained							
Definition	0	0	0					

- D0: SOF Interrupt enable (sof)
- D1: USB Bus reset assert interrupt enable (busrst_ass)
- D2: USB Bus reset de-assert interrupt enable (busrst_des)
- D3: Device suspend state interrupt enable (suspend)
- D4: Device awake state interrupt enable (awake)

(24) Frame number LSB register (FRAMELSB)

Address	1A2D
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	Frame number LSB							

This is valid when containing an end point in the isochronous transfer mode. When a start of frame (SOF) packet is transmitted by the host, USB controller automatically writes into the FRAMELSB and FRAMEMSB registers.

(25) Frame number MSB register (FRAMEMSB)

Address	1A2E
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	0	0	0	0	0	Frame number MSB		

This is valid when containing an end point in the isochronous transfer mode. When a start of frame (SOF) packet is transmitted by the host, USB controller automatically writes into the FRAMELSB and FRAMEMSB registers.

(26) System control register (SYSCON)

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	The previous value is retained							0
Definition	0	0						

Address: 1A2F[H]
R/W Access : R/W

- D0: Software reset (R/W) (sreset)
- D1: Power-down mode (R/W) (power)
  - 0 = Power saving is not done in the suspend mode.
  - 1 = Power saving is done in the suspend mode.
- D2: EP Mode (R/W) (ep_mod)
  - 0 = EP0 to EP5
  - 1 = EP0 to EP4
- D3: Pull-up control (R/W) (plup)
- D4: Remote wakeup (R/W)
- D5: PLL Enable (pll_enable)

Software reset: Write-only bit. Even when this bit is read out, it will be fixed at "0". A system reset is executed when a "1" is written in this bit. This is functionally equivalent to a hardware reset. However, this bit itself will always remain "0".

Power-down mode: Read/Write bit. When this bit is "0", the oscillations will not be stopped even during the suspend mode. When this bit is made "1"', the oscillations will be stopped in the suspend mode and the device goes into the power save mode.

EP mode: Read/Write bit. The 6EP mode is selected when this bit is 0 and the 5EP mode is selected when this bit is "1".

Pull-up control: When this bit is "1", the internal switch becomes ON and the PUCTL pin is pulled up to the Vcc level. When this bit is "0", the internal switch is made OFF and the PUCTL pin goes into the Hi-Z output.

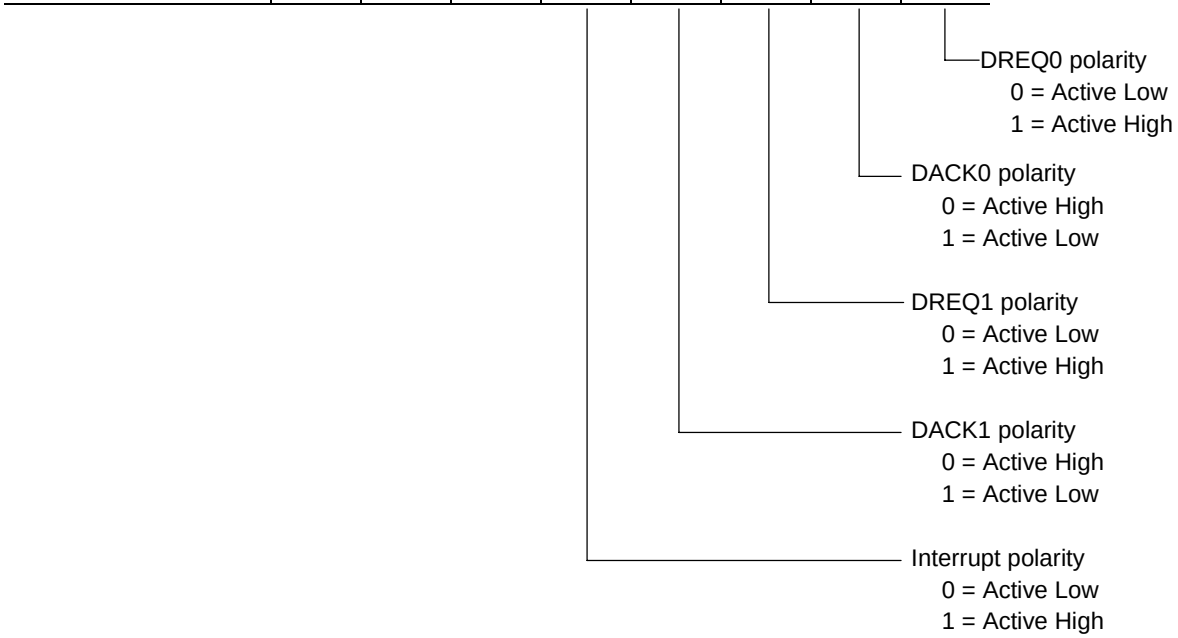
Remote wakeup: A remote wakeup is executed when a "1" is written in this bit. However, this bit itself will always remain "0".

PLL enable: This bit is for enabling the internal PLL. Be sure to set this bit to "1" before USB communication. If this bit is "0", the internal PLL circuit is not used and the USB control function operates under 2-divided clock input to the OSC0 pin, so that USB communication is not made properly.

(27) Polarity selection register (POLSEL)

Address	1A30
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	The previous value is retained							
Definition	0	0	0					



[NOTE] When using as normal operation mode, please keep initial value (All data is "0").

(28) EP0 configuration register (EP0CONF)

Address	1A40
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	1	0	0	0	0
Definition	0	0	0		0	0	0	0

Transfer type (read only)
00 = Control transfer

Configuration bit (read only)

Transfer type: Although these bits indicate the type of transfer, since EP0 has been fixed for control transfer in USB controller, this value is always fixed at 00h.

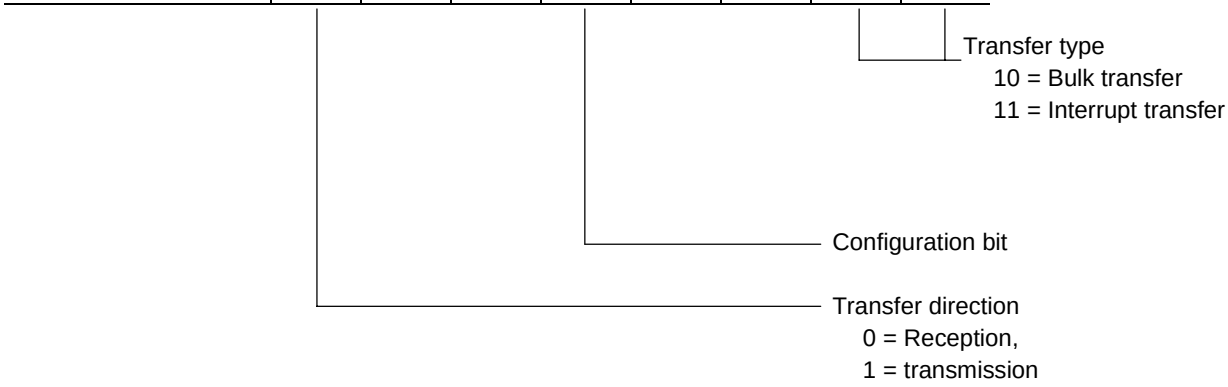
Configuration bit: The configuration bit of EP0 becomes "1" after a USB bus reset.

When this bit is "1", the data transmitted by the host to the end point can be received and also data can be transmitted from the end point to the host. When this bit is "0", this LSI will not respond to any transaction targeting this end point.

(29) EP1, 2, 3 configuration registers (EP1, 2, 3CONF)

Address	1A41 to 1A43
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition		0	0		0	0		



Transfer type: These bits indicate the type of transfer.

Configuration bit: When a Set Configuration request to make that EP active is received from the host, write a "1" into this bit during the status stage in the control transfer mode.

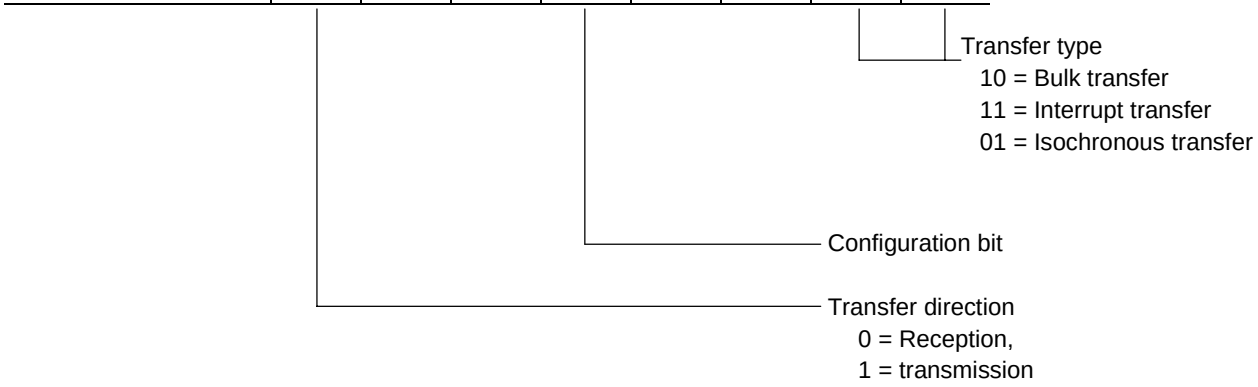
Data transmission and reception can be made between the host and the EP when this bit is "1". When this bit is "0", this LSI will not respond to the transactions targeted at that EP.

Transfer direction: Set the direction of data transfer using this bit.

(30) EP4, 5 configuration registers (EP4, 5CONF)

Address	1A44, 1A45
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition		0	0		0	0		



Transfer type: These bits indicate the type of transfer.

Configuration bit: When a Set Configuration request to make that EP active is received from the host, write a "1" into this bit during the status stage in the control transfer mode.

Data transmission and reception can be made between the host and the EP when this bit is "1". When this bit is "0", this LSI will not respond to the transactions targeted at that EP.

Transfer direction: Set the direction of data transfer using this bit.

(31) EP0 control register (EP0CONT)

Address	1A48
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	x	0	0	x	0
After a bus reset	0	0	0	x	0	0	x	0
Definition	0	0	0		0	0		

Stall bit (R/W)

Data sequence toggle bit (reception)

Data sequence toggle bit (transmission)

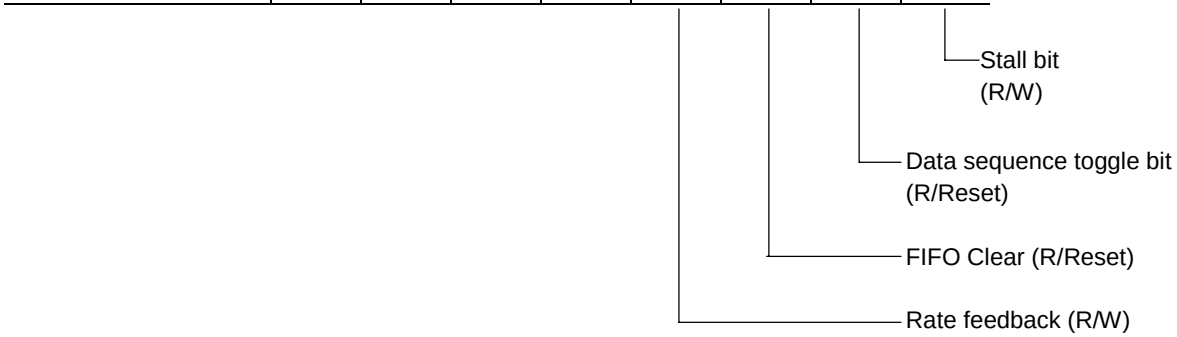
Stall bit: During EP0 reception (data stage of a control write transfer), if a packet with a number of bytes exceeding the maximum packet size specified in EP0PLD is received (or if the EOP packet is missing), USB controller automatically sets this bit to "1". In order to conform to the Protocol Stall of USB Rev. 1.1, this bit is reset automatically to "0" when a setup packet is received.

Data sequence toggle bits: USB controller automatically carries out synchronization using the data sequence toggle mechanism. Further, any write operation to these bits (D4 and D1) will be invalid.

(32) EP1, 2, 3, 4, 5 control registers (EP1, 2, 3, 4, 5CONT)

Address	1A49 to 1A4D
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	0	0	0	0				



Stall bit: During EP0 reception (data stage of a control write transfer), if a packet with a number of bytes exceeding the maximum packet size specified in EP0PLD is received (or if the EOP packet is missing), USB controller automatically sets this bit to "1".

Data sequence toggle bit: A reset will be made when a "1" is written in this bit. At the time of initializing the EP, reset the toggle bit of the data packet by writing a "1" to this bit, and specify PID of DATA0 (this bit too will become "0"). Thereafter, the synchronization operation using the data sequence toggle mechanism will be made automatically.

FIFO Clear: The EP will be valid only when it has been set for transmission by the EP control register. When a "1" is written in this bit, the transmit FIFO of that EP will be cleared. (However, this bit itself will remain "0".)

Rate feedback: This bit is valid only in the case of EP3. This bit will be fixed at "0" in all other EPs.

(33) EP0 payload register (EP0PLD)

Address	1A50
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	1	0	0	0	0	0
After a bus reset	0	0	1	0	0	0	0	0
Definition	0	0	1	0	0	0	0	0

Maximum packet size

Maximum packet size: Since the FIFO of EP0 in USB controller has a size of 32 bytes, write 20h into the byte bMaxPacketSize0 of the device descriptor. The maximum packet size is fixed at 32 bytes in this EP0PLD register. When a packet with more than 32 bytes is received, the stall bit in the EP0 status register is asserted and the stall handshake is returned to the host.

(34) EP1, 2 payload registers (EP1, 2PLD)

Address	1A51, 1A52
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	0	Maximum packet size (R/W)						

Maximum packet size: Write in this register from the CPU the value of the descriptor wMaxPacketSize of the end point selected by the Set_Configuration request from the host. The size of all packets other than a short packet is specified here in units of a byte.

When the EP has been assigned for reception, if a data packet with a number of bytes exceeding the maximum packet size specified in this register is received, the receive packet ready status bit is not asserted, but the stall bit is set in EOP and the stall handshake is returned to the host.

On the other hand, when the EP has been assigned for transmission, the transmit packet ready bit is set automatically when writing by the DMA controller of data with the maximum packet size specified in this register is completed. The content of this register is don't care during non-DMA transmission of data.

(35) EP3 payload register (EP3PLD)

Address	1A53
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	0	0						

Maximum packet size (R/W)

Maximum packet size: Write in this register the value of the descriptor wMaxPacketSize of the end point selected by the Set_Configuration request from the host. The size of all packets other than a short packet is specified here in units of a byte. Set 20h (32 bytes) or less because the FIFO size is 32 bytes.

When EP3 has been assigned for reception, if a data packet with a number of bytes exceeding the maximum packet size specified in this register is received, the receive packet ready bit is not asserted, but the stall bit is set in EOP and the stall handshake is returned to the host.

There is no need to use this register when EP3 has been assigned for transmission.

(36) EP4, 5 payload registers (EP4, 5PLD)

Address	1A54, 1A55
Access type	Read/Write

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	Maximum packet size (R/W)							

Maximum packet size: Write in this register the value of the descriptor wMaxPacketSize of the end point selected by the Set_Configuration request from the host. The maximum packet size is specified in units of a byte.

When the EP has been assigned for reception, if a data packet with a number of bytes exceeding the maximum packet size specified in these registers is received, the receive packet ready bit is not asserted, but the stall bit is set in EOP and the stall handshake is returned to the host.

On the other hand, when the EP has been assigned for transmission, the transmit packet ready bit is set automatically when writing by the DMA controller of data with the maximum packet size specified in this register is completed.

(37) EP0 receive byte count register (EP0RXCNT)

Address	1A58
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	0	0	Receive byte count (R)					

USB controller automatically counts the number of bytes in the packet being received. Although the counting is done only up to the number of bytes equal to the maximum packet size specified in the payload register in the case of a full packet, the count will be less than that size in the case of a short packet. Refers to this value and reads out the data of one packet from the EP0 Receive FIFO.

The EP0RXCNT register is cleared under the following conditions.

1. When the EP receive packet ready bit is reset.
2. When a setup packet is received.
3. When a "0" is written into the stall bit.

(38) EP1, 2 receive byte count registers (EP1, 2RXCNT)

Address	1A59, 1A5A
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	0	Receive byte count (R)						

USB controller automatically counts the number of bytes in the packet being received. Although the counting is done only up to the number of bytes equal to the maximum packet size specified in the payload register in the case of a full packet, the count will be less than that size in the case of a short packet. Refers to this value and reads out the data of one packet from the EP1/2 Receive FIFO.

This register will be invalid when the transfer direction of the EP is set for transmission.

The EP1,2RXCNT register is cleared under the following conditions.

1. When an OUT token is received for the EP.
2. When the EP receive packet ready bit is reset.
3. When a "0" is written into the stall bit.

(39) EP3 receive byte count register (EP3RXCNT)

Address	1A5B
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	0	0	Receive byte count (R)					

USB controller automatically counts the number of bytes in the packet being received. Although the counting is done only up to the number of bytes equal to the maximum packet size specified in the payload register in the case of a full packet, the count will be less than that size in the case of a short packet. Refers to this value and reads out the data of one packet from the EP3 Receive FIFO.

This register will be invalid when the transfer direction of the EP is set for transmission.

The EP3RXCNT register is cleared under the following conditions.

1. When an OUT token is received for EP3.
2. When the EP receive packet ready bit is reset.
3. When a "0" is written into the stall bit.

(40) EP4, 5 receive byte count registers (EP4, 5RXCNT)

Address	1A5C, 1A5D
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	Receive byte count LSB (R)							

USB controller automatically counts the number of bytes in the packet being received. Although the counting is done only up to the number of bytes equal to the maximum packet size specified in the payload register in the case of a full packet, the count will be less than that size in the case of a short packet. Refers to this value and reads out the data of one packet from the EP4/5 Receive FIFO.

This register will be invalid when the transfer direction of the EP is set for transmission.

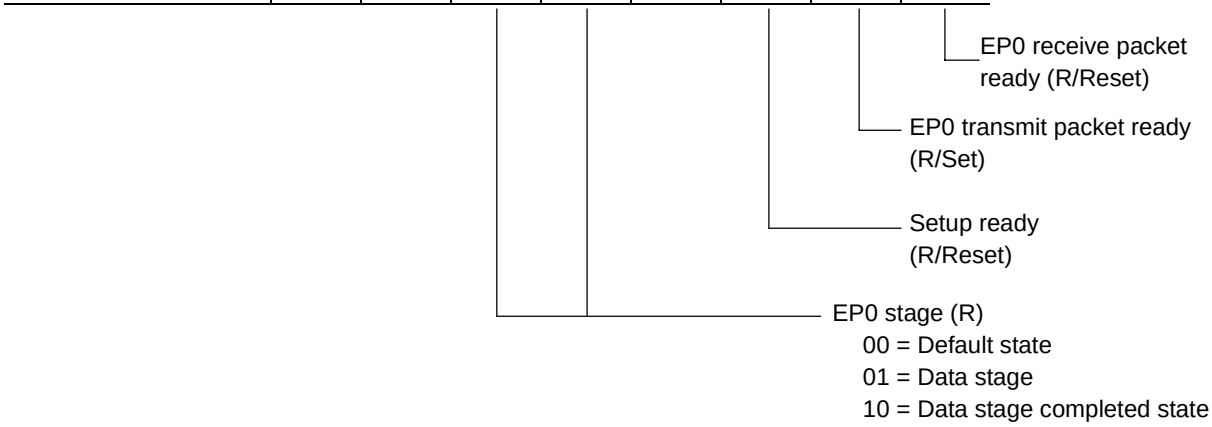
The EP4,5RXCNT register is cleared under the following conditions.

1. When an OUT token is received for the EP.
2. When the EP receive packet ready bit is reset.
3. When a "0" is written into the stall bit.

(41) EP0 status register (EP0STAT)

Address	1A60
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	0	0			0			



Setup ready: This bit is set to "1" automatically when a setup packet normally arrives in the 8-byte setup register, and the EP0 Receive FIFO is locked.

If INTENBL1(0) is asserted, the interrupt is also asserted automatically when this bit is set to "1". Write a "1" into this bit after reading out the 8-byte setup data.

When this is performed, the setup ready bit is reset and the internal interrupt signal also is de-asserted. During a control write transfer, the packet ready bit of EP0 is reset simultaneously and the lock condition is released, and it becomes possible to receive packets by EP0 during the data stage. The register will not change even if a "0" is written in this bit.

EP0 transmit packet ready bit (D1): Writing when D1 = 1 sets this bit to "1". The asserting and de-asserting conditions are described below.

Bit name	Asserting condition	Operation when asserted
EP0 transmit packet ready (D1)	When this bit is set to "1"	Data can be transmitted from EP0.

Bit name	De-asserting condition	Operation when de-asserted
EP0 transmit packet ready (D1)	1. When an ACK is received from the host for data transmission 2. When a setup packet is received	EP0 is locked. That is, an NAK is automatically returned when an IN token is sent from the host.

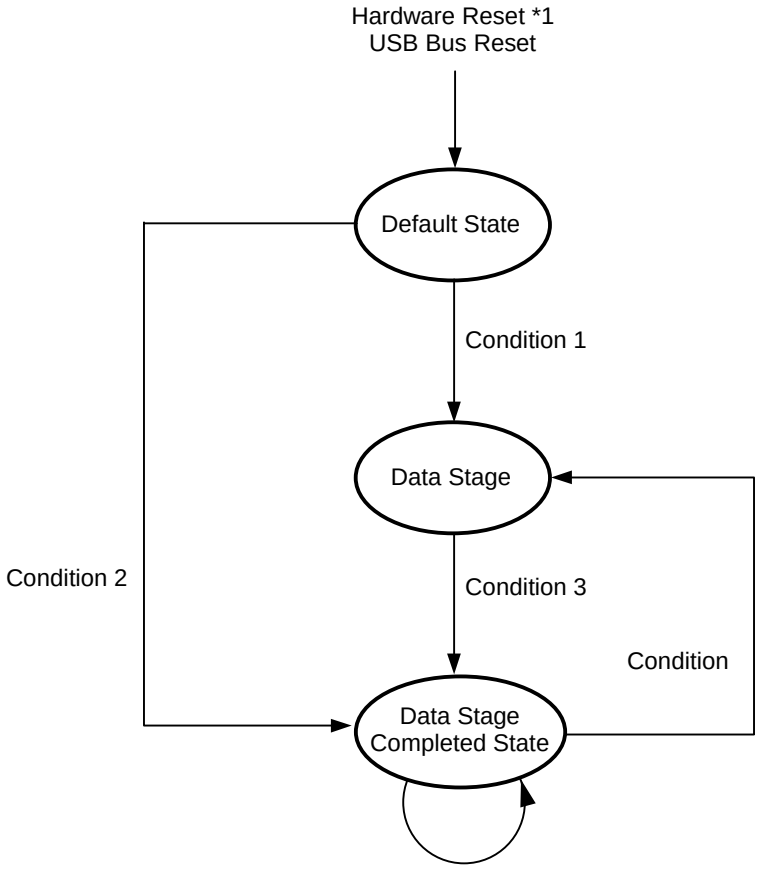
EP0 receive packet ready bit (D0): Writing when D1 = 1 sets this bit to "0". The asserting and de-asserting conditions are described below.

Bit name	Asserting condition	Operation when asserted
EP0 receive packet ready (D0)	1. When data is received by EP0 and stored in FIFO 2. When a setup packet is received during control Read transfer or control Write transfer	EP0 is locked (an NAK is automatically returned when a data packet is received from the host).

Bit name	De-asserting condition	Operation when de-asserted
EP0 receive packet ready (D0)	1. When this bit is reset (a "1" is written in this bit) 2. When the setup ready bit is reset during control Write transfer	EP0 can receive data.

EP0 Stage (D5, D4): These bits indicate the stage transition during control transfer.

The flowchart of the stage transition is shown below.



Condition 1: Reception of a setup packet of control READ transfer or control WRITE transfer.

Condition 2: Reception of a setup packet of control transfer without data.

Condition 3: Reception of a token (IN/OUT) of a direction opposite to the data flow in the data stage.

*1 A hardware reset on the USB core is controlled by bit 0 (USBRST) of the internal control register (P5IO), not by the external reset input pin (RESn). (See Section 13.4.)

When reset (n signal input, execution of a BRK instruction, overflow of the watchdog timer, or opcode trap), USBRST, the hardware reset control flag for the USB core, is set to "0" to reset the USB core. It is therefore required to release the reset state of the USB core by setting USBRST to "1" by the firmware.

(42) EP1, 2, 4, 5 status registers (EP1, 2, 4, 5STAT)

Address	1A61, 1A62, 1A64, 1A65
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	0	0	0	0	0	0		

EP Receive packet ready (Read/Reset)

EP Transmit packet ready (Read/Set)

This register is valid only when the corresponding EP has been set for bulk or interrupt transfer.

EP1, 2, 4, 5 Receive packet ready bit (D0):

This bit can be made "0" by writing a "1" into bit D0. The asserting and de-asserting conditions of this bit are as given below. The FIFOs of EP1, EP2, EP4, and EP5 have a 2-layer structure and also there are independent packet ready bits for layer A and layer B. The switching between these two layers is done automatically by USB controller.

Bit name	Asserting condition	Operation when asserted
EP1 Receive packet ready (D0)	When an error-free packet is received in either layer A or layer B.	Can read the EP1 Receive FIFO. EP1 is locked in the condition in which data packets have been received by both layer A and layer B.

Bit name	De-asserting condition	Operation when de-asserted
EP0 receive packet ready (D0)	When the bits of both layer A and layer B are reset (when a "1" is written in).	Reception can be made by EP1 when the bit of either layer A or layer B has been reset.

* This bit is automatically de-asserted when received data in both layer A and layer B has all been transferred by the DMA controller.

EP1, 2, 4, 5 Transmit packet ready bit (D1):

This bit can be made "1" by writing a "1" into bit D1. The asserting and de-asserting conditions of this bit are as given below. The FIFO of EP1 has a 2-layer structure and also there are independent packet ready bits for layer A and layer B. The switching between these two layers is done automatically by USB controller.

Bit name	Asserting condition	Operation when asserted
EP1 Transmit packet ready (D1)	When the bits of both layer A and layer B are set to "1".	Transmission can be made from EP1 when either layer A or layer B has been asserted.

* This bit is automatically asserted when as much data as the number of bytes specified by EP1, EP2, EP4, EP5 and PLD has been transferred into both layer A and layer B by the DMA controller.

Bit name	De-asserting condition	Operation when de-asserted
EP1 Transmit packet ready (D1)	When an ACK message is received from the host for the data transmission to either layer A or layer B.	EP1 is locked when transmit data has not been prepared for both layer A and layer B.

(43) EP3 status register (EP3STAT)

Address	1A63
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	0	0	0	0	0	0		

EP3 Receive packet ready (Read/Reset)

EP3 Transmit packet ready (Read/Set)

This register is valid only when EP3 has been set for bulk or interrupt transfer.

EP3 Receive packet ready bit (D0):

This bit can be made "0" by writing a "1" into bit D0. The asserting and de-asserting conditions of this bit are as given below.

Bit name	Asserting condition	Operation when asserted
EP3 Receive packet ready (D0)	When an error-free packet is received.	EP3 is locked.

Bit name	De-asserting condition	Operation when de-asserted
EP3 Receive packet ready (D0)	When this bit is reset . (written a "1" in)	Reception can be made by EP3.

EP3 Transmit packet ready bit (D1):

This bit can be made "1" by writing a "1" into this bit. The asserting and de-asserting conditions of this bit are as given below.

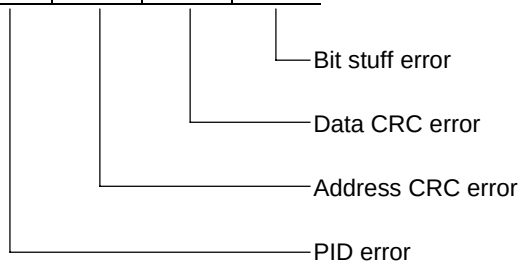
Bit name	Asserting condition	Operation when asserted
EP3 Transmit packet ready (D1)	When this bit is set to "1".	Transmission can be made from EP3.

Bit name	De-asserting condition	Operation when de-asserted
EP3 Transmit packet ready (D1)	When an ACK message is received from the host for the data transmission from EP3.	EP3 is locked.

(44) Packet error register (PKTERR)

Address	1A39
Access type	R

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	0	0	0
Definition	0	0	0	0				



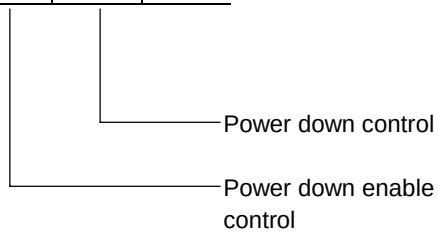
Each of the above bits is asserted when the corresponding error occurs, and is de-asserted when SOP is received.

The error information is reported by this register. There is no particular need to access this register other than for testing during development or for developing a product for making reports of measurement of error occurrence frequency.

(45) OSC test register (OSCTEST)

Address	1A3A
Access type	R/W

	D7	D6	D5	D4	D3	D2	D1	D0
After a hardware reset	0	0	0	0	0	0	0	0
After a bus reset	0	0	0	0	0	The previous value is retained		
Definition	0	0	0	0	0			0



Power down control:

Writing a "0" to this bit puts the OSC oscillation circuit and PLL into power-down mode forcibly.

Writing a "1" to this bit enables the OSC oscillation circuit and PLL.

Note: This bit setting is valid when bit 2 (power down enable control) is set to "1".

Power down enable control:

Writing a "1" to this bit enables the setting of power down control (bit 1).

Writing a "0" to this bit disables the setting of power down control (bit 1).

[Note]

This power-down setting can stop the OSC oscillation circuit only when OSCS (bit 3 of SBYCON) is "1" and the CPU is either in the STOP mode or operating under the subclock (XTCLK).

If OSCS is "1" but the CPU is not in the STOP mode, or if OSCS is "1" but the CPU is not operating under the subclock (XTCLK), the OSC oscillation circuit does not stop, because a clock must be supplied to the CPU. (See Chapter 3.)

17.5 USB Transceiver

17.5.1 Overview of USB Transceiver Circuit

The USB transceiver uses a differential input/output circuit with D+ as the positive logic signal and D– as the negative logic signal.

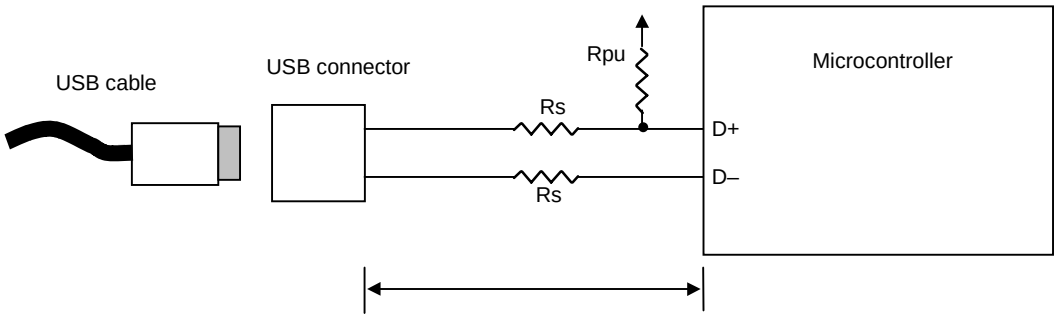
The input circuit is provided with a differential input receiver of the sense amplifier type and a single-end receiver of TTL Schmitt trigger input.

17.5.2 Method for and Notes on Connecting the USB Cable to USB Terminal (D+/D–)

This section explains how to connect the USB cable to USB terminal (D+/D–) and notes on the connection.

The section also shows recommended values for external parts. Evaluate the values with the printed circuit board for an application product well before setting them.

(1) Length of the wiring between the USB connector and USB terminal (D+/D–)



(2) External series resistor

The external series resistor Rs is necessary between the USB connector and USB terminal (D+/D–).

[Recommended value] Rs = 22 Ω (Precision: ±5%, Watt: 1/8 W)

(3) Pull-up resistor on the D+ line (high-speed data transfer)

The value of the pull-up resistor Rpu on the D+ line to transfer data at a high speed is as follows:

[Recommended value] Rpu = 1.5 kΩ (Precision: ±5%, Watt: 1/8 W)

(4) Procedure required if the USB cable has not been connected

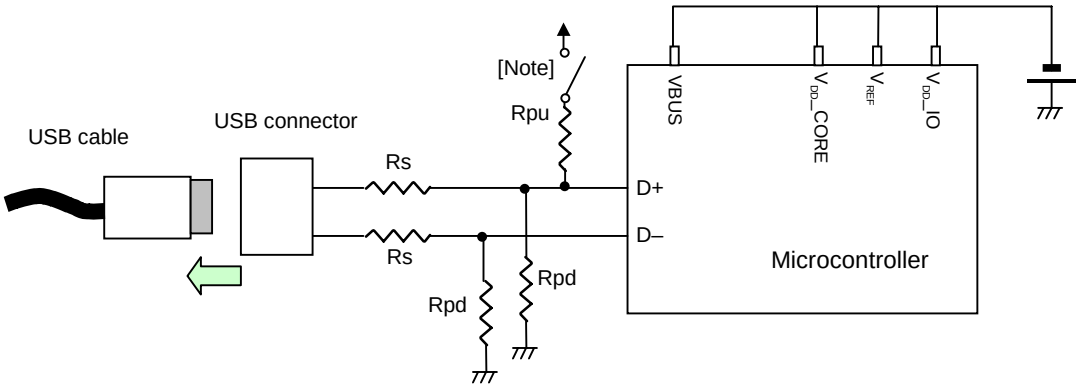
If the USB cable has not been connected (in other words, the USB terminal (D+/D–) is floating), through-current may flow to the single-end receiver of the USB receiver. In this case, take the following procedure:

When self-power supply is used:

The USB terminal (D+/D–) floats and through-current flows to the input gate of the single-end receiver of the USB transceiver in the following condition:

The USB block does not use power supplied through the USB cable. Instead, it uses self-power supply as the power supply (VBUS).

In this case, connect a pull-down resistor to the D+ and D– lines each to prevent the through-current.

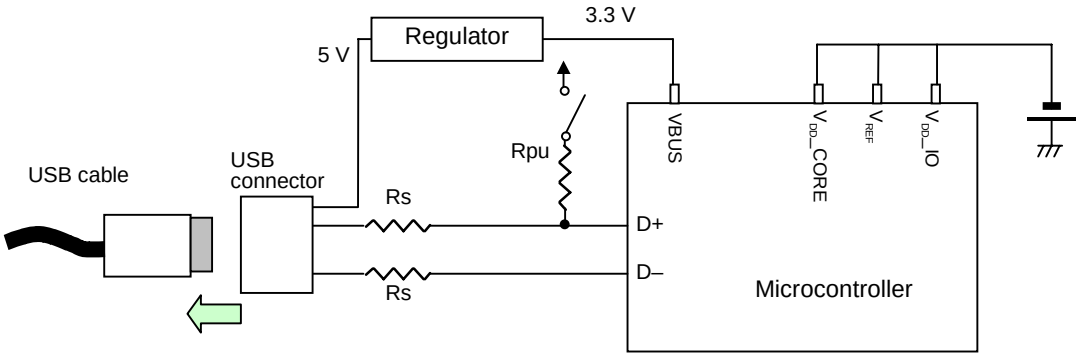


[Recommended value] Rpd = 470 kΩ (Precision ±5%, Watt: 1/8 W)

[Note] If the USB cable has not been connected, the pull-up resistor Rpu needs to be disconnected from the power supply.

When power is supplied through the cable:

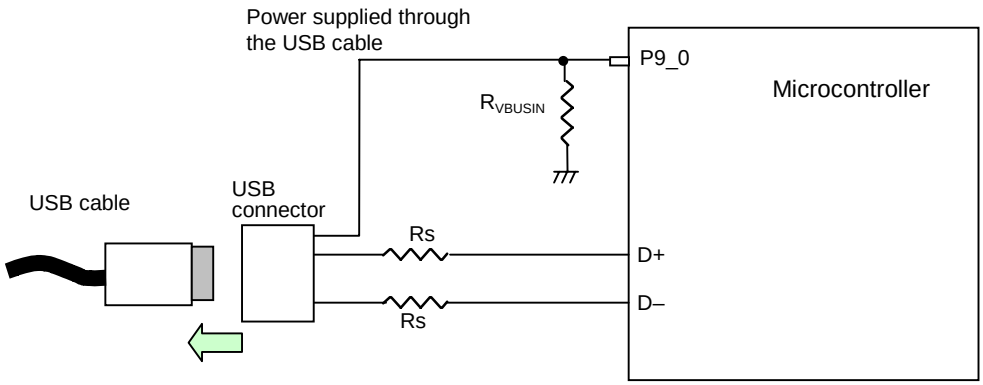
If the USB block uses the power supply (VBUS) through the USB cable, the USB terminal (D+/D–) floats while the USB cable is not connected. In this case, power is not supplied to the USB transceiver (and the USB control function block), and through-current is not generated at the input gate of the single-end receiver. Thus, the D+ and D– lines do not need the pull-down resistor Rpd.



17.6 Notes on Using the USB

To detect whether or not the USB cable is connected, input the power supplied through the USB cable to P9_0. Power of 5 V can be input directly to P9_0.

If the USB cable has not been connected, P9_0 floats and through-current flows to the input gate of P9_0. To prevent the through-current, connect the external pull-down resistor R_{VBUSIN} to P9_0.



[Recommended value] R_{VBUSIN} = 1 MΩ

While the USB cable is connected, current always flows to the pull-down resistor R_{VBUSIN}. In consideration of the current, evaluate the value with the printed circuit board of an application product well.

Chapter 18

Media Control Function

18. Media Control Function

18.1 Overview

A media control circuit is built in this device so that high speed data transfer can be made to the external NAND flash memory.

This circuit allows data to be written into or read from the external NAND flash memory in units of one byte, and the use of data transfer memory (buffer RAM) allows data to be written into or read from in units of multiple bytes.

The memory space for data transfer is present in the expanded RAM area (address 1200H to 19FFH) and is divided into the four banks 0 to 3 of 512 bytes each.

18.2 Registers for Media Control Functions

Table 18-1 lists a summary of the SFRs for the media control functions.

Table 18-1 Summary of SFRs for the Media Control Functions

Address [H]	Name	Symbol (BYTE)	Symbol (WORD)	R/W	8/16 operation	Initial Value[H]	Reference page
1B04	Media sequencer control register	MSCTRL	—	R/W	8	00	18-3
1B06*	Media sequencer wait register	MSWAIT	—	R/W	8	00	18-7
1B08*	Media sequencer status register	MSSTS	—	R	8	1E	18-8
1B0A*	Media sequencer error status register	MSERR	—	R	8	00	18-9
1B0C	Media command register	MMCMD	—	W	8	0	18-10
1B0E	Media address register	MMADR	—	W	8	0	18-10
1B10	Media data register	MMDATA	—	R/W	8	0	18-11
1B12*	Media select register	MMSEL	—	R/W	8	00	18-11
1B14	ECC1 Line parity register	—	ECC1LP	R	16	FFFF	18-12
1B15		—					
1B16	ECC2 Line parity register	—	ECC2LP	R	16	FFFF	18-12
1B17		—					
1B18*	ECC1 Column parity register	—	ECC1CP	R	16	003F	18-13
1B19		—					
1B1A*	ECC2 Column parity register	—	ECC2CP	R	16	003F	18-13
1B1B		—					
1B1C*	ECC1 Error pointer register	—	ECC1ERR	R	16	0000	18-14
1B1D		—					
1B1E*	ECC2 Error pointer register	—	ECC2ERR	R	16	0000	18-14
1B1F		—					
1B20	Redundancy part reserved data 1 register	FHD0	HREV1	R/W	8/16	0000	18-15
1B21		FHD1					
1B22	Redundancy part reserved data 2 register	FHD2	HREV2	R/W	8/16	0000	18-15
1B23		FHD3					
1B24	Redundancy part data/block status register	FHD4	HSTATS	R/W	8/16	0000	18-16
1B25		FHD5					
1B26	Redundancy part block address 1 register	FHD6	HBADR1	R/W	8/16	0000	18-17
1B27		FHD7					
1B28	Redundancy part ECC2-High register	FHD8	HECC2H	R/W	8/16	0000	18-18
1B29		FHD9					
1B2A	Redundancy part ECC2-Low/block address 2 register	FHD10	HECC2LA	R/W	8/16	0000	18-19
1B2B		FHD11					
1B2C	Redundancy part ECC1-High/block address 2 register	FHD12	HECC1HA	R/W	8/16	0000	18-20
1B2D		FHD13					
1B2E	Redundancy part ECC1-Low register	FHD14	HECC1L	R/W	8/16	0000	18-21
1B2F		FHD15					

[Notes]
1. Addresses are not consecutive in some places.
2. An asterisk in the address column indicates a missing bit.
3. For details, refer to Chapter 22 "Special Function Registers (SFRs)".
4. The redundancy part registers (FHD0 to FHD15) correspond to the redundancy part of the SmartMedia™ physical format. Refer to "SmartMedia™ Physical Format" for the data format of the redundancy part.

18.2.1 Description of the Registers for the Media Control Functions

(1) Media sequencer control register (MSCTRL)

This register specifies the operation of the media sequencer.

Bit 0 shows start of data transfer, bit 1 shows data transfer direction, bits 2 and 3 show Data length (the number of data bytes transferred between the media sequencer and the RAM), bits 4 and 5 show presence or absence of read/write of the redundancy part, bit 6 shows ECC Enabled or disabled, and bit 7 shows Parity enabled or disabled (control of the parity bit of the block address during the SmartMedia format of the redundancy part).

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

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), MSCTRL becomes 00H.

Figure 18-1 shows the MSCTRL configuration.

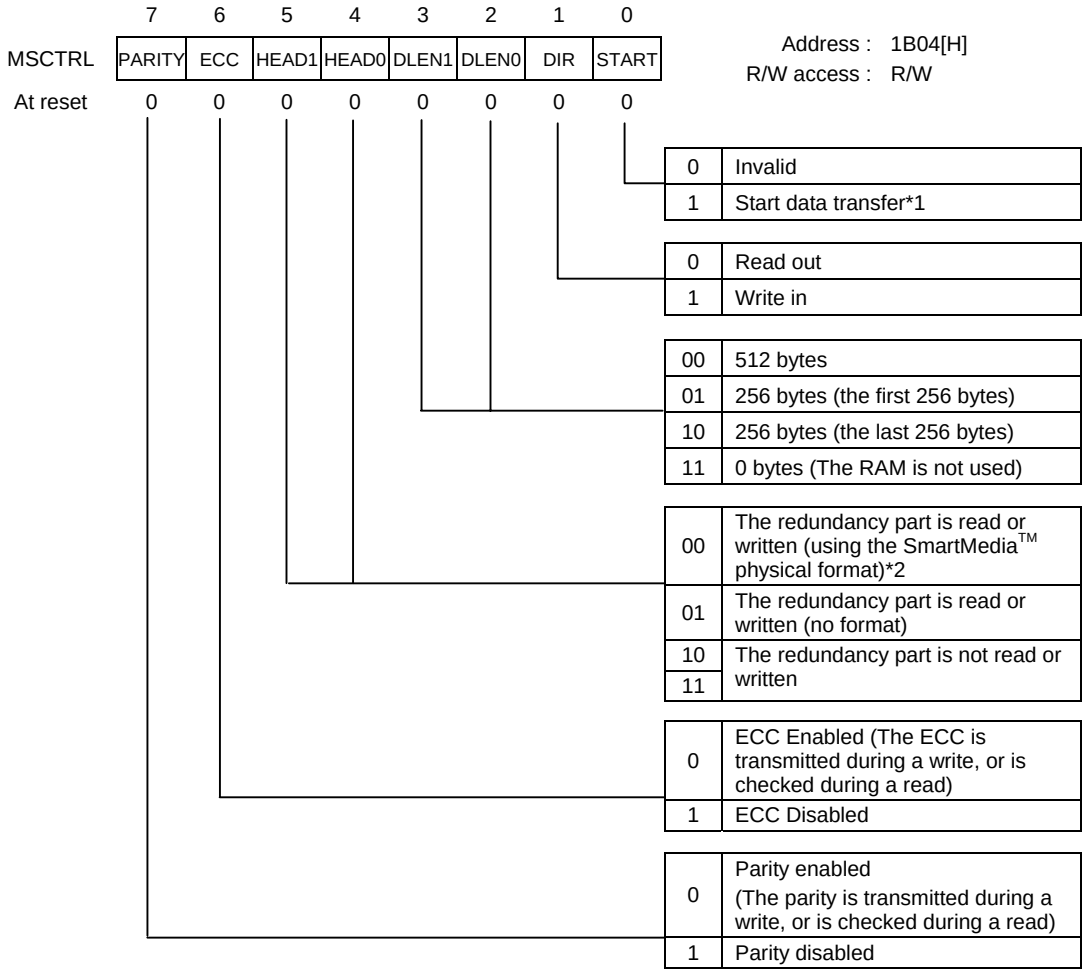


Figure 18-1 MSCTRL Configuration

Start of data transfer

*1 When a "1" is written, the data transfer is started. Writing a "0" is invalid. Start data transfer after confirming that the sequencer is stopped by reading the media sequencer status register (MSSTS). The operation is unpredictable if bit 0 (START) is set to "1" while the sequencer is operating. When read its value will be "0".

*2 When bits 4 and 5 (HEAD0, 1) are set to 00h, during a write operation, the fixed value FFFFFFFFh is written in the reserved area and the fixed value 00010b is written to the first 5 bits of the block address area according to the redundancy part format of SmartMedia™. Also, the same value as the block address area 1 is written in the block address area 2. No control is carried out during a read operation.

[Note]

When "Parity enabled" is set, a parity check that differs from the one described in SMIL (SmartMedia Interface Library) Version 1.00 will be performed. If conforming to SMIL Version 1.00 is required, set the bit to "Parity disabled" and generate a parity bit by firmware to write to the redundancy part.

For reference, the SmartMedia™ physical format is shown in Figure 18-2. For details, refer to the SmartMedia™ standards.

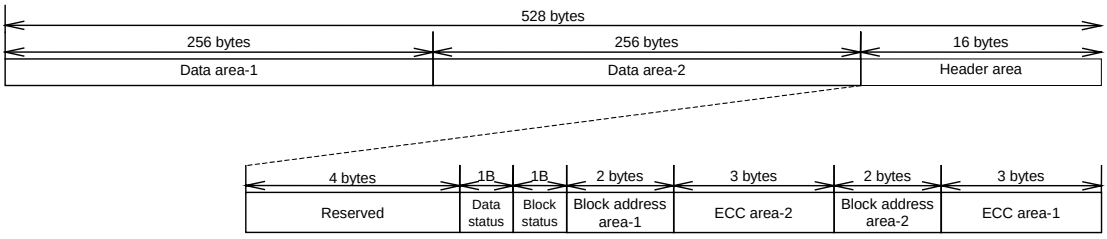
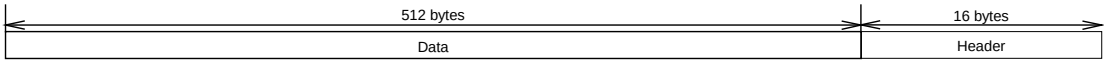


Figure 18-2 SmartMedia™ Physical Format

The different modes of data transfer for the sequencer and the purposes of using them are given below.

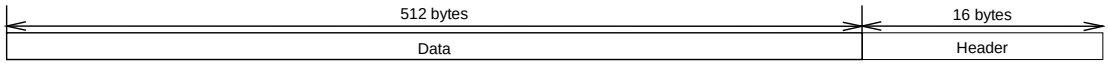
1) Data transfer mode 1 (MSCTRL register: HEAD = 00b, DLEN = 00b)
   Normal data transfer (read/write) in the SmartMedia format.



- Data transfer memory: 512 bytes transferred
- Redundancy part processing: Read/write is done according to the SmartMedia™ physical format.

Figure 18-3 Data Transfer Mode 1

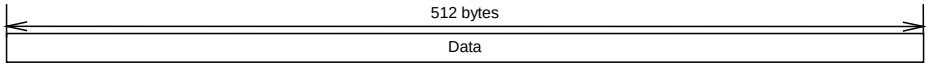
2) Data transfer mode 2 (MSCTRL register: HEAD = 01b, DLEN = 00b)
   Normal data transfer (read/write).



- Data transfer memory: 512 bytes transferred
- Redundancy part processing: Read/write is done without recognizing the format.

Figure 18-4 Data Transfer Mode 2

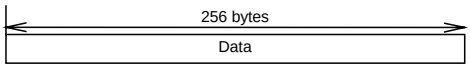
3) Data transfer mode 3 (MSCTRL register: HEAD = 1xb, DLEN = 00b)
   Data read for debugging.



- Data transfer memory: 512 bytes transferred
- Redundancy part processing: None

Figure 18-5 Data Transfer Mode 3

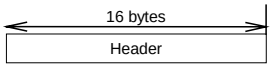
4) Data transfer mode 4 (MSCTRL register: HEAD = 1xb, DLEN = 01b or 10b)
Data read for debugging (reading Data Area 1 or Data Area 2).



- Data transfer memory: 256 bytes transferred
- Redundancy part processing: None

Figure 18-6 Data Transfer Mode 4

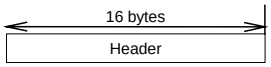
5) Data transfer mode 5 (MSCTRL register: HEAD = 00b, DLEN = 11b)
Normal redundancy part read in the SmartMedia format.



- Data transfer memory: 0 bytes transferred
- Redundancy part processing: Reading is done by recognizing the SmartMedia™ physical format.

Figure 18-7 Data Transfer Mode 5

6) Data transfer mode 6 (MSCTRL register: HEAD = 01b, DLEN = 11b)
Normal reading of the redundancy part.



- Data transfer memory: 0 bytes transferred
- Redundancy part processing: Reading is done without recognizing the format.

Figure 18-8 Data Transfer Mode 6

(2) Media sequencer wait register (MSWAIT)

This register controls the wait function of the media sequencer.

Bits 0 and 1 sets the wait function when the sequencer reads data from the flash memory and bits 2 and 3 sets the wait function when the sequencer writes data to the flash memory.

MSWAIT can be read from and written to by the program. However, write operations are invalid for bits 5 to 7. Also, if writing to bit 4, it must be written as "0". If read, a value of "0" will always be obtained for bits 4 to 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), MSWAIT becomes 00H.

Figure 18-9 shows the MSWAIT configuration.

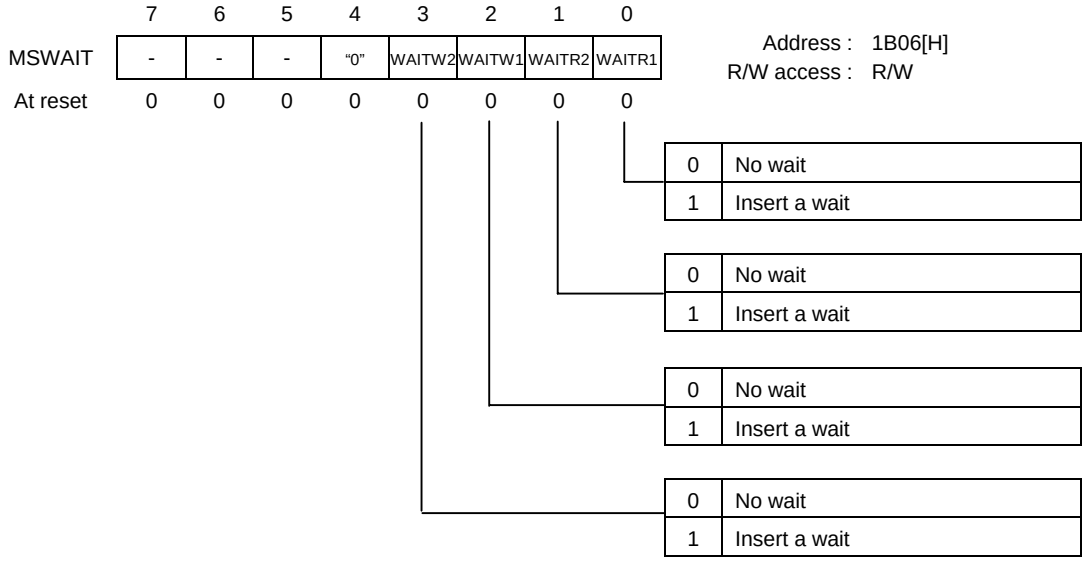


Figure 18-9 MSWAIT Configuration

When WAITW1 and WAITW2 are both set to "1", the write wait becomes "2"; when WAITR1 and WAITR2 are both set to "1", the read wait becomes "2".

For the details of the wait function, refer to Section 18.3.

(3) Media sequencer status register (MSSTS)

This register indicates the state of the media sequencer.

Bit 0 indicates the sequencer operation has been completed, bit 1 indicates Media Sequencer Ready/Busy, bit 2 indicates that no ECC error was detected, bit 3 indicates that no parity error was detected, bit 4 indicates the ready or busy state of the flash memory that has been connected.

The bits other than bit 0 (MSCOMP) can only be read by the program. Write operations to the bits other than bit 0 are invalid. If read, a value of "0" will always be obtained for bits 5 to 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of MSSTS becomes 1EH.

Figure 18-10 shows the MSSTS configuration.

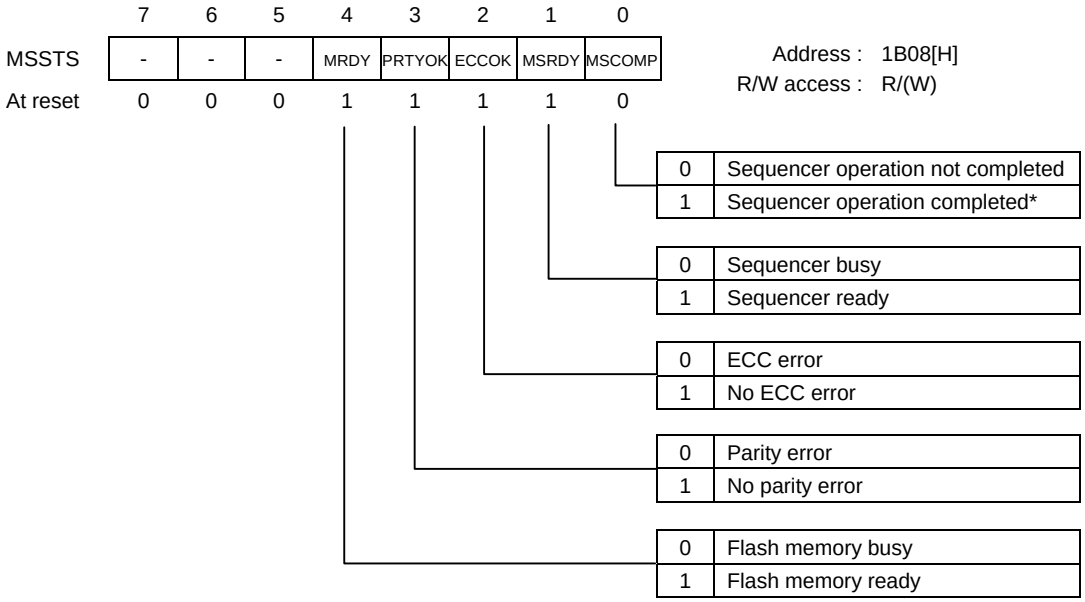


Figure 18-10 MSSTS Configuration

* Sequencer operation completed
When bit 0 (MSCOMP) is set to "1", it indicates that the sequencer operation has been completed and the interrupt signal to the CPU (interrupt from the internal Media contoller) has been asserted.
Writing a "1" to bit 0 clears the interrupt, but writing a "0" does not.

[Notes]

1: The result of parity checking of the block address area when reading the redundancy part in the SmartMedia format.
However, in SMIL (SmartMedia Interface Library) Version 1.00, a method is written that the parity bit for the block address includes a fixed value 00010b and that even parity for the entire block address area is to be generated. In this case, it is necessary to generate a parity bit by firmware and write to the redundancy part.
2: The result of ECC operation mentioned here is based on the SmartMedia specifications. According to the ECC specifications, the situations of wrong correction, wrong detection, or non-detection of errors can occur.

(4) Media sequencer error status register (MSERR)

This register indicates the error information of the media sequencer.

Bits 0 to 3 indicate the result of ECC operation. Bits 4 and 5 indicate the result of parity checking when the redundancy part is the SmartMedia format.

MSERR can only be read by the program. Write operations are invalid. If read, a value of "0" will always be obtained for bits 6 and 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of MSERR becomes 00H.

Figure 18-11 shows the MSERR configuration.

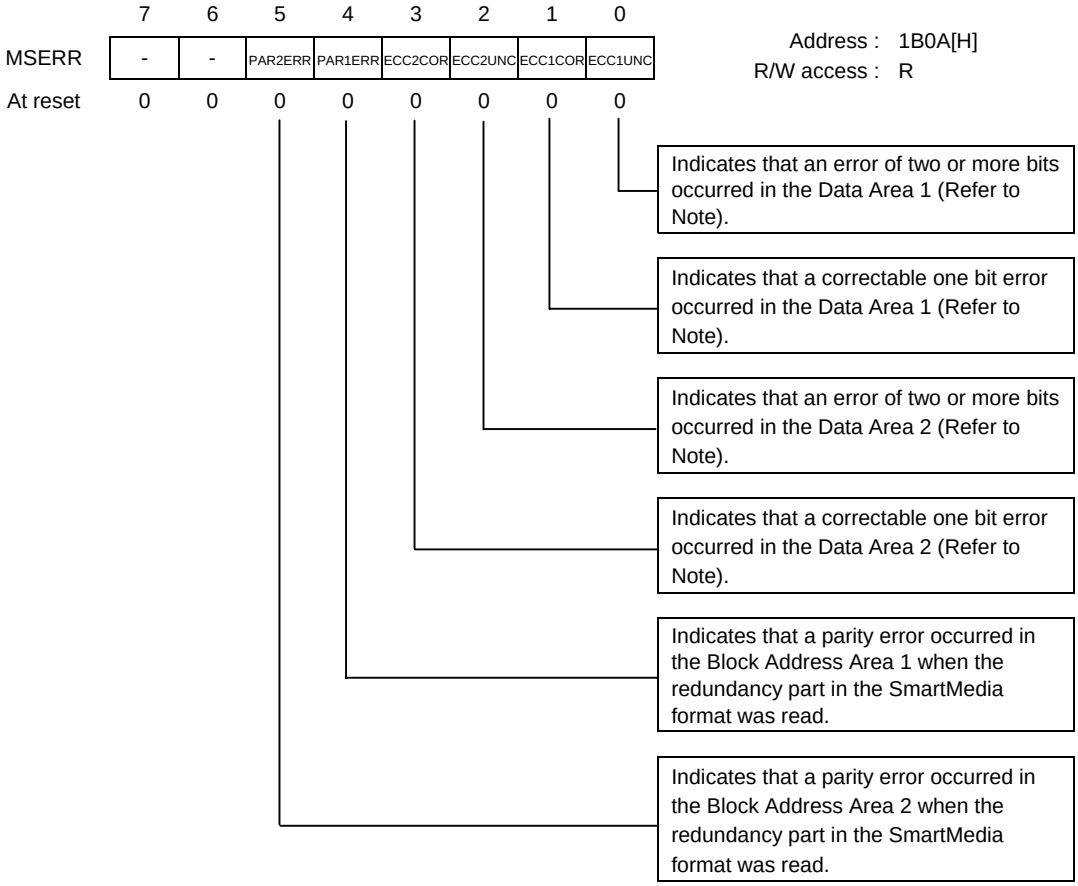


Figure 18-11 MSERR Configuration

[Note]

The result of ECC operation mentioned here is based on the SmartMedia specifications. According to the ECC specifications, the situations of wrong correction, wrong detection, or non-detection of errors can occur.

(5) Media command register (MMCMD)

This register is used for writing a command to the external NAND flash memory setting CLE = 1 and ALE = 0.

MMCMD can only be written by the program.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of MMCMD is undefined.

Figure 18-12 shows the MMCMD configuration.

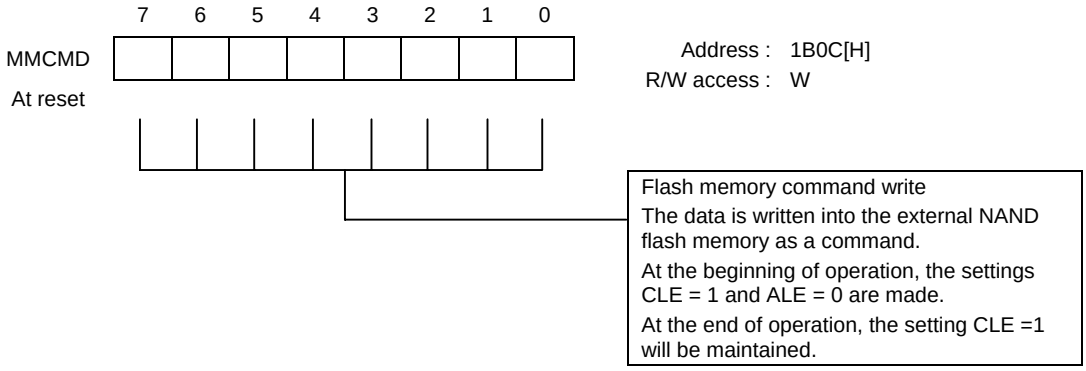


Figure 18-12 MMCMD Configuration

(6) Media address register (MMADR)

This register is used for writing an address to the flash external NAND memory setting CLE = 0 and ALE = 0.

MMADR can only be written by the program.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of MMADR is undefined.

Figure 18-13 shows the MMADR configuration.

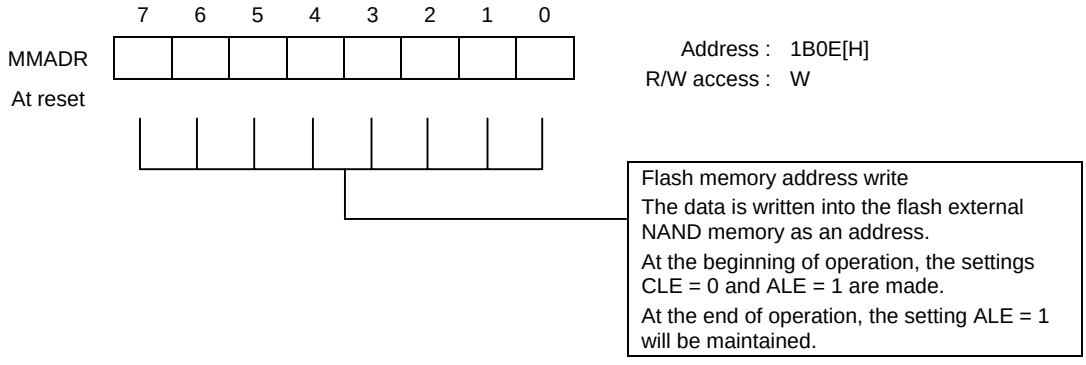


Figure 18-13 MMADR Configuration

(7) Media data register (MMDATA)

This register is used for writing data into the external NAND flash memory or for reading out data from the external NAND flash memory.

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

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), MMDATA is undefined.

Figure 18-14 shows the MMDATA configuration.

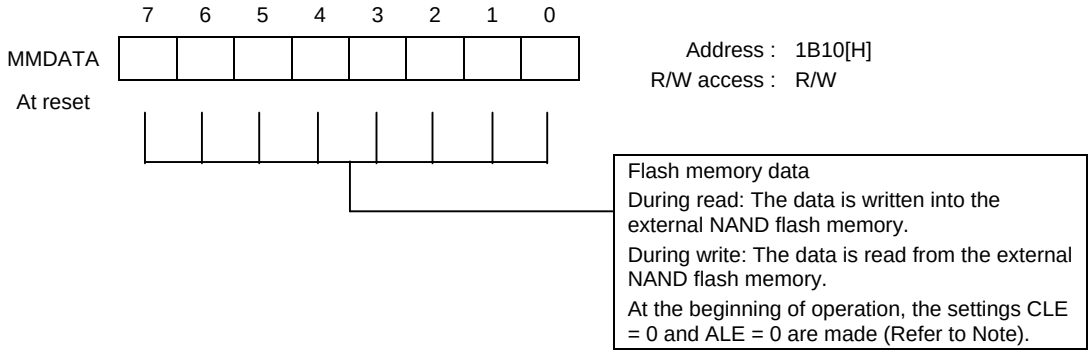


Figure 18-14 MMDATA Configuration

[Note]

Since the clearing of CLE and ALE cannot be done in time for the read signal during a read operation, it is necessary to clear both CLE and ALE to 0 before starting the operation.

(8) Media selector register (MMSEL)

This register is used for controlling the CLE and ALE signals of the NAND flash memory.

Bit 0 is control of the CLE signal of the flash memory and bit 1 is control of the ALE signal of the NAND flash memory.

MMSEL can be read from and written to by the program. Howeve, write operations are invalid for bits 2 to 7. If read, a value of "0" will always be obtained for bits 2 to 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), or at the time of starting the sequencer MMSEL becomes 00H.

Figure 18-15 shows the MMSEL configuration.

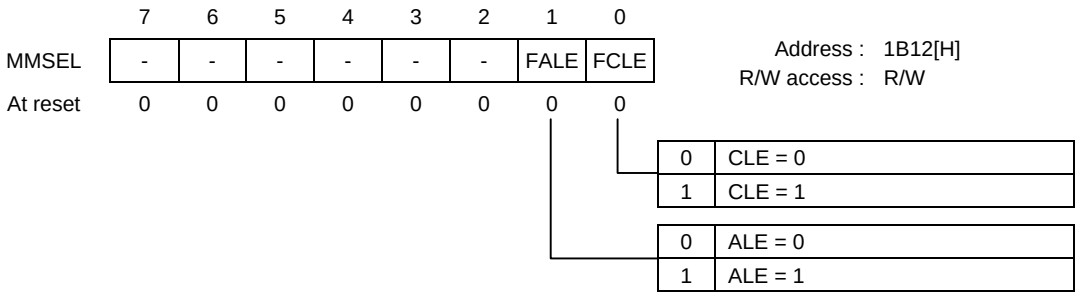


Figure 18-15 MMSEL Configuration

(9) ECC1 line parity register (ECC1LP)

This register stores the line parity of ECC1.

ECC1LP can only be read by the program. Write operations are invalid.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of ECC1LP becomes FFFFH.

Figure 18-16 shows the ECC1LP configuration.

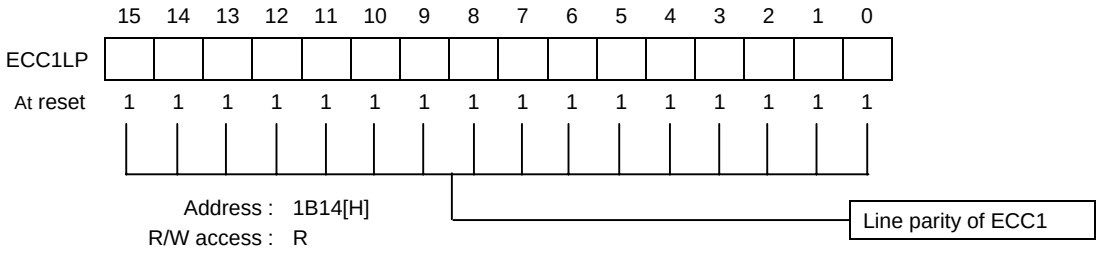


Figure 18-16 ECC1LP Configuration

(10) ECC2 line parity register (ECC2LP)

This register stores the line parity of ECC2.

ECC2LP can only be read by the program. Write operations are invalid.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of ECC2LP becomes FFFFH.

Figure 18-17 shows the ECC2LP configuration.

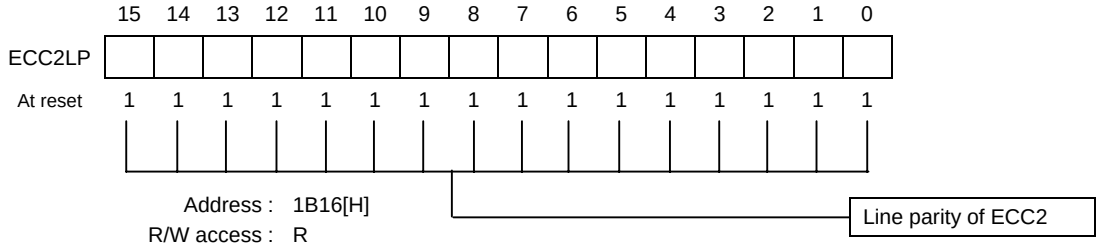


Figure 18-17 ECC2LP Configuration

(11) ECC1 column parity register (ECC1CP)

This register stores the column parity of ECC1.

ECC1CP can only be read by the program. Write operations are invalid. If read, a value of "0" will always be obtained for bits 6 to 15.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of ECC1CP becomes 003FH.

Figure 18-18 shows the ECC1CP configuration.

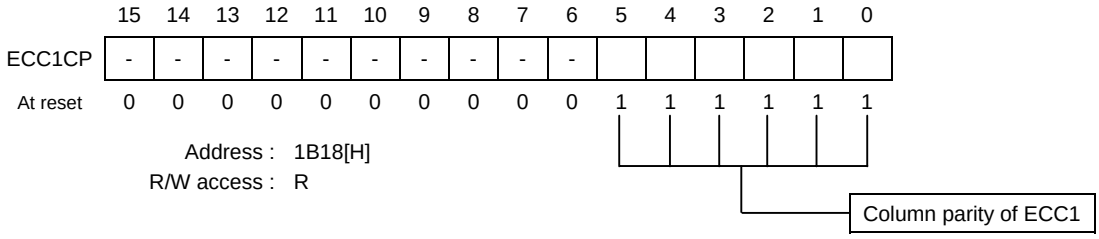


Figure 18-18 ECC1CP Configuration

(12) ECC2 column parity register (ECC2CP)

This register stores the column parity of ECC2.

ECC2CP can only be read by the program. Write operations are invalid. If read, a value of "0" will always be obtained for bits 6 to 15.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of ECC2CP becomes 003FH.

Figure 18-19 shows the ECC2CP configuration.

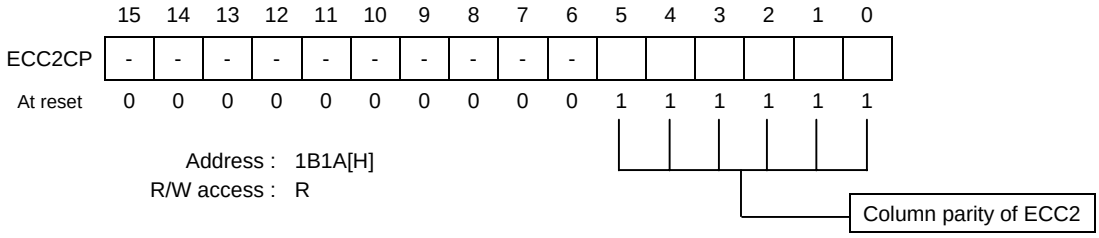


Figure 18-19 ECC2CP Configuration

(13) ECC1 error pointer register (ECC1ERR)

This register stores the pointer to the result of error detection of ECC1.

ECC1ERR can only be read by the program. Write operations are invalid. If read, a value of "0" will always be obtained for bits 11 to 15.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of ECC1ERR becomes 0000H.

Figure 18-20 shows the ECC1ERR configuration.

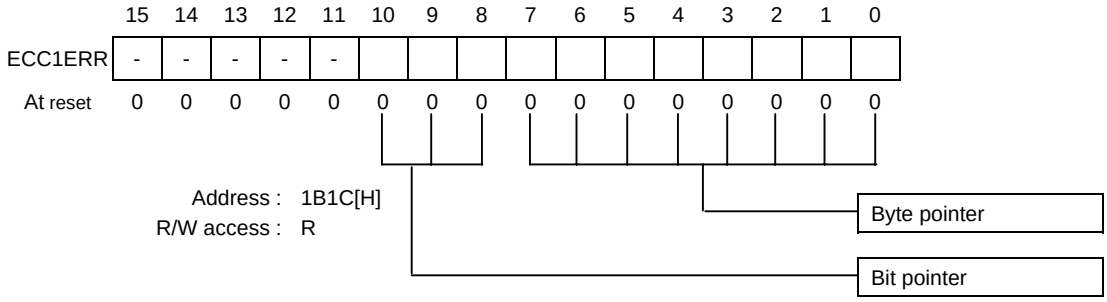


Figure 18-20 ECC1ERR Configuration

(14) ECC2 error pointer register (ECC2ERR)

This register stores the pointer to the result of error detection of ECC2.

ECC2ERR can only be read by the program. Write operations are invalid. If read, a value of "0" will always be obtained for bits 11 to 15.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of ECC2ERR becomes 0000H.

Figure 18-21 shows the ECC2ERR configuration.

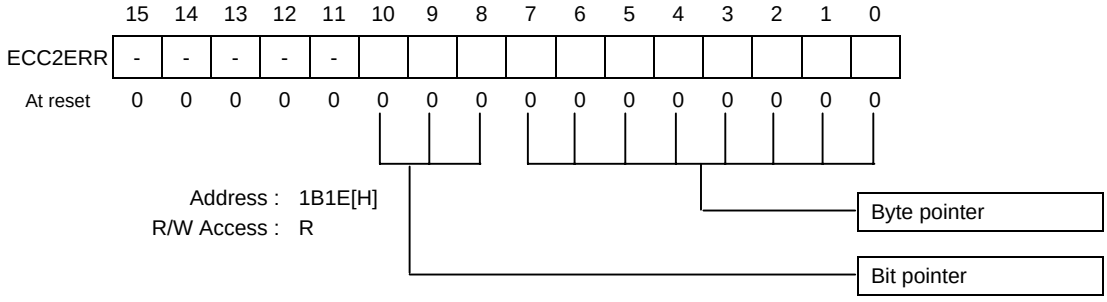


Figure 18-21 ECC2ERR Configuration

(15) Redundancy part reserved data 1 register (HREV1)

When the sequencer reads the redundancy part, the bytes 512 and 513 of the redundancy part are stored in this register. Also, when the sequencer writes the redundancy part, the bytes 512 and 513 of the redundancy part are set beforehand in this register.
This corresponds to the reserved data in the SmartMedia format. When the sequencer writes data in the SmartMedia format (when bits 4 and 5 (HEAD0, 1) of the MSCTRL register are both set to "0"), a fixed value of FFFFh is written into this register.
HREV1 can be read from and written to by the program.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), HREV1 becomes 0000H.
Figure 18-22 shows the HREV1 configuration.

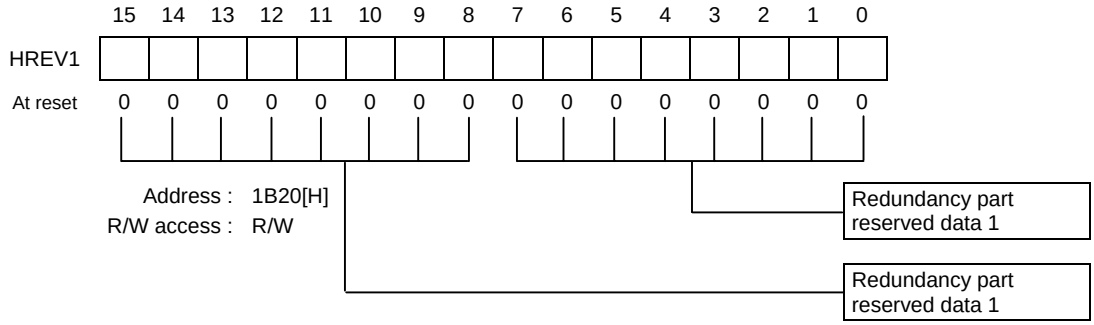


Figure 18-22 HREV1 Configuration

(16) Redundancy part reserved data 2 register (HREV2)

When the sequencer reads the redundancy part, the bytes 512 and 513 of the redundancy part are stored in this register. Also, when the sequencer writes the redundancy part, the bytes 512 and 513 of the redundancy part are set beforehand in this register.
This corresponds to the reserved data in the SmartMedia format. When the sequencer writes data in the SmartMedia format (when bits 4 and 5 (HEAD0, 1) of the MSCTRL register are both set to "0"), a fixed value of FFFFh is written into this register.
HREV2 can be read from and written to by the program.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), HREV2 becomes 0000H.
Figure 18-23 shows the HREV2 configuration.

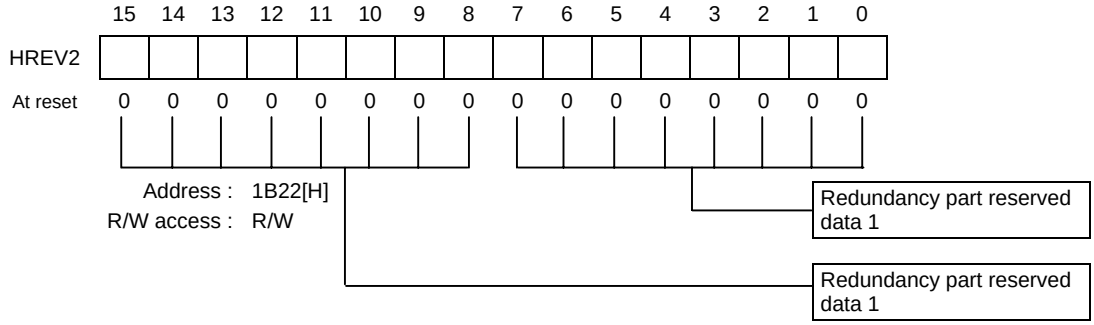


Figure 18-23 HREV2 Configuration

(17) Redundancy part data/block status register (HSTATS)

When the sequencer reads the redundancy part, the bytes 516 and 517 of the redundancy part are stored in this register. Also, when the sequencer writes the redundancy part, the bytes 516 and 517 of the redundancy part are set beforehand in this register.

This corresponds to the block status and the data status in the SmartMedia format.
HSTATS can be read from and written to by the program.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), HSTATS becomes 0000H.

Figure 18-24 shows the HSTATS configuration.

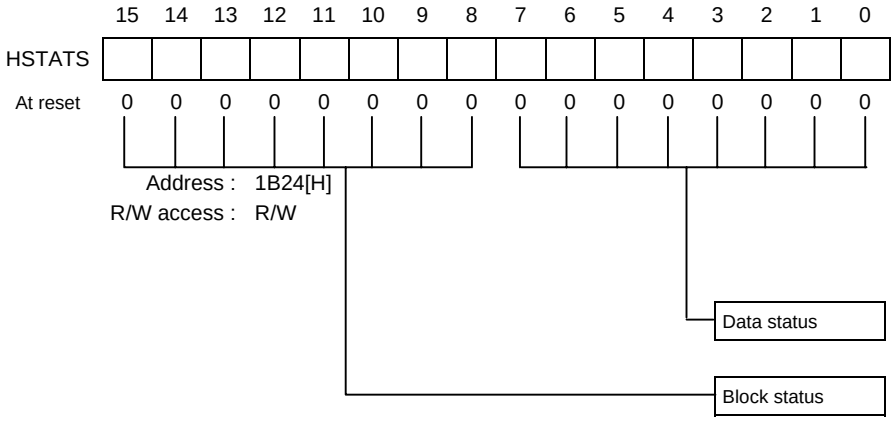


Figure 18-24 STATS Configuration

(18) Redundancy part block address 1 register (HBADR1)

When the sequencer reads the redundancy part, the bytes 518 and 519 of the redundancy part are stored in this register. Also, when the sequencer writes the redundancy part, the bytes 518 and 519 of the redundancy part are set beforehand in this register.

This corresponds to the block address 1 in the SmartMedia format. During a write operation, the parity generated from the block address (BA1ADH/BA1ADL) is written to the bit 8. If during a read operation the parity generated from the block address (BA1ADH/BA1ADL) is different from the value of the bit 8, the PARITY bit of the MSSTS register is set to 1.

When the sequencer writes data in the SmartMedia format (when bits 4 and 5 (HEAD0, 1) of the MSCTRL register are both set to "0"), afixed value of 00010b is written into the block address 1.

HBADR1 can be read from and written to by the program.
When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), HBADR1 becomes 0000H.

Figure 18-25 shows the HBADR1 configuration.

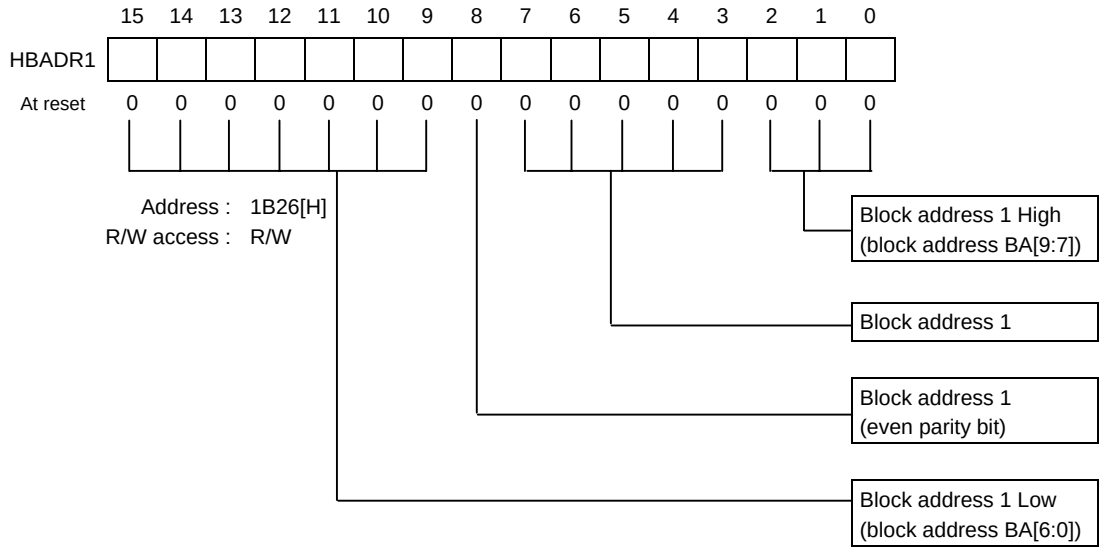


Figure 18-25 HBADR1 Configuration

(19) Redundancy part ECC2-High register (HECC2H)

When the sequencer reads the redundancy part, the bytes 520 and 521 of the redundancy part are stored in this register. Also, when the sequencer writes the redundancy part, the bytes 520 and 521 of the redundancy part are set beforehand in this register.

This corresponds to the high-order two bytes of the ECC code of Data Area 2 in the SmartMedia format.

HECC2H can be read from and written to by the program.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), HECC2H becomes 0000H.

Figure 18-26 shows the HECC2H configuration.

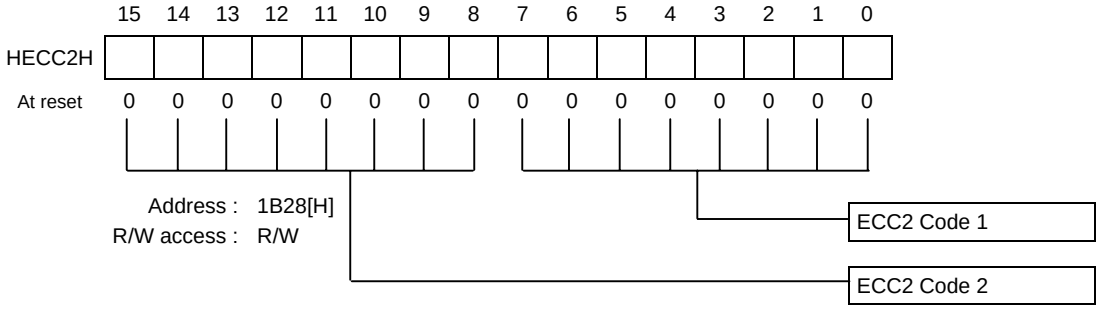


Figure 18-26 HECC2H Configuration

(20) Redundancy part ECC2-Low/block address 2 register (HECC2LA)

When the sequencer reads the redundancy part, the bytes 522 and 523 of the redundancy part are stored in this register. Also, when the sequencer writes the redundancy part, the bytes 522 and 523 of the redundancy part are set beforehand in this register.

This corresponds to the lower one byte of the ECC code of Data Area 2 and the high-order one byte of the block address 2 in the SmartMedia format. When writing the redundancy part in the SmartMedia format, the contents of this register will be ignored (block address 2 will have the same value as that of block address 1).

HECC2LA can be read from and written to by the program.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), HECC2LA becomes 0000H.

Figure 18-27 shows the HECC2LA configuration.

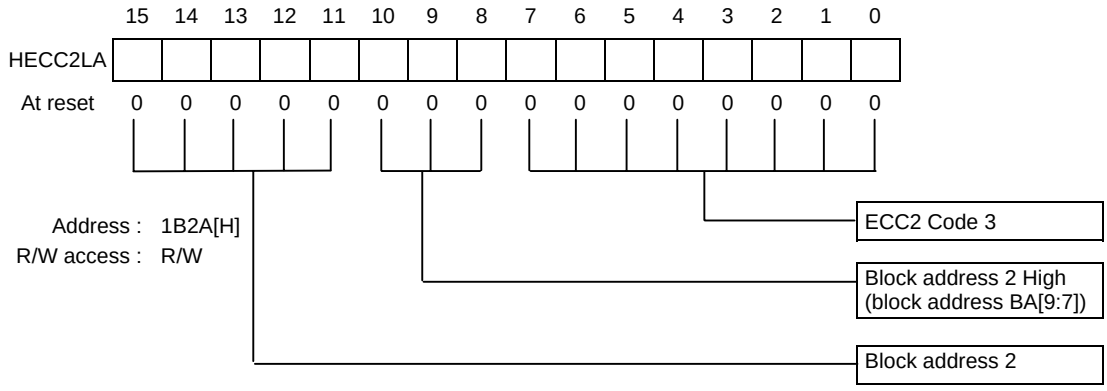


Figure 18-27 HECC2LA Configuration

(21) Redundancy part ECC1-High/block address 2 register (HECC1HA)

When the sequencer reads the redundancy part, the bytes 524 and 525 of the redundancy part are stored in this register. Also, when the sequencer writes the redundancy part, the bytes 524 and 525 of the redundancy part are set beforehand in this register.

This corresponds to the low-order one byte of the block address 1 and the uppermost one byte of the ECC code of Data Area 1 in the SmartMedia format. When writing the redundancy part in the SmartMedia format, the contents of this register will be ignored (block address 2 will have the same value as that of block address 1).

HECC1HA can be read from and written to by the program.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), HECC1HA becomes 0000H.

Figure 18-28 shows the HECC1HA configuration.

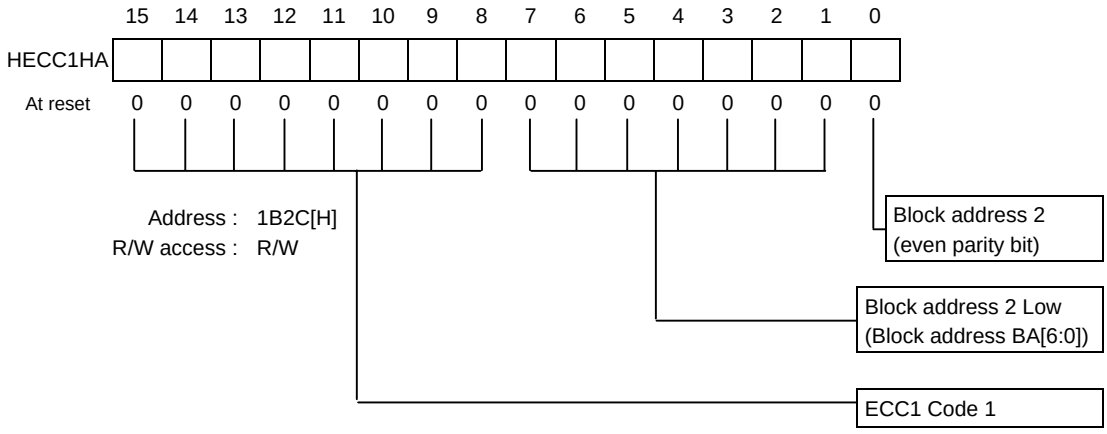


Figure 18-28 HECC1HA Configuration

(22) Redundancy part ECC1-Low register (HECC1L)

When the sequencer reads the redundancy part, the bytes 526 and 527 of the redundancy part are stored in this register. Also, when the sequencer writes the redundancy part, the bytes 526 and 527 of the redundancy part are set beforehand in this register.

This corresponds to the low-order two bytes of the ECC code of Data Area 1 in the SmartMedia format. When writing the redundancy part in the SmartMedia format, the contents of this register will be ignored.

HECC1L can be read from and written to by the program.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), HECC1L becomes 0000H.

Figure 18-29 shows the HECC1L configuration.

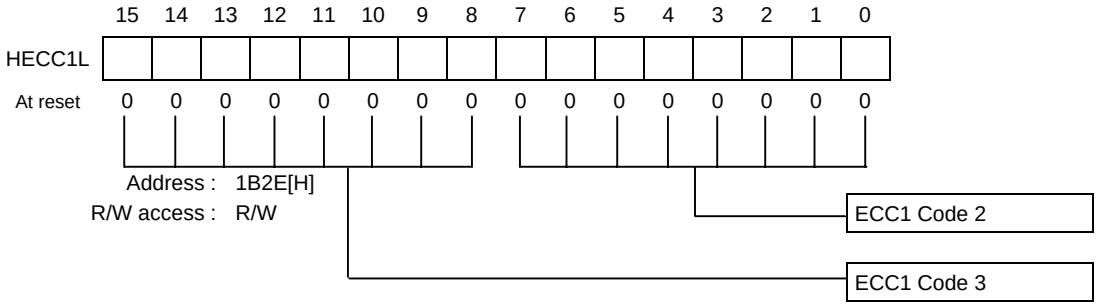


Figure 18-29 HECC1L Configuration

18.3 Wait Function

It is possible to specify the method of outputting the flash memory control signal during data transfer using the sequencer, depending on the operating frequency or the specifications of the external NAND flash memory to be connected. This specification of the outputting method is done using bits 0, 1, 2 and 3 (WAITR1, 2 and WAITW1, 2) of the media sequencer wait register (MSWAIT).

18.3.1 Wait Functions during Write Operations

(1) No write wait (MSWAIT register: WAITW1 = 0b, WAITW2 = 0b)
Figure 18-30 shows the data write timing in the case of no write wait.

In the case of no write wait, the write operation is done in one cycle of two clocks.

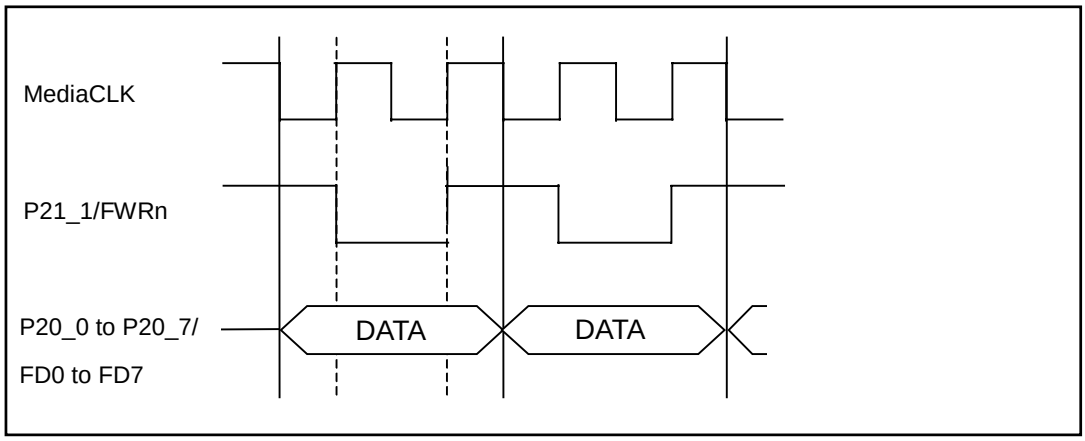


Figure 18-30 Data Write Timing in the Case of No Write Wait

(2) Write wait 1 (MSWAIT register: WAITW1 = 1b, WAITW2 = 0b)
Figure 18-31 shows the data write timing in the case of write wait 1.

In the case of write wait 1, the write operation is done in one cycle of three clocks.

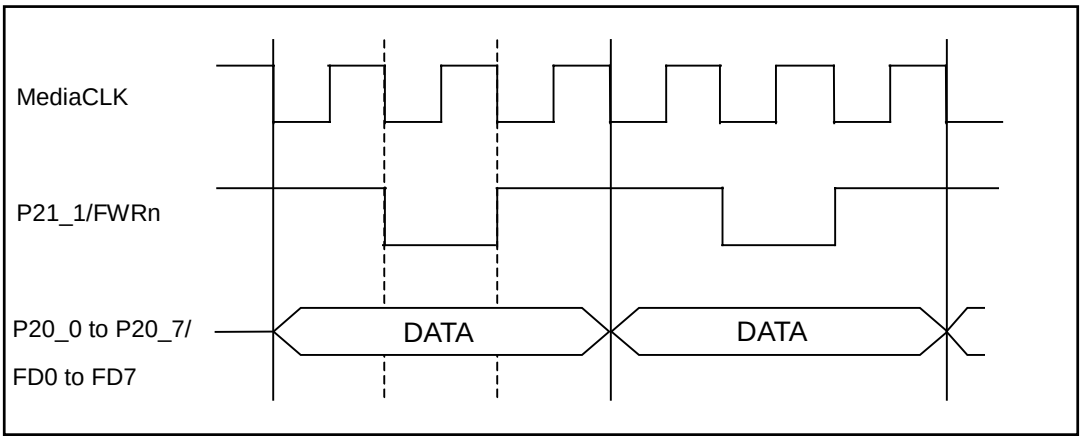


Figure 18-31 Data Write Timing in the Case of Write Wait 1

(3) Write wait 2 (MSWAIT register: WAITW1 = Xb, WAITW2 = 1b)

Figure 18-32 shows the data write timing in the case of write wait 2.

In the case of write wait 2, the write operation is done in one cycle of three clocks.

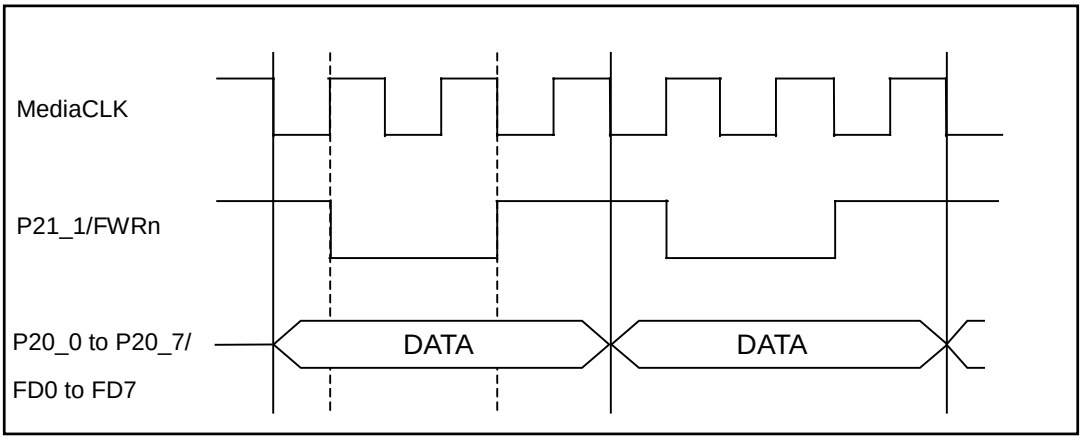


Figure 18-32 Data Write Timing in the Case of Write Wait 2

18.3.2 Wait Functions during Read Operations

(1) No read wait (MSWAIT register: WAITR1 = 0b, WAITR2 = 0b)
Figure 18-33 shows the data read timing in the case of no read wait.

In the case of no read wait, the read operation is done in one cycle of two clocks. The sequencer reads data at the falling edge of the clock.

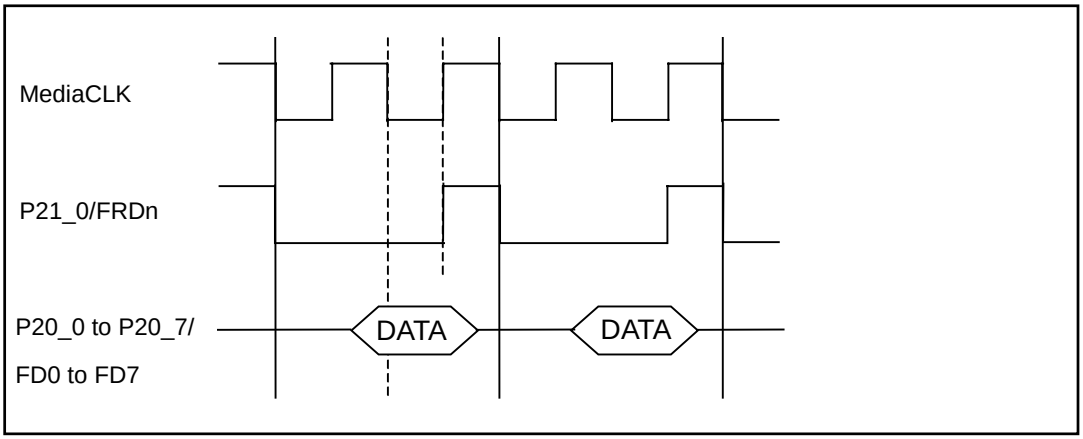


Figure 18-33 Data Read Timing in the Case of No Read Wait

(2) Read wait 1 (MSWAIT register: WAITR1 = 1b, WAITR2 = 0b)
Figure 18-34 shows the data read timing in the case of read wait 1.

In the case of read wait 1, the read operation is done in one cycle of three clocks. The sequencer reads data at the falling edge of the clock.

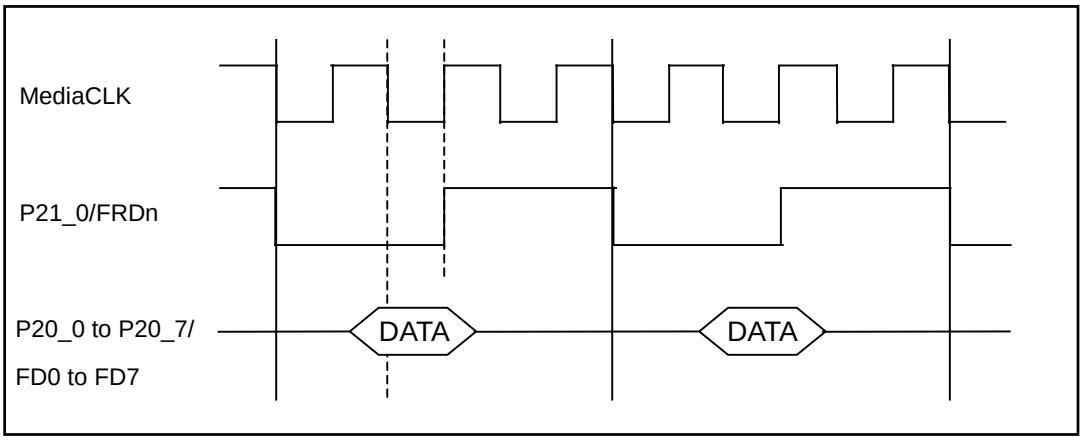


Figure 18-34 Data Read Timing in the Case of Read Wait 1

(3) Read wait 2 (MSWAIT register: WAITR1 = Xb, WAITR2 = 1b)

Figure 18-35 shows the data read timing in the case of read wait 2.

In the case of read wait 2, the read operation is done in one cycle of three clocks. The sequencer reads data at the falling edge of the clock.

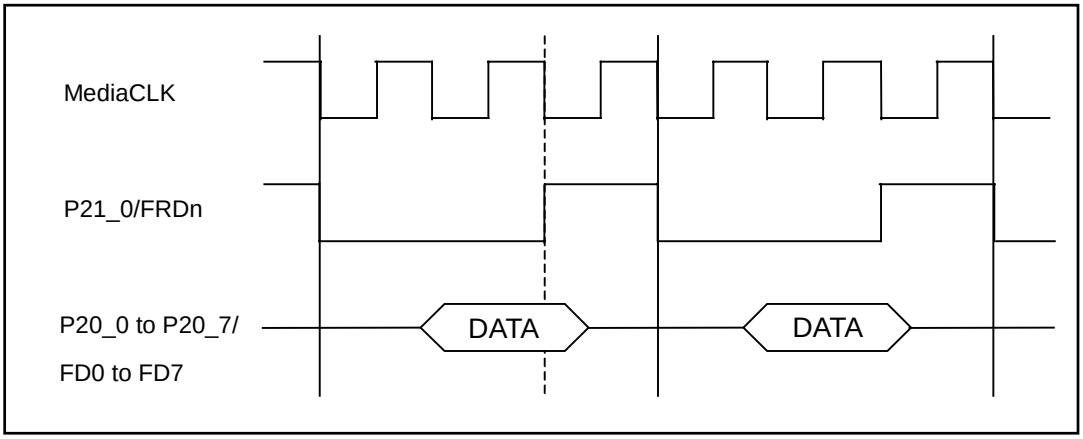


Figure 18-35 Data Read Timing in the Case of Read Wait 2

Chapter 19

Internal DMA Control Function

19. Internal DMA Control Function

19.1 Overview

A memory for data transfer and DMA for data transfer are used for the transfer of data between the USB control circuit and the media control circuit.

The memory space for data transfer (Buffer RAM) is present in the expanded RAM area (address 1200H to 19FFH) and is divided into the four banks 0 to 3 of 512 bytes each.

Figure19-1 shows a memory map of the data transfer memory space.

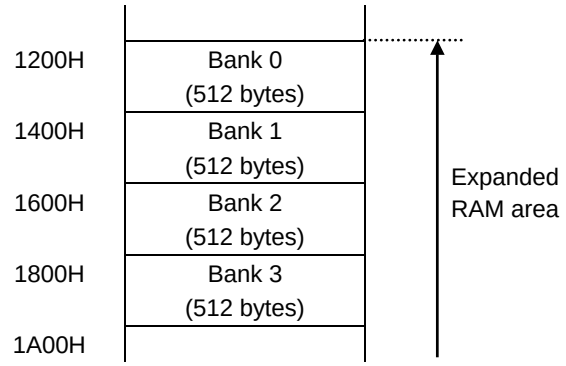


Figure 19-1 Memory Map of Data Transfer Memory Space

The following four types of data transfer are available.

(1) From the USB section to the memory for data transfer
(2) From the memory for data transfer to the USB section
(3) From the media control section to the memory for data transfer
(4) From the memory for data transfer to the media control section

19.2 Registers for the Internal DMA Control Function

Table 19-1 lists a summary of the SFRs for the internal DMA control function.

Table 19-1 Summary of SFRs for the Internal DMA Control Function

Address [H]	Name	Symbol (BYTE)	Symbol (WORD)	R/W	8/16 operation	Initial value [H]	Reference page
1A80	Channel 0 current address register	—	CH0ADDRESS	R/W	16	0000	19-3
1A81*							
1A82	Channel 1 current address register	—	CH1ADDRESS	R/W	16	0000	19-3
1A83*							
1A84	Channel 0 current word count register	—	CH0WDCNT	R/W	16	0000	19-4
1A85*							
1A86	Channel 1 current word count register	—	CH1WDCNT	R/W	16	0000	19-4
1A87*							
1A88*	Mode register	MODE	—	R/W	8	00	19-5
1A8A*	Channel 0 mask register	CH0MSK	—	R/W	8	01	19-6
1A8C*	Channel 1 mask register	CH1MSK	—	R/W	8	01	19-6
1A90*	Interrupt status register	INTSTAT	—	R/W	8	00	19-7
1A92*	Interrupt enable register	INTENBL	—	R/W	8	00	19-8
1A94*	DREQ monitor register	DREQMON	—	R	8	Un-defined	19-9
1AA0*	Option register	OPTION	—	R/W	8	00	19-10
1AA2*	Channel 0 packet size register	CH0PKTSZ	—	R/W	8	00	19-11
1AA4*	Channel 1 packet size register	CH1PKTSZ	—	R/W	8	00	19-11
1AA6*	Channel 0 maximum packet size register	CH0MXPKTSZ	—	R/W	8	00	19-12
1AA8*	Channel 1 maximum packet size register	CH1MXPKTSZ	—	R/W	8	00	19-12
1AB0*	Channel 0 first byte detection count register	CH0RXCNT	—	R/W	8	00	19-13
1AB2*	Channel 1 first byte detection count register	CH1RXCNT	—	R/W	8	00	19-13
1B00*	USB bank register	UBANK	—	R/W	8	00	19-14
1B02*	Media bank register	MBANK	—	R/W	8	00	19-15

[Notes]

1. Addresses are not consecutive in some places.
2. An asterisk in the address column indicates a missing bit.
3. For details, refer to Chapter 22 "Special Function Registers (SFRs)".

19.2.1 Description of the Registers for the Internal DMA Control Function

(1) Current address registers (CH0ADDRESS/CH1ADDRESS)

Each channel has a 9-bit current address register which stores the transfer address during DMA transfer and is incremented by 1 at every DMA cycle.

CH0ADDRESS/CH1ADDRESS can be read from and written to by the program. However, write operations are invalid for bits 9 to 15. If read, a value of "0" will always be obtained for bits 9 to 15.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), CH0ADDRESS/CH1ADDRESS becomes 0000H.

Figure 19-2 and 19-3 show the CH0ADDRESS and CH1ADDRESS configuration.

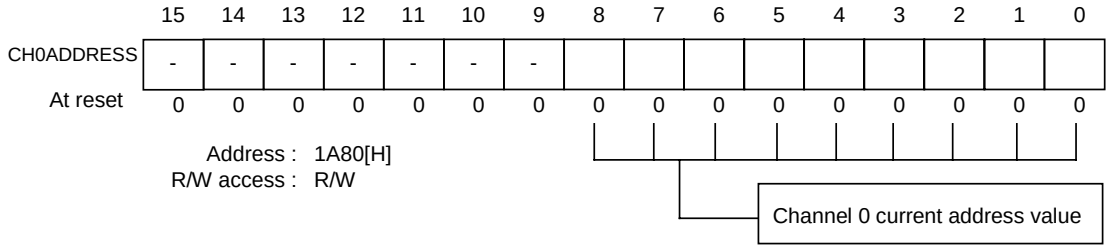


Figure 19-2 CH0ADDRESS Configuration

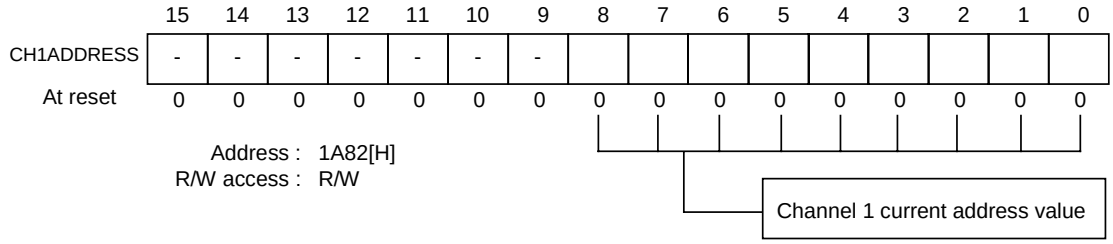


Figure 19-3 CH1ADDRESS Configuration

(2) Current word count registers (CH0WDCNT/CH1WDCNT)

Each channel has a 10-bit current word count register which stores the current number of transfers completed during DMA transfer and is decremented by 1 at every DMA cycle.

A terminal count condition is generated when the word count value changes from 0001H to 0000H. At this time, the TC (Terminal Count) interrupt is asserted, and at the same time, that channel will be masked automatically.

CH0WDCNT/CH1WDCNT can be read from and written to by the program. However, write operations are invalid for bits 10 to 15. If read, a value of "0" will always be obtained for bits 10 to 15.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), CH0WDCNT/CH1WDCNT becomes 0000H.

Figure 19-4 and 19-5 show the CH0WDCNT and CH1WDCNT configuration.

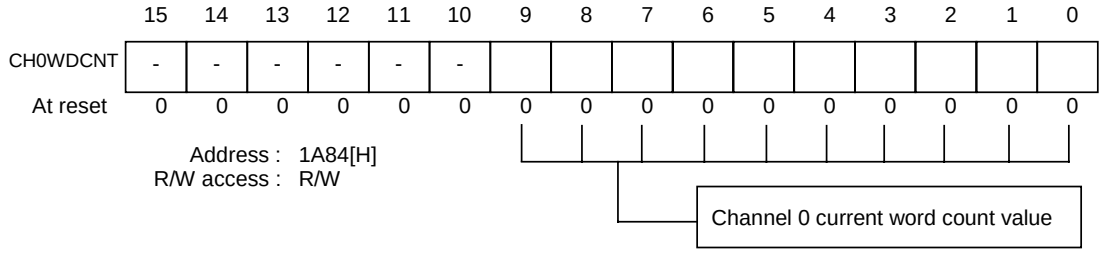


Figure 19-4 CH0WDCNT Configuration

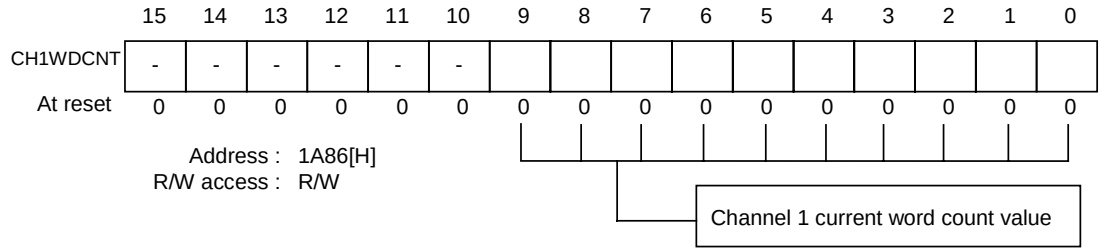


Figure 19-5 CH1WDCNT Configuration

(3) Mode register (MODE)

Each channel has a mode register with the following two bits.

Bits 0 and 1 specify the transfer direction (USB → the buffer RAM or the buffer RAM → USB), and bits 2 and 3 specify the highest-order address value of the buffer RAM to which DMA transfer is provided. The bits 2 and 3 are available when both bits 0 and 1 of the USB bank register (UBANK) are set to 1s.

MODE can be read from and written to by the program. However, write operations are invalid for bits 4 to 7. If read, a value of "0" will always be obtained for bits 4 to 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), MODE becomes 00H.

Figure 19-6 shows the MODE configuration.

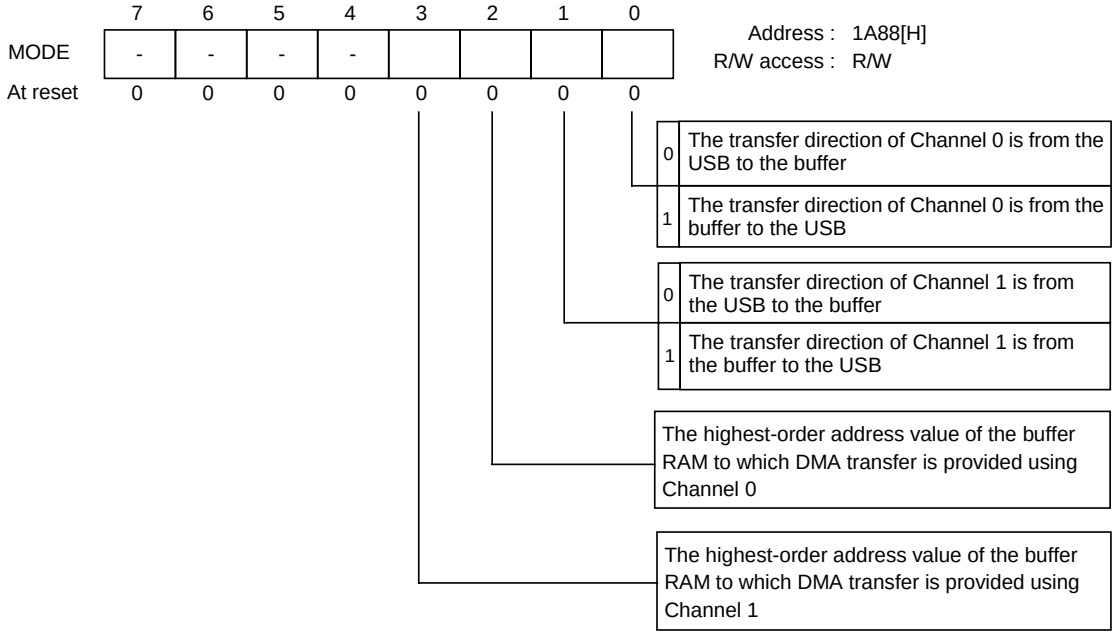


Figure 19-6 MODE Configuration

(4) Mask registers (CH0MSK/CH1MSK)

Each channel has a 1-bit mask bit, which controls the enabling and disabling of DMA transfer. Each mask bit is set automatically either when that channel reaches the terminal count or when the transfer of a short packet is done with the option register as described later set.

CH0MSK/CH1MSK can be read from and written to by the program. However, write operations are invalid for bits 1 to 7. If read, a value of "0" will always be obtained for bits 1 to 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), CH0MSK/CH1MSK becomes 01H.

Figure 19-7 and 19-8 show the CH0MSK and CH1MSK configuration.

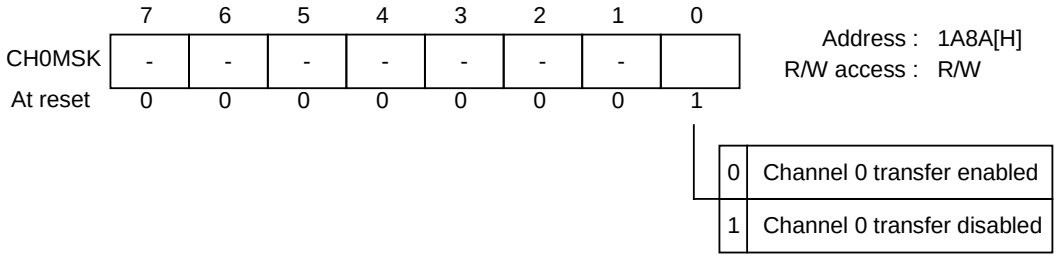


Figure 19-7 CH0MSK Configuration

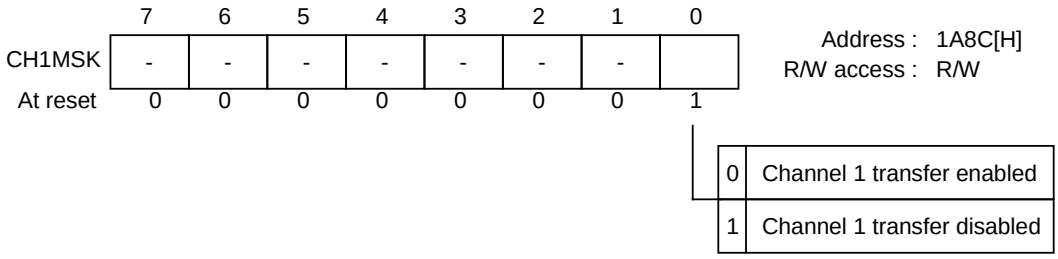


Figure 19-8 CH1MSK Configuration

(5) Interrupt status register (INTSTAT)

This register is used for the CPU to read the status of the DMA controller.

The bits 0 and 1 indicate whether the corresponding channel has reached the TC (terminal count) condition or not, and are set when the terminal count condition has been reached (When there is a transition of the word count value from 0001H to 0000H).

The bits 2 and 3 indicate that a short packet has been received in the EP of the corresponding channel. These bits are set when all the data of the short packet has been transferred to the buffer RAM.

The bit 7 is set to "1" when the mask registers of both channels have been set to "1" and also the DMA transfer has been stopped. In this condition, the access from the CPU to the registers of the USB controller becomes enabled.

In the case of bits 0 to 3, the corresponding interrupt request can be cleared by writing 1s to these bits. The interrupt request cannot be cleared by writing 0s to these bits.

In order to clear the interrupt request of bit 7, bit 7 of the interrupt enable register (INTENBL) must be reset to a "0".

INTSTAT 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 "0" will always be obtained for bits 4 to 6.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), INTSTAT becomes 00H.

Figure 19-9 shows the INTSTAT configuration.

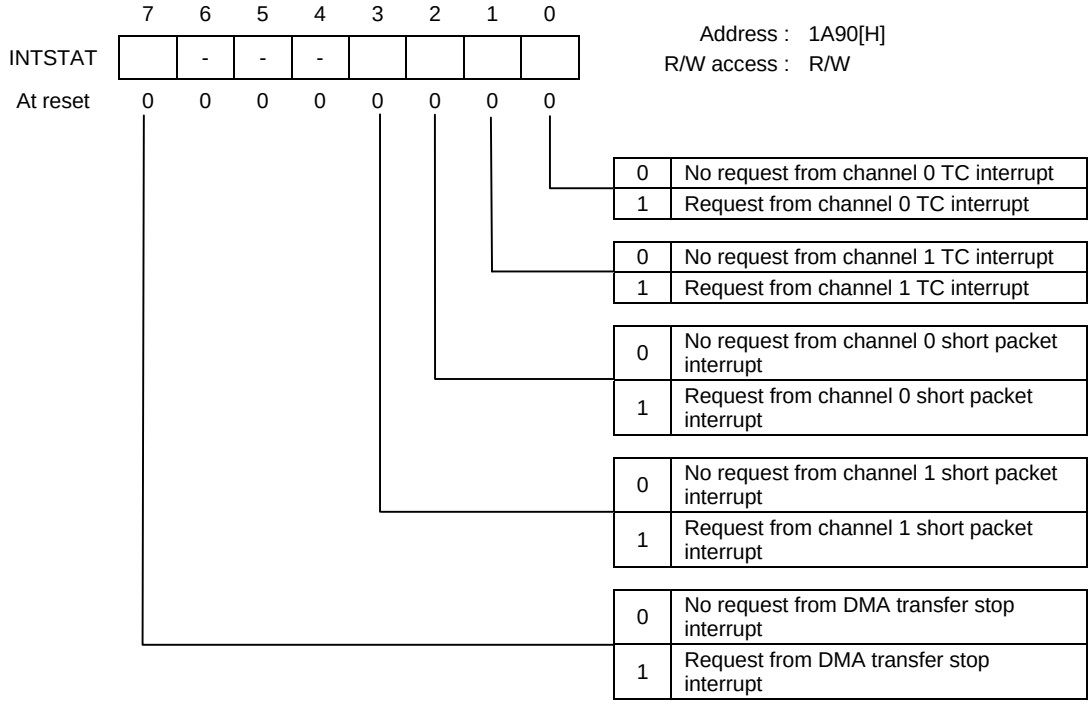


Figure 19-9 INTSTAT Configuration

(6) Interrupt enable register (INTENBL)

INTENBL 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 "0" will always be obtained for bits 4 to 6.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), INTENBL becomes 00H.

Figure 19-10 shows the INTENBL configuration.

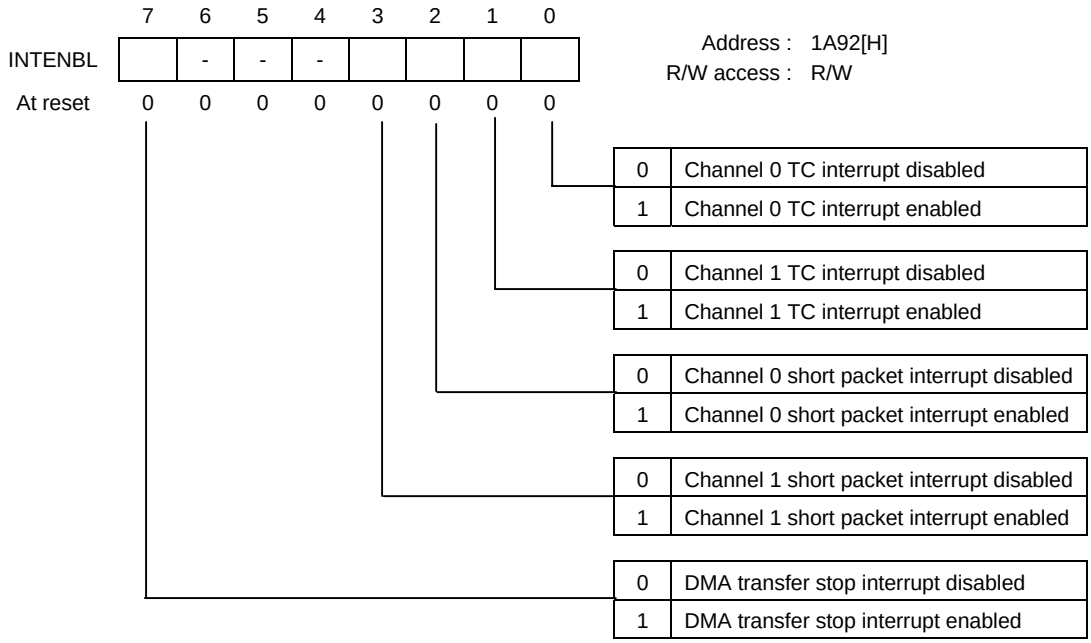


Figure 19-10 INTENBL Configuration

[Note]

To enable interrupts during DMA transfer, set to "0" (disabled) the corresponding bit of the interrupt enable register 1 (INTENBL1) of the EP to be the object of the DMA transfer.

(7) DREQ monitor register (DREQMON)

This register indicates the DREQ input signal condition for each channel.

The DREQ 0 (1) input signal resets bit 0 (1) to 0 when the data transfer request occurs for the DMA controller from the USB controller.

DREQMON can only be read by the program. Write operations are invalid. If read, a value of "0" will always be obtained for bits 2 to 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), the value of DREQMON is undefined.

Figure 19-11 shows the DREQMON configuration.

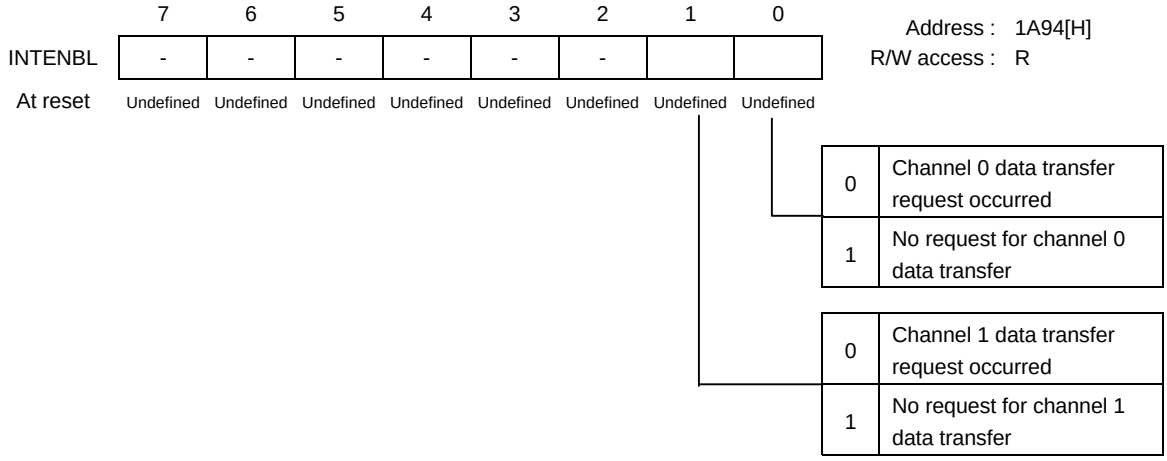


Figure 19-11 DREQMON Configuration

(8) Option register (OPTION)

The short packet reception detection function of this option register can be used in combination with a specific setting of the DMA control register within the USB controller (the mode in which the "Byte count (packet size)" is inserted at the beginning of a DMA transfer).

When the short packet reception detection function has been enabled, the first byte of each received packet during the DMA transfer will not be transferred to the RAM, but will be stored in the packet size register described below. A short packet is judged to have been received when this value is less than the value that has previously been set in the maximum packet size register.

When all the data of a short packet has been transferred to the buffer RAM, a short packet interrupt is generated and that channel will be masked automatically.

OPTION can be read from and written to by the program. However, write operations are invalid for bits 1 to 7. If read, a value of "0" will always be obtained for bits 1 to 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), OPTION becomes 00H.

Figure 19-12 shows the OPTION configuration.

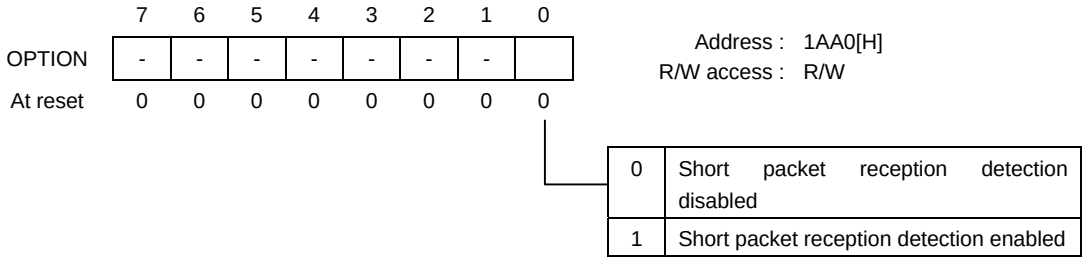


Figure 19-12 OPTION Configuration

(9) Packet size registers (CH0PKTSZ/CH1PKTSZ)

The packet size of the packet that has been received in the EP for each channel is automatically stored into these registers.

CH0PKTSZ/CH1PKTSZ can be read from and written to by the program. However, write operations are invalid for bit 7. If read, a value of "0" will always be obtained for bit 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), CH0PKTSZ/CH1PKTSZ becomes 00H.

Figure 19-13 and 19-14 show the CH0PKTSZ and CH1PKTSZ configuration.

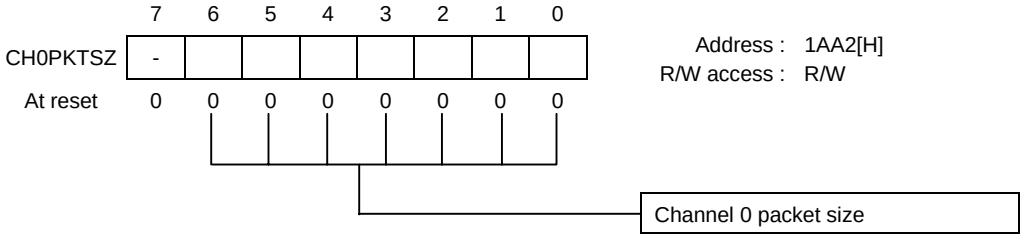


Figure 19-13 CH0PKTSZ Configuration

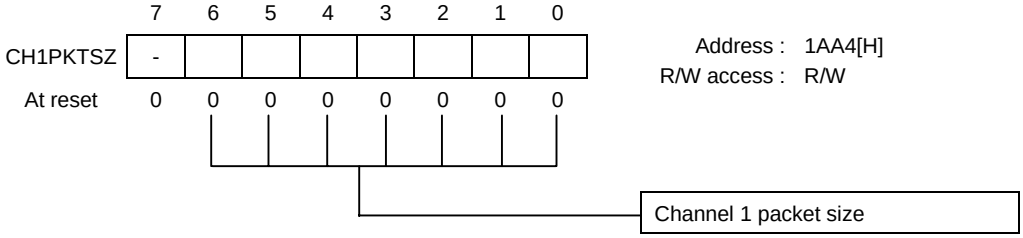


Figure 19-14 CH1PKTSZ Configuration

(10) Maximum packet size registers (CH0MXPKTSZ/CH1MXPKTSZ)

The normal setting is 64 bytes (the largest size of a packet for bulk transfer).

CH0MXPKTSZ/CH1MXPKTSZ can be read from and written to by the program. However, write operations are invalid for bit 7. If read, a value of "0" will always be obtained for bit 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), CH0MXPKTSZ/CH1MXPKTSZ becomes 00H.

Figure 19-15 and 19-16 show the CH0MXPKTSZ and CH1MXPKTSZ configuration.

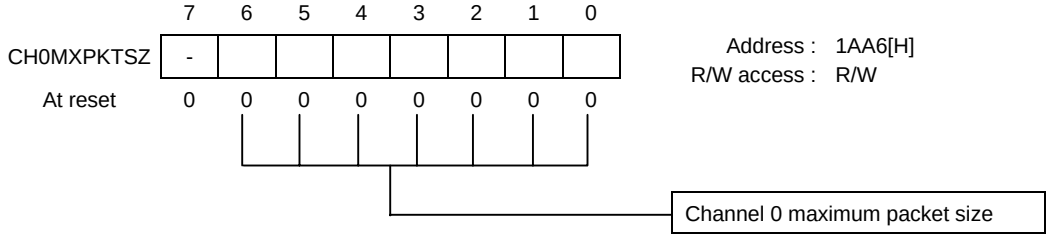


Figure 19-15 CH0MXPKTSZ Configuration

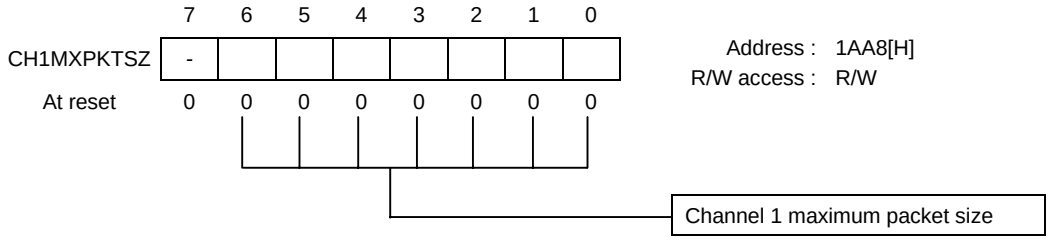


Figure 19-16 CH1MXPKTSZ Configuration

(11) First byte detection count registers (CH0RXCNT/CH1RXCNT)

This counter is incremented by 1 during each DMA cycle (including when the packet size is read from the USB controller). When the value of this counter is 0, the byte transferred next is taken as the packet size (that is, the first byte of the packet). When the value in this register is the same as the value in the maximum packet size register described above, the next byte is taken as the last byte in the packet, and when the transfer of that byte has been completed, the value in this counter will be reset to 0. When a write operation is made to this register, the count value will be reset irrespective of the value of the input data.

CH0RXCNT/CH1RXCNT can be read from and written to by the program. However, write operations are invalid for bit 7. If read, a value of "0" will always be obtained for bit 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), CH0RXCNT/CH1RXCNT becomes 00H.

Figure 19-17 and 19-18 show the CH0RXCNT and CH1RXCNT configuration.

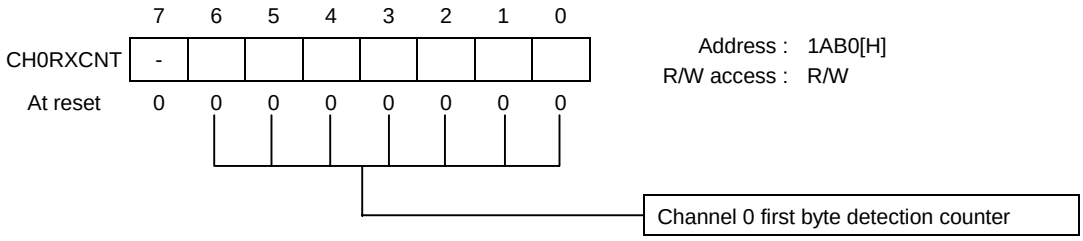


Figure 19-17 CH0RXCNT Configuration

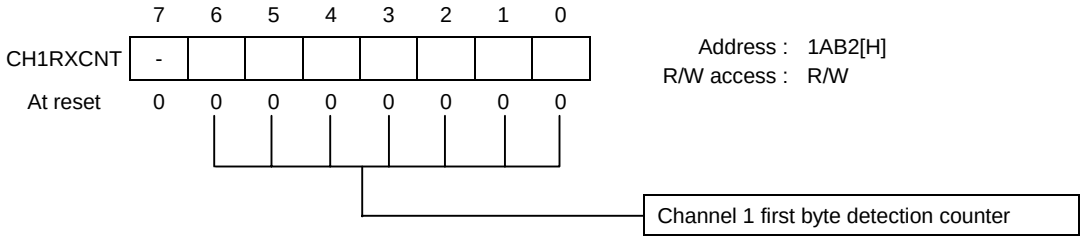


Figure 19-18 CH1RXCNT Configuration

(12) USB bank register (UBANK)

The bank to be used during the data transfer between the USB and the RAM is specified by the USB bank register.

It is possible to specify "Bank 0", or "Bank 1", or "Bank 0 and Bank 1" using the USB bank register.

UBANK can be read from and written to by the program. However, write operations are invalid for bits 2 to 7. If read, a value of "0" will always be obtained for bits 2 to 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), UBANK becomes 00H.

Figure 19-19 shows the UBANK configuration.

[Note]

When UBANK > 11, Bank 1 are assigned for the data transfer between the USB and the RAM; however, it is bits 2 and 3 of the mode register (MODE) that specify which bank is to be used for the actual data transfer.

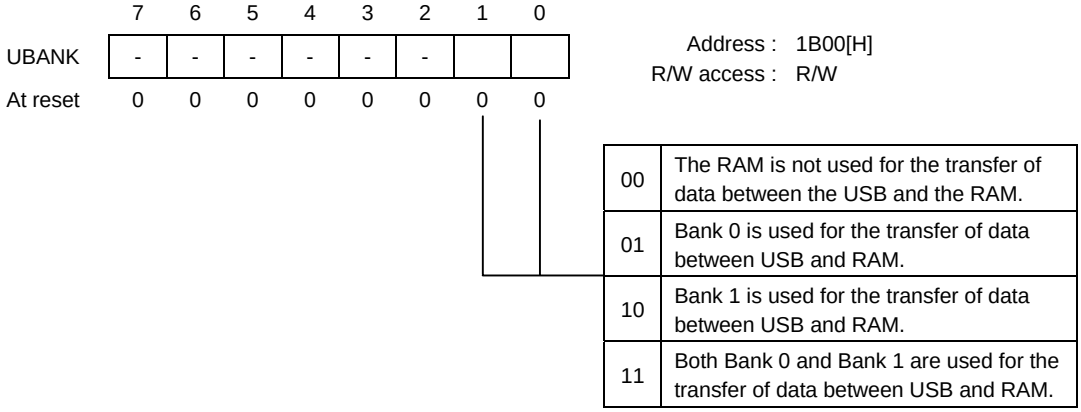


Figure 19-19 UBANK Configuration

(13) Media bank register (MBANK)

The bank to be used during the data transfer between the media and the RAM is specified by the media bank register. It is possible to specify from Bank 0 to Bank 3 in the media bank register.

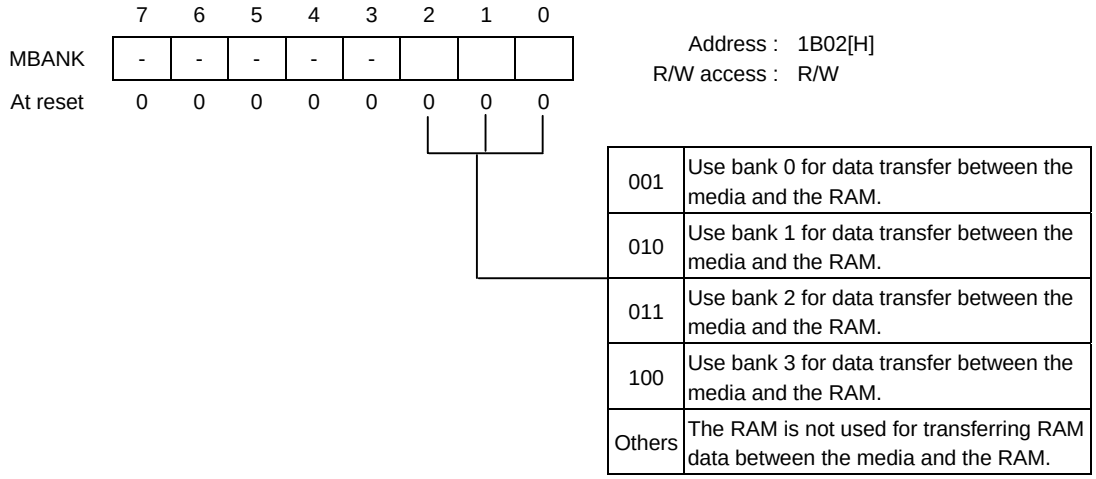
MBANK can be read from and written to by the program. However, write operations are invalid for bits 3 to 7. If read, a value of "0" will always be obtained for bits 3 to 7.

When reset (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, opcode trap), MBANK becomes 00H.

Figure 19-20 shows the MBANK configuration.

[Note]

The banks specified by the USB bank register and the media bank register cannot be accessed from the CPU. Also, when the same bank is specified in both the USB bank register and the media bank register, the specification in the USB bank register will be given priority.



001	Use bank 0 for data transfer between the media and the RAM.
010	Use bank 1 for data transfer between the media and the RAM.
011	Use bank 2 for data transfer between the media and the RAM.
100	Use bank 3 for data transfer between the media and the RAM.
Others	The RAM is not used for transferring RAM data between the media and the RAM.

Figure 19-20 MBANK Configuration

Chapter 20

Flash Memory

20. Flash Memory

20.1 Overview

The ML66Q525A, a member of the ML66525 family devices, is equipped with an electrically programmable non-volatile memory (128 KB flash memory) as the internal program memory. With three types of flash memory programming modes, the ML66Q525A can be programmed even after being installed in a system.

20.2 Features

- Power supply voltage
  ML66Q525A (2.4 to 3.6 V operation) … Can be programmed with a single 2.4 to 3.6 V power supply.

- Programming modes
  Flash memory has the following three programming modes.
  - Parallel mode … Can be programmed with a Minato Electronics' PROM writer.
  - Serial mode … Can be programmed with a flash memory writer.
  - User mode … Can be programmed by program execution.

- 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. 50 ms (at 2.4 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.

  The security function cannot be set in the parallel mode or in the user mode.

20.3 Programming Modes

Flash memory of the ML66Q525 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 Minato Electronics' 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. MTP66525) to Minato Electronics Co.-make 1893 (ver. 1.20d or later) or 1931 (ver. 1.20d or later) writer and then perform the programming.

(2) Serial mode

Programming in this mode is performed with a general-purpose 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.

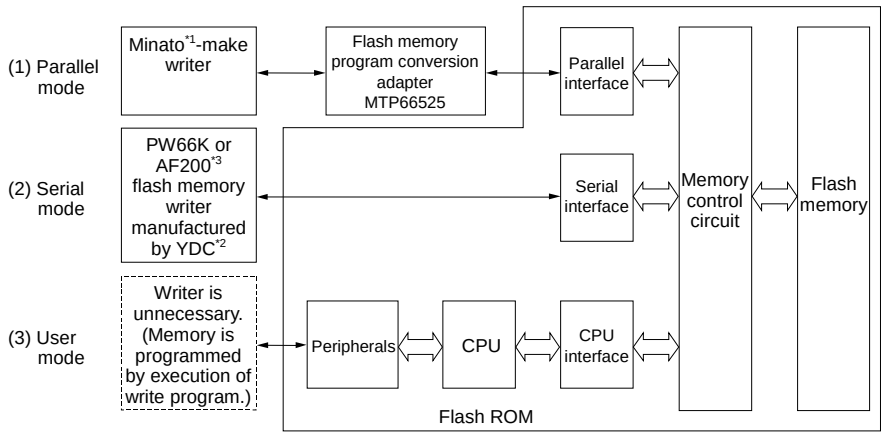
In the serial mode, connect a flash memory writer (model no. PW66K) or a flash microcontroller programmer (model no. AF200) manufactured by YDC Co., Ltd. to the two microcontroller pins (P7_6, P7_7), the FLAMOD pin, and V_{DD} and GND pins, and then perform the programming. Programming is performed while the microcontroller is in the reset or stop mode.

(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 20-1 shows a block diagram of the flash memory programming modes.



*1: Minato Electronics Co., Ltd.
*2: YDC is Yokogawa Digital Computer
*3: AF200 is a trademark of YDC.

Figure 20-1 Block Diagram of Programming Modes

20.4 Parallel Mode

20.4.1 Overview of the Parallel Mode

Programming in the parallel mode is performed with a PROM. The writing and reading of programs is performed by connecting Oki Electric's flash memory program conversion adapter (MTP66525) to a Minato Electronics-make writer 1893 (ver. 1.20d or later) or 1931 (ver. 1.20d or later). Figure 20-2 shows a connection diagram. Since an auto-erase function is provided, flash memory does not have to be erased prior to programming.

[Note]

The security function cannot be set in the parallel mode.

20.4.2 PROM Writer Setting

For setting the PROM writer, Refer to the PROM writer manual.

For writing, be sure to use the device code ML66Q525, which has been registered exclusively for the ML66Q525A. Do not use the Silicon Signature mode, which automatically assigns a device code when an LSI is inserted. Refer to the PROM writer manual for details.

20.4.3 Flash Memory Programming Conversion Adapter

Use Oki Electric's flash memory program conversion adapter (MTP66525).

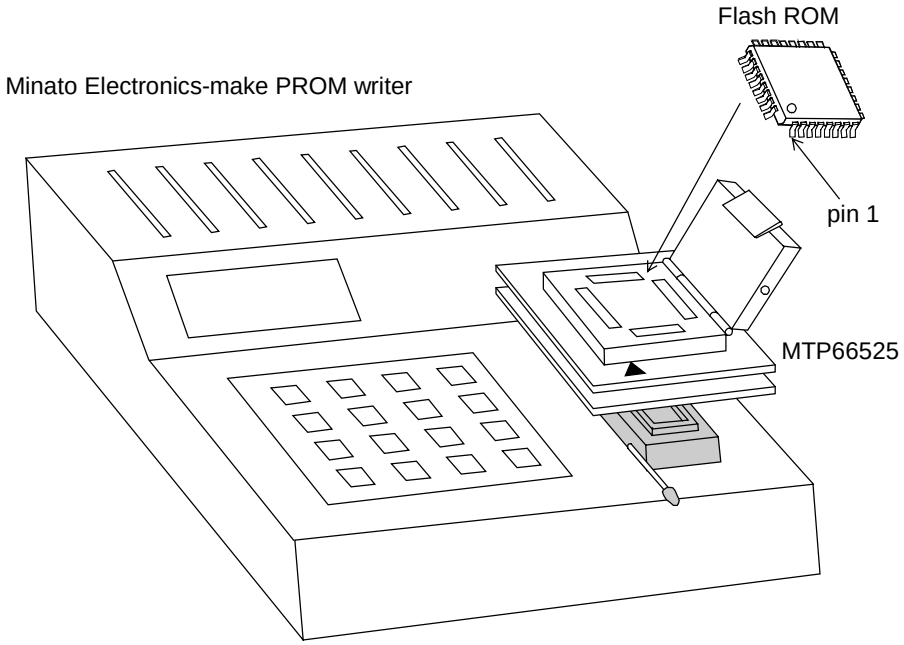


Figure 20-2 Parallel Mode Connection Diagram

20.5 Serial Mode

20.5.1 Overview of the Serial Mode

Programming in the serial mode is performed with a general-purpose serial 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 general-purpose flash microcontroller programmer (AF200) manufactured by YDC, to the two microcontroller pins (P7_6, P7_7), the FLAMOD pin, and V_{DD} and GND pins. Programming and reading are performed while the microcontroller is in the reset. Since an auto-erase function is provided, flash memory does not have to be erased prior to programming.

20.5.2 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 20-1 lists the pins used in the serial mode.

The serial mode can only be set while the CPU is in reset. The GND level is input to the FLAMOD pin to set the serial mode. V_{DD} is connected to monitor V_{DD} of the user system.

Table 20-1 List of Pins Used in Serial Mode

Pin name	Flash memory function
P7_6	FLACK (serial clock input)
P7_7	FLADAT (serial data I/O)
FLAMOD	FLAMOD (high voltage input to set serial mode)
V_{DD}	User system V_{DD} monitor
GND	Ground

[Note]

The PW66K/AF200 are provided with a pin to which a voltage higher than the power supply (approx. 9 V) is applied during the serial mode, for writing to other Flash ROM.

Be sure to do connection correctly.

(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 P7_6, P7_7, FLAMOD, V_{DD} and GND pins of the microcontroller 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 20-3 shows serial mode connection circuit example 1.

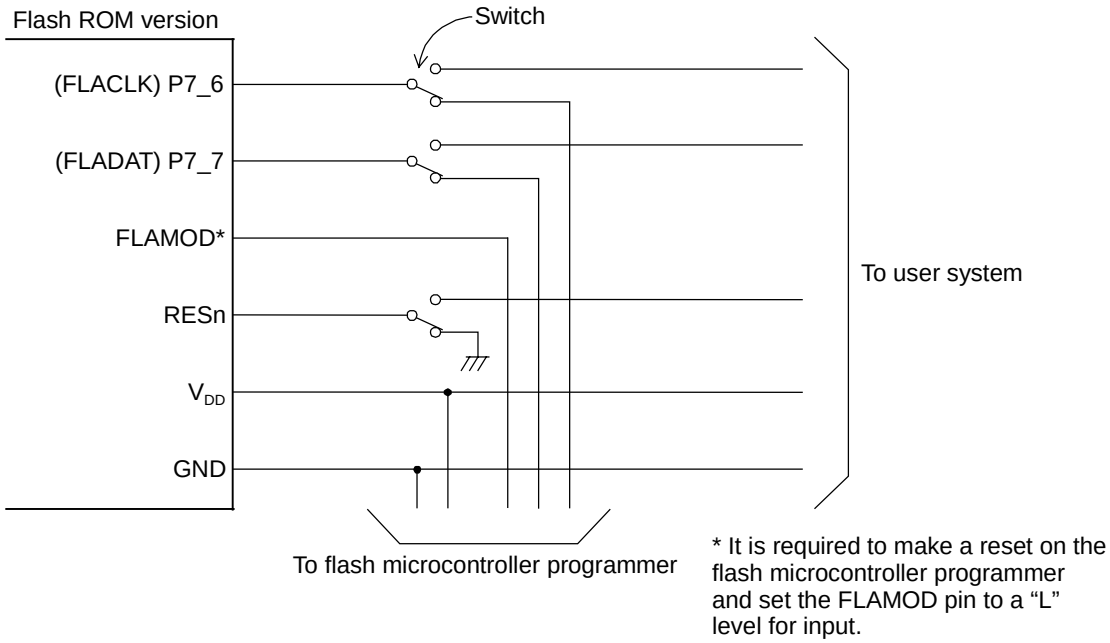


Figure 20-3 Serial Mode Connection Circuit Example 1

If not possible to install a switch in the user system, do not use pins P7_6 and P7_7 with the user system and connect them only to the flash microcontroller programmer. Also, connect each of the P7_6 and P7_7 pins through a resistor of approximately 100 kΩ to V_{DD}.

Figure 20-4 shows serial mode connection circuit example 2.

[Note]

Programming and reading in the serial mode are only available while the microcontroller is in the reset. To execute the programming/reading, apply "L" level to the RESn pin. In addition, it is required to make a reset on the flash microcontroller programmer and make a connection so that a "L" level will be applied to the FLAMOD pin.

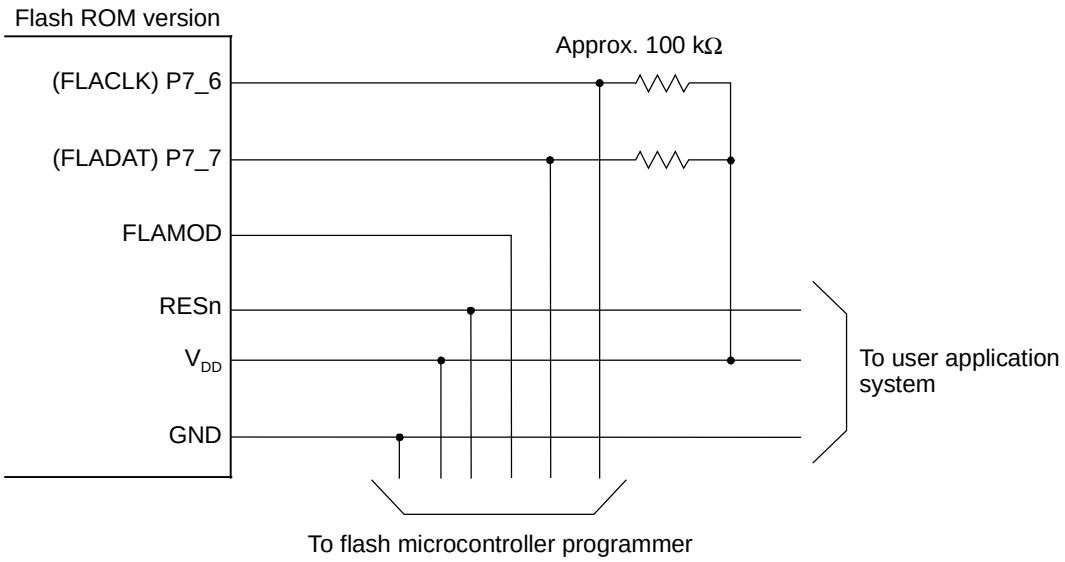


Figure 20-4 Serial Mode Connection Circuit Example 2

(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) .

The procedure for programming with the flash microcontroller programmer is shown 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 P7_6, P7_7, FLAMOD, V_{DD} and GND pins of the microcontroller.
2) Set the microcontroller to the reset. (Apply a "L" level to the RESn pin.)
   The flash microcontroller will generate a protocol error if other than reset is set.
3) Make a reset on the flash microcontroller programmer and apply a "L" level to the FLAMOD pin.
4) Perform the programming or read operation with the flash microcontroller programmer.
   The serial mode has been set by the settings of 2) and 3).
5) Verify that operation of the flash microcontroller programmer has been completed correctly.
6) Release reset on the flash microcontroller programmer (FLAMOD = "pulled-up" level) and on the microcontroller (RESn = "pulled-up" level).
   The serial mode is released automatically.
7) Release reset on the microcontroller (RESn = "pulled-up" level).
   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.

[Note]
Do not repeat security function resetting one after another: Each time the security function is reset, all the sectors are erased and the endurance of the flash ROM decreases. Moreover, the reliability becomes lower when data is written next time.

20.6 User Mode

20.6.1 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 20-5 shows a block diagram of the user mode.

[Note] The security function cannot be set in the user mode.

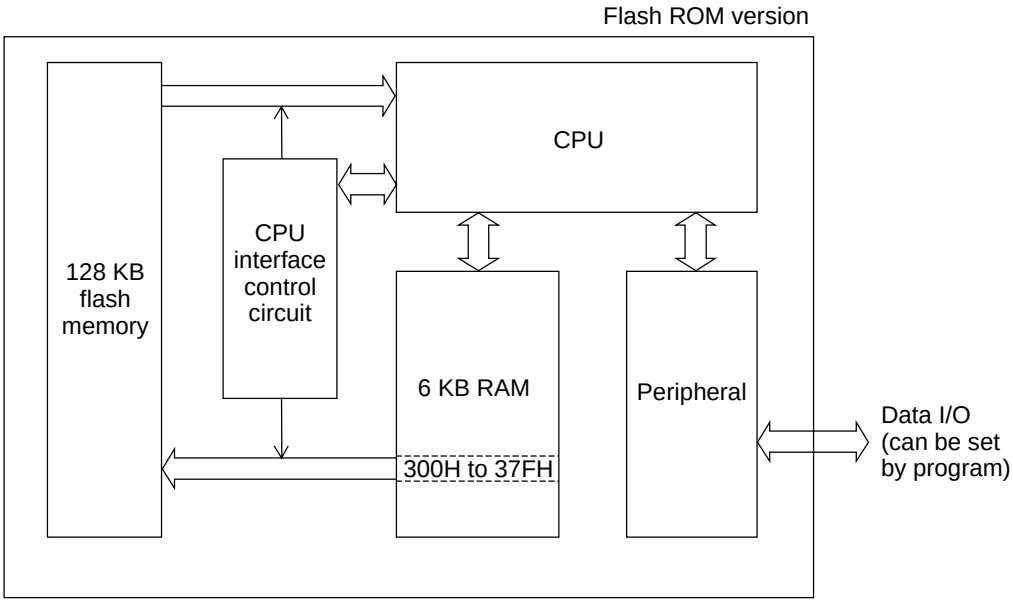


Figure 20-5 User Mode Block Diagram

20.6.2 User Mode Programming Registers

The ML66Q525 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 20-2 lists a summary of the SFRs for the user mode.

Table 20-2 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"	20-9
00F1☆	Flash memory control register	FLACON	—	R/W	8	66/E6	20-9
00F2☆	Flash memory address register	—	FLAADRS	R/W	16	Undefined	20-8
00F3☆		—					

[Notes]

1. A star (☆) in the address column indicates a missing bit.
2. For details, refer to Chapter 22, "Special Function Registers (SFRs)".

20.6.3 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 20-6 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
16	15	14	13	12	11	10	9	8	7	
0	0	0	0	0	0	0	0	0	0	0000H to 007FH
0	0	0	0	0	0	0	0	0	1	0080H to 00FFH
0	0	0	0	0	0	0	0	1	0	0100H to 017FH
1	1	1	1	1	1	1	1	0	1	1FE80H to 1FEFFH
1	1	1	1	1	1	1	1	1	0	1FF00H to 1FF7FH
1	1	1	1	1	1	1	1	1	1	1FF80H to 1FFFFH

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

Figure 20-6 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 20-7 shows the FLAACP configuration.

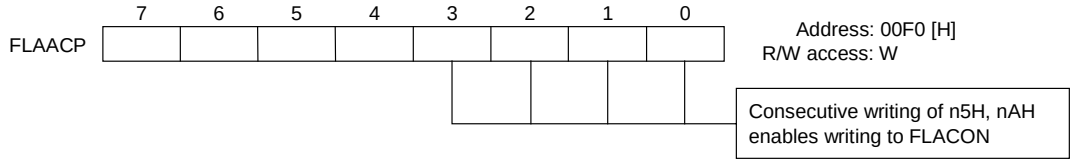


Figure 20-7 FLAACP Configuration

(3) Flash memory control register (FLACON)

FLACON is a 4-bit register that controls programming and operation of the flash memory.

Figure 20-8 shows the FLACON configuration.

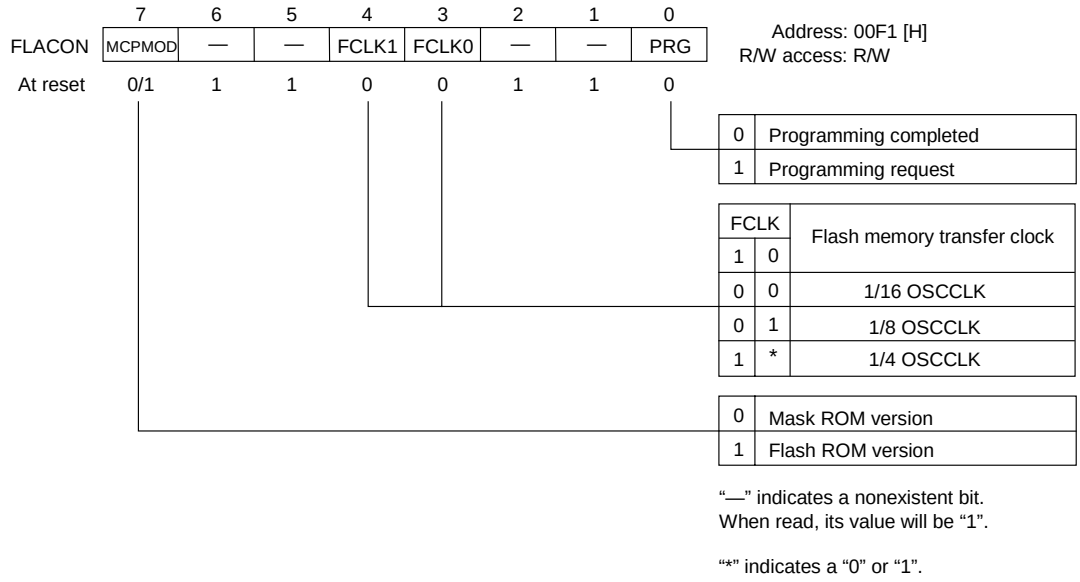


Figure 20-8 FLACON Configuration

Writes to bits 0, 3, and 4 (PRG, FCLK0, FCLK1) 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.

Bit 7 (MCPMOD) of the FLACON is read-only. The write operation is ignored.

[Description of each bit]

- PRG (bit 0)

If PRG (bit 0) of FLACON is set to "1" and bit 1 (HLT) of the standby control register (SBYCON) is set to "1", at the beginning of the next instruction, the CPU will enter the HALT mode and data at internal RAM addresses 300H through 37FH will be transferred to the flash memory. Then, a flash memory block will be cleared by the auto-erase function and the write operation performed.

When the HALT mode is released by a Timer 0 interrupt, 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 other then Timer 0 interrupt occurs during programming of the flash memory, the interrupt is processed. However, the flash memory area will be incorrectly programmed. Also, the CPU program will get out of control after the interrupt is processed.
- If reset is initiated by input to the RESn 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 be incorrectly programmed. If reset is initiated during programming, reprogram the flash memory area that was in the process of being programmed.

Figure 20-9 shows the relationship between internal RAM and flash memory.

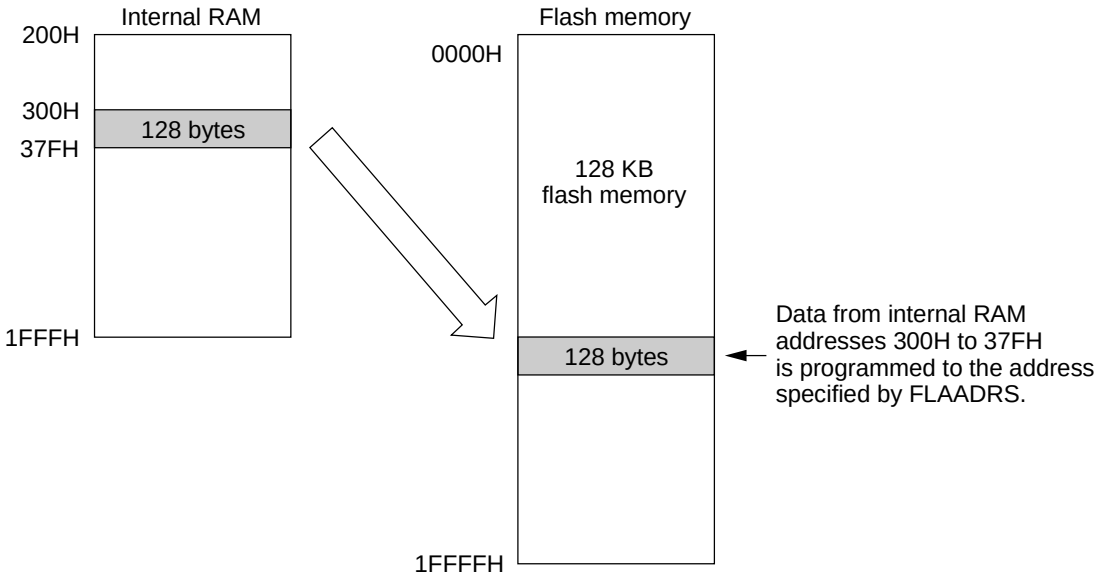


Figure 20-9 Relation between Internal RAM and Flash Memory

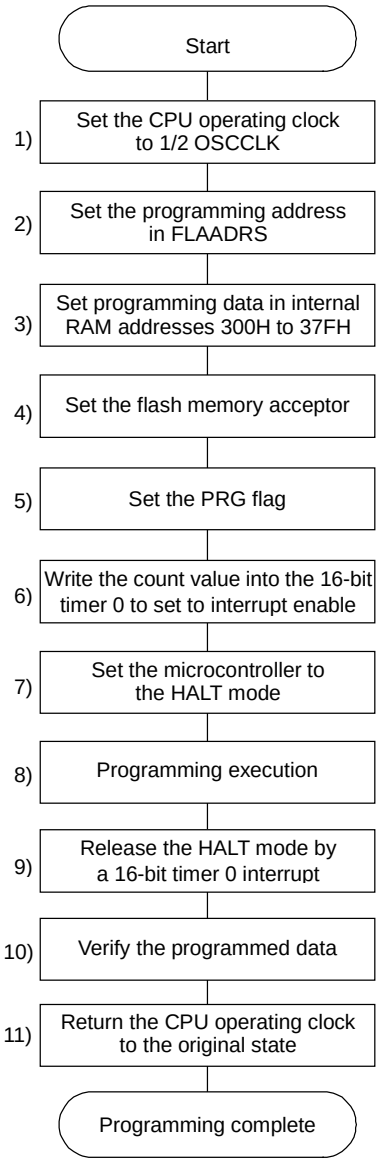
- FCLK0, FCLK1 (bits 3,4)
  FCLK0 (bit 3) and FCLK1 (bit 4) of the FLACON register are bits that set the clock that transfers data from internal RAM addresses 300H through 37FH to flash memory. At reset, since both FCLK0 and FCLK1 become "0", 1/16CLK will be selected as the transfer clock.
- MCPMOD (bit 7)
  MCPMOD (bit 7) of FLACON is a bit that is used to know whether the on-chip memory is Mask ROM or Flash ROM.
  When "0" is read, the on-chip memory is Mask ROM.
  When "1" is read, it is Flash memory.
  This bit is read-only. The write operation is ignored.

20.6.4 User Mode Programming Example

(1) User mode programming flowchart example

Figure 20-10 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) Write a "0" and a "1" to bit 7 and bit 6 of the standby control register (SBYCON) respectively.

2) Set FLAADRS with the upper 10 bits (FA7 to FA16) of the address desired to write to in 128-byte units.

3) Using serial I/O or external memory access, set the programming data in internal RAM addresses 300H through 37FH.

4) Write the consecutive values of n5H, nAH (n = 0 to F) to FLAACP to set the flash memory acceptor to "1". After completion of flash memory programming, the flash memory acceptor is reset to "0" by a 16-bit timer 0 interrupt.

5) Set the program (PRG) flag of FLACON. A 16-bit timer 0 interrupt causes the PRG flag to be reset to "0".

6) Since the flash memory programming time is managed by the 16-bit timer 0, enable 16-bit timer 0 interrupts only. Disable all the other interrupts.

7) The microcontroller enters the HALT mode at the beginning of the next instruction.

8) When the CPU enters the HALT mode, data in internal RAM addresses 300H through 37FH is transferred to the flash memory, and programming (with the auto-erase function) is performed.

9) After completion of flash memory programming, release the HALT mode by a 16-bit timer 0 interrupt.

10) Verify that data has been written to the flash memory. ROM table reference instructions may be used to verify the data.

11) Return the CPU operating clock to the clock used by the program.

(Note) It is impossible to write data or verify the written data using an external program.

Figure 20-10 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.

```
        MOVB    SBYCON,#40H     Set the CPU operating clock to 1/2 OSCCLK.
        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.
        MOV     TM0C,#6700H     Set the timer start value for the timer 0 to start
                                counting.
        MOV     TM0R,#0000H     Write the timer 0 reload value.
        MOVB    TM0CON,#08H     Make timer 0 start counting.
        RB      QTM0OV          Clear timer 0 interrupt request.
        MOVB    IE1,#01H        Enable timer 0 interrupts.
        SB      SBYCON.1        Set the HLT flag.
        NOP                     At the beginning of the instruction, the CPU enters
                                the HALT mode and programming begins. The
                                HALT mode is released by timer 0 interrupt.

        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]

It is impossible to write data or verify the written data using an external program.

(3) Timer 0 counter count start value

The time of writing data in the flash memory must be controlled by the 16-bit timer 0 counter.

As the flash memory data write time is up to 50 ms (at 2.4 to 3.6 V) for every 128 bytes, set a timer value for starting counting of the timer 0 counter so that the time period between the start of programming (after the PRG flag is set to go into the HALT mode) and the release of the HALT mode (or release of the write mode) is 50 ms or longer.

(Equation for calculating the overflow time of the timer 0 counter)

The time (tTM0) to when the timer 0 counter overflows is calculated from:

$$tTM0 = (1/f) \times T \times N \times (65536 - TM0C) \quad [\mu s] \quad (TM0C: 0 \text{ to } 65535)$$

where

f = Frequency (MHz) of the basic clock (CPUCLK)

T = Count clock of the timer 0 counter (divide value of TBCCLK)

N = Divide value of the 1/n counter

TM0C = Count starting value of the timer 0 counter

Table 20-2 lists TM0C starting values to set a flash memory data write time of 50 ms when OSCCLK is 24 MHz.

Table 20-2 Set Values for Flash Memory Data Write Time (OSCCLK = 24 MHz)

CPUCLK	N	Count clock selected	TM0C start value [H]
1/2 OSCCLK	1	1/16 TBCCLK	6D80
	2	1/8 TBCCLK	6D80
	3	1/4 TBCCLK	3CA0
	5	1/2 TBCCLK	1590
	10	TBCCLK	1590
1/4 OSCCLK	1	1/8 TBCCLK	6D80
	2	1/4 TBCCLK	6D80
	3	1/2 TBCCLK	3CA0
	5	TBCCLK	1590

[Note]

Do not set CPUCLK of 1/1 OSCCLK for writing in the User mode. If set, the CPU control will run out of control after a write is completed.

20.6.5 Notes on Use of User Mode

Note the following items when generating a program to be used with the user mode.

- If an interrupt other than Timer 0 interrupt occurs during programming of the flash memory, the interrupt is processed. However, the flash memory area will be incorrectly programmed. Also, the CPU program will get out of control after the interrupt is processed.
- If reset is initiated by input to the RESn 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.)
- When the flash memory is programmed in the user mode while the watchdog timer (WDT) is running, the WDT is designed to stop the count clock during HALT mode is in progress.
- After programming of the flash memory in the user mode, the flash memory is changed from programming mode to read mode. To read data from the first flash memory, set CPUCLK = 1/2 OSCCLK in advance before programming of the flash memory in the user mode.
  (If CPUCLK = 1/1 OSCCLK, the CPU program control will run out of control after programming is completed.)
- Development tools (emulator) cannot evaluate programming or erasing.

20.7 Notes on Program

(1) Programming of flash memory immediately after power-on

Programming to flash memory is automatically disabled for approximately 20 ms (for 2.4 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) When the supply voltage drops

If the supply voltage drops to a value below the guaranteed value (2.4 V) during programming to the flash memory, data may not be programmed correctly. So take care that the supply voltage during programming does not drop below the guaranteed value. After programming, be sure to check the data in the flash memory to see if data has been programmed correctly.

Chapter 21

Electrical Characteristics

21. Electrical Characteristics

21.1 Absolute Maximum Ratings

Parameter	Symbol	Condition		Rated value	Unit
Digital power supply voltage	V_{DD}_CORE V_{DD}_IO VBUS	GND = AGND = 0 V Ta = 25°C		–0.3 to +4.6	V
Input voltage	V_I	Other than P9_0		–0.3 to V_{DD}_IO + 0.3	V
		P9_0 (5 V tolerant input)		–0.3 to +0.6	V
Output voltage	V_O			–0.3 to V_{DD}_IO + 0.3	V
Analog reference voltage	V_{REF}			–0.3 to +4.6	V
Analog input voltage	V_{AI}			–0.3 to V_{REF}	V
Power dissipation	P_D	Ta = 70°C per package	100-pin TQFP	680	mW
			144-pin LFBGA	595	mW
Storage temperature	T_{STG}	—		–50 to +150	°C

21.2 Recommended Operating Conditions

Parameter	Symbol	Condition		Range	Unit
Digital power supply voltage	V_{DD}_CORE V_{DD}_IO	f_{OSC} ≤ 24 MHz V_{DD}_CORE ≤ V_{DD}_IO		2.4 to 3.6	V
Analog reference voltage	V_{REF}	V_{DD}_CORE ≤ V_{REF}		2.4 to 3.6	V
Analog input voltage	V_{AI}	—		AGND to V_{REF}	V
VBUS input voltage	VBUS	—		3.0 to 3.6	V
Memory hold voltage	V_{DDH}	f_{OSC} = 0 Hz		2.0 to 3.6	V
Operating frequency	f_{OSC}	USB is used		12, 16, 24	MHz
		USB is unused		2 to 24	
	f_{XT}	—		32.768	kHz
Ambient temperature	Ta	—		–30 to +70	°C
Fan out	N	MOS load		20	—
		TTL load	P7, P10_0 to P10_2	6	—
			P0, P1, P2, P3, P4, P6, P8, P9, P10_3 to P10_5, P15, P20, P21	1	—

21.3 Allowable Output Current Values

(V_{DD}_IO = 2.4 to 3.6 V, Ta = –30 to +70°C)

Parameter	Pin	Symbol	Min.	Typ.	Max.	Unit
"H" output pin (1 pin)	All output pins	I_{OH}	—	—	–10	mA
"H" output pins (sum total)	Sum total of all output pins	ΣI_{OH}	—	—	–70	
"L" output pin (1 pin)	All output pins	I_{OL}	—	—	10	
"L" output pins (sum total)	Sum total of P0, P3	ΣI_{OL}	—	—	35	
	Sum total of P1, P2, P4					
	Sum total of P6, P7, P8, P9					
	Sum total of P10, P15					
	Sum total of P20, P21				70	
	Sum total of all output pins				160	

[Note]

Connect all V_{DD}_CORE and V_{DD}_IO pins to the power supply voltage and all GND pins to the ground voltage. If there is a pin or pins that are not connected to the power supply voltage or ground voltage, the device cannot be guaranteed for normal operation.

21.4 Internal Flash ROM Programming Conditions

Parameter	Symbol	Condition	Rating	Unit
Supply voltage	V_{DD}_CORE V_{DD}_IO	V_{DD}_CORE ≤ V_{DD}_IO	2.4 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

21.5 DC Characteristics

21.5.1 DC Characteristics (Except USB port)

(V_{DD}_CORE = V_{DD}_IO = V_{REF} = 2.4 to 3.6 V, GND = AGND = 0 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
"H" input voltage *1	V_{IH}	—	0.80 V_{DD}	—	5.5	V
"H" input voltage			0.80 V_{DD}	—	V_{DD} + 0.3	
"L" input voltage	V_{IL}	—	–0.3	—	0.2V_{DD}	
"H" output voltage *2	V_{OH}	I_O = –400 µA	V_{DD} – 0.4	—	—	
		I_O = –2.0 mA	V_{DD} – 0.8	—	—	
"H" output voltage *3		I_O = –200 µA	V_{DD} – 0.4	—	—	
		I_O = –1.0 mA	V_{DD} – 0.8	—	—	
"L" output voltage *2	V_{OL}	I_O = 3.2 mA	—	—	0.5	
		I_O = 5.0 mA	—	—	0.9	
"L" output voltage *3		I_O = 1.6 mA	—	—	0.5	
		I_O = 2.5 mA	—	—	0.9	
Input leakage current *4, *6	I_{IH}/I_{IL}	V_I = V_{DD}/0 V	—	—	1/–1	µA
Input current *5			—	—	1/–90	
Input current *7			—	—	15/–15	
Output leakage current *2, *3	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	—	1.8	5	mA
		When A/D is stopped	—	—	5	µA

V_{DD} = V_{DD}_IO

*1. Applicable to P9_0 (5 V tolerant input)

*2. Applicable to P7 and P10_0 to P10_2

*3. Applicable to P0, P1, P2, P3, P4, P6, P8, P9, P10_3 to P10_5, P15, P20 and P21

*4. Applicable to P12 and P13

*5. Applicable to RESn and FLAMOD

*6. Applicable to EAn, NMI, and TEST

*7. Applicable to OSC0

Supply current

- ML66525A

(V_{DD}_CORE = V_{DD}_IO = V_{REF} = 2.4 to 3.6 V, VBUS = 3.0 to 3.6 V, GND = AGND = 0 V, Ta = –30 to +70°C)

Mode	Symbol	Condition		Min.	Typ.	Max.	Unit	Applicable power supply
CPU operation mode	I_{DD}	fosc = 24 MHz, No load		—	28	60	mA	V_{DD}_CORE + V_{DD}_IO
		fosc = 24 MHz, DMA/media control stopped. No load			18	50		
		f_{XT} = 32.768 kHz, DMA/media control stopped. No load *1		—	100	300	µA	
USB operation mode	I_{BUS}	Setting of 48 MHz for multiplication selection. No Load		—	25	45	mA	VBUS
HALT mode	I_{DDH}	fosc = 24 MHz, DMA/media control stopped. No load		—	9	18	mA	V_{DD}_CORE + V_{DD}_IO
STOP mode	I_{DDS}	OSC is stopped *1	XT is used *2	—	15	160	µA	V_{DD}_CORE + V_{DD}_IO
			XT is not used *2	—	10	150		
Suspend current	I_{SUSP}	Suspend state OSC is stopped, XT is not used *1		—	1	100	µA	VBUS

The values in the Typ. Columm indicate reference values at 25°C and 3.0 V (The VBUS currents indicate values at 3.3 V).
*1: The temperature condition ranges from –30 to +50°C
*2: Ports configured to be inputs should be connected to V_{DD}_IO or 0 V. No load should be applied to other ports.

- ML66Q525A

(V_{DD}_CORE = V_{DD}_IO = V_{REF} = 2.4 to 3.6 V, VBUS = 3.0 to 3.6 V, GND = AGND = 0 V, Ta = –30 to +70°C)

Mode	Symbol	Condition		Min.	Typ.	Max.	Unit	Applicable power supply
CPU operation mode	I_{DD}	fosc = 24 MHz, No load		—	28	60	mA	V_{DD}_CORE + V_{DD}_IO
		fosc = 24 MHz, DMA/media control stopped. No load			18	50		
		f_{XT} = 32.768 kHz, DMA/media control stopped. No load *1		—	100	300	µA	
USB operation mode	I_{BUS}	Setting of 48 MHz for multiplication selection No Load		—	25	45	mA	VBUS
HALT mode	I_{DDH}	fosc = 24 MHz, DMA/media control stopped. No load		—	10	20	mA	V_{DD}_CORE + V_{DD}_IO
STOP mode	I_{DDS}	OSC is stopped *1	XT is used *2	—	15	160	µA	V_{DD}_CORE + V_{DD}_IO
			XT is not used *2	—	10	150		
Suspend current	I_{SUSP}	Suspend state, D+/D– fixed OSC is stopped, XT is not used *1		—	1	100	µA	VBUS

The values in the Typ. Columm indicate reference values at 25°C and 3.0 V (The VBUS currents indicate values at 3.3 V).
*1: The temperature condition ranges from –30 to +50°C
*2: Ports configured to be inputs should be connected to V_{DD}_IO or 0 V. No load should be applied to other ports.

21.5.2 DC Characteristics (USB port)

(VBUS = 3.0 to 3.6 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit	Applicable pin
Differential input sensitivity	V_{DI}	|(D+) – (D–)|	0.2	—	—	V	D+, D–
Differential common mode range	V_{CM}	Includes V_{DI}	0.8	—	2.5		
Single ended receiver threshold	V_{SE}		0.8	—	2.0		
"H" output voltage	V_{OH}	15 kΩ to GND	2.8	—	—	V	D+, D–
		I_{OH} = –100 µA	VBUS – 0.2	—	—	V	PUCTL
		I_{OH} = 4 mA	2.4	—	—		
"L" output voltage	V_{OL}	1.5 kΩ to 3.6 V	—	—	0.3	V	D+, D–
Output leakage current	I_{LO}	V_O = VBUS/0 V	—	—	±10	µA	D+, D–
		V_O = VBUS/0 V	—	—	±10		PUCTL

21.6 AC Characteristics

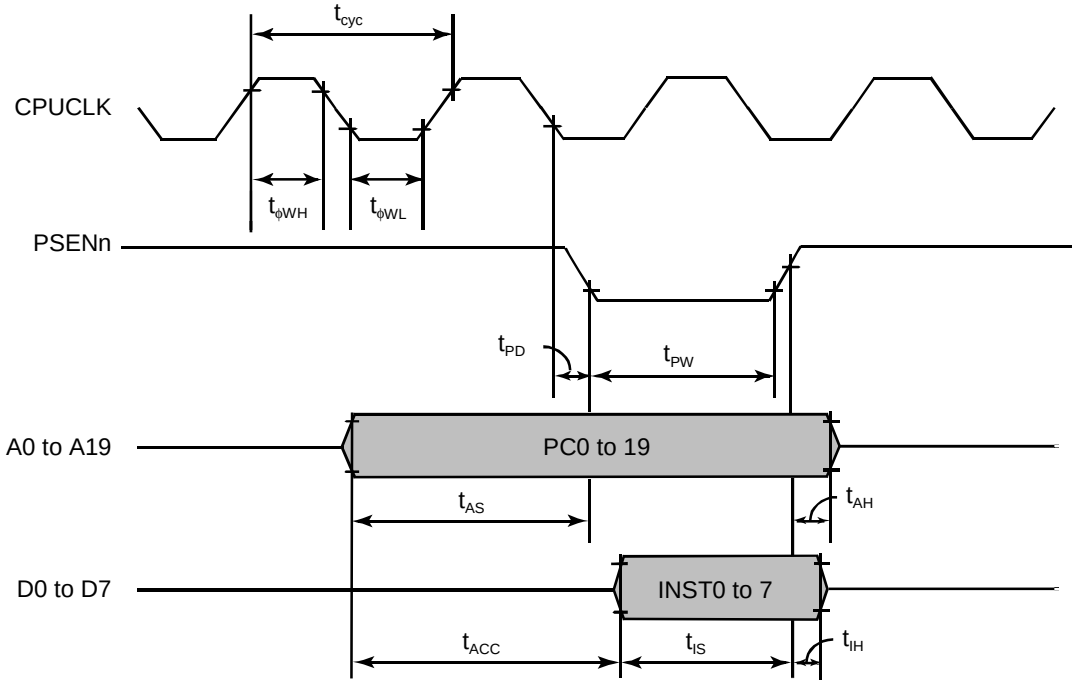
21.6.1 AC Characteristics (Except USB port)

(1) External program memory control

(V_{DD}_CORE = V_{DD}_IO = V_{REF} = 2.4 to 3.6 V, GND = AGND = 0 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 24 MHz	41.67	—	ns
Clock pulse width (HIGH level)	$t_{\phi WH}$	V_{DD}_CORE = C_L = 50 pF	16.25	—	
Clock pulse width (LOW level)	$t_{\phi WL}$		16.25	—	
PSENn pulse width	t_{PW}		(2 + 2n)tϕ – 25	—	
PSENn pulse delay time	t_{PD}		—	55	
Address setup time	t_{AS}		2tϕ – 25	—	
Address hold time	t_{AH}		–10	—	
Instruction setup time	t_{IS}		40	—	
Instruction hold time	t_{IH}		0	—	
Read data access time	t_{ACC}		—	(3 + 2n)tϕ – 50	

(Note) tϕ = t_{cyc}/2
N = 0 to 3 (n wait cycles inserted)



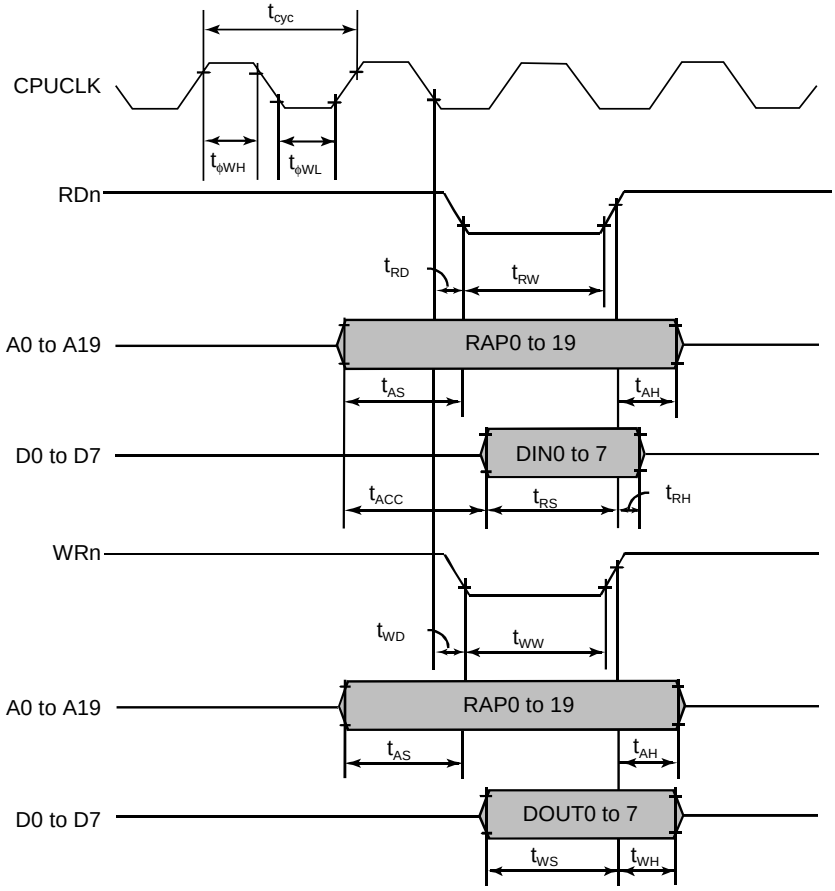
Bus timing during no wait cycle time

(2) External data memory control

(V_{DD}_CORE = V_{DD}_IO = V_{REF} = 2.4 to 3.6 V, GND = AGND = 0 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 24 MHz	41.67	—	ns
Clock pulse width (HIGH level)	$t_{\phi WH}$	C_L = 50 pF	16.25	—	
Clock pulse width (LOW level)	$t_{\phi WL}$		16.25	—	
RDn pulse width	t_{RW}		(2 + 2n)tϕ – 25	—	
WRn pulse width	t_{WW}		(2 + 2n)tϕ – 25	—	
RDn pulse delay time	t_{RD}		—	55	
WRn pulse delay time	t_{WD}		—	55	
Address setup time	t_{AS}		tϕ – 20	—	
Address hold time	t_{AH}		tϕ – 20	—	
Read data setup time	t_{RS}		40	—	
Read data hold time	t_{RH}		0	—	
Read data access time	t_{ACC}		—	(3 + 2n)tϕ – 50	
Write data setup time	t_{WS}		2tϕ – 30	—	
Write data hold time	t_{WH}		tϕ – 6	—	

(Note) tϕ = t_{cyc}/2
N = 0 to 7 (n wait cycles inserted)



Bus timing during no wait cycle time

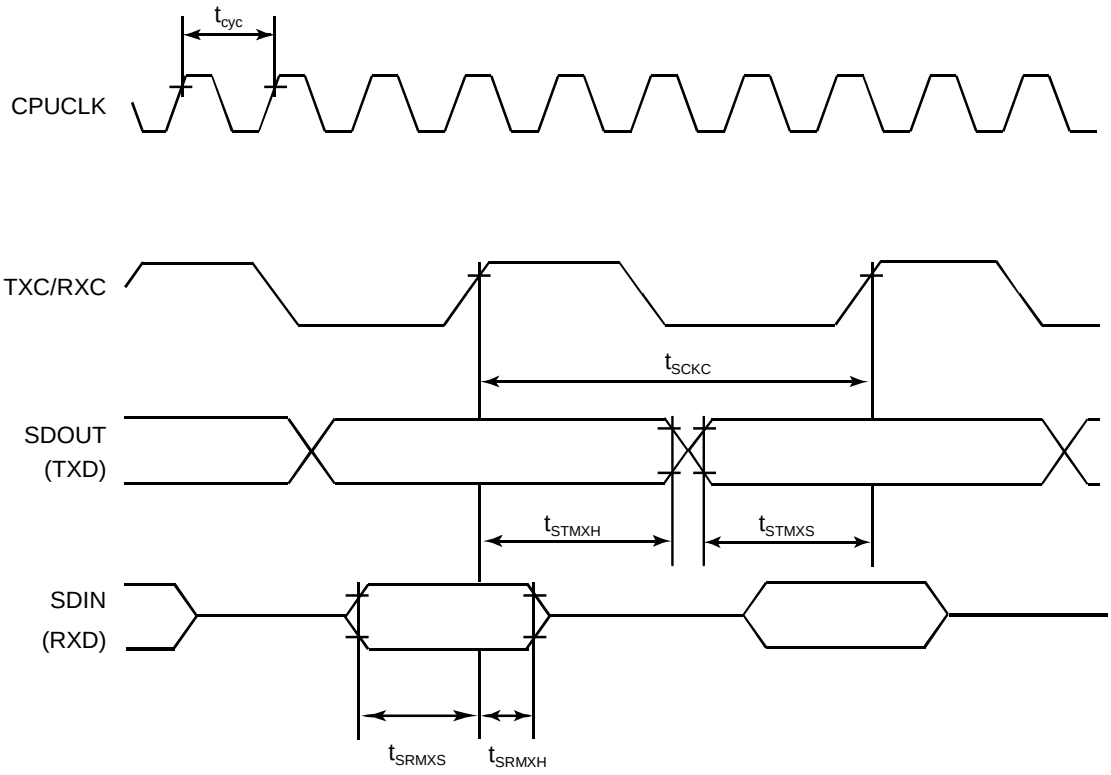
(3) Serial port control

1. Serial port 1, 6 (SIO1, 6)
   Master mode (Clock synchronous serial port)

(V_{DD}_CORE = V_{DD}_IO = V_{REF} = 2.4 to 3.6 V, GND = AGND = 0 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 24 MHz	41.67	—	ns
Serial clock cycle time	t_{SCKC}	C_L = 50 pF	4 t_{cyc}	—	
Output data setup time	t_{STMXS}		2tϕ – 10	—	
Output data hold time	t_{STMXH}		5tϕ – 20	—	
Input data setup time	t_{SRMXS}		21	—	
Input data hold time	t_{SRMXH}		0	—	

(Note) tϕ = t_{cyc}/2

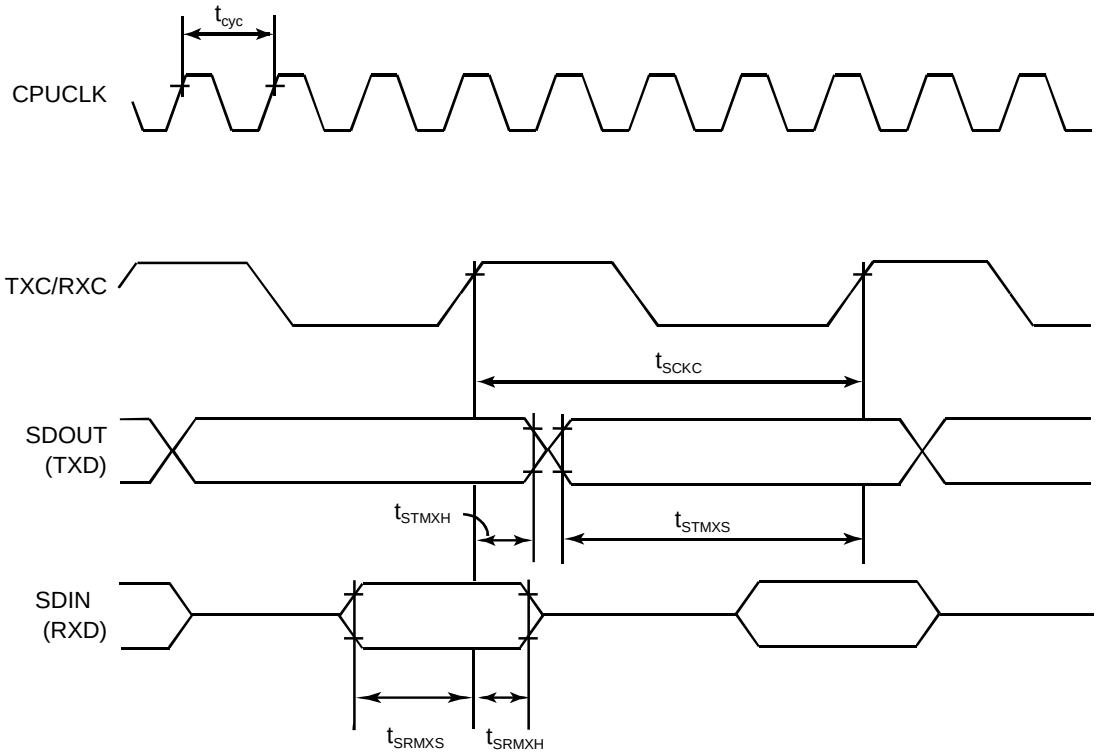


Slave mode (Clock synchronous serial port)

(V_{DD}_CORE = V_{DD}_IO = V_{REF} = 2.4 to 3.6 V, GND = AGND = 0 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 24 MHz	41.67	—	ns
Serial clock cycle time	t_{SCKC}	C_L = 50 pF	$4t_{cyc}$	—	
Output data setup time	t_{STMXS}		$2t\phi - 30$	—	
Output data hold time	t_{STMXH}		$4t\phi - 20$	—	
Input data setup time	t_{SRMXS}		21	—	
Input data hold time	t_{SRMXH}		7	—	

(Note) $t\phi = t_{cyc}/2$

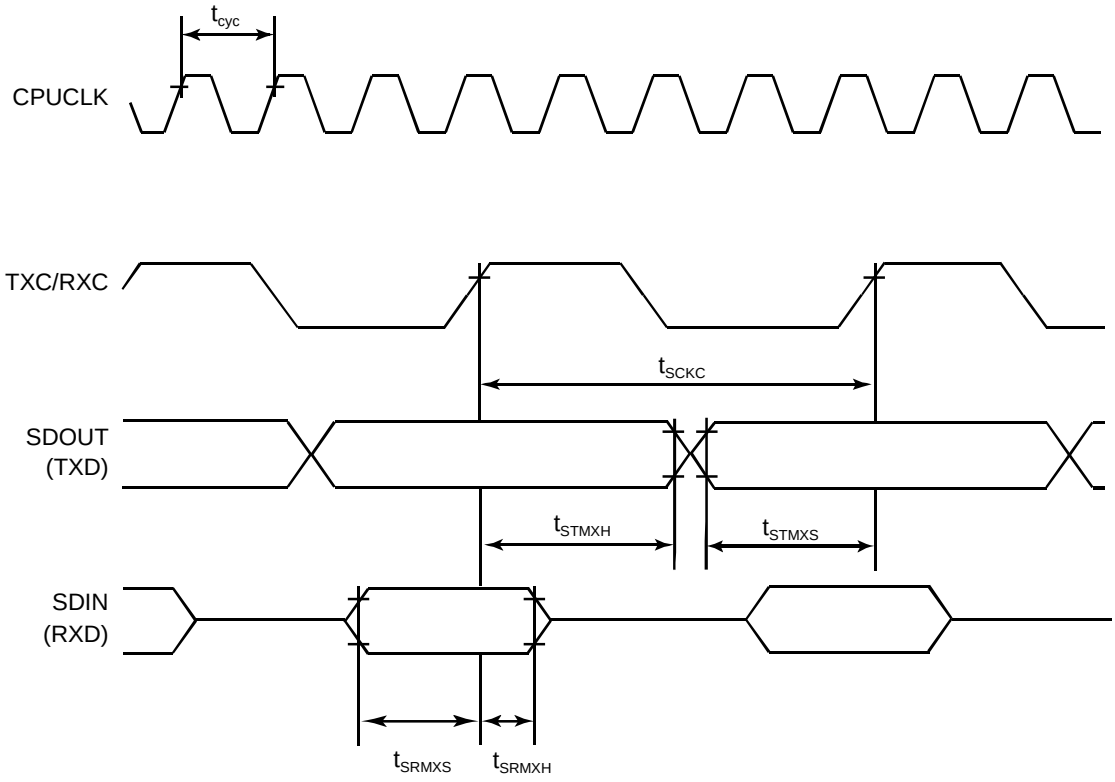


2. Serial port 4 (SIO4)
   Master mode (Clock synchronous serial port)

(V_{DD}_CORE = V_{DD}_IO = V_{REF} = 2.4 to 3.6 V, GND = AGND = 0 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 24 MHz	41.67	—	ns
Serial clock cycle time	t_{SCKC}	C_L = 50 pF	400	—	
Output data setup time	t_{STMXS}		190	—	
Output data hold time	t_{STMXH}		130	—	
Input data setup time	t_{SRMXS}		21	—	
Input data hold time	t_{SRMXH}		0	—	

(Note) $t\phi = t_{cyc}/2$

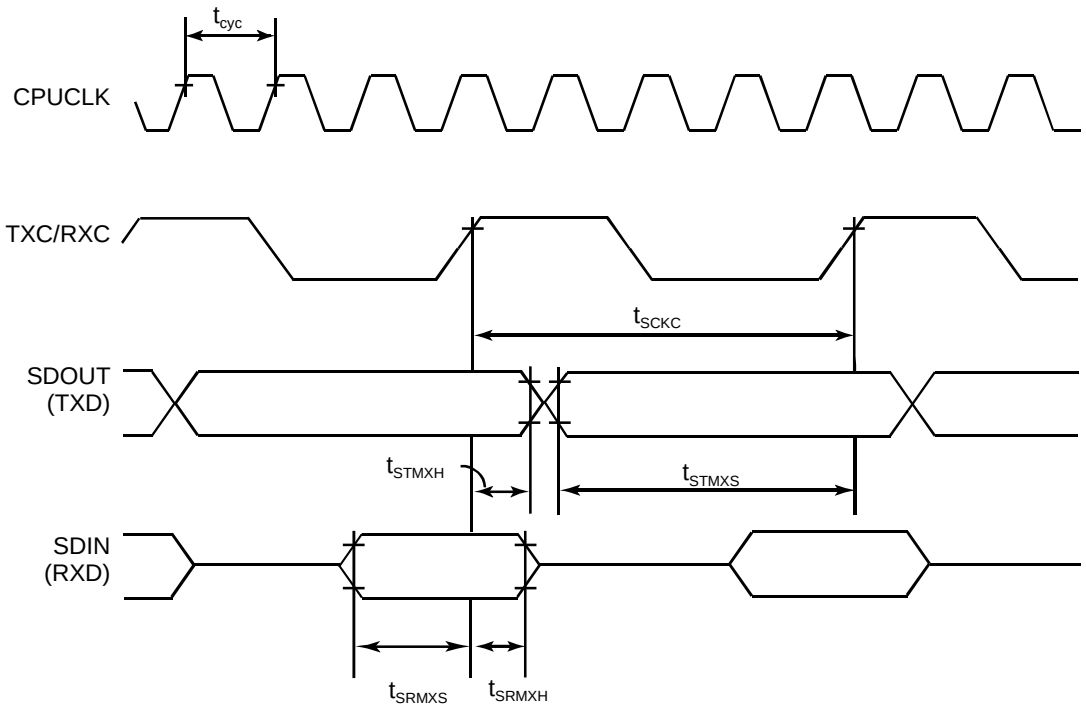


Slave mode (Clock synchronous serial port)

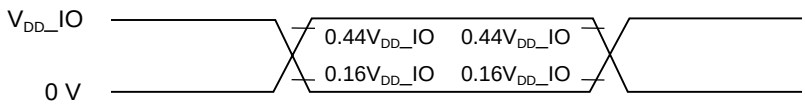
(V_{DD}_CORE = V_{DD}_IO = V_{REF} = 2.4 to 3.6 V, GND = AGND = 0 V, Ta = –30 to +70°C)

Parameter	Symbol	Condition	Min.	Max.	Unit
Cycle time	t_{cyc}	f_{OSC} = 24 MHz	41.67	—	ns
Serial clock cycle time	t_{SCKC}	C_L = 50 pF	400	—	
Output data setup time	t_{STMXS}		70	—	
Output data hold time	t_{STMXH}		180	—	
Input data setup time	t_{SRMXS}		21	—	
Input data hold time	t_{SRMXH}		7	—	

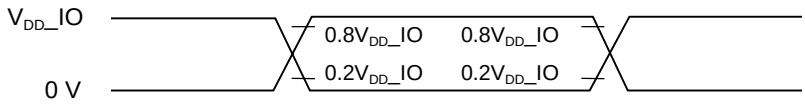
(Note) $t\phi = t_{cyc}/2$



Measurement points for AC timing (except the serial port)



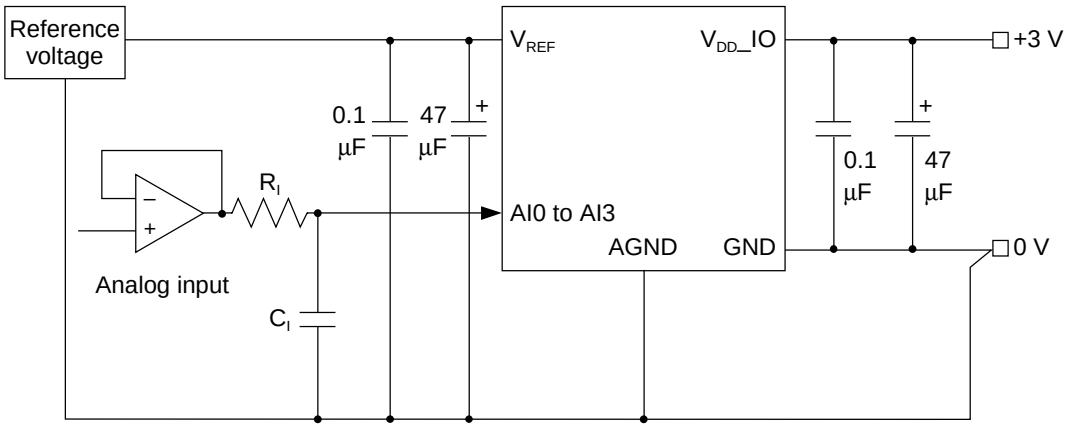
Measurement points for AC timing (the serial port)



21.7 A/D Converter Characteristics

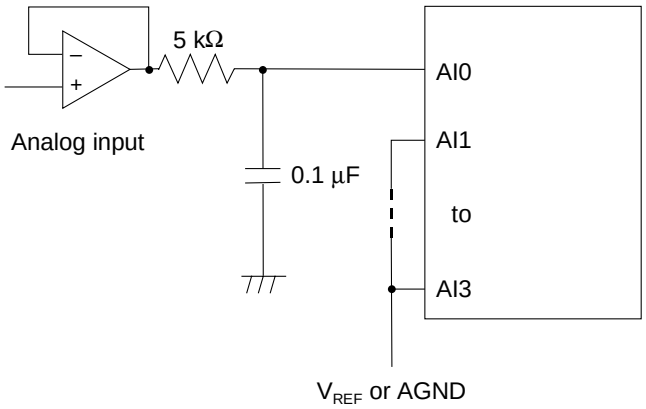
(Ta = –30 to +70°C, V_{REF} = 2.4 to 3.6 V, AGND = GND = 0 V)

Parameter	Symbol	Condition	Min.	Typ.	Max.	Unit
Resolution	n	Refer to measurement circuit 1 Analog input source impedance $R_I \leq 5$ kΩ	—	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 2	—	—	±1	
Conversion time	t_{CONV}	Set according to ADTM set data	16	—	3906.3	µs/ch



R_I (impedance of analog input source) ≤ 5 kΩ
C_I ≅ 0.1 µF

Measurement Circuit 1



Cross talk is the difference between the A/D conversion results when the same analog input is applied to AI0 through AI3 and the A/D conversion results of the circuit to the left.

Measurement Circuit 2

Definition of Terminology

1. Resolution
   Resolution is the value of minimum discernible analog input.
   With 10 bits, since $2^{10} = 1024$, resolution of $(V_{REF} - AGND) \div 1024$ 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 $1LSB = (V_{REF} - AGND) \div 1024$. 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.

Chapter 22

Special Function Registers (SFRs)

22. Special Function Registers (SFRs)

22.1 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)

22.2 List of SFRs

Tables 22-1 and 22-2 list the SFRs. Terms in the tables 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 (RESn signal input, execution of a BRK instruction, overflow of the watchdog timer, or an opcode trap). 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 22-1 SFR List (1/5)

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	2-22
0001		—												FF	
0002	Local register base	LRBL	LRB										8/16	Undefined	2-21
0003		LRBH												Undefined	
0004	Program status word	PSWL	PSW	MAB	F1	BCB1	BCB0	F0	SCB2	SCB1	SCB0		8/16	00	2-17
0005		PSWH		CY	ZF	HC	DD	S	F2	OV	MIE			00	
0006	Accumulator	ACCL	ACC										8/16	00	2-16
0007		ACCH												00	
0008	Table segment register	TSR	DSTSR	0	0	0	0						8/16	00	2-25
0009	Data segment register	DSR		0	0	0	0							00	2-26
000A	EXPANDED RAM ready control register	XPDRDY	(*1)	(1)	(1)	(1)	(1)	(1)	XRDY2	XRDY1	SRDY0		8	FF	4-7
000B	ROM window register	ROMWIN			1	1							8	Undefined	4-2
000C	ROM ready control register	ROMRDY	(*2)	(1)	0	0	0	(1)	0	ORDY1	ORDY0		8	8B	4-4
000D	RAM ready control register	RAMRDY		(1)	ARDY12	ARDY11	ARDY10	(1)	ADRY02	ARDY01	ARDY00		8	FF	4-5
000E	Stop code acceptor	STPACP	—									W	8	"0"	3-3
000F	Stadby 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		8	FC	2-1
0013	Real-time counter control register	RTCCON	—	(1)	(1)	(1)	(1)	SELRTI1	SELRTI0	RTCIE	RTCCL		8	F0	9-2
0015	Peripheral control register	PRPHCON	—	(1)	0	0	EXTXT	(1)	(1)	0	0		8	8C	13-2
0018	Port 0 data register	P0	—	P0_7	P0_6	P0_5	P0_4	P0_3	P0_2	P0_1	P0_0		8	00	5-16
0019	Port 1 data register	P1	—	P1_7	P1_6	P1_5	P1_4	P1_3	P1_2	P1_1	P1_0		8	00	5-18
001A	Port 2 data register	P2	—	(0)	(0)	(0)	(0)	P2_3	P2_2	P2_1	P2_0		8	00	5-20
001B	Port 3 data register	P3	—	(0)	(0)	(0)	(0)	P3_3	P3_2	P3_1	P3_0		8	00	5-22
001C	Port 4 data register	P4	—	P4_7	P4_6	P4_5	P4_4	P4_3	P4_2	P4_1	P4_0		8	00	5-24
001E	Port 6 data register	P6	—	(0)	(0)	(0)	(0)	P6_3	P6_2	P6_1	P6_0	R/W	8	00	5-26
001F	Port 7 data register	P7	—	P7_7	P7_6	(0)	(0)	(0)	(0)	(0)	(0)		8	00	5-28
0020	Port 0 mode register	P0IO	—	P0IO7	P0IO6	P0IO5	P0IO4	P0IO3	P0IO2	P0IO1	P0IO0		8	00/FF	5-16
0021	Port 1 mode register	P1IO	—	P0IO7	P0IO6	P0IO5	P0IO4	P0IO3	P0IO2	P0IO1	P0IO0		8	00/FF	5-18
0022	Port 2 mode register	P2IO	—	(0)	(0)	(0)	(0)	P2IO3	P2IO2	P2IO1	P2IO0		8	00/0F	5-20
0023	Port 3 mode register	P3IO	—	(0)	(0)	(0)	(0)	P3IO3	P3IO2	P3IO1	(0)		8	00/02	5-22
0024	Port 4 mode register	P4IO	—	P4IO7	P4IO6	P4IO5	P4IO4	P4IO3	P4IO2	P4IO1	P4IO0		8	00/FF	5-24
0025	Internal control register	P5IO	(*3)	FSEL1	FSEL0	SCK4IO	SCK4INV	STBSEL	MECKEN	UDCKEN	USBRST		8	00	13-3
0026	Port 6 mode register	P6IO	—	(0)	(0)	(0)	(0)	P6IO3	P6IO2	P6IO1	P6IO0		8	00	5-26

Table 22-1 SFR List (2/5)

Address [H]	Function	Byte	Word	Bit Symbol 7	6	5	4	3	2	1	0	R/W	8/16	Initial value [H]	Reference page
0027	Port 7 mode register	P7IO	—	P7IO7	P7IO6	(0)	(0)	(0)	(0)	(0)	(0)	R/W	8	00	5-28
0028	Port 0 secondary function control register	P0SF	—	XAD7	XAD6	XAD5	XAD4	XAD3	XAD2	XAD1	XAD0		8	00/FF	5-16
				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	5-18
				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/FF	5-20
								P2SF3	P2SF2	P2SF1	P2SF0				
002B	Port 3 secondary function control register	P3SF	—	(0)	(0)	(0)	(0)	WR	RD	PSEN	(0)		8	00/02	5-22
								P3SF3	P3SF2	P3SF1					
002C	Port 4 secondary function control register	P4SF	—	XDM7	XDM6	XDM5	XDM4	XDM3	XDM2	XDM1	XDM0		8	00/FF	5-24
				P4SF7	P4SF6	P4SF5	P4SF4	P4SF3	P4SF2	P4SF1	P4SF0				
002E	Port 6 secondary function control register	P6SF	—	(0)	(0)	(0)	(0)	P6SF3	P6SF2	P6SF1	P6SF0		8	00	5-26
002F	Port 7 secondary function control register	P7SF	—	PWM1OUT	PWM0OUT	(0)	(0)	(0)	(0)	(0)	(0)		8	00	5-28
				P7SF7	P7SF6										
0030	Interrupt request register 0	IRQ0	—	0	0	0	0	0	0	0	QINT0		8	00	15-12
0031	Interrupt request register 1	IRQ1	—	0	QTM3OV	0	0	QINT3	QINT2	QINT1	QTM0OV		8	00	15-13
0032	Interrupt request register 2	IRQ2	—	QSIO1	QTM4OV	0	QTM7OV	QINT7	QINT6	QINT5	QINT4		8	00	15-14
0033	Interrupt request register 3	IRQ3	—	QRTC	QINT8	QAD	QTM6OV	QSIO4	QSIO3	0	QTM5OV		8	00	15-15
									QSIO6						
0034	Interrupt enable register 0	IE0	—	0	0	0	0	0	0	0	EINT0		8	00	15-17
0035	Interrupt enable register 1	IE1	—	0	ETM3OV	0	0	EINT3	EINT2	EINT1	ETM0OV		8	00	15-18
0036	Interrupt enable register 2	IE2	—	ESIO1	ETM4OV	0	ETM7OV	EINT7	EINT6	EINT5	EINT4		8	00	15-19
0037	Interrupt enable register 3	IE3	—	ERTC	EINT8	EAD	ETM6OV	ESIO4	ESIO3	0	ETM5OV		8	00	15-20
									ESIO6						
0038	Interrupt priority control register 0	IP0	—	0	0	0	0	0	0	P1INT0	P0INT0		8	00	15-22
003A	Interrupt priority control register 2	IP2	—	P1INT3	P0INT3	P1INT2	P0INT2	P1INT1	P0INT1	P1TM0OV	P0TM0OV		8	00	15-23
003B	Interrupt priority control register 3	IP3	—	0	0	P1TM3OV	P0TM3OV	0	0	0	0		8	00	15-24
003C	Interrupt priority control register 4	IP4	—	P1INT7	P0INT7	P1INT6	P0INT6	P1INT5	P0INT5	P1INT4	P0INT4		8	00	15-25
003D	Interrupt priority control register 5	IP5	—	P1SIO1	P0SIO1	P1TM4OV	P0TM4OV	0	0	P1TM7OV	P0TM7OV		8	00	15-26
003E	Interrupt priority control register 6	IP6	—	P1SIO4	P0SIO4	P1SIO3	P0SIO3	0	0	P1TM5OV	P0TM5OV		8	00	15-27
						P1SIO6	P0SIO6								
003F	Interrupt priority control register 7	IP7	—	P1RTC	P0RTC	P1INT8	P0INT8	P1AD	P0AD	P1TM6OV	P0TM6OV		8	00	15-28
0058	External interrupt control register 0	EXI0CON	—	EX3M1	EX3M0	EX2M1	EX2M0	EX1M1	EX1M0	EX0M1	EX0M0		8	00	14-2
0059	External interrupt control register 1	EXI1CON	—	EX7M1	EX7M0	EX6M1	EX6M0	EX5M1	EX5M0	EX4M1	EX4M0		8	00	14-3

Table 22-1 SFR List (3/5)

Address [H]	Function	Byte	Word	Bit Symbol 7	6	5	4	3	2	1	0	R/W	8/16	Initial value [H]	Reference page
005A	External interrupt control register 2	EXI2CON	(*4)	MIPF	NMIRD	NMIM1	NMIM0	(1)	(1)	EX8M1	EX8M0	R/W	8	0C/4C	14-4
005B	External interrupt control register	EX8ICON	—	0	0	0	0	0	0	EXI9S	EXI8S		8	00	14-5
005C	Interrupt request register 4	IRQ4	—	(1)	(1)	(1)	QTM9OV	0	0	QPWM1	QPWM0		8	E0	15-16
005D	Interrupt enable register 4	IE4	—	(1)	(1)	(1)	ETM9OV	0	0	EPWM1	EPWM0		8	E0	15-21
005E	Interrupt priority control register 8	IP8	—	0	0	0	0	P1PWM1	P0PWM1	P1PWM0	P0PWM0		8	00	15-29
005F	Interrupt priority control register 9	IP9	—	(1)	(1)	(1)	(1)	(1)	(1)	P1TM9OV	P0TM9OV		8	FC	15-30
0060	TBC clock divider 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	
0064	General-purpose 16-bit timer 0 register	—	TM0R										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
0070	General-purpose 8-bit timer 3 counter	TM3C	—										8	Undefined	8-10
0071	General-purpose 8-bit timer 3 register	TM3R	—										8	Undefined	8-10
0072	General-purpose 8-bit timer 3 control register	TM3CON	—	TM3OUT	(1)	(1)	(1)	TM3RUN	TM3C2	TM3C1	TM3C0		8	70	8-10
0074	General-purpose 8-bit timer 4 counter	TM4C	—										8	Undefined	8-16
0075	General-purpose 8-bit timer 4 register	TM4R	—										8	Undefined	8-16
0076	General-purpose 8-bit timer 4 control register	TM4CON	—	TM4OUT	(1)	(1)	(1)	TM4RUN	TM4C2	TM4C1	TM4C0		8	70	8-16
0078	General-purpose 8-bit timer 5 counter	TM5C	—										8	Undefined	8-22
0079	General-purpose 8-bit timer 5 register	TM5R	—										8	Undefined	8-22
007A	General-purpose 8-bit timer 5 control register	TM5CON	—	TM5OUT	(1)	(1)	(1)	TM5RUN	TM5C2	TM5C1	TM5C0		8	70	8-22
007C	General-purpose 8-bit timer 6 counter	TM6C	—										8	Undefined	8-28
007D	General-purpose 8-bit timer 6 register	TM6R	—										8/16	Undefined	8-28
007E	General-purpose 8-bit timer 6 control register	TM6CON	(*5)	MODWDT	WDTLDE	WDTRUN	(1)	ATMRUN	WDTC2	WDTC1	WDTC0		8	10	8-29
0084	SIO1 transmit control register	ST1CON	—	TR1NIE	TR1MIE	ST1ODD	ST1PEN	ST1STB / ST1SLV	(1)	ST1LN	ST1MOD		8	04	11-16
0085	SIO1 receive control register	SR1CON	—	SR1REN	RC1IE	SR1ODD	SR1PEN	SR1SLV	S1EXC	SR1LN	SR1MOD		8	00	11-18
0086	SIO1 transmit-receive buffer register	S1BUF	—										8	Undefined	11-22
0087	SIO1 status register	S1STAT	—	(0)	(0)	RC1END	TR1END	TR1EMP	PERR1	OERR1	FERR1		8	00	11-20
008A	SIO3 control register	SIO3CON	—	(1)	SFT3CT2	SFT3CT1	SFT3CT0	S3BUSY	SFT3SLV	SFT3CK1	SFT3CK0		8	80	11-41
008B	SIO3 register	SIO3R	—										8	Undefined	11-43
008C	SIO4 control register	SIO4CON	—	(1)	BUSY4	ICK4	TEN4	SIO4SL	SIO4C2	SIO4C1	SIO4C0		8	80	11-48
008D	FIFO control register	FIFOCON	(*6)	(1)	(1)	(1)	(1)	SRES4	ORE4	FUL4	EMP4		8	F1	11-50

Table 22-1 SFR List (4/5)

Address [H]	Function	Byte	Word	Bit Symbol 7	6	5	4	3	2	1	0	R/W	8/16	Initial value [H]	Reference page
008E	SIO4 serial input FIFO data register	SIN4										R	8	Undefined	11-52
008F	SIO4 serial output FIFO data register	SOUT4										W	8	Undefined	11-52
0090	PWM register 0	PWR0	PWR01									R/W	8/16	00	10-4
0091	PWM register 1	PWR1												00	
0094	PWM cycle register 0	PWCY0	PWCY										8/16	00	10-3
0095	PWM cycle register 1	PWCY1												00	
0096	PWM counter 0	PWC0	PWC										8/16	00	10-3
0097	PWM counter 1	PWC1												00	
0098	PWM control register 0	PWCON0		PWC10V	PWCK11	PWCK10	PW1RUN	PWC0OV	PWCK01	PWCK00	PW0RUN		8	00	10-4
0099	PWM control register 1	PWCON1		(1)	(1)	(1)	(1)	(1)	(1)	(1)	PWHSM		8	FE	10-6
009D	A/D control register 0H	ADCON0H		ADTM02	ADTM01	ADTM00	STS0	0	0	ADSTM01	ADSTM00		8	00	12-3
009E	A/D interrupt control register 0	ADINT0		(1)	(1)	(1)	(1)	ADSTIE0	0	INSTST0	0		8	F0	12-5
00A0	A/D result register 00		ADR00									R	16	Undefined	12-6
00A1				(0)	(0)	(0)	(0)	(0)	(0)					Undefined	
00A2	A/D result register 01		ADR01										16	Undefined	12-6
00A3				(0)	(0)	(0)	(0)	(0)	(0)					Undefined	
00A4	A/D result register 02		ADR02										16	Undefined	12-6
00A5				(0)	(0)	(0)	(0)	(0)	(0)					Undefined	
00A6	A/D result register 03		ADR03										16	Undefined	12-6
00A7				(0)	(0)	(0)	(0)	(0)	(0)					Undefined	
00B8	Port 8 data register	P8		(0)	(0)	(0)	(0)	P8_3	P8_2	P8_1	P8_0	R/W	8	00	5-30
00B9	Port 9 data register	P9		(0)	(0)	(0)	(0)	(0)	(0)	(0)	P9_0		8	00	5-32
00BA	Port 10 data register	P10		(0)	(0)	P10_5	P10_4	P10_3	P10_2	P10_1	P10_0		8	00	5-34
00BC	Port 12 data register	P12		(0)	(0)	(0)	(0)	P12_3	P12_2	P12_1	P12_0	R	8	Undefined	5-37
00BD	Port 13 data register	P13		(0)	(0)	(0)	(0)	(0)	(0)	P13_1	P13_0		8	Undefined	5-38
00BF	Port 15 data register	P15		(0)	(0)	(0)	(0)	P15_3	P15_2	P15_1	P15_0		8	00	5-39
00C0	Port 8 mode register	P8IO		(0)	(0)	(0)	(0)	P8IO3	P8IO2	P8IO1	P8IO0		8	00	5-30
00C1	Port 9 mode register	P9IO		(0)	(0)	(0)	(0)	(0)	(0)	(0)	P9IO0		8	00	5-32
00C2	Port 10 mode register	P10IO		(0)	(0)	P10IO5	P10IO4	P10IO3	P10IO2	P10IO1	P10IO0		8	00	5-34
00C3	Port 15 mode register	P15IO		(0)	(0)	(0)	(0)	P15IO3	P15IO2	P15IO1	P15IO0	R/W	8	00	5-39
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		(0)	(0)	(0)	(0)	TXC1	RXC1	TXD1	P8SF0		8	00	5-30
								P8SF3	P8SF2	P8SF1					

Table 22-1 SFR List (5/5)

Address [H]	Function	Byte	Word	Bit Symbol 7	6	5	4	3	2	1	0	R/W	8/16	Initial value [H]	Reference page
00C9	Port 9 secondary function control register	P9SF	—	(0)	(0)	(0)	(0)	(0)	(0)	(0)	P9SF0	R/W	0	00	5-32
00CA	Port 10 secondary function control register	P10SF	—	(0)	(0)	P10SF5	SIOO4 P10SF4	SIOCK4 P10SF3	SIOO3 P10SF2	P10SF1	SIOCK3 P10SF0		8	00	5-34
00CC	General-purpose 8-bit timer 9 counter	TM9C	—										8	Undefined	8-37
00CD	General-purpose 8 bit timer 9 register	TM9R	—										8	Undefined	8-37
00CE	General-purpose 8-bit timer 9 control register	TM9CON	—	TM9OUT	(1)	(1)	(1)	TM9RUN	TM9C2	TM9C1	TM9C0		8	70	8-37
00D0	General-purpose 16-bit timer 7 counter	—	TM7C										16	Undefined	8-43
00D1		—												Undefined	
0064	General-purpose 16-bit timer 7 register	—	TM7R										16	Undefined	8-43
0065		—												Undefined	
0066	General-purpose 16-bit timer 7 control register	TM7CON	—	TM7OUT	(1)	(1)	(1)	TM7RUN	TM7C2	TM7C1	TM7C0		8	70	8-43
☆ 00F0	Flash memory acceptor	FLAACP	—									W	8	"0"	20-9
☆ 00F1	Flash memory control register	FLACON	(*7)	MCPMOD	(1)	(1)	FCLK1	FCLK0	(1)	(1)	PRG	R/W	8	66/E6	20-9
☆ 00F2	Flash memory address register	—	FLAADRS	FA11	FA10	FA9	FA8	FA7	(1)	(1)	(1)		16	Undefined	20-8
☆ 00F3		—		(1)	(1)	(1)	FA16	FA15	FA14	FA13	FA12				
00F4	SIO6 transmit control register	ST6CON	—	TR6NIE	TR6MIE	ST6ODD	ST6PEN	ST6STB SR6SLV	(1)	ST6LN	ST6MOD		8	04	11-4
00F5	SIO6 receive control register	SR6CON	—	SR6REN	RC6IE	SR6ODD	SR6PEN	SR6SLV	S6EXC	SR6LN	SR6MOD		8	00	11-6
00F6	SIO6 transmit-receive buffer register	S6BUF	—										8	Undefined	11-10
00F7	SIO6 status register	S6STAT	—	(0)	(0)	RC6END	TR6END	TR6EMP	PERR6	OERR6	FERR6		8	00	11-8

The meanings other than symbols in the Bit Symbol column are described below.

Blank
: An I/O function register is assigned to this bit. However, a bit symbol is not needed for this bit because this bit does not have to be accessed in bits or cannot be accessed in bits.

(1) An I/O function register is not assigned to this bit. The write operation is ignored. "1" is always read from this bit.

(0) Any write operation to this bit is ignored. "0" is always read from this bit. "0" is read from bits 10 to 15 of ADR00 to ADR07.

1 Be sure to write "1" to this bit.

0 Be sure to write "0" to this bit.

Initial value [H]
: This value indicates the content of each SFR at reset (RESn signal input, BRK instruction execution, overflow of watchdog timer, or opcode trap).
This value is expressed in hexadecial numbers.

(*1) Be sure to write "1" to bits 5 and 4 of ROMWIN.

(*2) Be sure to write "0" to bits 6 to 4 of ROMRDY.

(*3) Each flag of P5IO has the following functions. This is not the mode register for I/O switching of port 5.
FSEL1 (bit 7), FSEL0 (bit 6):
These are flags for multiplication selection of PLL in the USB macro.
(0, 0) = x4, (0, 1) = x8, (1, 0) = x6, (1, 1) = prohibited
SCK4IO (bit 5):
This is an I/O switching flag for P10_3/SIOCK4 pin. It is also required to do I/O switching through P10IO3. "0" = input, "1" = output
SCK4INV (bit 4):
This is a polarity selection flag for P10_3/SIOCK4 pin. "0" = positive, "1" = negative
STBSEL (bit 3):
This is an 80-system/68-system selection flag for secondary function output of P3_2/RDn pin.
"0" = 80-system, "1" = 68-system
MECKEN (bit 2):
This is a clock supply selection signal for Media control block.
"0" = clock supply, "1" = clock stop
UDCKN (bit 1):
This is a clock supply selection signal for USB DMA controller.
"0" = clock supply, "1" = clock stop
USBRST (bit 0):
This is a hardware reset control signal for USB macro.
"0" = USB reset, "1" = USB reset released

(*4) Bit 6 (NMIRD) of EXI2CON is a read-only bit. The write operation to this bit is ignored. "0" is read from this bit.

(*5) Bits 6 (WDTLDE) and 5 (WDTRUN) of TM6CON are read-only bits.
The write operation to these bits are ignored.

(*6) Bit 2 (ORE4), bit 1 (FUL4) and bit 0 (EMP4) of FIFOCON are read only.
The write operation to these bits is ignored.

(*7) Bit 7 (MCPMOD) is of FLACON is a read-only bit.
The write operation to this bit is ignored.
"0" is read for Mask ROM version and "1" is for Flash-ROM version.

☆ This register is used only for Flash-ROM version (ML66Q525A). For details, refer to Chapter 20, "Flash Memory".

Table 22-2 SFRs for Expanded RAM Area (1/4)

Address [H]	Function	Byte	Word	Bit Symbol 7	6	5	4	3	2	1	0	R/W	8/16	Initial value [H]	Reference page
1200 to 13FF	Buffer RAM Bank 0												8-16	Undefined	2-14
1400 to 15FF	Buffer RAM Bank 1											R/W	8/16	Undefined	2-14
1600 to 17FF	Buffer RAM Bank 2												8/16	Undefined	2-14
1800 to 19FF	Buffer RAM Bank 3												8/16	Undefined	2-14
1A00	bmRequest type setup register	bmRequestType											8	00	17-26
1A01	bRequest setup register	bRequest											8	00	17-26
1A02	wValueLSB setup register	wValueLSB											8	00	17-27
1A03	wValueMSB setup register	wValueMSB										R	8	00	17-27
1A04	wIndexLSB setup register	wIndexLSB											8	00	17-28
1A05	wIndexMSB setup register	wIndexMSB											8	00	17-28
1A06	wLengthLSB setup register	wLengthLSB											8	00	17-29
1A07	wLengthMSB setup register	wLengthMSB											8	00	17-29
1A10	DMA0 control register	DMA0CON											8	00	17-30
1A11	DMA0 interval register	DMA0INTVL										R/W	8	00	17-31
1A12	DMA1 control register	DMA1CON											8	00	17-30
1A13	DMA1 interval register	DMA1INTVL											8	00	17-31
1A20	Device address register	DVCADR		(0)									8	00	17-32
1A21	Interrupt status register 1	INTSTAT1										R	8	00	17-33
1A22	Interrupt status register 2	INTSTAT2		(0)	(0)	(0)							8	00	17-34
1A24	Interrupt enable register 1	INTENBL1										R/W	8	01	17-35
1A25	Interrupt enable register 2	INTENBL2		(0)	(0)	(0)							8	00	17-36
1A2D	Frame number LSB register	FRAMELSB										R	8	00	17-37
1A2E	Frame number MSB register	FRAMEMSB		(0)	(0)	(0)	(0)	(0)					8	00	17-37
1A2F	System control register	SYSCON		(0)								R/W	8	00	17-38
1A30	Polarity selection register	POLSEL		(0)	(0)	(0)							8	00	17-39
1A39	Packet error register 1	PKTERR		(0)	(0)	(0)	(0)					R	8	00	17-55
1A3A	OSC test	OSCTEST		(0)	(0)	(0)	(0)	(0)					8	00	17-56
1A40	EP0 configuration register	EP0CONF		(0)	(0)	(0)		(0)	(0)	(0)	(0)	R/W	8	00	17-40
1A41	EP1 configuration register	EP1CONF			(0)	(0)		(0)	(0)				8	00	17-41
1A42	EP2 configuration register	EP2CONF			(0)	(0)		(0)	(0)				8	00	17-41

Table 22-2 SFRs for Expanded RAM Area (2/4)

Address [H]	Function	Byte	Word	Bit Symbol 7	6	5	4	3	2	1	0	R/W	8/16	Initial value [H]	Reference page
1A43	EP3 configuration register	EP3CONF			(0)	(0)		(0)	(0)			R/W	8	00	17-41
1A44	EP4 configuration register	EP4CONF			(0)	(0)		(0)	(0)				8	00	17-42
1A45	EP5 configuration register	EP5CONF			(0)	(0)		(0)	(0)				8	00	17-42
1A48	EP0 control register	EP0CONT		(0)	(0)	(0)		(0)	(0)				8	Undefined	17-43
1A49	EP1 control register	EP1CONT		(0)	(0)	(0)	(0)						8	00	17-44
1A4A	EP2 control register	EP2CONT		(0)	(0)	(0)	(0)						8	00	17-44
1A4B	EP3 control register	EP3CONT		(0)	(0)	(0)	(0)						8	00	17-44
1A4C	EP4 control register	EP4CONT		(0)	(0)	(0)	(0)						8	00	17-44
1A4D	EP5 control register	EP5CONT		(0)	(0)	(0)	(0)						8	00	17-44
1A50	EP0 payload register	EP0PLD		(0)	(0)	(1)	(0)	(0)	(0)	(0)	(0)	R	8	20	17-45
1A51	EP1 payload register	EP1PLD		(0)									8	00	17-45
1A52	EP2 payload register	EP2PLD		(0)								R/W	8	00	17-45
1A53	EP3 payload register	EP3PLD		(0)	(0)								8	00	17-46
1A54	EP4 payload register	EP4PLD											8	00	17-47
1A55	EP5 payload register	EP5PLD											8	00	17-47
1A58	EP0 receive byte count register	EP0RXCNT		(0)	(0)								8	00	17-48
1A59	EP1 receive byte count register	EP1RXCNT		(0)									8	00	17-48
1A5A	EP2 receive byte count register	EP2RXCNT		(0)									8	00	17-48
1A5B	EP3 receive byte count register	EP3RXCNT		(0)	(0)								8	00	17-49
1A5C	EP4 receive byte count register	EP4RXCNT										R	8	00	17-49
1A5D	EP5 receive byte count register	EP5RXCNT											8	00	17-49
1A60	EP0 status register	EP0STAT		(0)	(0)			(0)					8	00	17-50
1A61	EP1 status register	EP1STAT		(0)	(0)	(0)	(0)	(0)	(0)				8	00	17-53
1A62	EP2 status register	EP2STAT		(0)	(0)	(0)	(0)	(0)	(0)				8	00	17-53
1A63	EP3 status register	EP3STAT		(0)	(0)	(0)	(0)	(0)	(0)				8	00	17-54
1A64	EP4 status register	EP4STAT		(0)	(0)	(0)	(0)	(0)	(0)			R	8	00	17-53
1A65	EP5 status register	EP5STAT		(0)	(0)	(0)	(0)	(0)	(0)				8	00	17-53
1A70	EP0 transmit FIFO	EP0TXFIFO										W	8	00	17-22
1A79	EP0 receive FIFO	EP0RXFIFO										R	8	Undefined	17-22
1A79	EP1 transmit/receive FIFO	EP1FIFO											8	Undefined	17-23
1A7A	EP2 transmit/receive FIFO	EP2FIFO											8	Undefined	17-23
1A7B	EP3 transmit/receive FIFO	EP3FIFO										R/W	8	Undefined	17-24
1A7C	EP4 transmit/receive FIFO	EP4FIFO											8	Undefined	17-24
1A7D	EP5 transmit/receive FIFO	EP5FIFO											8	Undefined	17-25

Table 22-2 SFRs for Expanded RAM Area (3/4)

Address [H]	Function	Byte	Word	Bit Symbol 7	6	5	4	3	2	1	0	R/W	8/16	Initial value [H]	Reference page
1A80	Channel 0 current address register		CH0 ADDRESS										16	0000	19-3
1A81				(0)	(0)	(0)	(0)	(0)	(0)	(0)					
1A82	Channel 1 current address register		CH1 ADDRESS										16	0000	19-3
1A83				(0)	(0)	(0)	(0)	(0)	(0)	(0)					
1A84	Channel 0 current word count register		CH0 WDCNT									R/W	16	0000	19-4
1A85				(0)	(0)	(0)	(0)	(0)	(0)						
1A86	Channel 1 current word count register		CH1 WDCNT										16	0000	19-4
1A87				(0)	(0)	(0)	(0)	(0)	(0)						
1A88	Mode register	MODE		(0)	(0)	(0)	(0)						8	00	19-5
1A8A	Channel 0 mask register	CH0MSK		(0)	(0)	(0)	(0)	(0)	(0)	(0)			8	01	19-6
1A8C	Channel 1 mask register	CH1MSK		(0)	(0)	(0)	(0)	(0)	(0)	(0)			8	01	19-6
1A90	Interrupt status register	INTSTAT			(0)	(0)	(0)					R	8	00	19-7
1A92	Interrupt enable register	INTENBL			(0)	(0)	(0)					R/W	8	00	19-8
1A94	DREQ monitor register	DREQMON		(0)	(0)	(0)	(0)	(0)	(0)			R	8	Undefined	19-9
1AA0	Option register	OPTION		(0)	(0)	(0)	(0)	(0)	(0)	(0)			8	00	19-10
1AA2	Channel 0 packet size register	CH0PKTSZ		(0)									8	00	19-11
1AA4	Channel 1 packet size register	CH1PKTSZ		(0)								R/W	8	00	19-11
1AA6	Channel 0 maximum packet size register	CH0MXPKTSZ		(0)									8	00	19-12
1AA8	Channel 1 maximum packet size register	CH1MXPKTSZ		(0)									8	00	19-12
1AB0	Channel 0 first byte detection counter	CH0RXCNT		(0)								W	8	00	19-13
1AB2	Channel 1 first byte detection counter	CH1RXCNT		(0)									8	00	19-13
1B00	USB bank register	UBANK		(0)	(0)	(0)	(0)	(0)	(0)				8	00	19-14
1B02	Media bank register	MBANK		(0)	(0)	(0)	(0)	(0)	(0)			R/W	8	00	19-15
1B04	Media sequencer control register	MSCTRL		PARITY	ECC	HEAD1	HEAD0	DLEN1	DLEN0	DDIR	START		8	00	18-3
1B06	Media sequencer wait register	MSWAIT		(0)	(0)	(0)	0	WAITW2	WAITW1	WAITR2	WAITR1		8	00	18-7
1B08	Media sequencer status register	MSSTS		(0)	(0)	(0)	MRDY	PRTYOK	ECCOK	MSRDY	MSCOMP	R	8	1E	18-8
1B0A	Media sequencer error status register	MSERR		(0)	(0)	PAR2ERR	PAR1ERR	ECC2COR	ECC2UNC	ECC1COR	ECC1UNC		8	00	18-9
1B0C	Media command register	MMCMD										W	8		18-10
1B0E	Media address register	MMADR											8		18-10
1B10	Media data register	MMDATA										R/W	8	FF	18-11
1B12	Media select register	MMSEL		(0)	(0)	(0)	(0)	(0)	(0)	FALE	FCLE		8	00	18-11
1B14	ECC1 line parity register	ECC1LP										R/W	16	FFFF	18-12
1B15															

Table 22-2 SFRs for Expanded RAM Area (4/4)

Address [H]	Function	Byte	Word	Bit Symbol 7	6	5	4	3	2	1	0	R/W	8/16	Initial value [H]	Reference page
1B16	ECC2 line parity register		ECC2LP									R	16	FFFF	18-12
1B17															
1B18	ECC1 column parity register		ECC1CP	(0)	(0)								16	003F	18-13
1B19				(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)				
1B1A	ECC2 column parity register		ECC2CP	(0)	(0)								16	003F	18-13
1B1B				(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)				
1B1C	ECC1 error pointer register		ECC1ERR										16	0000	18-14
1B1D				(0)	(0)	(0)	(0)	(0)							
1B1E	ECC2 error pointer register		ECC2ERR										16	0000	18-14
1B1F				(0)	(0)	(0)	(0)	(0)							
1B20	Redundancy part reserved data 1 register	FHD0	HREV1									R/W	8/16	0000	18-15
1B21		FHD1													
1B22	Redundancy part reserved data 2 register	FHD2	HREV2										8/16	0000	18-15
1B23		FHD3													
1B24	Redundancy part data/block status register	FHD4	HSTATS									R/W	8/16	0000	18-16
1B25		FHD5													
1B26	Redundancy part block address 1 register	FHD6	HBADR1										8/16	0000	18-17
1B27		FHD7													
1B28	Redundancy part ECC2-High register	FHD8	HECC2H										8/16	0000	18-18
1B29		FHD9													
1B2A	Redundancy part ECC2-Low/block address 2 register	FHD10	HECC2LA										8/16	0000	18-19
1B2B		FHD11													
1B2C	Redundancy part ECC1-High/block address 2 register	FHD12	HECC1HA										8/16	0000	18-20
1B2D		FHD13													
1B2E	Redundancy part ECC1-Low register	FHD14	HECC1L										8/16	0000	18-21
1B2F		FHD15													
1B80	Port 20 data register	P20		P20_7	P20_6	P20_5	P20_4	P20_3	P20_2	P20_1	P20_0		8	00	5-41
1B81	Port 21 data register	P21		(0)	(0)	(0)	P21_4	P21_3	P21_2	P21_1	P21_0		8	00	5-43
1B82	Port 20 mode register	P20IO		P20IO7	P20IO6	P20IO5	P20IO4	P20IO3	P20IO2	P20IO1	P20IO0		8	00	5-41
1B83	Port 21 mode register	P21IO		(0)	(0)	(0)	P21IO4	P21IO3	P21IO2	P21IO1	P21IO0		8	00	5-43
1B84	Port 20 secondary function control register	P20SF		P20SF7	P20SF6	P20SF5	P20SF4	P20SF3	P20SF2	P20SF1	P20SF0		8	00	5-41
1B85	Port 21 secondary function control register	P21SF		(0)	(0)	(0)	P20SF4	P20SF3	P20SF2	P20SF1	P20SF0		8	00	5-43

[Note]

With respect to the 2 KByte area of buffer RAM assigned to 1200H to 19FFH, do not read the banks specified by the USB bank register and the media bank register (write operations are invalid).

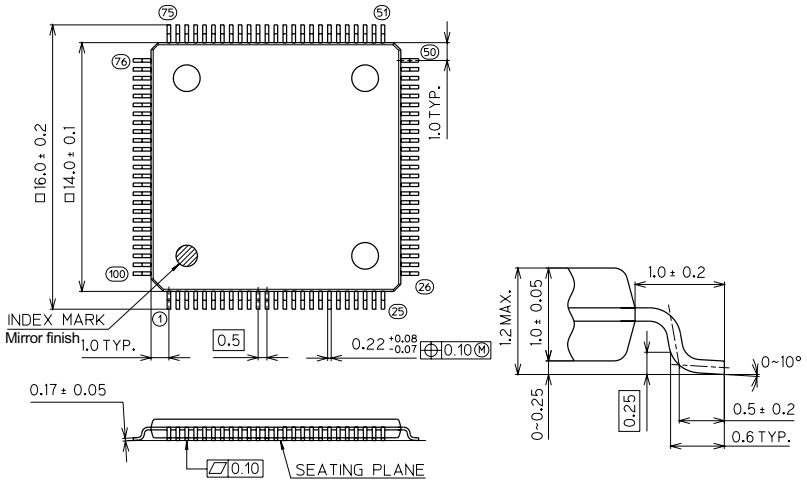
Chapter 23

Package Dimensions

23. Package Dimensions

(Unit: mm)

TQFP100-P-1414-0.50-K



Package material	Epoxy resin
Lead frame material	42 alloy
Pin treatment	Solder plating (≥5µm)
Package weight (g)	0.55 TYP.
Rev. No./Last Revised	4/Oct. 28, 1996

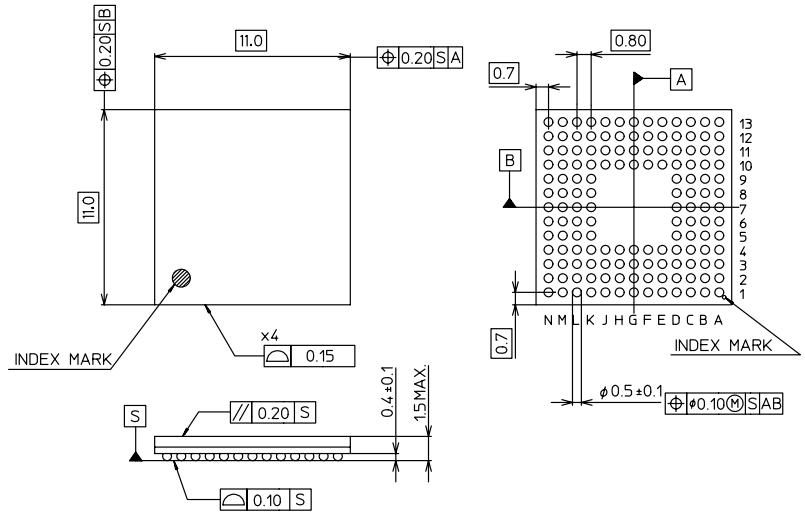
Oki Electric Industry Co., Ltd.

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).

(Unit: mm)

P-LFBGA144-1111-0.80



Package material	Epoxy resin
Ball material	Sn/Pb
Package weight (g)	0.30 TYP.
Rev. No./Last Revised	1/Aug. 25, 1999

Oki Electric Industry Co., Ltd.

Notes for Mounting the Surface Mounting Type Package

The P-LFBGA 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).

Chapter 24

Revision History

Revision History

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
PEUL66525-01	Dec. 2000	–	–	Preliminary edition 1
PEUL66525-02	May 2001	–	–	Preliminary edition 2
FEUL66525-01	Feb. 8, 2002	–	–	Final Edition
		1-1	1-1	Changed the device name from ML66Q525 to ML66Q525A.
		1-1	1-1	Made a change of description in Sec. 1.2.
		1-4	1-4	Made changes in Table 1-1.
		1-10	1-10	Made a few corrections in Table 1-1.
		1-10	1-10	Corrected the note contents.
		1-12	1-12	Changed the pin symbol V_{DD} to V_{DD}_CORE.
		3-1	3-1	Partly changed the note contents in Sec. 3.2.
		3-2	3-2	Added a few items to Table 3-1.
		3-2	3-2	Corrected a reference number in Table 3-1.
		3-5	3-5	Changed the note contents.
		3-11	3-11	Made a change to the description of reset operation.
		3-11	3-11	Made a few corrections in Fig. 3-6.
		3-11	3-11	Added a description to the note for Fig. 3-6.
		4-7	4-7	Added a note.
		5-2	5-2	Put an asterisk against pin symbol P3_2.
		5-2	5-2	Made a change of I/O attribute.
		5-2	5-2	Added a note.
		6-3	6-3	Added Table 6-2 and notes.
		6-3	6-4	Moved Fig. 6-3 and Notes to page 6-4.
		6-4	6-4	Partly corrected the text.
		6-5	6-5	Modified Fig. 6-6 and partly changed the note contents.
		7-1	7-1	Made a few corrections to the markings in Fig. 7-1.
		7-3	7-3	Made a minor correction in Fig. 7-3.
		10-9	10-9	Made changes in Sec. 10.4.1.
		10-10	10-10	Made changes in Sec. 10.4.2.
		11-29	11-29	Added one item to the notes.

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL66525-01	Feb. 8, 2002	11-36	11-36	Added a note.
		11-47	11-47	Added a description to Sec. 11.8.
		11-49	11-49	Added a note.
		11-50	11-50	Partly changed the contents of item (2).
		12-3	12-3	Made changes in the "Description of each bit".
		12-4	12-4	Made corrections to Fig. 12-2.
		12-6	12-6	Made changes in item (2).
		12-7	12-7	Made changes in Sec. 12.5.1.
		12-8	12-8	Partly changed the contents of item 2).
		13-5	13-5	Changed the device name from ML66525 to ML66525A.
		14-1	14-1	Made changes in Sec. 14.1.
		14-3	14-3	Made changes in Fig. 14-2.
		14-3	14-3	Added notes.
		14-4	14-4	Modified Fig. 14-3.
		14-5	14-5	Added a note.
		15-10	15-10	Partly changed the contents of item (2).
		15-11	15-11	Made a minor correction to Fig. 15-3.
		16-2	16-2	Added two note items.
		16-3	16-3	Modified the ground symbol.
		16-4	16-4	Made a minor correction to Fig. 16-2.
		17-3	17-3	Made changes in Sec. 17.3.3.
		17-5	17-5	Made a few corrections in Sec. 17.3.5.
		17-7	17-7	Partly corrected the contents of item (1).
		17-8	17-8	Made a few corrections in item (1).
		17-12	17-12	Corrected a description in the table.
		17-15	17-15	Made changes in Sec. 17.3.10.
		17-16	17-16	Inserted an indicator bar (missing in the previous ed.) in the table.
		17-16	17-16	Made corrections in item (1).
		17-16	17-16	Added a note.
		17-17	17-17	Made corrections in item (2).
		17-17	17-17	Added a note.
		17-33	17-33	Made changes in item (20).
		17-38	17-38	Partly changed the contents of item (26).

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL66525-01	Feb. 8, 2002	17-42	17-42	Added an explanation to item (30).
		17-46	17-46	Partly corrected the contents of item (35).
		17-47	17-47	Partly corrected the contents of item (36).
		17-50	17-50	Made a few corrections in item (41).
		17-51	17-51	Partly corrected the contents of item (41).
		17-52	17-52	Partly corrected the description in the figure.
		17-53	17-53	Added two note items.
		17-53	17-53	Partly corrected the contents of item (42).
		17-54	17-54	Partly corrected the contents of item (43).
		17-56	17-56	Made changes in item (45).
		17-56	17-56	Added a note.
		17-57	17-57	Made changes in Sec. 17.5.2.
		17-58	17-58	Made a change of description in the figure.
		17-59	17-59	Made a change of description in each of the figures.
		17-59	17-59	Deleted an extra pin symbol (P9_0).
		18-4	18-4	Added a note.
		18-8	18-8	Added explanation to the note contents.
		18-22	18-22	Corrected a symbol in Fig. 18-30.
		18-23	18-23	Corrected a symbol in Fig. 18-31 and 18-32.
		18-24	18-24	Corrected a symbol in Fig. 18-33 and 18-34.
		18-25	18-25	Corrected a symbol in Fig. 18-35.
		19-8	19-8	Added a note.
		20-1	20-1	Changed the device name from ML66Q525 to ML66Q525A.
		20-3	20-3	Changed the device name from ML66Q525 to ML66Q525A.
		20-3	20-3	Made changes in Sec. 20.4.2.
		20-5	20-5	Made changes in Fig. 20-3.
		20-5	20-5	Made changes to the note contents.
		20-6	20-6	Made changes in item (3).
		20-6	20-6	Added a note.
		20-15	20-15	Partly changed the contents of Sec. 20.6.5.
		20-15	20-15	Changed the contents of item (2) in Sec. 20.7.

Document No.	Date	Page		Description
		Previous Edition	Current Edition	
FEUL66525-01	Feb. 8, 2002	Title page of chap.	Title page of chap.	Deleted "(Preliminary)" from the title.
		21-1	21-1	Deleted "(Preliminary)" from the chapter title.
		21-2	21-2	Deleted "ML66525/Q525".
		21-3	21-3	Deleted "ML66525/Q525".
		21-4	21-4	Changed the device names ML66525 and ML66Q525 to ML66525A and ML66Q525A respectively.
		21-4	21-4	Made corrections in the note contents.
		21-6	21-6	Deleted "ML66525/Q525".
		21-7	21-7	Deleted "ML66525/Q525".
		21-8	21-8	Deleted "ML66525/Q525".
		21-9	21-9	Deleted "ML66525/Q525".
		21-10	21-10	Deleted "ML66525/Q525".
		21-11	21-11	Deleted "ML66525/Q525".
		21-12	21-12	Deleted "ML66525/Q525".
		22-3	22-3	Partly corrected the contents of the table.
		22-4	22-4	Partly corrected the contents of the table.
		22-5	22-5	Added a reference number.
		22-6	22-6	Partly corrected the contents of the table.
		22-7	22-7	Made a change of reference number in the table.
		22-9	22-9	Made changes in the note contents.
		22-9	22-9	Changed the device name from ML66Q525 to ML66Q525A.
		23-2	23-2	Corrected the product name.

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.