Description

The M30218 group of single-chip microcomputers are built using the high-performance silicon gate CMOS process using a M16C/60 Series CPU core and are packaged in a 100-pin plastic molded QFP. These single-chip microcomputers operate using sophisticated instructions featuring a high level of instruction efficiency. With 1M bytes of address space, they are capable of executing instructions at high speed. They also feature a built-in multiplier and DMAC, making them ideal for controlling musical instruments, household appliances and other high-speed processing applications.

The M30218 group includes a wide range of products with different internal memory types and sizes and various package types.

Features

- Basic machine instructions ..... Compatible with the M16C/60 series
- Memory capacity ..... ROM / RAM (See figure memory expansion)
- Shortest instruction execution time . 100ns (f(X_{IN})=10MHz)
- Supply voltage ..... 4.0V to 5.5V (f(X_{IN})=10MHz)
  2.7V to 5.5V (f(X_{IN})=3.5MHz)(Note)
- Interrupts ..... 19 internal and 6 external interrupt sources, 4 software
- Multifunction 16-bit timer ..... Timer A X 5, Timer B X 3
- FLD conrtoller ..... total 56 pins
  (high-breakdown-voltage P-channel open-drain output : 52pins)
- Serial I/O ..... 2 channels for UART or clock synchronous,
  1 channels for clock synchronous
  (max.256 bytes automatic transfer function)
- DMAC ..... 2 channels (triggers: 15 sources)
- A-D converter ..... 10 bits X 8 channels
- D-A converter ..... 8 bits X 2 channels
- CRC calculation circuit ..... 1 circuit
- Watchdog timer ..... 1 pin
- Programmable I/O ..... 48 pins
- High-breakdown-voltage output ..... 52 pins
- Clock generating circuit ..... 2 built-in clock generation circuit
  (built-in feedback resistor, and external ceramic or quartz oscillator)

Note: Only mask ROM version.

Applications

Household appliances, office equipment, Audio etc.

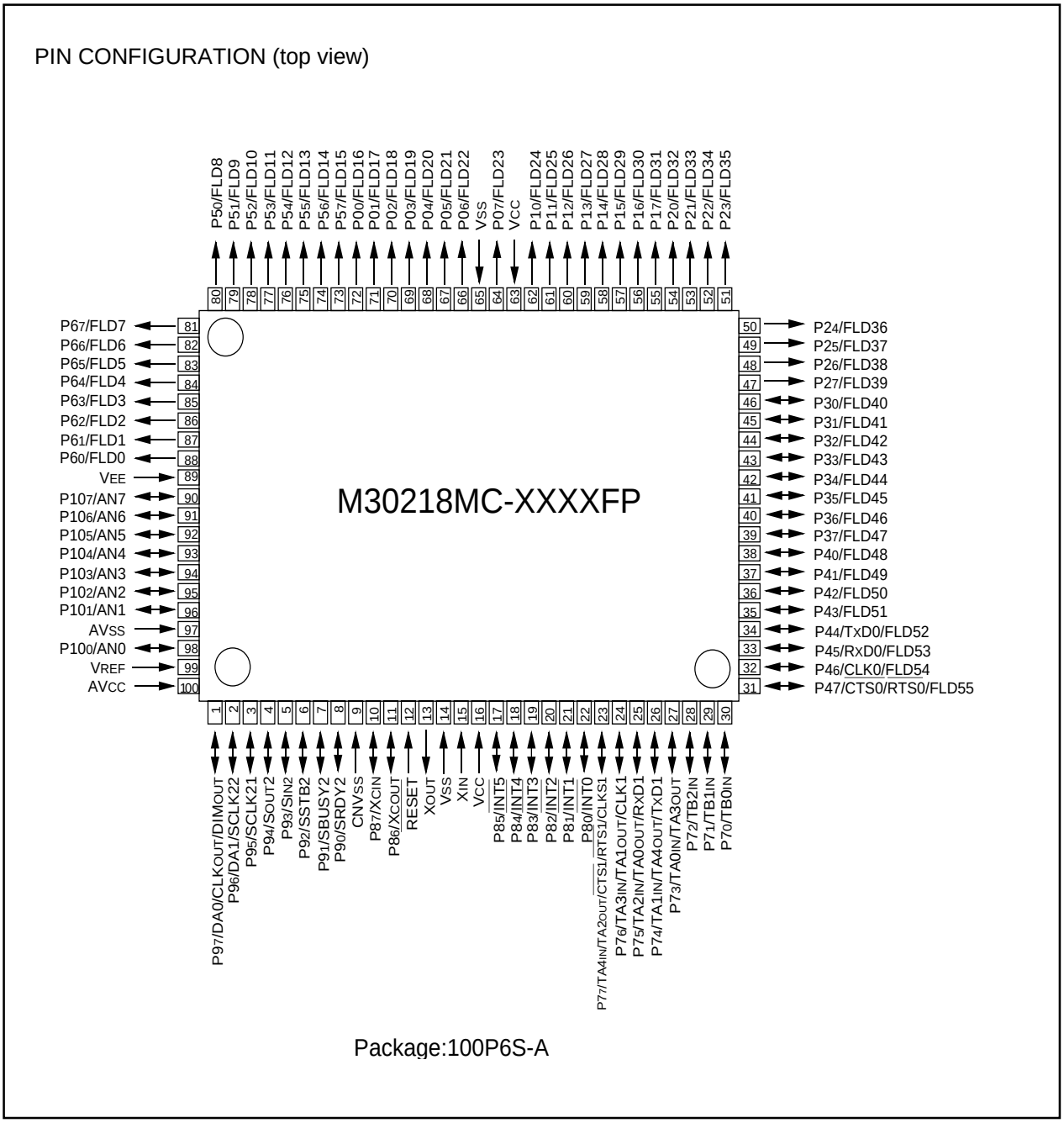
Specifications written in this manual are believed to be accurate, but are not guaranteed to be entirely free of error.
Specifications in this manual may be changed for functional or performance improvements. Please make sure your manual is the latest edition.

------Table of Contents------

Central Processing Unit (CPU) 10
Reset 14
Clock Generating Circuit 18
Protection 26
Interrupts 27
Watchdog Timer 45
DMAC 47
FLD controller 53
Timer 70
Serial I/O 87
A-D Converter 114
D-A Converter 124
CRC Calculation Circuit 126
Programmable I/O Ports 128
Flash memory version 152

Pin Configuration

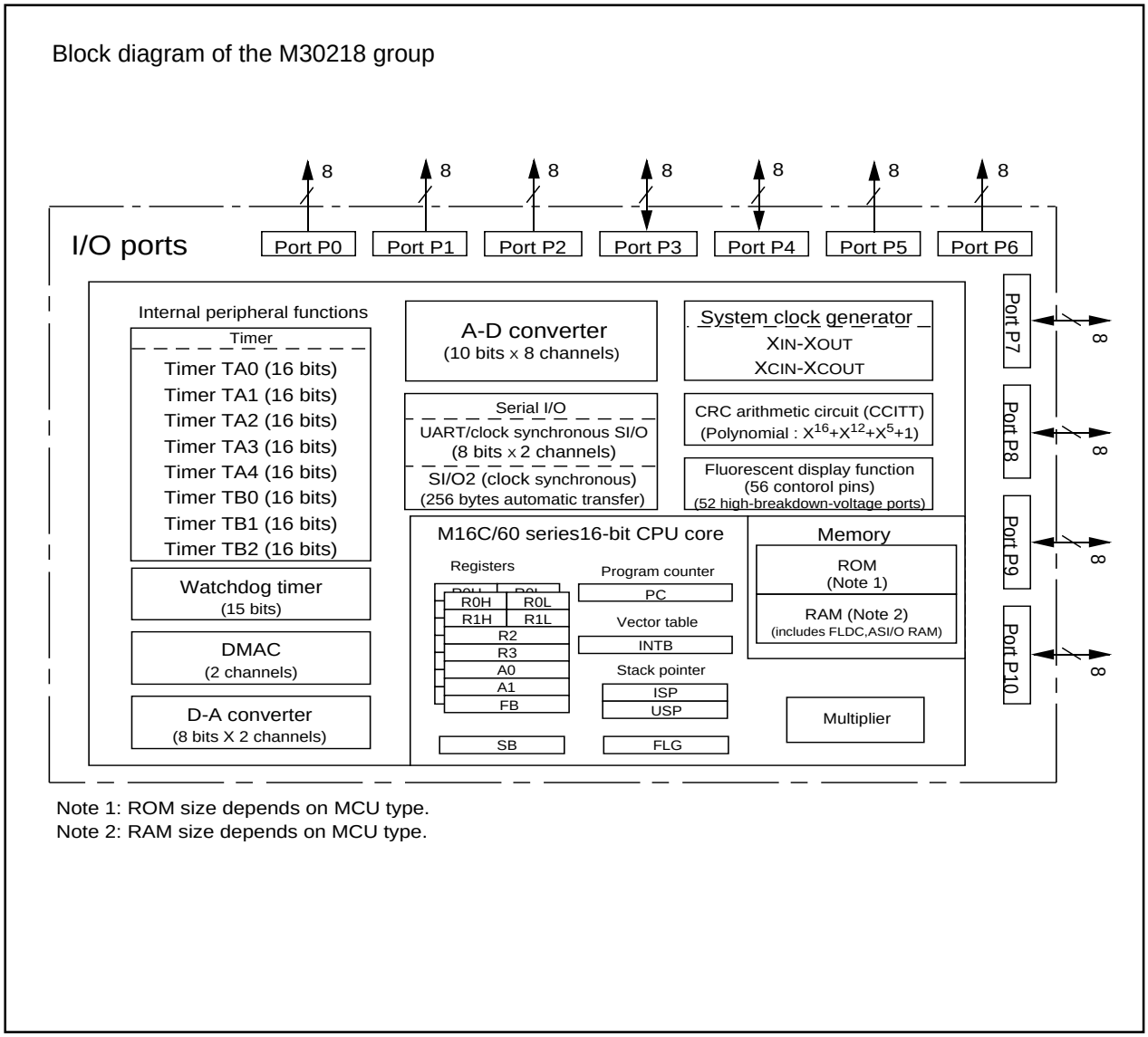
Figures AA-1 show the pin configurations (top view).



FigureAA-1. Pin configuration (top view)

Block Diagram

Figure AA-2 is a block diagram of the M30218 group.



Note 1: ROM size depends on MCU type.
Note 2: RAM size depends on MCU type.

FigureAA-2. Block diagram of M30218 group

Performance Outline

Table AA-1 is a performance outline of M30218 group.

Table AA-1. Performance outline of M30218 group

Item			Performance
Number of basic instructions			91 instructions
Shortest instruction execution time			100ns(f(XIN)=10MHz)
Memory capacity	ROM		See figure memory expansion
	RAM		See figure memory expansion
I/O port	P3, P4, P7 to P10		8 bits x 6
Output port	P0 to P2, P5, P6		8 bit x 5
Multifunction timer	TA0, TA1, TA2, TA3, TA4		16 bits x 5
	TB0, TB1, TB2		16 bits x 3
Serial I/O	UART0, UART1		(UART or clock synchronous) x 2
	SI/O2		(Clock synchronous) x 1 (with automatic transfer function)
Fluorescent display			56 pins
A-D converter			10 bits x 8 channels
D-A converter			8 bits x 2
DMAC			2 channels (triggers :15 sources)
CRC calculation circuit			1 circuit (polynomial: $X^{16} + X^{12} + X^5 + 1$)
Watchdog timer			15 bits x 1 (with prescaler)
Interrupt			19 internal and 6 external sources, 4 software sources, 7 levels
Clock generating circuit			2 built-in clock generation circuits (built-in feedback resistor, and external ceramic or quartz oscillator)
Supply voltage			4.0 to 5.5V (f(XIN)=10MHz) 2.7 to 5.5V (f(XIN)=3.5MHz) (Note)
Power consumption			18 mW (VCC=3V, f(XIN)=5MHz)
I/O characteristics	I/O withstand voltage		VCC-48V (output ports : P0 to P2, P5, P6, I/O ports : P3, P40 to P43)
			0 to VCC (I/O ports :P44 to P47, P7 to P10)
	Output current	H	- 18mA (P0 to P3, P40 to P43, P5, P6) :high-breakdown-voltage, P-channel open-drain
			- 5mA (P44 to P47, P7 to P10)
		L	5mA (P44 to P47, P7 to P10)
Operating ambient temperature			–20 to 85°C
Device configuration			CMOS silicon gate
Package			100-pin plastic mold QFP

Note: Only mask ROM version.

Description

Mitsubishi plans to release the following products in the M30218 group:

(1) Support for mask ROM version and flash memory version

(2) Memory capacity

(3) Package

100P6S : Plastic molded QFP (mask ROM version and flash memory version)

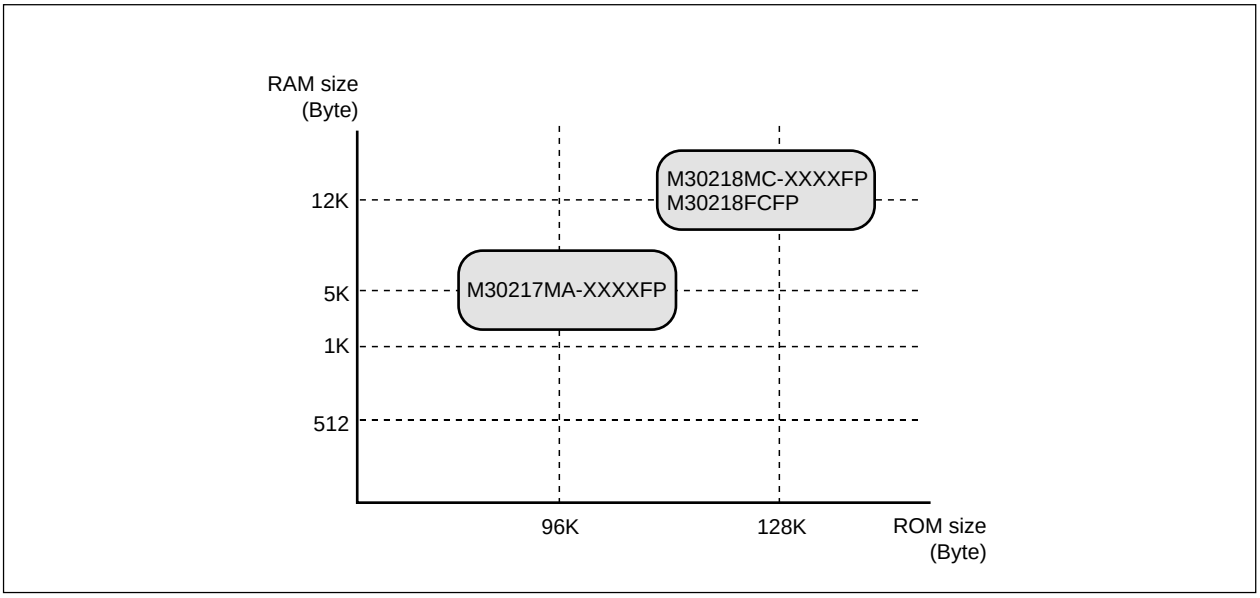


Figure AA-3. ROM expansion

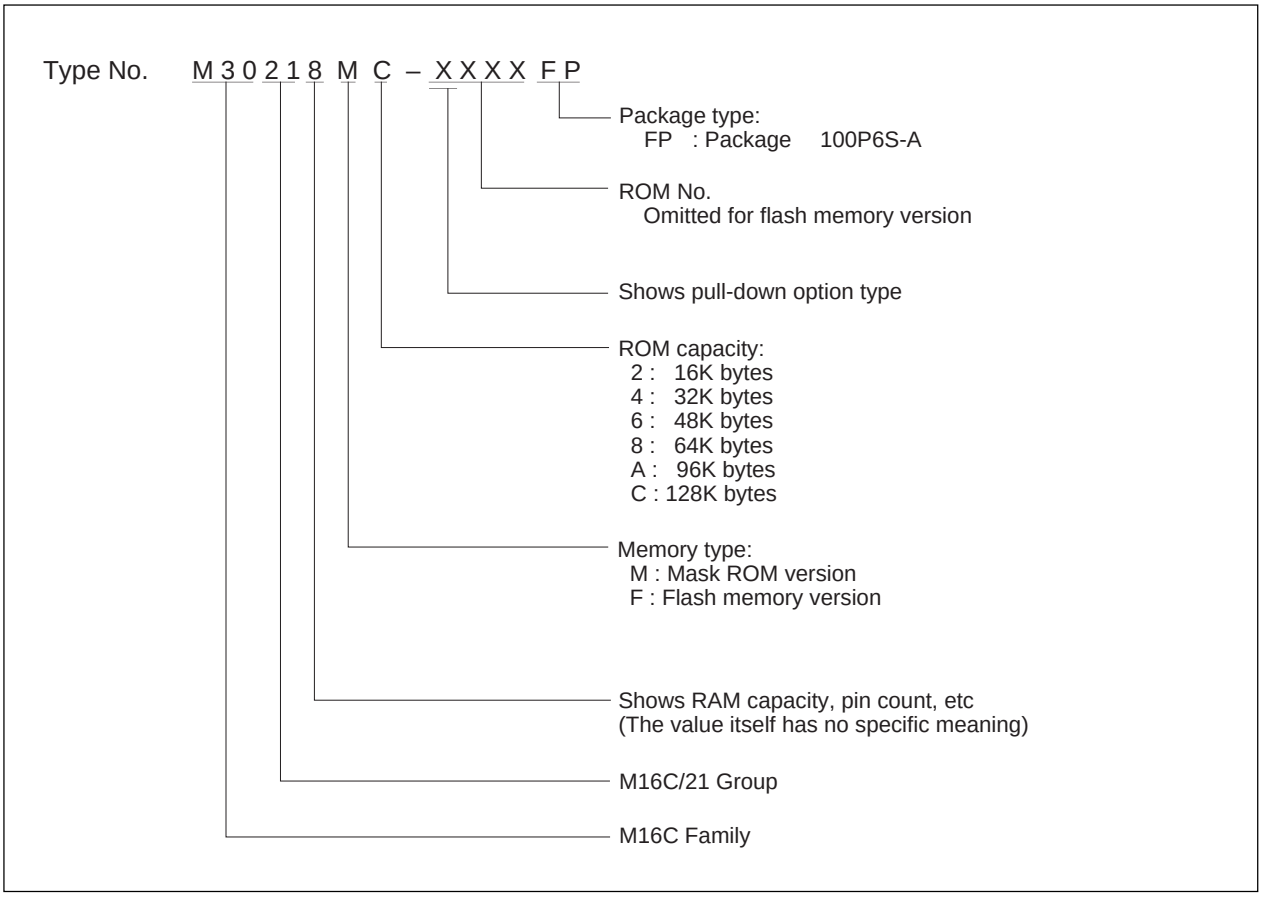


Figure AA-4. Type No., memory size, and package

Pin Description

Pin name	Signal name	I/O type	Function
V_{CC}, V_{SS}	Power supply input		Supply 2.7V(Note1) to 5.5 V to the V_{CC} pin. Supply 0 V to the V_{SS} pin. Connect a bypass capacitor across the V_{CC} pin and V_{SS} pin.
CNV$_{SS}$	CNV$_{SS}$	Input	Connect it to the V_{SS} pin.
$\overline{\text{RESET}}$	Reset input	Input	A “L” on this input resets the microcomputer.
X_{IN} X_{OUT}	Clock input Clock output	Input Output	These pins are provided for the main clock generating circuit.Connect a ceramic resonator or crystal between the X_{IN} and the X_{OUT} pins. To use an externally derived clock, input it to the X_{IN} pin and leave the X_{OUT} pin open.
AV$_{CC}$	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to V_{CC}.
AV$_{SS}$	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to V_{SS}.
V_{REF}	Reference voltage input	Input	This pin is a reference voltage input for the A-D converter.
V_{EE}	pull-down power source		Apply voltage supplied to pull-down resistors of ports P0 to P1,P5,P6.
$P0_0$/FLD_{16} to $P0_7$/FLD_{23}	Output port P0	Output	This is an 8-bit CMOS output port and high-breakdown-voltage P-channel open-drain output structure. A pull-down resistor is built in between port P0 and V_{EE} pin. At reset, this port is set to V_{EE} level. P0 function as FLD controller output pins as selected by software.
$P1_0$/FLD_{24} to $P1_7$/FLD_{31}	Output port P1	Output	This is an 8-bit output port equivalent to P0. Pins in this port also function as FLD controller output pins as selected by software.
$P2_0$/FLD_{32} to $P2_7$/FLD_{39}	Output port P2	Output	This is an 8-bit output port equivalent to P0. A pull-down resistor is not built in between P2 and V_{EE} pin (Note2). Pins in this port also function as FLD controller output pins as selected by software.
$P3_0$/FLD_{40} to $P3_7$/FLD_{47}	I/O port P3	Input/output	This is an 8-bit I/O port. A pull-down resistor is not built in between P3 and V_{EE} pin (Note2). It has an input/output port direction register that allows the user to set each pin for input or output. This is low-voltage input level, and high-breakdown-voltage P-channel open-drain output structure. Pins in this port also function as FLD controller output pins as selected by software.
$P4_0$/FLD_{48} to $P4_7$/FLD_{56}	I/O port P4	Input/output	This is an 8-bit I/O port equivalent to P3. This is low-voltage input level. $P4_0$ to $P4_3$ is high-breakdown-voltage P-channel open-drain output structure, $P4_4$ to $P4_7$ is CMOS output. A pull-down resistor is not built in between P4($P4_0$ to $P4_3$) and V_{EE} pin (Note2). Pins in this port also function as FLD controller output pins as selected by software. $P4_4$ to $P4_7$ also function as UART0 I/O pins as selected by software. When set for input, the user can specify in units of four bits by software whether or not they are tied to a pull-up resistor.
$P5_0$/FLD_8 to $P5_7$/FLD_{15}	Output port P5	Output	This is an 8-bit output port equivalent to P0. Pins in this port also function as FLD controller output pins as selected by software.
$P6_0$/FLD_0 to $P6_7$/FLD_7	Output port P6	Output	This is an 8-bit output port equivalent to P0. Pins in this port also function as FLD controller output pins as selected by software.

Pin Description

Pin name	Signal name	I/O type	Function
$P7_0$ to $P7_7$	I/O port P7	Input/output	This is an 8-bit I/O port equivalent to P3. This is CMOS input/output. When set for input, the user can specify in units of four bits by software whether or not they are tied to a pull-up resistor. $P7_0$ to $P7_2$ function as TimerB0 to B2 input pins as selected by software. $P7_3$ function as TimerA0 I/O pin as selected by software. $P7_4$ to $P7_7$ function as TimerA1 to A4 I/O pins, and UART1 I/O pins as selected by software.
$P8_0$ to $P8_7$	I/O port P8	Input/output	This is an 8-bit I/O port equivalent to P7. When set for input, the user can specify in units of four bits by software whether or not they are tied to a pull-up resistor. $P8_0$ to $P8_5$ function as external interrupt input pins as selected by software. $P8_6$,$P8_7$ function as sub-clock input pin as selected by software. In this case, connect a quarts oscillator between $P8_6$(X_{OUT} pin) and $P8_7$(X_{CIN} pin)
$P9_0$ to $P9_7$	I/O port P9	Input/output	This is an 8-bit I/O port equivalent to P7. When set for input, the user can specify in units of four bits by software whether or not they are tied to a pull-up resistor. $P9_7$ function as D-A converter output pins, clock output pins (same frequency of X_{IN}/8, X_{IN}/32 or X_{CIN}) and DIM signal output pin of FLD controller as selected by software. $P9_6$ function as D-A converter output pins and clock I/O pin of serial I/O with automatic transfer as selected by software. $P9_0$ to $P9_5$ function as I/O pin of serial I/O with automatic transfer as selected by software.
$P10_0$ to $P10_7$	I/O port P10	Input/output	This is an 8-bit I/O port equivalent to P7. When set for input, the user can specify in units of four bits by software whether or not they are tied to a pull-up resistor. Pins in this port also function as A-D converter input pins as selected by software.

Note 1: Supply 4.0V to 5.5V to the V_{CC} pin in flash memory version.
Note 2: Port $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, and $P4_0$ to $P4_3$ can be selected whether pull-down resistors are built-in or not by the mask option specification. Flash memory version does not have this option.

Operation of Functional Blocks

The M30218 group accommodates certain units in a single chip. These units include ROM and RAM to store instructions and data and the central processing unit (CPU) to execute arithmetic/logic operations. Also included are peripheral units such as timers, FLD controller, serial I/O, D-A converter, DMAC, CRC calculation circuit, A-D converter, and I/O ports.

The following explains each unit.

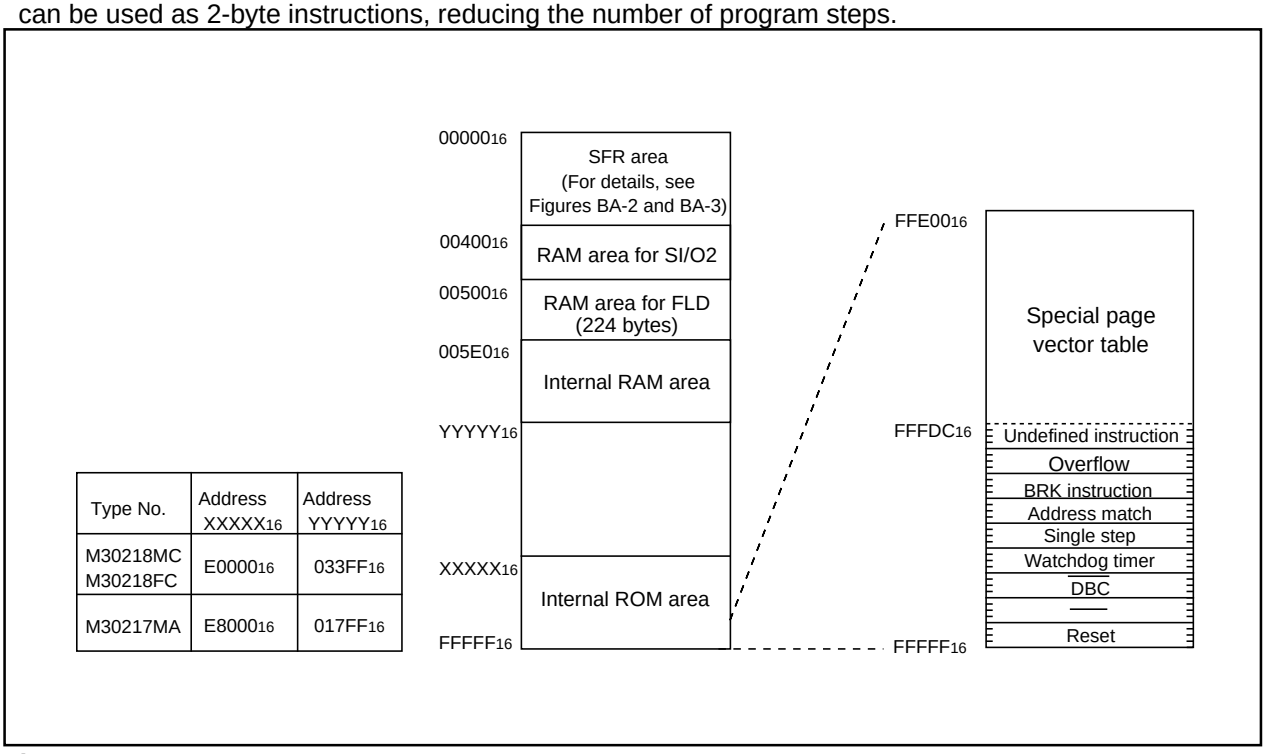
Memory

Figure BA-1 is a memory map of the M30218 group. The address space extends the 1M bytes from address 00000_{16} to $FFFFF_{16}$. From $FFFFF_{16}$ down is ROM. For example, in the M30218MC-XXXFP, there is 128K bytes of internal ROM from $E0000_{16}$ to $FFFFF_{16}$. The vector table for fixed interrupts such as the reset are mapped to $FFFDC_{16}$ to $FFFFF_{16}$. The starting address of the interrupt routine is stored here. The address of the vector table for timer interrupts, etc., can be set as desired using the internal register (INTB). See the section on interrupts for details.

From 00400_{16} up is RAM. For example, in the M30218MC-XXXFP, there is 12K bytes of internal RAM from 00400_{16} to $033FF_{16}$. In addition to storing data, the RAM also stores the stack used when calling subroutines and when interrupts are generated. (From 00400_{16} to $004FF_{16}$ is RAM for SIO2. From 00500_{16} to $005DF_{16}$ is RAM for FLD.)

The SFR area is mapped to 00000_{16} to $003FF_{16}$. This area accommodates the control registers for peripheral devices such as I/O ports, A-D converter, serial I/O, and timers, etc. Any part of the SFR area that is not occupied is reserved and cannot be used for other purposes.

The special page vector table is mapped to $FFE00_{16}$ to $FFFDB_{16}$. If the starting addresses of subroutines or the destination addresses of jumps are stored here, subroutine call instructions and jump instructions can be used as 2-byte instructions, reducing the number of program steps.



Type No.	Address XXXXX$_{16}$	Address YYYYY$_{16}$
M30218MC M30218FC	$E0000_{16}$	$033FF_{16}$
M30217MA	$E8000_{16}$	$017FF_{16}$

Figure BA-1. Memory map

0000₁₆	
0001₁₆	
0002₁₆	
0003₁₆	
0004₁₆	Processor mode register 0 (PM0)
0005₁₆	Processor mode register 1(PM1)
0006₁₆	System clock control register 0 (CM0)
0007₁₆	System clock control register 1 (CM1)
0008₁₆	
0009₁₆	Address match interrupt enable register (AIER)
000A₁₆	Protect register (PRCR)
000B₁₆	
000C₁₆	
000D₁₆	
000E₁₆	Watchdog timer start register (WDTS)
000F₁₆	Watchdog timer control register (WDC)
0010₁₆	Address match interrupt register 0 (RMAD0)
0011₁₆	
0012₁₆	
0013₁₆	
0014₁₆	Address match interrupt register 1 (RMAD1)
0015₁₆	
0016₁₆	
0017₁₆	
0018₁₆	
0019₁₆	
001A₁₆	
001B₁₆	
001C₁₆	
001D₁₆	
001E₁₆	
001F₁₆	
0020₁₆	DMA0 source pointer (SAR0)
0021₁₆	
0022₁₆	
0023₁₆	
0024₁₆	DMA0 destination pointer (DAR0)
0025₁₆	
0026₁₆	
0027₁₆	
0028₁₆	DMA0 transfer counter (TCR0)
0029₁₆	
002A₁₆	
002B₁₆	
002C₁₆	DMA0 control register (DM0CON)
002D₁₆	
002E₁₆	
002F₁₆	
0030₁₆	DMA1 source pointer (SAR1)
0031₁₆	
0032₁₆	
0033₁₆	
0034₁₆	DMA1 destination pointer (DAR1)
0035₁₆	
0036₁₆	
0037₁₆	
0038₁₆	DMA1 transfer counter (TCR1)
0039₁₆	
003A₁₆	
003B₁₆	
003C₁₆	DMA1 control register (DM1CON)
003D₁₆	
003E₁₆	
003F₁₆	

0040₁₆	
0041₁₆	
0042₁₆	
0043₁₆	
0044₁₆	
0045₁₆	
0046₁₆	
0047₁₆	$\overline{\text{INT3}}$ interrupt control register (INT3IC)
0048₁₆	$\overline{\text{INT4}}$ interrupt control register (INT4IC)
0049₁₆	$\overline{\text{INT5}}$ interrupt control register (INT5IC)
004A₁₆	
004B₁₆	DMA0 interrupt control register (DM0IC)
004C₁₆	DMA1 interrupt control register (DM1IC)
004D₁₆	
004E₁₆	A-D conversion interrupt control register (ADIC)
004F₁₆	SI/O automatic transfer interrupt control register (ASIOIC)
0050₁₆	FLD interrupt control register (FLDIC)
0051₁₆	UART0 transmit interrupt control register (S0TIC)
0052₁₆	UART0 receive interrupt control register (S0RIC)
0053₁₆	UART1 transmit interrupt control register (S1TIC)
0054₁₆	UART1 receive interrupt control register (S1RIC)
0055₁₆	Timer A0 interrupt control register (TA0IC)
0056₁₆	Timer A1 interrupt control register (TA1IC)
0057₁₆	Timer A2 interrupt control register (TA2IC)
0058₁₆	Timer A3 interrupt control register (TA3IC)
0059₁₆	Timer A4 interrupt control register (TA4IC)
005A₁₆	Timer B0 interrupt control register (TB0IC)
005B₁₆	Timer B1 interrupt control register (TB1IC)
005C₁₆	Timer B2 interrupt control register (TB2IC)
005D₁₆	$\overline{\text{INT0}}$ interrupt control register (INT0IC)
005E₁₆	$\overline{\text{INT1}}$ interrupt control register (INT1IC)
005F₁₆	$\overline{\text{INT2}}$ interrupt control register (INT2IC)

0340₁₆	Serial I/O2 automatic transfer data pointer (SIO2DP)
0341₁₆	
0342₁₆	Serial I/O2 control register 1 (SIO2CON1)
0343₁₆	
0344₁₆	Serial I/O2 control register 2 (SIO2CON2)
0345₁₆	
0346₁₆	Serial I/O2 register / transfer counter (SIO2)
0347₁₆	
0348₁₆	Serial I/O2 control register 3 (SIO2CON3)
0349₁₆	
034A₁₆	
034B₁₆	
034C₁₆	
034D₁₆	
034E₁₆	
034F₁₆	
0350₁₆	FLD mode register (FLDM)
0351₁₆	FLD output control register (FLDCON)
0352₁₆	Tdisp time set register (TDISP)
0353₁₆	
0354₁₆	Toff1 time set register (TOFF1)
0355₁₆	
0356₁₆	Toff2 time set register (TOFF2)
0357₁₆	
0358₁₆	FLD data pointer (FLDDP)
0359₁₆	P2 FLD/port switch register (P2FPR)
035A₁₆	P3 FLD/port switch register (P3FPR)
035B₁₆	P4 FLD/port switch register (P4FPR)
035C₁₆	P5 digit output set register (P5DOR)
035D₁₆	P6 digit output set register (P6DOR)
035E₁₆	
035F₁₆	

Figure BA-2. Location of peripheral unit control registers (1)

038016	Count start flag (TABSR)
038116	Clock prescaler reset flag (CPSRF)
038216	One-shot start flag (ONSF)
038316	Trigger select register (TRGSR)
038416	Up-down flag (UDF)
038516	
038616	Timer A0 (TA0)
038716	
038816	Timer A1 (TA1)
038916	
038A16	Timer A2 (TA2)
038B16	
038C16	Timer A3 (TA3)
038D16	
038E16	Timer A4 (TA4)
038F16	
039016	Timer B0 (TB0)
039116	
039216	Timer B1 (TB1)
039316	
039416	Timer B2 (TB2)
039516	
039616	Timer A0 mode register (TA0MR)
039716	Timer A1 mode register (TA1MR)
039816	Timer A2 mode register (TA2MR)
039916	Timer A3 mode register (TA3MR)
039A16	Timer A4 mode register (TA4MR)
039B16	Timer B0 mode register (TB0MR)
039C16	Timer B1 mode register (TB1MR)
039D16	Timer B2 mode register (TB2MR)
039E16	
039F16	
03A016	UART0 transmit/receive mode register (U0MR)
03A116	UART0 bit rate generator (U0BRG)
03A216	UART0 transmit buffer register (U0TB)
03A316	
03A416	UART0 transmit/receive control register 0 (U0C0)
03A516	UART0 transmit/receive control register 1 (U0C1)
03A616	UART0 receive buffer register (U0RB)
03A716	
03A816	UART1 transmit/receive mode register (U1MR)
03A916	UART1 bit rate generator (U1BRG)
03AA16	UART1 transmit buffer register (U1TB)
03AB16	
03AC16	UART1 transmit/receive control register 0 (U1C0)
03AD16	UART1 transmit/receive control register 1 (U1C1)
03AE16	UART1 receive buffer register (U1RB)
03AF16	
03B016	UART transmit/receive control register 2 (UCON)
03B116	
03B216	
03B316	
03B416	Flash memory control register 0 (FCON0) (Note)
03B516	Flash memory control register 1 (FCON1) (Note)
03B616	Flash command register (FCMD) (Note)
03B716	
03B816	DMA0 request cause select register (DM0SL)
03B916	
03BA16	DMA1 request cause select register (DM1SL)
03BB16	
03BC16	CRC data register (CRCD)
03BD16	
03BE16	CRC input register (CRCIN)
03BF16	

03C016	A-D register 0 (AD0)
03C116	
03C216	A-D register 1 (AD1)
03C316	
03C416	A-D register 2 (AD2)
03C516	
03C616	A-D register 3 (AD3)
03C716	
03C816	A-D register 4 (AD4)
03C916	
03CA16	A-D register 5 (AD5)
03CB16	
03CC16	A-D register 6 (AD6)
03CD16	
03CE16	A-D register 7 (AD7)
03CF16	
03D016	
03D116	
03D216	
03D316	
03D416	A-D control register 2 (ADCON2)
03D516	
03D616	A-D control register 0 (ADCON0)
03D716	A-D control register 1 (ADCON1)
03D816	D-A register 0 (DA0)
03D916	
03DA16	D-A register 1 (DA1)
03DB16	
03DC16	D-A control register (DACON)
03DD16	
03DE16	
03DF16	
03E016	Port P0 (P0)
03E116	Port P1 (P1)
03E216	
03E316	
03E416	Port P2 (P2)
03E516	Port P3 (P3)
03E616	
03E716	Port P3 direction register (PD3)
03E816	Port P4 (P4)
03E916	Port P5 (P5)
03EA16	Port P4 direction register (PD4)
03EB16	
03EC16	Port P6 (P6)
03ED16	Port P7 (P7)
03EE16	
03EF16	Port P7 direction register (PD7)
03F016	Port P8 (P8)
03F116	Port P9 (P9)
03F216	Port P8 direction register (PD8)
03F316	Port P9 direction register (PD9)
03F416	Port P10 (P10)
03F516	
03F616	Port P10 direction register (PD10)
03F716	
03F816	
03F916	
03FA16	
03FB16	
03FC16	
03FD16	Pull-up control register 0 (PUR0)
03FE16	Pull-up control register 1 (PUR1)
03FF16	

Note: This register is only exist in flash memory version.

Figure BA-3. Location of peripheral unit control registers (2)

Central Processing Unit (CPU)

The CPU has a total of 13 registers shown in Figure CA-1. Seven of these registers (R0, R1, R2, R3, A0, A1, and FB) come in two sets; therefore, these have two register banks.

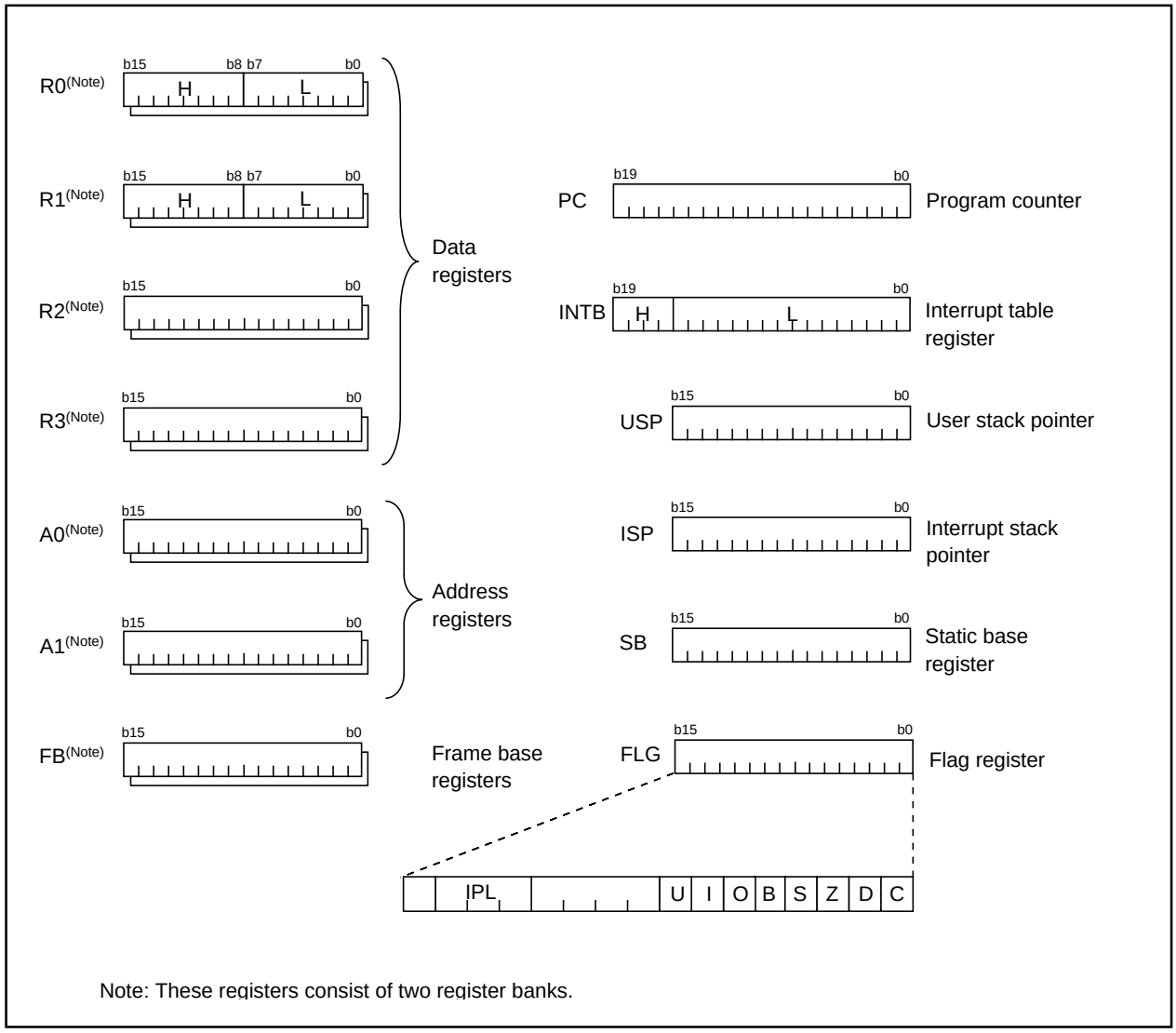


Figure CA-1. Central processing unit register

(1) Data registers (R0, R0H, R0L, R1, R1H, R1L, R2, and R3)

Data registers (R0, R1, R2, and R3) are configured with 16 bits, and are used primarily for transfer and arithmetic/logic operations.

Registers R0 and R1 each can be used as separate 8-bit data registers, high-order bits as (R0H, R1H), and low-order bits as (R0L, R1L). In some instructions, registers R2 and R0, as well as R3 and R1 can use as 32-bit data registers (R2R0, R3R1).

(2) Address registers (A0 and A1)

Address registers (A0 and A1) are configured with 16 bits, and have functions equivalent to those of data registers. These registers can also be used for address register indirect addressing and address register relative addressing.

In some instructions, registers A1 and A0 can be combined for use as a 32-bit address register (A1A0).

(3) Frame base register (FB)

Frame base register (FB) is configured with 16 bits, and is used for FB relative addressing.

(4) Program counter (PC)

Program counter (PC) is configured with 20 bits, indicating the address of an instruction to be executed.

(5) Interrupt table register (INTB)

Interrupt table register (INTB) is configured with 20 bits, indicating the start address of an interrupt vector table.

(6) Stack pointer (USP/ISP)

Stack pointer comes in two types: user stack pointer (USP) and interrupt stack pointer (ISP), each configured with 16 bits.

Your desired type of stack pointer (USP or ISP) can be selected by a stack pointer select flag (U flag). This flag is located at the position of bit 7 in the flag register (FLG).

(7) Static base register (SB)

Static base register (SB) is configured with 16 bits, and is used for SB relative addressing.

(8) Flag register (FLG)

Flag register (FLG) is configured with 11 bits, each bit is used as a flag. Figure CA-2 shows the flag register (FLG). The following explains the function of each flag:

- **Bit 0: Carry flag (C flag)**
  This flag retains a carry, borrow, or shift-out bit that has occurred in the arithmetic/logic unit.
- **Bit 1: Debug flag (D flag)**
  This flag enables a single-step interrupt.
  When this flag is “1”, a single-step interrupt is generated after instruction execution. This flag is cleared to “0” when the interrupt is acknowledged.
- **Bit 2: Zero flag (Z flag)**
  This flag is set to “1” when an arithmetic operation resulted in 0; otherwise, cleared to “0”.
- **Bit 3: Sign flag (S flag)**
  This flag is set to “1” when an arithmetic operation resulted in a negative value; otherwise, cleared to “0”.
- **Bit 4: Register bank select flag (B flag)**
  This flag chooses a register bank. Register bank 0 is selected when this flag is “0” ; register bank 1 is selected when this flag is “1”.
- **Bit 5: Overflow flag (O flag)**
  This flag is set to “1” when an arithmetic operation resulted in overflow; otherwise, cleared to “0”.
- **Bit 6: Interrupt enable flag (I flag)**
  This flag enables a maskable interrupt.
  An interrupt is disabled when this flag is “0”, and is enabled when this flag is “1”. This flag is cleared to “0” when the interrupt is acknowledged.

- **Bit 7: Stack pointer select flag (U flag)**
  Interrupt stack pointer (ISP) is selected when this flag is “0” ; user stack pointer (USP) is selected when this flag is “1”.
  This flag is cleared to “0” when a hardware interrupt is acknowledged or an INT instruction of software interrupt Nos. 0 to 31 is executed.
- **Bits 8 to 11: Reserved area**
- **Bits 12 to 14: Processor interrupt priority level (IPL)**
  Processor interrupt priority level (IPL) is configured with three bits, for specification of up to eight processor interrupt priority levels from level 0 to level 7.
  If a requested interrupt has priority greater than the processor interrupt priority level (IPL), the interrupt is enabled.
- **Bit 15: Reserved area**

The C, Z, S, and O flags are changed when instructions are executed. See the software manual for details.

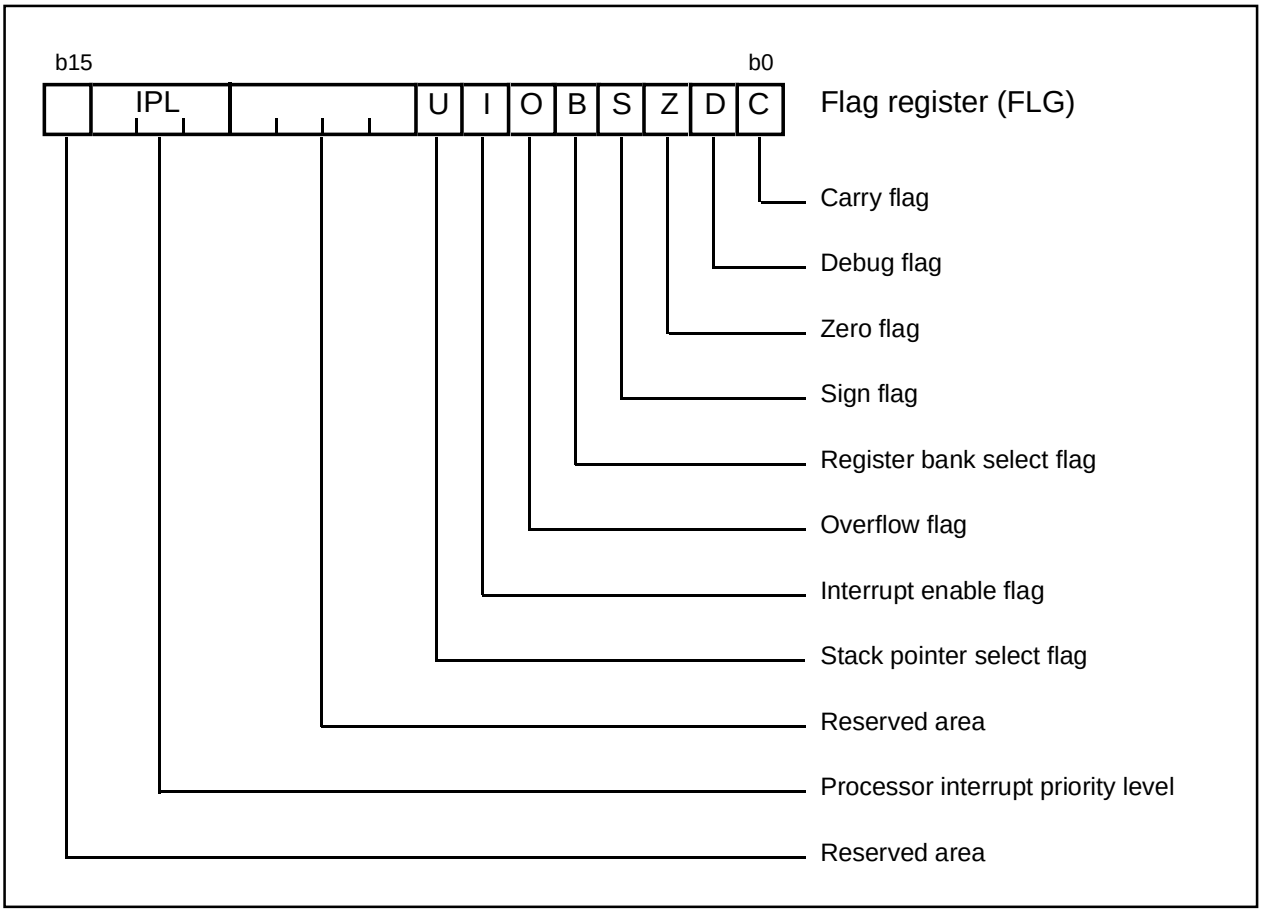


Figure CA-2. Flag register (FLG)

Reset

There are two kinds of resets; hardware and software. In both cases, operation is the same after the reset. (See “Software Reset” for details of software resets.) This section explains on hardware resets.

When the supply voltage is in the range where operation is guaranteed, a reset is effected by holding the reset pin level “L” (0.2VCC max.) for at least 20 cycles. When the reset pin level is then returned to the “H” level while main clock is stable, the reset status is cancelled and program execution resumes from the address in the reset vector table.

Figure DA-1 shows the example reset circuit. Figure DA-2 shows the reset sequence.

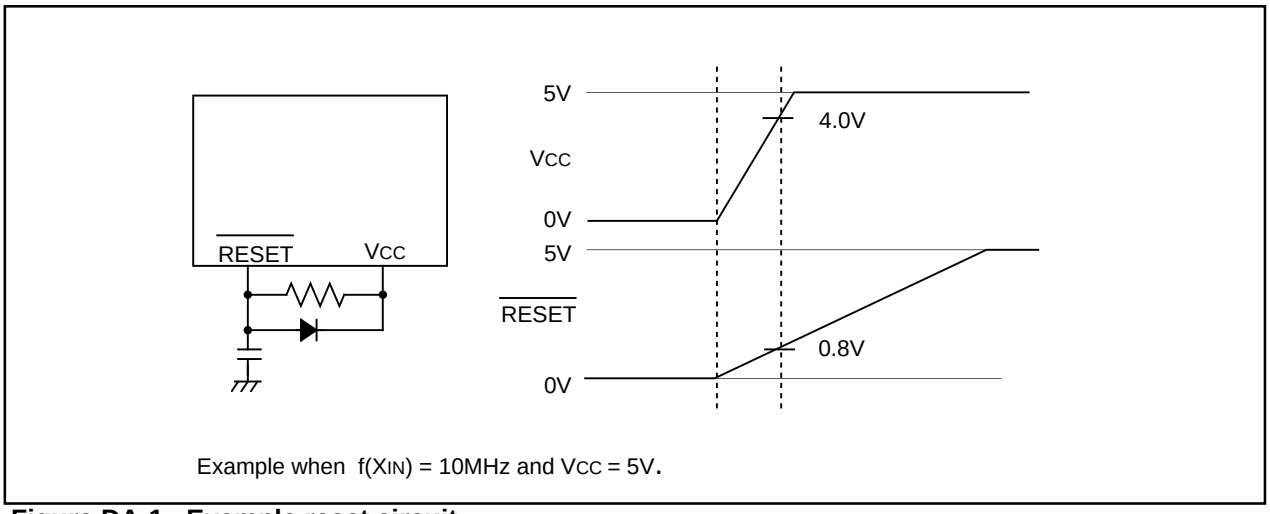


Figure DA-1. Example reset circuit

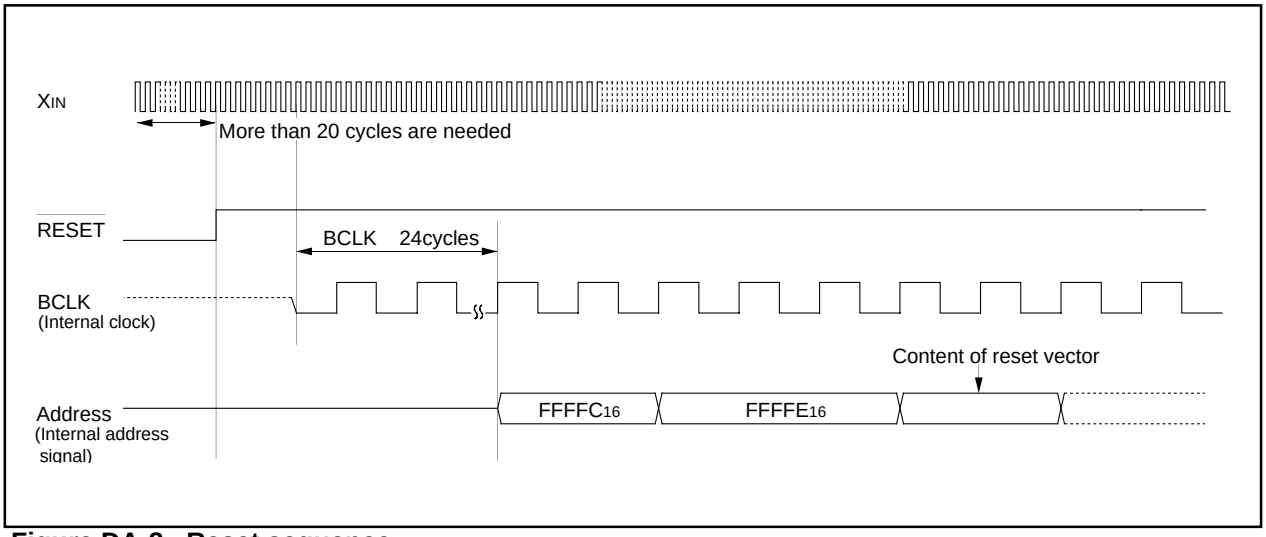


Figure DA-2. Reset sequence

(1) Processor mode register 0	(0004_{16})	0 0 0 0	
(2) Processor mode register 1	(0005_{16})	0 0 0	
(3) System clock control register 0	(0006_{16})	0 1 0 0 1 0 0 0	
(4) System clock control register 1	(0007_{16})	0 0 1 0 0 0 0 0	
(5) Address match interrupt enable register	(0009_{16})	0 0	
(6) Protect register	($000A_{16}$)	0 0 0	
(7) Watchdog timer control register	($000F_{16}$)	0 0 0 ? ? ? ? ?	
(8) Address match interrupt register 0	(0010_{16})	00_{16}	
	(0011_{16})	00_{16}	
	(0012_{16})	0 0 0 0	
(9) Address match interrupt register 1	(0014_{16})	00_{16}	
	(0015_{16})	00_{16}	
	(0016_{16})	0 0 0 0	
(10) DMA0 control register	($002C_{16}$)	0 0 0 0 0 ? 0 0	
(11) DMA1 control register	($003C_{16}$)	0 0 0 0 0 ? 0 0	
(12) INT3 interrupt control register	(0044_{16})	0 0 ? 0 0 0	
(13) INT4 interrupt control register	(0048_{16})	0 0 ? 0 0 0	
(14) INT5 interrupt control register	(0049_{16})	0 0 ? 0 0 0	
(15) DMA0 interrupt control register	($004B_{16}$)	? 0 0 0	
(16) DMA1 interrupt control register	($004C_{16}$)	? 0 0 0	
(17) A-D conversion interrupt control register	($004E_{16}$)	? 0 0 0	
(18) SI/O automatic transfer interrupt control register	($004F_{16}$)	? 0 0 0	
(19) FLD interrupt control register	(0050_{16})	? 0 0 0	
(20) UART0 transmit interrupt control register	(0051_{16})	? 0 0 0	
(21) UART0 receive interrupt control register	(0052_{16})	? 0 0 0	
(22) UART1 transmit interrupt control register	(0053_{16})	? 0 0 0	
(23) UART1 receive interrupt control register	(0054_{16})	? 0 0 0	
(24) Timer A0 interrupt control register	(0055_{16})	? 0 0 0	
(25) Timer A1 interrupt control register	(0056_{16})	? 0 0 0	
(26) Timer A2 interrupt control register	(0057_{16})	? 0 0 0	
(27) Timer A3 interrupt control register	(0058_{16})	? 0 0 0	
(28) Timer A4 interrupt control register	(0059_{16})	? 0 0 0	
(29) Timer B0 interrupt control register	($005A_{16}$)	? 0 0 0	
(30) Timer B1 interrupt control register	($005B_{16}$)	? 0 0 0	
(31) Timer B2 interrupt control register	($005C_{16}$)	? 0 0 0	
(32) INT0 interrupt control register	($005D_{16}$)	0 0 ? 0 0 0	
(33) INT1 interrupt control register	($005E_{16}$)	0 0 ? 0 0 0	
(34) INT2 interrupt control register	($005F_{16}$)	0 0 ? 0 0 0	
(35) Serial I/O 2 control register 1	(0342_{16})	00_{16}	
(36) Serial I/O 2 control register 2	(0344_{16})	00_{16}	
(37) Serial I/O 2 control register 3	(0348_{16})	00_{16}	
(38) FLDC mode register	(0350_{16})	00_{16}	
(39) FLD output control register	(0351_{16})	00_{16}	
(40) Tdisp time set register	(0352_{16})	00_{16}	
(41) Toff1 time set register	(0354_{16})	FF_{16}	
(42) Toff2 time set register	(0356_{16})	FF_{16}	
(43) P2 FLD/port switch register	(0359_{16})	00_{16}	
(44) P3 FLD/port switch register	($035A_{16}$)	00_{16}	
(45) P4 FLD/port switch register	($035B_{16}$)	00_{16}	
(46) P5 digit output set register	($035C_{16}$)	00_{16}	
(47) P6 digit output set register	($035D_{16}$)	00_{16}	

The content of other registers and RAM is undefined when the microcomputer is reset.
The initial values must therefore be set.

x : Nothing is mapped to this bit
? : Undefined

Figure DA-3. Device's internal status after a reset is cleared

(48) Count start flag	(0380$_{16}$)···	00$_{16}$	
(49) Clock prescaler reset flag	(0381$_{16}$)···	0 X X X X X X X	
(50) One-shot start flag	(0382$_{16}$)···	0 0 X 0 0 0 0 0	
(51) Trigger select flag	(0383$_{16}$)···	00$_{16}$	
(52) Up-down flag	(0384$_{16}$)···	00$_{16}$	
(53) Timer A0 mode register	(0396$_{16}$)···	00$_{16}$	
(54) Timer A1 mode register	(0397$_{16}$)···	00$_{16}$	
(55) Timer A2 mode register	(0398$_{16}$)···	00$_{16}$	
(56) Timer A3 mode register	(0399$_{16}$)···	00$_{16}$	
(57) Timer A4 mode register	(039A$_{16}$)···	00$_{16}$	
(58) Timer B0 mode register	(039B$_{16}$)···	0 0 ? X 0 0 0 0	
(59) Timer B1 mode register	(039C$_{16}$)···	0 0 ? X 0 0 0 0	
(60) Timer B2 mode register	(039D$_{16}$)···	0 0 ? X 0 0 0 0	
(61) UART0 transmit/receive mode register	(03A0$_{16}$)···	00$_{16}$	
(62) UART0 transmit/receive control register 0	(03A4$_{16}$)···	0 0 0 0 1 0 0 0	
(63) UART0 transmit/receive control register 1	(03A5$_{16}$)···	0 0 0 0 0 0 1 0	
(64) UART1 transmit/receive mode register	(03A8$_{16}$)···	00$_{16}$	
(65) UART1 transmit/receive control register 0	(03AC$_{16}$)···	0 0 0 0 1 0 0 0	
(66) UART1 transmit/receive control register 1	(03AD$_{16}$)···	0 0 0 0 0 0 1 0	
(67) UART transmit/receive control register 2	(03B0$_{16}$)···	X 0 0 0 0 0 0 0	
(68) Flash memory control register 0 (Note)	(03B4$_{16}$)···	0 0 1 0 0 0 0 0	
(69) Flash memory control register 1 (Note)	(03B5$_{16}$)···	X X X X X X 0 0	
(70) Flash command register (Note)	(03B6$_{16}$)···	00$_{16}$	
(71) DMA0 cause select register	(03B8$_{16}$)···	00$_{16}$	
(72) DMA1 cause select register	(03BA$_{16}$)···	00$_{16}$	
(73) A-D control register 2	(03D4$_{16}$)···	X X X X X X X 0	
(74) A-D control register 0	(03D6$_{16}$)···	0 0 0 0 0 ? ? ?	
(75) A-D control register 1	(03D7$_{16}$)···	00$_{16}$	
(76) D-A control register	(03DC$_{16}$)···	00$_{16}$	
(77) Port P3 direction register	(03E7$_{16}$)···	00$_{16}$	
(78) Port P4 direction register	(03EA$_{16}$)···	00$_{16}$	
(79) Port P7 direction register	(03EF$_{16}$)···	00$_{16}$	
(80) Port P8 direction register	(03F2$_{16}$)···	00$_{16}$	
(81) Port P9 direction register	(03F3$_{16}$)···	00$_{16}$	
(82) Port P10 direction register	(03F6$_{16}$)···	00$_{16}$	
(83) Pull-up control register 0	(03FD$_{16}$)···	00$_{16}$	
(84) Pull-up control register 1	(03FE$_{16}$)···	00$_{16}$	
(85) Data registers (R0/R1/R2/R3)		0000$_{16}$	
(86) Address registers (A0/A1)		0000$_{16}$	
(87) Frame base register (FB)		0000$_{16}$	
(88) Interrupt table register (INTB)		00000$_{16}$	
(89) User stack pointer (USP)		0000$_{16}$	
(90) Interrupt stack pointer (ISP)		0000$_{16}$	
(91) Static base register (SB)		0000$_{16}$	
(92) Flag register (FLG)		0000$_{16}$	

The content of other registers and RAM is undefined when the microcomputer is reset.
The initial values must therefore be set.

x : Nothing is mapped to this bit
? : Undefined
Note: This register is only exist in flash memory version.

Figure DA-4. Device's internal status after a reset is cleared

Software Reset

Writing “1” to bit 3 of the processor mode register 0 (address 0004_{16}) applies a (software reset) reset to the microcomputer. A software reset has almost the same effect as a hardware reset. The contents of internal RAM are preserved.

Figure DA-5 shows the processor mode register 0 and 1.

Processor mode register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
✕	✕	✕	✕	0	0	0	0

Symbol	Address	When reset
PM0	0004_{16}	$XXXX0000_2$

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to “0”	○	○
PM03	Software reset bit	The device is reset when this bit is set to “1”. The value of this bit is “0” when read.	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—

Note: Set bit 1 of the protect register (address $000A_{16}$) to “1” when writing new values to this register.

Processor mode register 1 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
0	✕	✕	✕	✕	✕	0	0

Symbol	Address	When reset
PM1	0005_{16}	$00XXXXX0_2$

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to “0”	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—
Reserved bit		Must always be set to “0”	○	○

Note: Set bit 1 of the protect register (address $000A_{16}$) to “1” when writing new values to this register.

Figure DA-5. Processor mode register 0 and 1.

Clock Generating Circuit

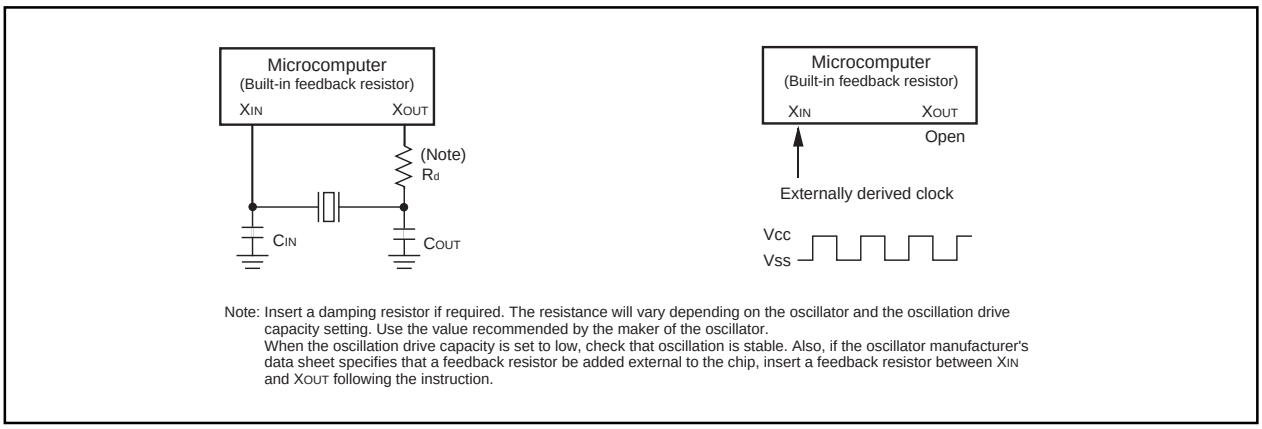
The clock generating circuit contains two oscillator circuits that supply the operating clock sources to the CPU and internal peripheral units.

Table WA-1. Main clock and sub clock generating circuits

	Main clock generating circuit	Sub clock generating circuit
Use of clock	• CPU's operating clock source • Internal peripheral units' operating clock source	• CPU's operating clock source • Timer A/B's count clock source
Usable oscillator	Ceramic or crystal oscillator	Crystal oscillator
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT
Oscillation stop/restart function	Available	Available
Oscillator status immediately after reset	Oscillating	Stopped
Other	Externally derived clock can be input	

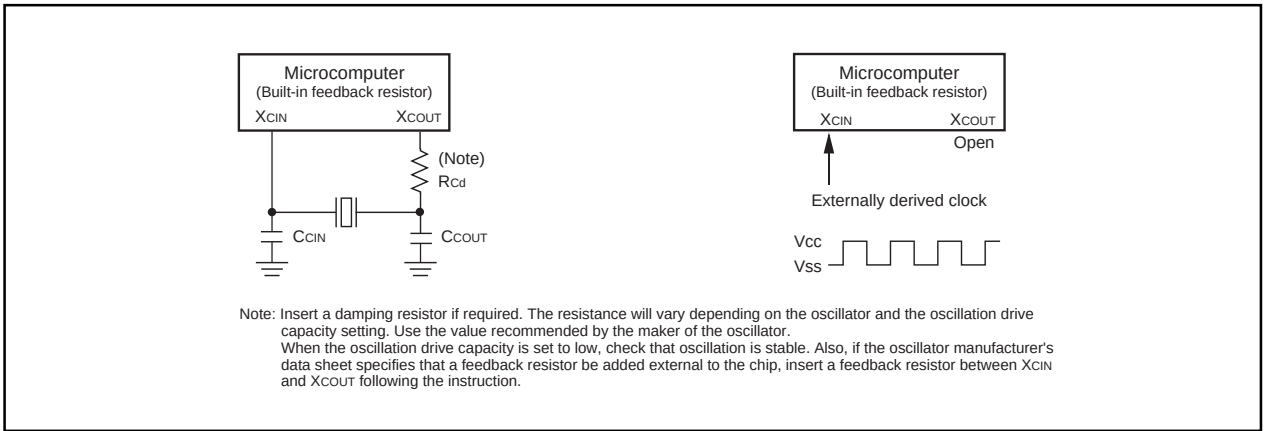
Example of oscillator circuit

Figure WA-1 shows some examples of the main clock circuit, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Figure WA-2 shows some examples of sub clock circuits, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Circuit constants in Figures WA-1 and WA-2 vary with each oscillator used. Use the values recommended by the manufacturer of your oscillator.



Note: Insert a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by the maker of the oscillator.
When the oscillation drive capacity is set to low, check that oscillation is stable. Also, if the oscillator manufacturer's data sheet specifies that a feedback resistor be added external to the chip, insert a feedback resistor between XIN and XOUT following the instruction.

Figure WA-1. Examples of main clock



Note: Insert a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by the maker of the oscillator.
When the oscillation drive capacity is set to low, check that oscillation is stable. Also, if the oscillator manufacturer's data sheet specifies that a feedback resistor be added external to the chip, insert a feedback resistor between XCIN and XCOUT following the instruction.

Figure WA-2. Examples of sub clock

Clock Control

Figure WA-3 shows the block diagram of the clock generating circuit.

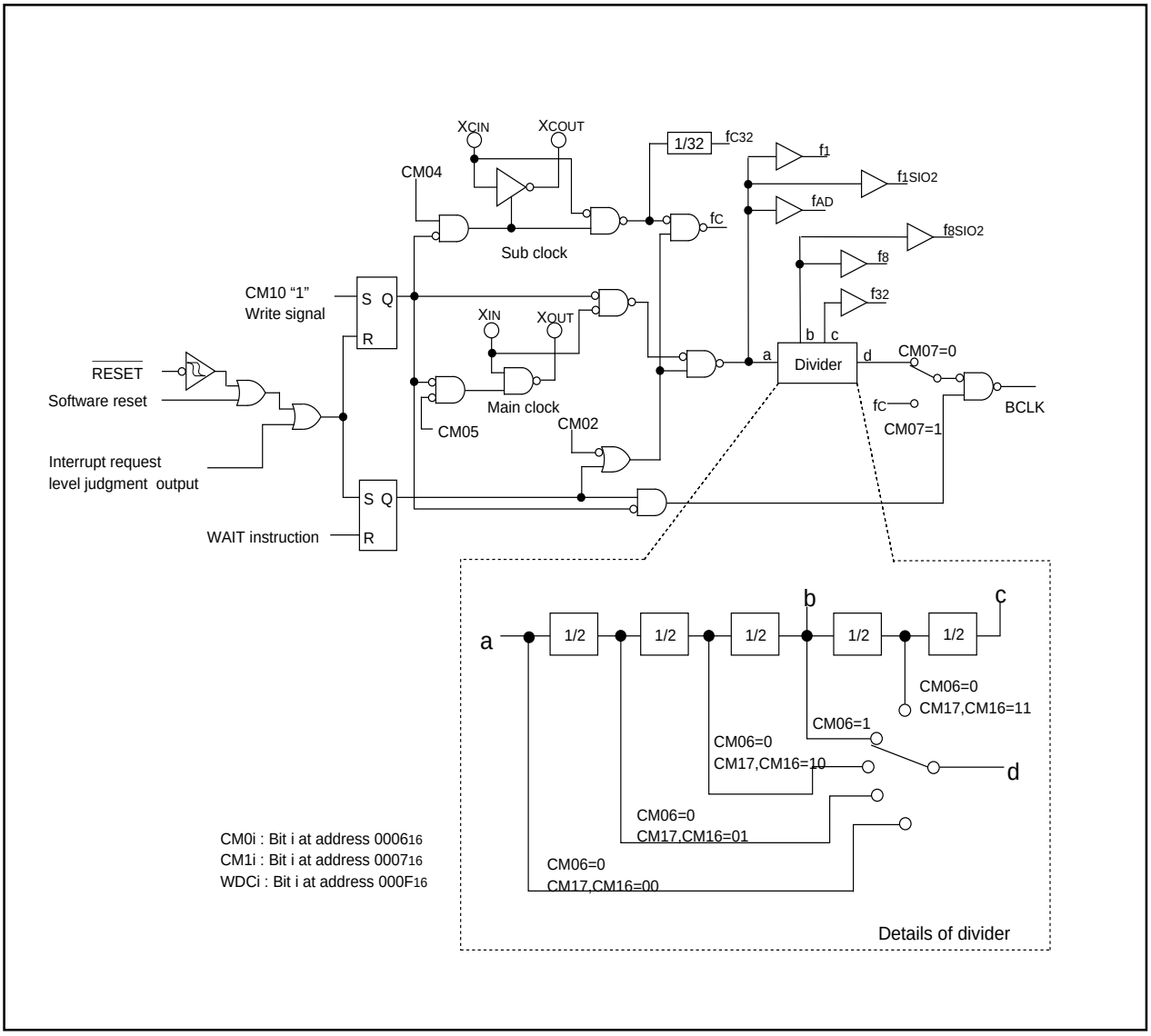


Figure WA-3. Clock generating circuit

The following paragraphs describes the clocks generated by the clock generating circuit.

(1) Main clock

The main clock is generated by the main clock oscillation circuit. After a reset, the clock is divided by 8 to the BCLK. The clock can be stopped using the main clock stop bit (bit 5 at address 0006_{16}). Stopping the clock, after switching the operating clock source of CPU to the sub-clock, reduces the power dissipation. After the oscillation of the main clock oscillation circuit has stabilized, the drive capacity of the main clock oscillation circuit can be reduced using the XIN-XOUT drive capacity select bit (bit 5 at address 0007_{16}). Reducing the drive capacity of the main clock oscillation circuit reduces the power dissipation. This bit changes to "1" when shifting from high-speed/medium-speed mode to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

(2) Sub-clock

The sub-clock is generated by the sub-clock oscillation circuit. No sub-clock is generated after a reset. After oscillation is started using the port Xc select bit (bit 4 at address 0006_{16}), the sub-clock can be selected as the BCLK by using the system clock select bit (bit 7 at address 0006_{16}). However, be sure that the sub-clock oscillation has fully stabilized before switching.

After the oscillation of the sub-clock oscillation circuit has stabilized, the drive capacity of the sub-clock oscillation circuit can be reduced using the XCIN-XCOUT drive capacity select bit (bit 3 at address 0006_{16}). Reducing the drive capacity of the sub-clock oscillation circuit reduces the power dissipation. This bit changes to "1" when shifting to stop mode and at a reset.

(3) BCLK

The BCLK is the clock that drives the CPU, and is fc or the clock is derived by dividing the main clock by 1, 2, 4, 8, or 16. The BCLK is derived by dividing the main clock by 8 after a reset. The BCLK signal can be output from BCLK pin by the BCLK output disable bit (bit 7 at address 0004_{16}) in the memory expansion and the microprocessor modes.

The main clock division select bit 0(bit 6 at address 0006_{16}) changes to "1" when shifting from high-speed/medium-speed to stop mode and at reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

(4) Peripheral function clock(f_1, f_8, f_{32}, f_{AD}, f_{1SIO2}, f_{8SIO2})

The clock for the peripheral devices is derived from the main clock or by dividing it by 1, 8, or 32. The peripheral function clock is stopped by stopping the main clock or by setting the WAIT peripheral function clock stop bit (bit 2 at 0006_{16}) to "1" and then executing a WAIT instruction.

(5) f_{C32}

This clock is derived by dividing the sub-clock by 32. It is used for the timer A and timer B counts.

(6) f_C

This clock has the same frequency as the sub-clock. It is used for the BCLK and for the watchdog timer.

Figure WA-4 shows the system clock control registers 0 and 1.

System clock control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: CM0 — Address: 0006$_{16}$ — When reset: 48$_{16}$

Bit symbol	Bit name	Function	R	W
CM00	Clock output function select bit (Valid only in single-chip mode)	b1 b0 0 0 : I/O port P9$_7$/DA$_0$ 0 1 : f$_C$ output 1 0 : f$_8$ output 1 1 : f$_{32}$ output	O	O
CM01			O	O
CM02	WAIT peripheral function clock stop bit	0 : Do not stop peripheral function clock in wait mode 1 : Stop peripheral function clock in wait mode (Note 8)	O	O
CM03	XCIN-XCOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	O	O
CM04	Port XC select bit	0 : I/O port 1 : XCIN-XCOUT generation	O	O
CM05	Main clock (XIN-XOUT) stop bit (Note 3, 4, 5)	0 : On 1 : Off	O	O
CM06	Main clock division select bit 0 (Note 7)	0 : CM16 and CM17 valid 1 : Division by 8 mode	O	O
CM07	System clock select bit (Note 6)	0 : XIN, XOUT 1 : XCIN, XCOUT	O	O

Note 1: Set bit 0 of the protect register (address 000A$_{16}$) to “1” before writing to this register.
Note 2: Changes to “1” when shiffing to stop mode and at a reset.
Note 3: When entering power saving mode, main clock stops using this bit. When returning from stop mode and operating with XIN, set this bit to “0”. When main clock oscillation is operating by itself, set system clock select bit (CM07) to “1” before setting this bit to “1”.
Note 4: When inputting external clock, only clock oscillation buffer is stopped and clock input is acceptable.
Note 5: If this bit is set to “1”, XOUT turns “H”. The built-in feedback resistor remains being connected, so XIN turns pulled up to XOUT (“H”) via the feedback resistor.
Note 6: Set port Xc select bit (CM04) to “1” and stabilize the sub-clock oscillating before setting to this bit from “0” to “1”. Do not write to both bits at the same time. And also, set the main clock stop bit (CM05) to “0” and stabilize the main clock oscillating before setting this bit from “1” to “0”.
Note 7: This bit changes to “1” when shifting from high-speed/medium-speed mode to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.
Note 8: f$_{C32}$ is not included.

System clock control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
			0	0	0	0	

Symbol: CM1 — Address: 0007$_{16}$ — When reset: 20$_{16}$

Bit symbol	Bit name	Function	R	W
CM10	All clock stop control bit (Note4)	0 : Clock on 1 : All clocks off (stop mode)	O	O
Reserved bit		Always set to “0”	O	O
Reserved bit		Always set to “0”	O	O
Reserved bit		Always set to “0”	O	O
Reserved bit		Always set to “0”	O	O
CM15	XIN-XOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	O	O
CM16	Main clock division select bit 1 (Note 3)	b7 b6 0 0 : No division mode 0 1 : Division by 2 mode 1 0 : Division by 4 mode 1 1 : Division by 16 mode	O	O
CM17				

Note 1: Set bit 0 of the protect register (address 000A$_{16}$) to “1” before writing to this register.
Note 2: This bit changes to “1” when shifting from high-speed/medium-speed mode to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.
Note 3: Can be selected when bit 6 of the system clock control register 0 (address 0006$_{16}$) is “0”. If “1”, division mode is fixed at 8.
Note 4: If this bit is set to “1”, XOUT turns “H”, and the built-in feedback resistor is cut off. XCIN and XCOUT turn high-impedance state.

Figure WA-4. Clock control registers 0 and 1

Clock Output

The clock output function select bit (bit 0,1 at address 0006_{16}) allows you to choose the clock from f_8, f_{32}, or f_C to be output from the P97/DA0/CLKOUT/DIMOUT pin. When the WAIT peripheral function clock stop bit (bit 2 at address 0006_{16}) is set to "1", the output of f_8 and f_{32} stop by executing of WAIT instruction.

Stop Mode

Writing "1" to the all-clock stop control bit (bit 0 at address 0007_{16}) stops all oscillation and the microcomputer enters stop mode. In stop mode, the content of the internal RAM is retained provided that V_{CC} remains above 2V.

Because the oscillation of BCLK, f_1 to f_{32}, fc, f_{C32}, and f_{AD} stops in stop mode, peripheral functions such as the fluorescent display function, serial I/O 2, A-D converter and watchdog timer do not function. However, timer A and timer B operate provided that the event counter mode is set to an external pulse, and UART0 and UART2 functions provided an external clock is selected. Table WA-2 shows the status of the ports in stop mode.

Stop mode is cancelled by a hardware reset or an interrupt. If an interrupt is to be used to cancel stop mode, that interrupt must first have been enabled. If returning by an interrupt, that interrupt routine is executed.

When shifting from high-speed/medium-speed mode to stop mode and at a reset, the main clock division select bit 0 (bit 6 at address 0006_{16}) is set to "1". When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

Table WA-2. Port status during stop mode

Pin		States
Port		Retains status before stop mode
CLKOUT	When fC selected	"H"
	When f_8, f_{32} selected	Retains status before stop mode

Wait Mode

When a WAIT instruction is executed, BCLK stops and the microcomputer enters the wait mode. In this mode, oscillation continues but BCLK and watchdog timer stop. Writing "1" to the WAIT peripheral function clock stop bit and executing a WAIT instruction stops the clock being supplied to the internal peripheral functions, allowing power dissipation to be reduced. Table WA-3 shows the status of the ports in wait mode.

Wait mode is cancelled by a hardware reset or an interrupt. If an interrupt is used to cancel wait mode, the microcomputer restarts from the interrupt routine using as BCLK, the clock that had been selected when the WAIT instruction was executed.

Table WA-3. Port status during wait mode

Pin		States
Port		Retains status before wait mode
CLKOUT	When fC selected	Does not stop
	When f_8, f_{32} selected	Does not stop when the WAIT peripheral function clock stop bit is "0". (Note) When the WAIT peripheral function clock stop bit is "1", the status immediately prior to entering wait mode is maintained.

Note: Attention that reducing the power dissipation is impossible.

Status Transition Of BCLK

Power dissipation can be reduced and low-voltage operation achieved by changing the count source for BCLK. Table WA-4 shows the operating modes corresponding to the settings of system clock control registers 0 and 1.

When reset, the device starts in division by 8 mode. The main clock division select bit 0(bit 6 at address 0006_{16}) changes to "1" when shifting from high-speed/medium-speed to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained. The following shows the operational modes of BCLK.

(1) Division by 2 mode

The main clock is divided by 2 to obtain the BCLK.

(2) Division by 4 mode

The main clock is divided by 4 to obtain the BCLK.

(3) Division by 8 mode

The main clock is divided by 8 to obtain the BCLK. When reset, the device starts operating from this mode. Before the user can go from this mode to no division mode, division by 2 mode, or division by 4 mode, the main clock must be oscillating stably. When going to low-speed or lower power consumption mode, make sure the sub-clock is oscillating stably.

(4) Division by 16 mode

The main clock is divided by 16 to obtain the BCLK.

(5) No-division mode

The main clock is divided by 1 to obtain the BCLK.

(6) Low-speed mode

f_C is used as the BCLK. Note that oscillation of both the main and sub-clocks must have stabilized before transferring from this mode to another or vice versa. At least 2 to 3 seconds are required after the sub-clock starts. Therefore, the program must be written to wait until this clock has stabilized immediately after powering up and after stop mode is cancelled.

(7) Low power dissipation mode

f_C is the BCLK and the main clock is stopped.

Note : Before the count source for BCLK can be changed from X_{IN} to X_{CIN} or vice versa, the clock to which the count source is going to be switched must be oscillating stably. Allow a wait time in software for the oscillation to stabilize before switching over the clock.

Table WA-4. Operating modes dictated by settings of system clock control registers 0 and 1

CM17	CM16	CM07	CM06	CM05	CM04	Operating mode of BCLK
0	1	0	0	0	Invalid	Division by 2 mode
1	0	0	0	0	Invalid	Division by 4 mode
Invalid	Invalid	0	1	0	Invalid	Division by 8 mode
1	1	0	0	0	Invalid	Division by 16 mode
0	0	0	0	0	Invalid	No-division mode
Invalid	Invalid	1	Invalid	0	1	Low-speed mode
Invalid	Invalid	1	Invalid	1	1	Low power dissipation mode

Power control

The following is a description of the three available power control modes:

Modes

Power control is available in three modes.

(a) Normal operation mode

- **High-speed mode**

  Divide-by-1 frequency of the main clock becomes the BCLK. The CPU operates with the internal clock selected. Each peripheral function operates according to its assigned clock.

- **Medium-speed mode**

  Divide-by-2, divide-by-4, divide-by-8, or divide-by-16 frequency of the main clock becomes the BCLK. The CPU operates according to the internal clock selected. Each peripheral function operates according to its assigned clock.

- **Low-speed mode**

  fC becomes the BCLK. The CPU operates according to the fc clock. The fc clock is supplied by the secondary clock. Each peripheral function operates according to its assigned clock.

- **Low power consumption mode**

  The main clock operating in low-speed mode is stopped. The CPU operates according to the fC clock. The fc clock is supplied by the secondary clock. The only peripheral functions that operate are those with the sub-clock selected as the count source.

(b) Wait mode

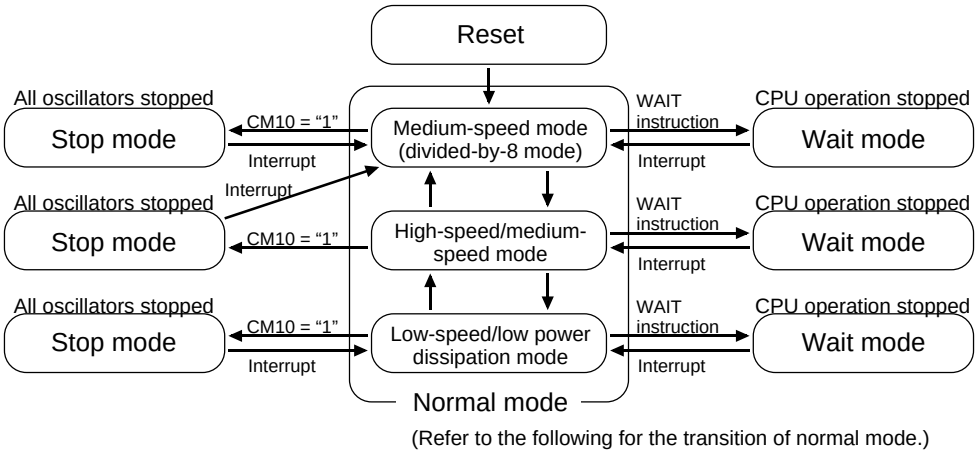
The CPU operation is stopped. The oscillators do not stop.

(c) Stop mode

All oscillators stop. The CPU and all built-in peripheral functions stop. This mode, among the three modes listed here, is the most effective in decreasing power consumption.

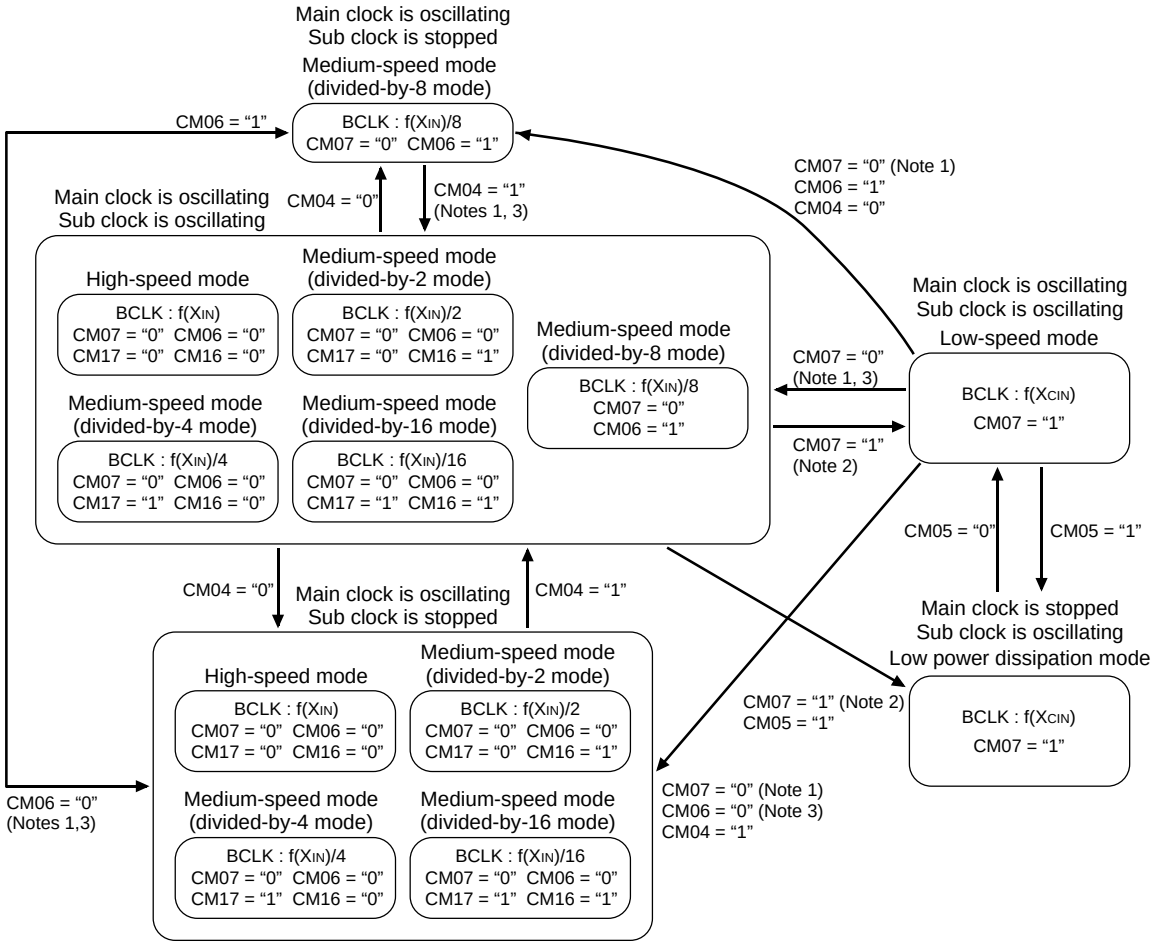
Figure WA-5 is the state transition diagram of the above modes.

Transition of stop mode, wait mode



(Refer to the following for the transition of normal mode.)

Transition of normal mode



Note 1: Switch clock after oscillation of main clock is sufficiently stable.
Note 2: Switch clock after oscillation of sub clock is sufficiently stable.
Note 3: Change CM06 after changing CM17 and CM16.
Note 4: Transit in accordance with arrow.

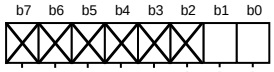
Figure WA-5. State transition diagram of Power control mode

Protection

The protection function is provided so that the values in important registers cannot be changed in the event that the program runs out of control. Figure WA-6 shows the protect register. The values in the processor mode register 0 (address 0004_{16}), processor mode register 1 (address 0005_{16}), system clock control register 0 (address 0006_{16}), and system clock control register 1 (address 0007_{16}) can only be changed when the respective bit in the protect register is set to “1”.

The system clock control registers 0 and 1 write-enable bit (bit 0 at address $000A_{16}$) and processor mode register 0 and 1 write-enable bit (bit 1 at address $000A_{16}$) do not automatically return to “0” after a value has been written to an address. The program must therefore be written to return these bits to “0”.

Protect register



Symbol: PRCR
Address: $000A_{16}$
When reset: $XXXXX000_2$

Bit symbol	Bit name	Function	R	W
PRC0	Enables writing to system clock control registers 0 and 1 (addresses 0006_{16} and 0007_{16})	0 : Write-inhibited 1 : Write-enabled	○	○
PRC1	Enables writing to processor mode registers 0 and 1 (addresses 0004_{16} and 0005_{16})	0 : Write-inhibited 1 : Write-enabled	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			–	–

Figure WA-6. Protect register

Overview of Interrupt

Type of Interrupts

Figure DD-1 lists the types of interrupts.

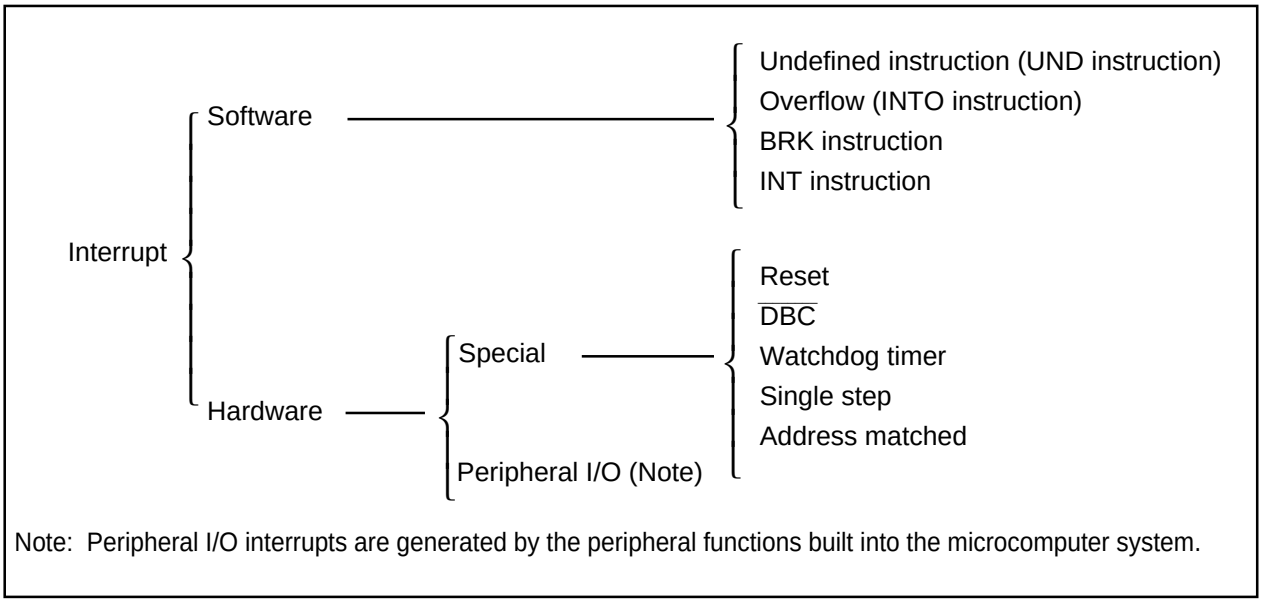


Figure DD-1. Classification of interrupts

- Maskable interrupt : An interrupt which can be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **can be changed** by priority level.
- Non-maskable interrupt : An interrupt which cannot be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **cannot be changed** by priority level.

Software Interrupts

A software interrupt occurs when executing certain instructions. Software interrupts are non-maskable interrupts.

- **Undefined instruction interrupt**

  An undefined instruction interrupt occurs when executing the UND instruction.

- **Overflow interrupt**

  An overflow interrupt occurs when executing the INTO instruction with the overflow flag (O flag) set to "1". The following are instructions whose O flag changes by arithmetic:
  ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, SUB

- **BRK interrupt**

  A BRK interrupt occurs when executing the BRK instruction.

- **INT interrupt**

  An INT interrupt occurs when specifying one of software interrupt numbers 0 through 63 and executing the INT instruction. Software interrupt numbers 0 through 31 are assigned to peripheral I/O interrupts, so executing the INT instruction allows executing the same interrupt routine that a peripheral I/O interrupt does.

  The stack pointer (SP) used for the INT interrupt is dependent on which software interrupt number is involved.

  So far as software interrupt numbers 0 through 31 are concerned, the microcomputer saves the stack pointer assignment flag (U flag) when it accepts an interrupt request. If change the U flag to "0" and select the interrupt stack pointer (ISP), and then execute an interrupt sequence. When returning from the interrupt routine, the U flag is returned to the state it was before the acceptance of interrupt request. So far as software numbers 32 through 63 are concerned, the stack pointer does not make a shift.

Hardware Interrupts

Hardware interrupts are classified into two types — special interrupts and peripheral I/O interrupts.

(1) Special interrupts

Special interrupts are non-maskable interrupts.

- **Reset**

  Reset occurs if an “L” is input to the $\overline{\text{RESET}}$ pin.

- **$\overline{\text{DBC}}$ interrupt**

  This interrupt is exclusively for the debugger, do not use it in other circumstances.

- **Watchdog timer interrupt**

  Generated by the watchdog timer.

- **Single-step interrupt**

  This interrupt is exclusively for the debugger, do not use it in other circumstances. With the debug flag (D flag) set to “1”, a single-step interrupt occurs after one instruction is executed.

- **Address match interrupt**

  An address match interrupt occurs immediately before the instruction held in the address indicated by the address match interrupt register is executed with the address match interrupt enable bit set to “1”. If an address other than the first address of the instruction in the address match interrupt register is set, no address match interrupt occurs.

(2) Peripheral I/O interrupts

A peripheral I/O interrupt is generated by one of built-in peripheral functions. Built-in peripheral functions are dependent on classes of products, so the interrupt factors too are dependent on classes of products. The interrupt vector table is the same as the one for software interrupt numbers 0 through 31 the INT instruction uses. Peripheral I/O interrupts are maskable interrupts.

- **DMA0 interrupt, DMA1 interrupt**

  These are interrupts that DMA generates.

- **A-D conversion interrupt**

  This is an interrupt that the A-D converter generates.

- **UART0 and UART1 transmission interrupt**

  These are interrupts that the serial I/O transmission generates.

- **UART0 and UART1 reception interrupt**

  These are interrupts that the serial I/O reception generates.

- **SI/O automatic transfer interrupt**

  This is an interrupt that the SI/O automatic transfer generates.

- **Timer A0 interrupt through timer A4 interrupt**

  These are interrupts that timer A generates

- **Timer B0 interrupt through timer B2 interrupt**

  These are interrupts that timer B generates.

- **$\overline{\text{INT0}}$ interrupt through $\overline{\text{INT5}}$ interrupt**

  An $\overline{\text{INT}}$ interrupt occurs if either a rising edge or a falling edge is input to the $\overline{\text{INT}}$ pin.

Interrupts and Interrupt Vector Tables

If an interrupt request is accepted, a program branches to the interrupt routine set in the interrupt vector table. Set the first address of the interrupt routine in each vector table. Figure DD-2 shows the format for specifying the address.

Two types of interrupt vector tables are available — fixed vector table in which addresses are fixed and variable vector table in which addresses can be varied by the setting.

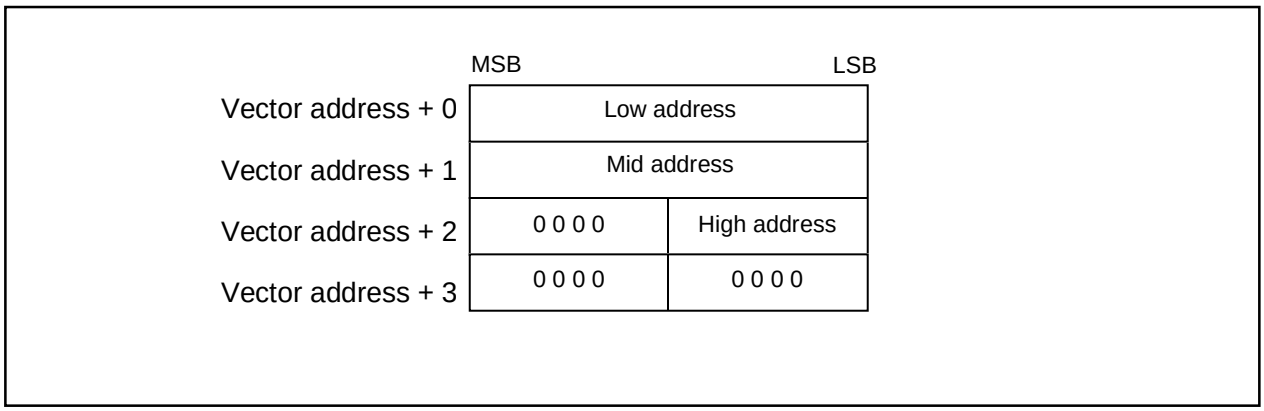


Figure DD-2. Format for specifying interrupt vector addresses

• Fixed vector tables

The fixed vector table is a table in which addresses are fixed. The vector tables are located in an area extending from FFFDC16 to FFFFF16. One vector table comprises four bytes. Set the first address of interrupt routine in each vector table. Table DD-1 shows the interrupts assigned to the fixed vector tables and addresses of vector tables.

Table DD-1. Interrupts assigned to the fixed vector tables and addresses of vector tables

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks
Undefined instruction	FFFDC16 to FFFDF16	Interrupt on UND instruction
Overflow	FFFE016 to FFFE316	Interrupt on INTO instruction
BRK instruction	FFFE416 to FFFE716	If the vector contains FF16, program execution starts from the address shown by the vector in the variable vector table
Address match	FFFE816 to FFFEB16	There is an address-matching interrupt enable bit
Single step (Note)	FFFEC16 to FFFEF16	Do not use
Watchdog timer	FFFF016 to FFFF316	
$\overline{\text{DBC}}$ (Note)	FFFF416 to FFFF716	Do not use
-	FFFF816 to FFFFB16	-
Reset	FFFFC16 to FFFFF16	

Note: Interrupts used for debugging purposes only.

• Variable vector tables

The addresses in the variable vector table can be modified, according to the user's settings. Indicate the first address using the interrupt table register (INTB). The 256-byte area subsequent to the address the INTB indicates becomes the area for the variable vector tables. One vector table comprises four bytes. Set the first address of the interrupt routine in each vector table. Table DD-2 shows the interrupts assigned to the variable vector tables and addresses of vector tables.

Table DD-2. Interrupts assigned to the variable vector tables and addresses of vector tables

Software interrupt number	Vector table address Address (L) to address (H)	Interrupt source	Remarks
Software interrupt number 0	+0 to +3 (Note)	BRK instruction	Cannot be masked I flag
——		——	
Software interrupt number 7	+28 to +31 (Note)	$\overline{\text{INT3}}$	
Software interrupt number 8	+32 to +35 (Note)	$\overline{\text{INT4}}$	
Software interrupt number 9	+36 to +39 (Note)	$\overline{\text{INT5}}$	
——		——	
Software interrupt number 11	+44 to +47 (Note)	DMA0	
Software interrupt number 12	+48 to +51 (Note)	DMA1	
——		——	
Software interrupt number 14	+56 to +59 (Note)	A-D	
Software interrupt number 15	+60 to +63 (Note)	SI/O automatic transfer	
Software interrupt number 16	+64 to +67 (Note)	FLD	
Software interrupt number 17	+68 to +71 (Note)	UART0 transmit	
Software interrupt number 18	+72 to +75 (Note)	UART0 receive	
Software interrupt number 19	+76 to +79 (Note)	UART1 transmit	
Software interrupt number 20	+80 to +83 (Note)	UART1 receive	
Software interrupt number 21	+84 to +87 (Note)	Timer A0	
Software interrupt number 22	+88 to +91 (Note)	Timer A1	
Software interrupt number 23	+92 to +95 (Note)	Timer A2	
Software interrupt number 24	+96 to +99 (Note)	Timer A3	
Software interrupt number 25	+100 to +103 (Note)	Timer A4	
Software interrupt number 26	+104 to +107 (Note)	Timer B0	
Software interrupt number 27	+108 to +111 (Note)	Timer B1	
Software interrupt number 28	+112 to +115 (Note)	Timer B2	
Software interrupt number 29	+116 to +119 (Note)	$\overline{\text{INT0}}$	
Software interrupt number 30	+120 to +123 (Note)	$\overline{\text{INT1}}$	
Software interrupt number 31	+124 to +127 (Note)	$\overline{\text{INT2}}$	
Software interrupt number 32 to Software interrupt number 63	+128 to +131 (Note) to +252 to +255 (Note)	Software interrupt	Cannot be masked I flag

Note : Address relative to address in interrupt table register (INTB).

Interrupt Control

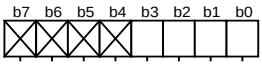
Descriptions are given here regarding how to enable or disable maskable interrupts and how to set the priority to be accepted. What is described here does not apply to non-maskable interrupts.

Enable or disable a maskable interrupt using the interrupt enable flag (I flag), interrupt priority level selection bit, or processor interrupt priority level (IPL). Whether an interrupt request is present or absent is indicated by the interrupt request bit. The interrupt request bit and the interrupt priority level selection bit are located in the interrupt control register of each interrupt. Also, the interrupt enable flag (I flag) and the IPL are located in the flag register (FLG).

Figure DD-3 shows the memory map of the interrupt control registers.

Interrupt control register(Note2)

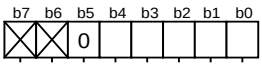
Symbol	Address	When reset
DMiIC(i=0, 1)	$004B_{16}$ to $004C_{16}$	$XXXXX000_2$
ADIC	$004E_{16}$	$XXXXX000_2$
ASIOIC	$004F_{16}$	$XXXXX000_2$
FLDIC	0050_{16}	$XXXXX000_2$
SiTIC(i=0, 1)	0051_{16}, 0053_{16}	$XXXXX000_2$
SiRIC(i=0, 1)	0052_{16}, 0054_{16}	$XXXXX000_2$
TAiIC(i=0 to 4)	0055_{16} to 0059_{16}	$XXXXX000_2$
TBiIC(i=0 to 2)	$005A_{16}$ to $005C_{16}$	$XXXXX000_2$



Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	○	○ (Note1)
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			–	–

Note 1: This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).

Note 2: To rewrite the interrupt control register, do so at a point that dose not generate the interrupt request for that register. For details, see the precautions for interrupts.



Symbol	Address	When reset
INTiIC(i=0 to 5)	$005D_{16}$ to $005F_{16}$ 0047_{16} to 0049_{16}	$XX00X000_2$

Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0: Interrupt not requested 1: Interrupt requested	○	○ (Note1)
POL	Polarity select bit	0 : Selects falling edge 1 : Selects rising edge	○	○
Reserved bit		Always set to “0”	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			–	–

Note1 : This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).

Note 2: To rewrite the interrupt control register, do so at a point that dose not generate the interrupt request for that register. For details, see the precautions for interrupts.

Figure DD-3. Interrupt control registers

Interrupt Enable Flag (I flag)

The interrupt enable flag (I flag) controls the enabling and disabling of maskable interrupts. Setting this flag to “1” enables all maskable interrupts; setting it to “0” disables all maskable interrupts. This flag is set to “0” after reset.

Interrupt Request Bit

The interrupt request bit is set to "1" by hardware when an interrupt is requested. After the interrupt is accepted and jumps to the corresponding interrupt vector, the request bit is set to "0" by hardware. The interrupt request bit can also be set to "0" by software. (Do not set this bit to "1").

Interrupt Priority Level Select Bit and Processor Interrupt Priority Level (IPL)

Set the interrupt priority level using the interrupt priority level select bit, which is one of the component bits of the interrupt control register. When an interrupt request occurs, the interrupt priority level is compared with the IPL. The interrupt is enabled only when the priority level of the interrupt is higher than the IPL. Therefore, setting the interrupt priority level to “0” disables the interrupt.
Table DD-3 shows the settings of interrupt priority levels and Table DD-4 shows the interrupt levels enabled, according to the consist of the IPL.

The following are conditions under which an interrupt is accepted:
· interrupt enable flag (I flag) = 1
· interrupt request bit = 1
· interrupt priority level > IPL

The interrupt enable flag (I flag), the interrupt request bit, the interrupt priority select bit, and the IPL are independent, and they are not affected by one another.

Table DD-3. Settings of interrupt priority levels

Interrupt priority level select bit (b2 b1 b0)	Interrupt priority level	Priority order
0 0 0	Level 0 (interrupt disabled)	———
0 0 1	Level 1	Low ↓
0 1 0	Level 2	
0 1 1	Level 3	
1 0 0	Level 4	
1 0 1	Level 5	
1 1 0	Level 6	
1 1 1	Level 7	High

Table DD-4. Interrupt levels enabled according to the contents of the IPL

IPL (IPL_2 IPL_1 IPL_0)	Enabled interrupt priority levels
0 0 0	Interrupt levels 1 and above are enabled
0 0 1	Interrupt levels 2 and above are enabled
0 1 0	Interrupt levels 3 and above are enabled
0 1 1	Interrupt levels 4 and above are enabled
1 0 0	Interrupt levels 5 and above are enabled
1 0 1	Interrupt levels 6 and above are enabled
1 1 0	Interrupt levels 7 and above are enabled
1 1 1	All maskable interrupts are disabled

Rewrite the interrupt control register

To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. If there is possibility of the interrupt request occur, rewrite the interrupt control register after the interrupt is disabled. The program examples are described as follow:

Example 1:

```
INT_SWITCH1:
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Clear TA0IC int. priority level and int. request bit.
    NOP                     ;
    NOP
    FSET    I               ; Enable interrupts.
```

Example 2:

```
INT_SWITCH2:
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Clear TA0IC int. priority level and int. request bit.
    MOV.W   MEM, R0         ; Dummy read.
    FSET    I               ; Enable interrupts.
```

Example 3:

```
INT_SWITCH3:
    PUSHC   FLG             ;
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Clear TA0IC int. priority level and int. request bit.
    POPC    FLG             ; Enable interrupts.
```

The reason why two NOP instructions or dummy read are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

When a instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : AND, OR, BCLR, BSET

Interrupt Sequence

An interrupt sequence — what are performed over a period from the instant an interrupt is accepted to the instant the interrupt routine is executed — is described here.

If an interrupt occurs during execution of an instruction, the processor determines its priority when the execution of the instruction is completed, and transfers control to the interrupt sequence from the next cycle. If an interrupt occurs during execution of either the SMOVB, SMOVF, SSTR or RMPA instruction, the processor temporarily suspends the instruction being executed, and transfers control to the interrupt sequence.

In the interrupt sequence, the processor carries out the following in sequence given:

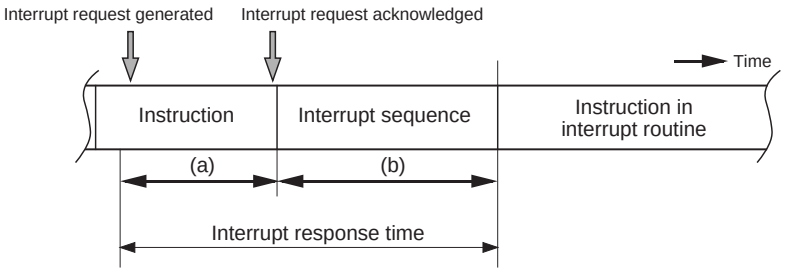
(1) CPU gets the interrupt information (the interrupt number and interrupt request level) by reading address 0000_{16}.
(2) Saves the content of the flag register (FLG) as it was immediately before the start of interrupt sequence in the temporary register (Note) within the CPU.
(3) Sets the interrupt enable flag (I flag), the debug flag (D flag), and the stack pointer select flag (U flag) to "0" (the U flag, however does not change if the INT instruction, in software interrupt numbers 32 through 63, is executed)
(4) Saves the content of the temporary register (Note) within the CPU in the stack area.
(5) Saves the content of the program counter (PC) in the stack area.
(6) Sets the interrupt priority level of the accepted instruction in the IPL.

After the interrupt sequence is completed, the processor resumes executing instructions from the first address of the interrupt routine.

Note: This register cannot be utilized by the user.

Interrupt Response Time

'Interrupt response time' is the period between the instant an interrupt occurs and the instant the first instruction within the interrupt routine has been executed. This time comprises the period from the occurrence of an interrupt to the completion of the instruction under execution at that moment (a) and the time required for executing the interrupt sequence (b). Figure DD-4 shows the interrupt response time.



(a) Time from interrupt request is generated to when the instruction then under execution is completed.
(b) Time in which the instruction sequence is executed.

Figure DD-4. Interrupt response time

Time (a) is dependent on the instruction under execution. Thirty cycles is the maximum required for the DIVX instruction.

Time (b) is as shown in Table DD-5.

Table DD-5. Time required for executing the interrupt sequence

Interrupt vector address	Stack pointer (SP) value	16-Bit bust	8-Bit bus
Even	Even	18 cycles (Note 1)	20 cycles (Note 1)
Even	Odd	19 cycles (Note 1)	20 cycles (Note 1)
Odd (Note 2)	Even	19 cycles (Note 1)	20 cycles (Note 1)
Odd (Note 2)	Odd	20 cycles (Note 1)	20 cycles (Note 1)

Note 1: Add 2 cycles in the case of a $\overline{DBC}$ interrupt; add 1 cycle in the case either of an address coincidence interrupt or of a single-step interrupt.

Note 2: Locate an interrupt vector address in an even address, if possible.

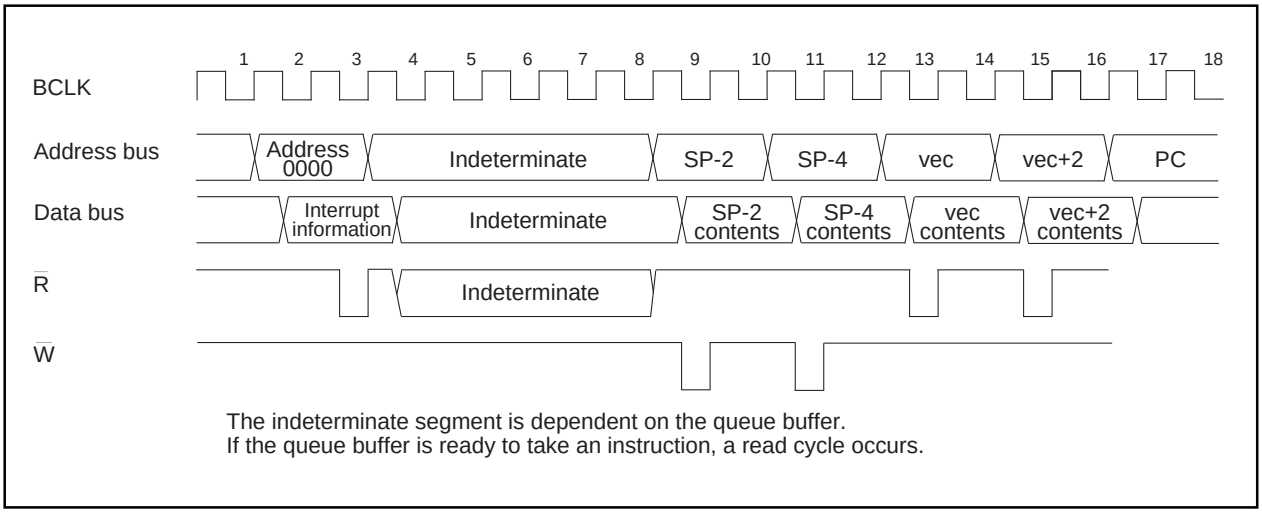


Figure DD-5. Time required for executing the interrupt sequence

Variation of IPL when Interrupt Request is Accepted

If an interrupt request is accepted, the interrupt priority level of the accepted interrupt is set in the IPL.

If an interrupt request, that does not have an interrupt priority level, is accepted, one of the values shown in Table DD-6 is set in the IPL.

Table DD-6. Relationship between interrupts without interrupt priority levels and IPL

Interrupt sources without priority levels	Value set in the IPL
Watchdog timer	7
Reset	0
Other	Not changed

Saving Registers

In the interrupt sequence, only the contents of the flag register (FLG) and that of the program counter (PC) are saved in the stack area.

First, the processor saves the four higher-order bits of the program counter, and 4 upper-order bits and 8 lower-order bits of the FLG register, 16 bits in total, in the stack area, then saves 16 lower-order bits of the program counter. Figure DD-6 shows the state of the stack as it was before the acceptance of the interrupt request, and the state the stack after the acceptance of the interrupt request.

Save other necessary registers at the beginning of the interrupt routine using software. Using the PUSHM instruction alone can save all the registers except the stack pointer (SP).

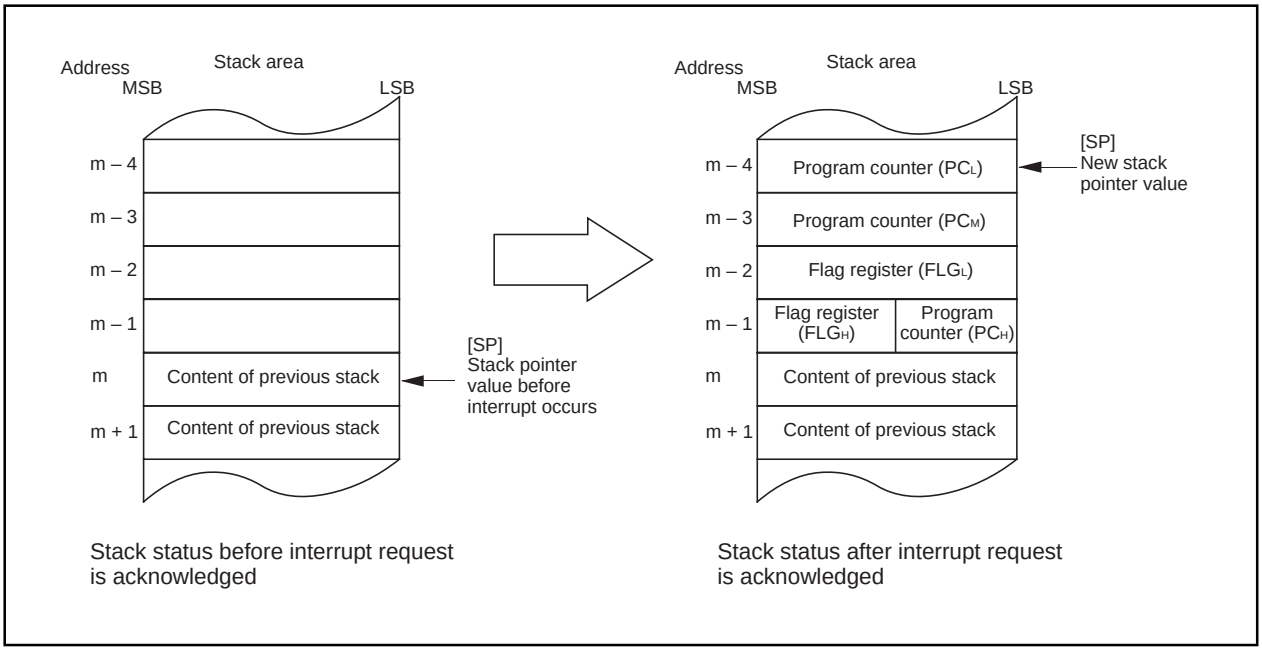
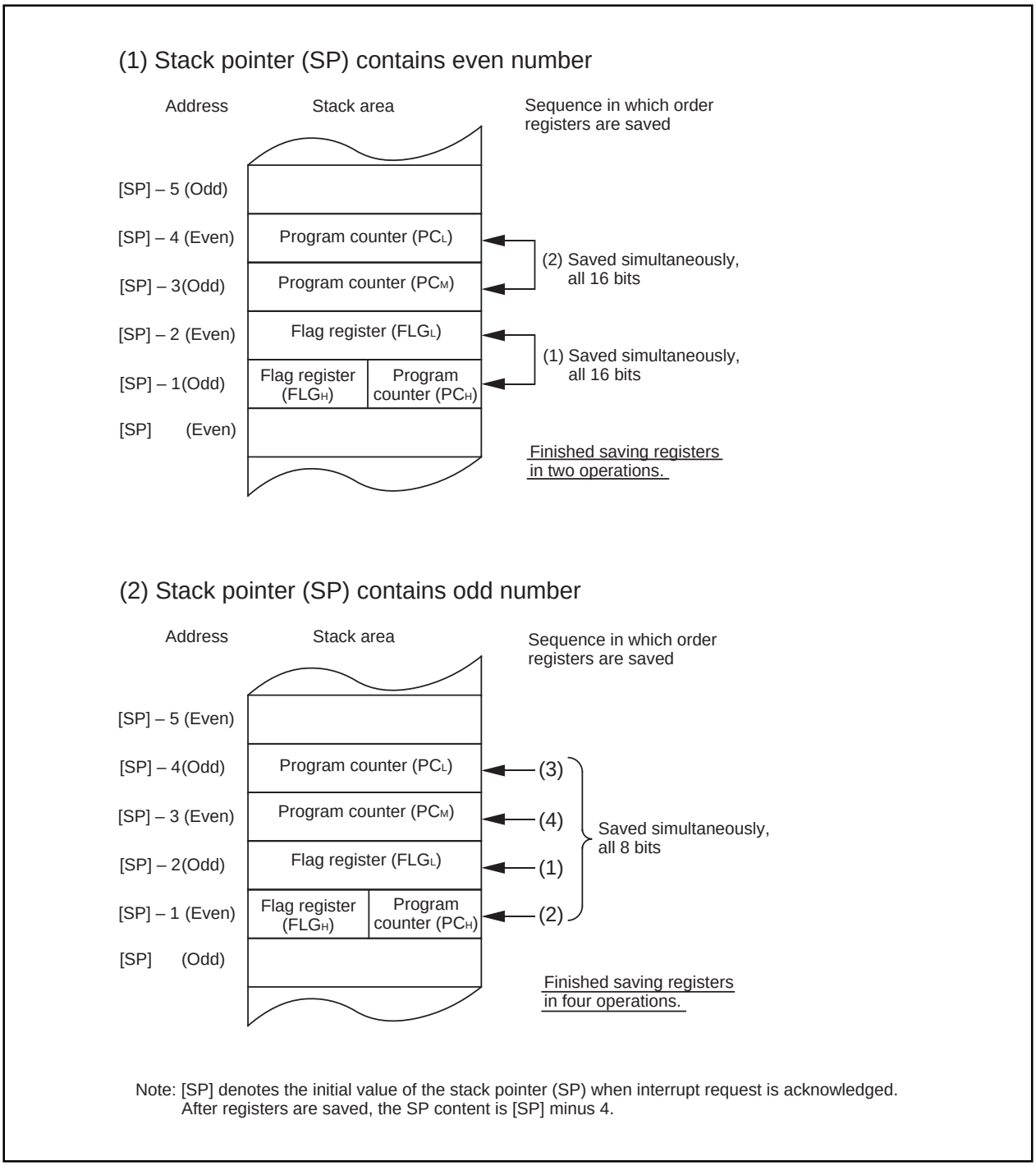


Figure DD-6. State of stack before and after acceptance of interrupt request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the content of the stack pointer, at the time of acceptance of an interrupt request, is even or oDD- If the content of the stack pointer (Note) is even, the content of the flag register (FLG) and the content of the program counter (PC) are saved, 16 bits at a time. If odd, their contents are saved in two steps, 8 bits at a time. Figure DD-7 shows the operation of the saving registers.

Note: Stack pointer indicated by U flag.



Note: [SP] denotes the initial value of the stack pointer (SP) when interrupt request is acknowledged.
After registers are saved, the SP content is [SP] minus 4.

Figure DD-7. Operation of saving registers

Returning from an Interrupt Routine

Executing the REIT instruction at the end of an interrupt routine returns the contents of the flag register (FLG) as it was immediately before the start of interrupt sequence and the contents of the program counter (PC), both of which have been saved in the stack area. Then control returns to the program that was being executed before the acceptance of the interrupt request, so that the suspended process resumes.

Return the other registers saved by software within the interrupt routine using the POPM or similar instruction before executing the REIT instruction.

Interrupt Priority

If there are two or more interrupt requests occurring at a point in time within a single sampling (checking whether interrupt requests are made), the interrupt assigned a higher priority is accepted.

Assign an arbitrary priority to maskable interrupts (peripheral I/O interrupts) using the interrupt priority level select bit. If the same interrupt priority level is assigned, however, the interrupt assigned a higher hardware priority is accepted.

Priorities of the special interrupts, such as Reset (dealt with as an interrupt assigned the highest priority), watchdog timer interrupt, etc. are regulated by hardware.

Figure DD-8 shows the priorities of hardware interrupts.

Software interrupts are not affected by the interrupt priority. If an instruction is executed, control branches invariably to the interrupt routine.

Reset > $\overline{\text{DBC}}$ > Watchdog timer > Peripheral I/O > Single step > Address match

Figure DD-8. Hardware interrupts priorities

Interrupt resolution circuit

When two or more interrupts are generated simultaneously, this circuit selects the interrupt with the highest priority level. Figure DD-9 shows the circuit that judges the interrupt priority level.

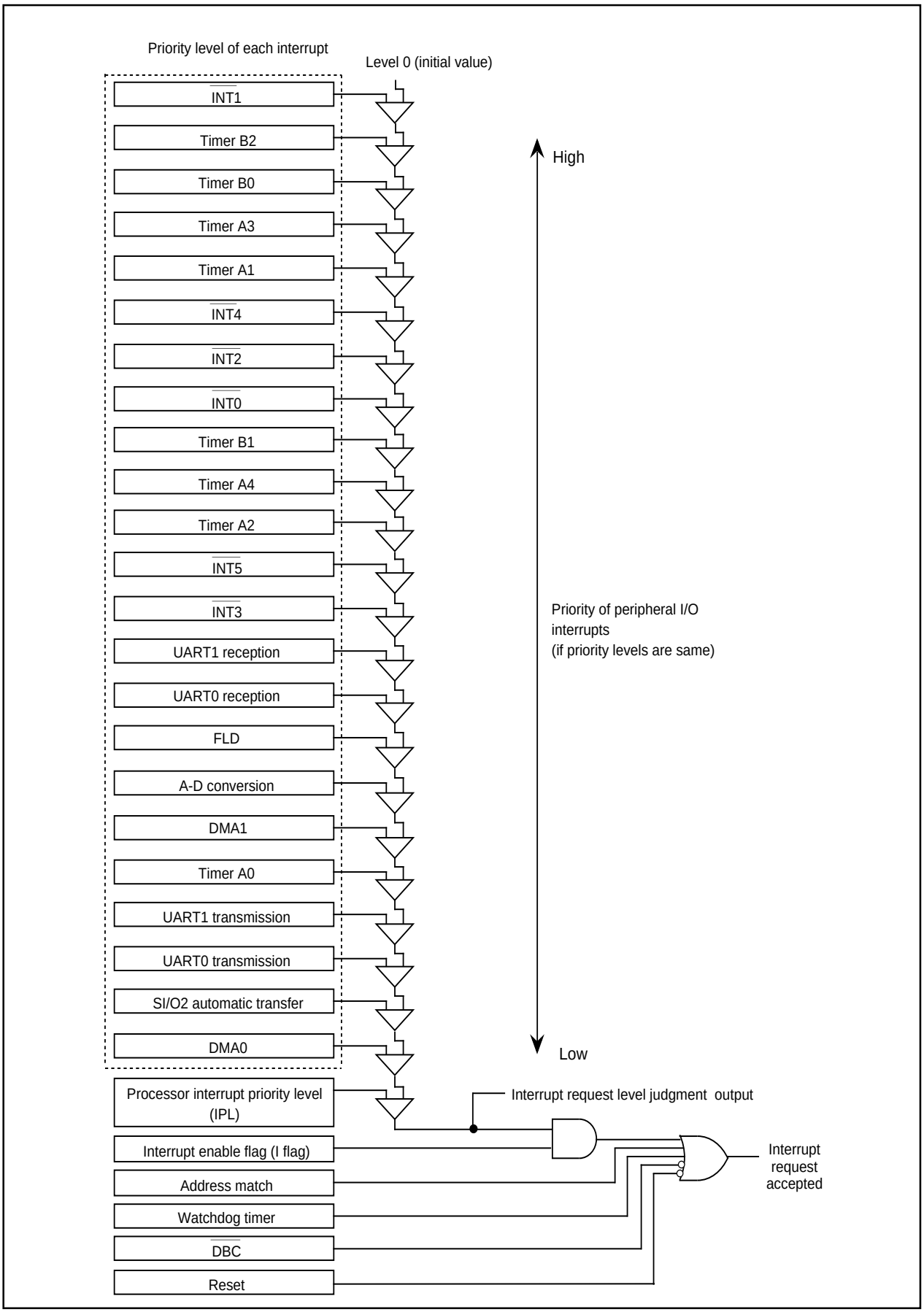


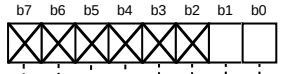
Figure DD-9. Maskable interrupts priorities

Address Match Interrupt

An address match interrupt is generated when the address match interrupt address register contents match the program counter value. Two address match interrupts can be set, each of which can be enabled and disabled by an address match interrupt enable bit. Address match interrupts are not affected by the interrupt enable flag (I flag) and processor interrupt priority level (IPL). The value of the program counter (PC) for an address match interrupt varies depending on the instruction being executed.

Figure DD-12 shows the address match interrupt-related registers.

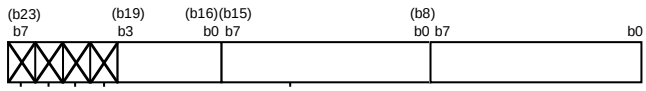
Address match interrupt enable register



Symbol	Address	When reset
AIER	0009_{16}	$XXXXXX00_2$

Bit symbol	Bit name	Function	R	W
AIER0	Address match interrupt 0 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	○	○
AIER1	Address match interrupt 1 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—

Address match interrupt register i (i = 0, 1)



Symbol	Address	When reset
RMAD0	0012_{16} to 0010_{16}	$X00000_{16}$
RMAD1	0016_{16} to 0014_{16}	$X00000_{16}$

Function	Values that can be set	R	W
Address setting register for address match interrupt	00000_{16} to $FFFFF_{16}$	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.		—	—

Figure DD-12. Address match interrupt-related registers

Precautions for Interrupts

(1) Reading address 00000$_{16}$

- When maskable interrupt is occurred, CPU read the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence.
  The interrupt request bit of the certain interrupt written in address 00000$_{16}$ will then be set to "0".
  Reading address 00000$_{16}$ by software sets enabled highest priority interrupt source request bit to "0".
  Though the interrupt is generated, the interrupt routine may not be executed.
  Do not read address 00000$_{16}$ by software.

(2) Setting the stack pointer

- The value of the stack pointer immediately after reset is initialized to 0000$_{16}$. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt.

(3) External interrupt

- Either an "L" level or an "H" level of at least 250 ns width is necessary for the signal input to pins $\overline{\text{INT}}_0$ through $\overline{\text{INT}}_5$ regardless of the CPU operation clock.
- When the polarity of the $\overline{\text{INT}}_0$ through $\overline{\text{INT}}_5$ pins is changed, the interrupt request bit is sometimes set to "1". After changing the polarity, set the interrupt request bit to "0". Figure DD-13 shows the procedure for changing the $\overline{\text{INT}}$ interrupt generate factor.

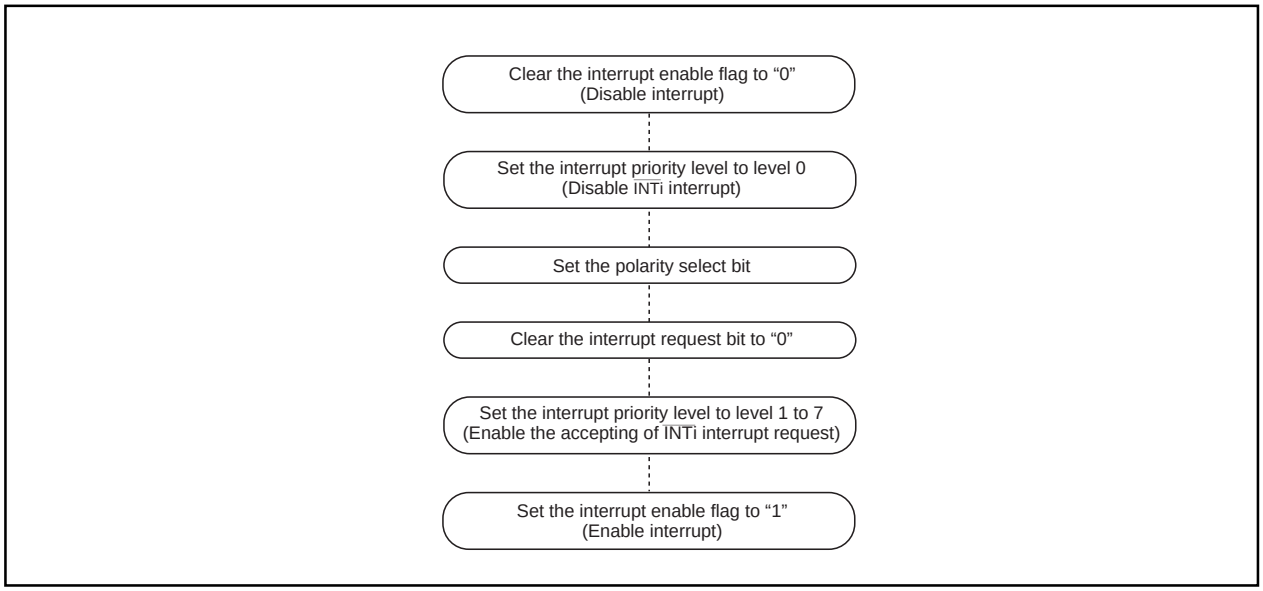


Figure DD-13. Switching condition of $\overline{\text{INT}}$ interrupt request

(5) Rewrite the interrupt control register

• To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. If there is possibility of the interrupt request occur, rewrite the interrupt control register after the interrupt is disabled. The program examples are described as follow:

Example 1:

```
INT_SWITCH1:
    FCLR    I              ; Disable interrupts.
    AND.B   #00h, 0055h    ; Clear TA0IC int. priority level and int. request bit.
    NOP                    ;
    NOP
    FSET    I              ; Enable interrupts.
```

Example 2:

```
INT_SWITCH2:
    FCLR    I              ; Disable interrupts.
    AND.B   #00h, 0055h    ; Clear TA0IC int. priority level and int. request bit.
    MOV.W   MEM, R0        ; Dummy read.
    FSET    I              ; Enable interrupts.
```

Example 3:

```
INT_SWITCH3:
    PUSHC   FLG            ;
    FCLR    I              ; Disable interrupts.
    AND.B   #00h, 0055h    ; Clear TA0IC int. priority level and int. request bit.
    POPC    FLG            ; Enable interrupts.
```

The reason why two NOP instructions or dummy read are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

• When a instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.
Instructions : AND, OR, BCLR, BSET

Watchdog Timer

The watchdog timer has the function of detecting when the program is out of control. The watchdog timer is a 15-bit counter which down-counts the clock derived by dividing the BCLK using the prescaler. A watchdog timer interrupt is generated when an underflow occurs in the watchdog timer. When X_{IN} is selected for the BCLK, bit 7 of the watchdog timer control register (address $000F_{16}$) selects the prescaler division ratio (by 16 or by 128). When X_{CIN} is selected as the BCLK, the prescaler is set for division by 2 regardless of bit 7 of the watchdog timer control register (address $000F_{16}$). Thus the watchdog timer's period can be calculated as given below. The watchdog timer's period is, however, subject to an error due to the prescaler.

With X_{IN} chosen for BCLK

$$\text{Watchdog timer period} = \frac{\text{prescaler dividing ratio (16 or 128) X watchdog timer count (32768)}}{\text{BCLK}}$$

With X_{CIN} chosen for BCLK

$$\text{Watchdog timer period} = \frac{\text{prescaler dividing ratio (2) X watchdog timer count (32768)}}{\text{BCLK}}$$

For example, suppose that BCLK runs at 10 MHz and that 16 has been chosen for the dividing ratio of the prescaler, then the watchdog timer's period becomes approximately 52.4 ms.

The watchdog timer is initialized by writing to the watchdog timer start register (address $000E_{16}$) and when a watchdog timer interrupt request is generated. The prescaler is initialized only when the microcomputer is reset. After a reset is cancelled, the watchdog timer and prescaler are both stopped. The count is started by writing to the watchdog timer start register (address $000E_{16}$).
Figure FA-1 shows the block diagram of the watchdog timer. Figure FA-2 shows the watchdog timer-related registers.

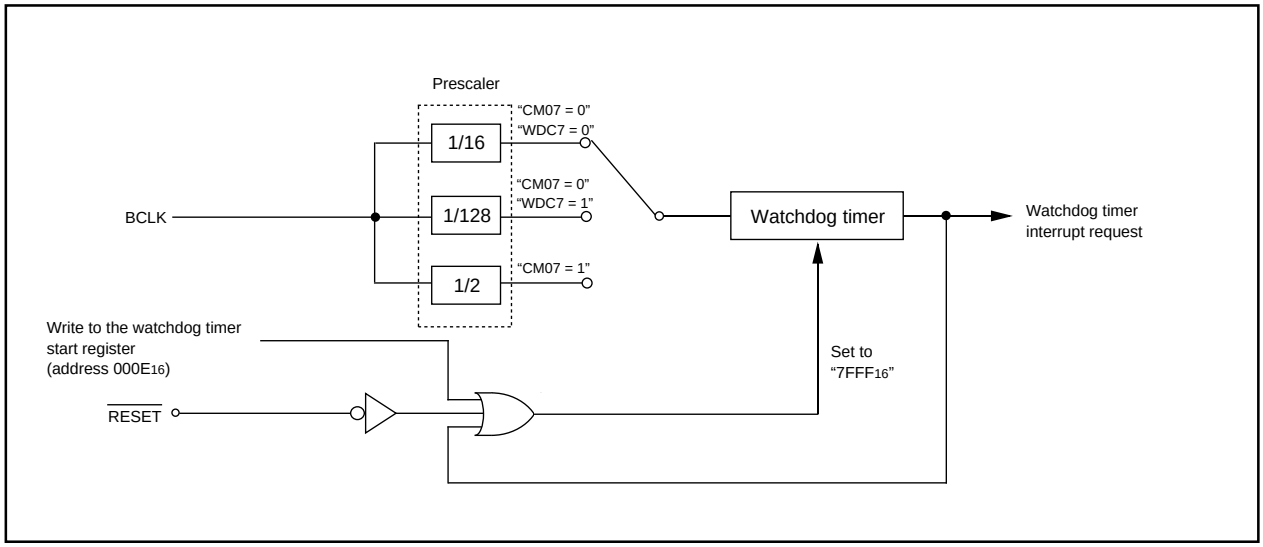


Figure FA-1. Block diagram of watchdog timer

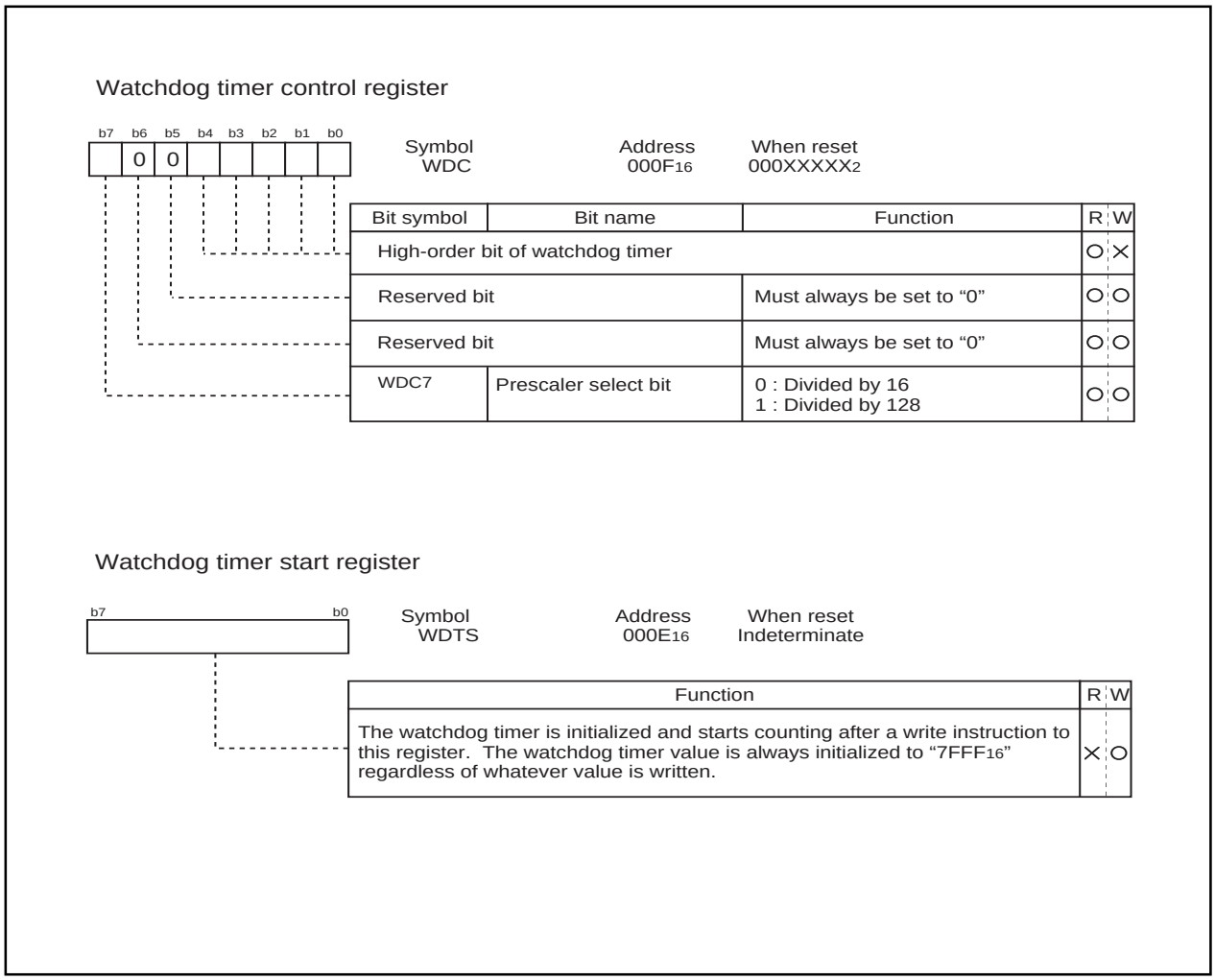


Figure FA-2. Watchdog timer control and start registers

DMAC

This microcomputer has two DMAC (direct memory access controller) channels that allow data to be sent to memory without using the CPU. DMAC shares the same data bus with the CPU. The DMAC is given a higher right of using the bus than the CPU, which leads to working the cycle stealing method. On this account, the operation from the occurrence of DMA transfer request signal to the completion of 1-word (16-bit) or 1-byte (8-bit) data transfer can be performed at high speed. Figure EC-1 shows the block diagram of the DMAC. Table EC-1 shows the DMAC specifications. Figure EC-2 to Figure EC-3 show the registers used by the DMAC.

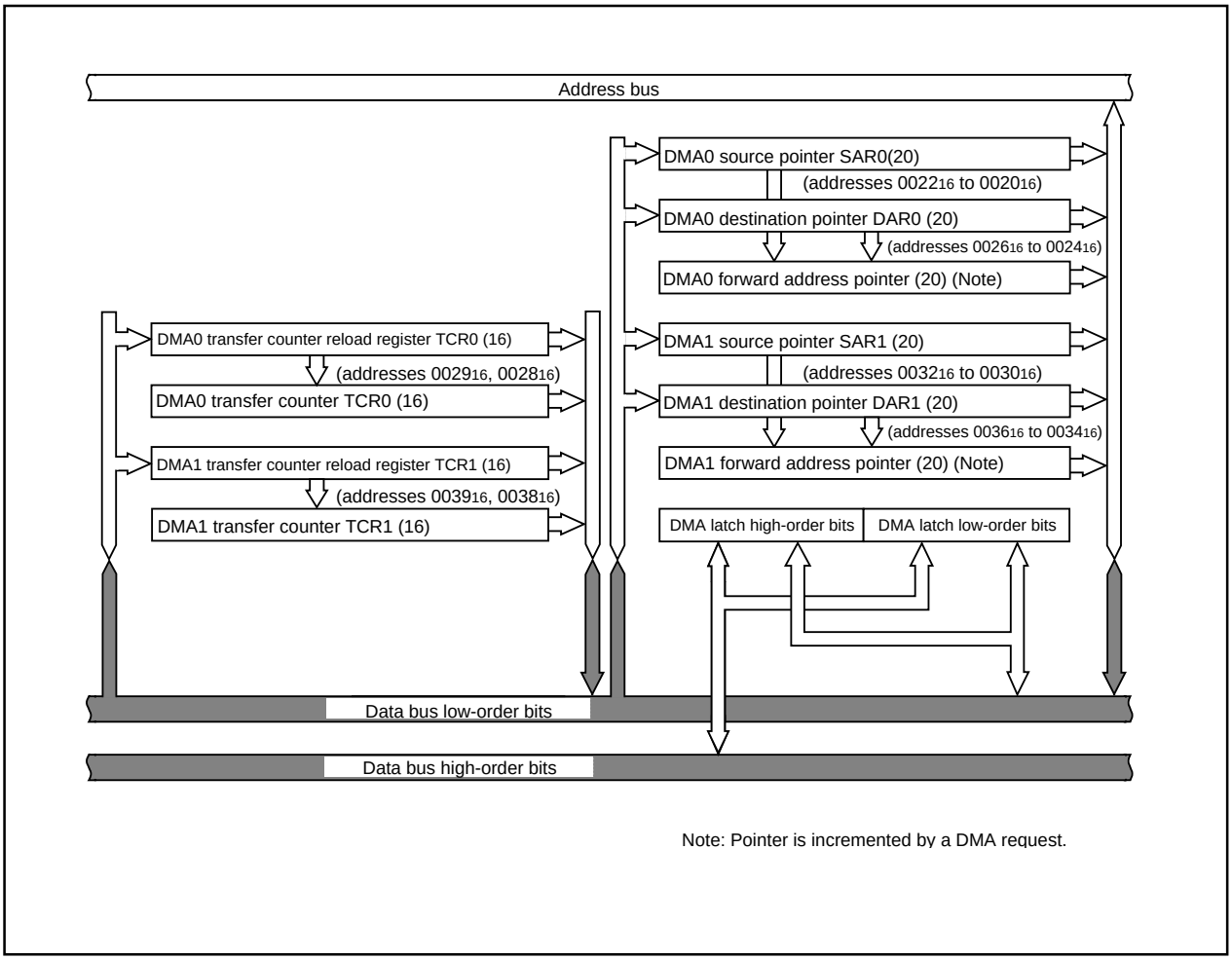


Figure EC-1. Block diagram of DMAC

Either a write signal to the software DMA request bit or an interrupt request signal is used as a DMA transfer request signal. But the DMA transfer is affected neither by the interrupt enable flag (I flag) nor by the interrupt priority level. The DMA transfer doesn't affect any interrupts either.

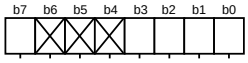
If the DMAC is active (the DMA enable bit is set to 1), data transfer starts every time a DMA transfer request signal occurs. If the cycle of the occurrences of DMA transfer request signals is higher than the DMA transfer cycle, there can be instances in which the number of transfer requests doesn't agree with the number of transfers. For details, see the description of the DMA request bit.

Table EC-1. DMAC specifications

Item	Specification
No. of channels	2 (cycle steal method)
Transfer memory space	• From any address in the 1M bytes space to a fixed address • From a fixed address to any address in the 1M bytes space • From a fixed address to a fixed address (Note that DMA-related registers [0020$_{16}$ to 003F$_{16}$] cannot be accessed)
Maximum No. of bytes transferred	128K bytes (with 16-bit transfers) or 64K bytes (with 8-bit transfers)
DMA request factors (Note)	Falling edge of $\overline{INT0}$ or $\overline{INT1}$ ($\overline{INT0}$ can be selected by DMA0, $\overline{INT1}$ by DMA1) Timer A0 to timer A4 interrupt requests Timer B0 to timer B2 interrupt requests UART0 transmission and reception interrupt requests UART1 transmission and reception interrupt requests A-D conversion interrupt requests Software triggers
Channel priority	DMA0 takes precedence if DMA0 and DMA1 requests are generated simultaneously
Transfer unit	8 bits or 16 bits
Transfer address direction	forward or fixed (forward direction cannot be specified for both source and destination simultaneously)
Transfer mode	• Single transfer mode After the transfer counter underflows, the DMA enable bit turns to "0", and the DMAC turns inactive • Repeat transfer mode After the transfer counter underflows, the value of the transfer counter reload register is reloaded to the transfer counter. The DMAC remains active unless a "0" is written to the DMA enable bit.
DMA interrupt request generation timing	When an underflow occurs in the transfer counter
Active	When the DMA enable bit is set to "1", the DMAC is active. When the DMAC is active, data transfer starts every time a DMA transfer request signal occurs.
Inactive	• When the DMA enable bit is set to "0", the DMAC is inactive. • After the transfer counter underflows in single transfer mode
Forward address pointer and load timing for transfer counter	At the time of starting data transfer immediately after turning the DMAC active, re the value of one of source pointer and destination pointer - the one specified for the forward direction - is reloaded to the forward direction address pointer,and the value of the transfer counter reload register is reloaded to the transfer counter.
Writing to register	Registers specified for forward direction transfer are always write enabled. Registers specified for fixed address transfer are write-enabled when the DMA enable bit is "0".
Reading the register	Can be read at any time. However, when the DMA enable bit is "1", reading the register set up as the forward register is the same as reading the value of the forward address pointer.

Note: DMA transfer is not effective to any interrupt. DMA transfer is affected neither by the interrupt enable flag (I flag) nor by the interrupt priority level.

DMAi request cause select register

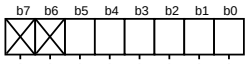


Symbol	Address	When reset
DMiSL(i=0,1)	$03B8_{16}$,$03BA_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
DSEL0	DMA request cause select bit	b3 b2 b1 b0 0 0 0 0 : Falling edge of $\overline{INT0}$ / $\overline{INT1}$ pin (Note) 0 0 0 1 : Software trigger 0 0 1 0 : Timer A0 0 0 1 1 : Timer A1 0 1 0 0 : Timer A2 0 1 0 1 : Timer A3 0 1 1 0 : Timer A4 0 1 1 1 : Timer B0 1 0 0 0 : Timer B1 1 0 0 1 : Timer B2 1 0 1 0 : UART0 transmit 1 0 1 1 : UART0 receive 1 1 0 0 : UART1 transmit 1 1 0 1 : UART1 receive 1 1 1 0 : A-D conversion 1 1 1 1 : Inhibited	○	○
DSEL1			○	○
DSEL2			○	○
DSEL3			○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.			–	–
DSR	Software DMA request bit	If software trigger is selected, a DMA request is generated by setting this bit to “1” (When read, the value of this bit is always “0”)	○	○

Note: Address $03B8_{16}$ is for $\overline{INT0}$; address $03BA_{16}$ is for $\overline{INT1}$.

DMAi control register



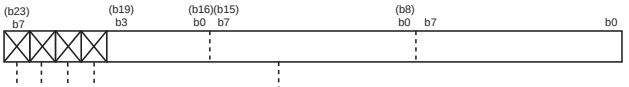
Symbol	Address	When reset
DMiCON(i=0,1)	$002C_{16}$, $003C_{16}$	$00000X00_2$

Bit symbol	Bit name	Function	R	W
DMBIT	Transfer unit bit select bit	0 : 16 bits 1 : 8 bits	○	○
DMASL	Repeat transfer mode select bit	0 : Single transfer 1 : Repeat transfer	○	○
DMAS	DMA request bit (Note 1)	0 : DMA not requested 1 : DMA requested	○	○ (Note 2)
DMAE	DMA enable bit	0 : Disabled 1 : Enabled	○	○
DSD	Source address direction select bit (Note 3)	0 : Fixed 1 : Forward	○	○
DAD	Destination address direction select bit (Note 3)	0 : Fixed 1 : Forward	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.			–	–

Note 1: DMA request can be cleared by resetting the bit.
Note 2: This bit can only be set to “0”.
Note 3: Source address direction select bit and destination address direction select bit cannot be set to “1” simultaneously.

Figure EC-2. DMAC-related registers (1)

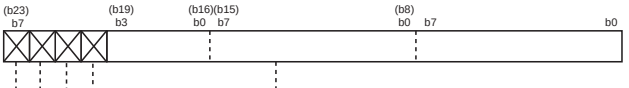
DMAi source pointer (i = 0, 1)



Symbol	Address	When reset
SAR0	0022_{16} to 0020_{16}	Indeterminate
SAR1	0032_{16} to 0030_{16}	Indeterminate

Function	Transfer count specification	R	W
• Source pointer Stores the source address	00000_{16} to $FFFFF_{16}$	○	○
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".		—	—

DMAi destination pointer (i = 0, 1)



Symbol	Address	When reset
DAR0	0026_{16} to 0024_{16}	Indeterminate
DAR1	0036_{16} to 0034_{16}	Indeterminate

Function	Transfer count specification	R	W
• Destination pointer Stores the destination address	00000_{16} to $FFFFF_{16}$	○	○
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".		—	—

DMAi transfer counter (i = 0, 1)

Symbol	Address	When reset
TCR0	0029_{16}, 0028_{16}	Indeterminate
TCR1	0039_{16}, 0038_{16}	Indeterminate

Function	Transfer count specification	R	W
• Transfer counter Set a value one less than the transfer count	0000_{16} to $FFFF_{16}$	○	○

Figure EC-3. DMAC-related registers (2)

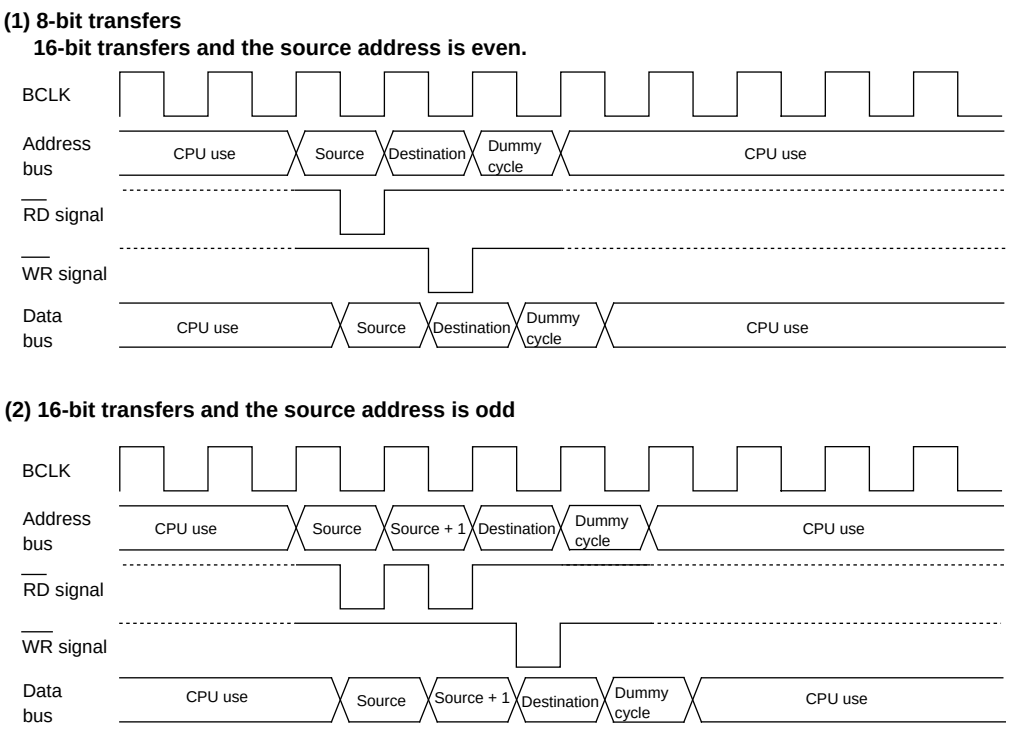
(1) Transfer cycle

The transfer cycle consists of the bus cycle in which data is read from memory or from the SFR area (source read) and the bus cycle in which the data is written to memory or to the SFR area (destination write). The number of read and write bus cycles depends on the source and destination addresses.

(a) Effect of source and destination addresses

When 16-bit data is transferred on a 16-bit data bus, and the source and destination both start at odd addresses, there are one more source read cycle and destination write cycle than when the source and destination both start at even addresses.

Figure EC-4 shows the example of the transfer cycles (a state of internal bus) for a source read. For convenience, the destination write cycle is shown as one cycle and the source read cycles for the different conditions are shown. In reality, the destination write cycle is subject to the same conditions as the source read cycle, with the transfer cycle changing accordingly. When calculating the transfer cycle, remember to apply the respective conditions to both the destination write cycle and the source read cycle.



Note: The same timing changes occur with the respective conditions at the destination as at the source.

Figure EC-4. Example of transfer cycles for a source read (the state of internal bus)

(2) DMAC Transfer

Any combination of even or odd transfer read and write addresses is possible. Table EC-2 shows the number of DMAC transfer cycles.

The number of DMAC transfer cycles can be calculated as follows:

No. of transfer cycles per transfer unit = No. of read cycles x j + No. of write cycles x k

Table EC-2. No. of DMAC transfer cycles

Transfer unit	Access address	singelchip mode	
		No. of read cycles	No. of write cycles
8-bit transfers (DMBIT="1")	Even	1	1
	Odd	1	1
16-bit transfers (DMBIT="0")	Even	1	1
	Odd	2	2

Coefficient j, k

Internal memory	
Internal ROM/RAM	SFR area
1	2

FLD Controller

The M30218 group has fluorescent display (FLD) drive and control circuits.

Table KA-0 shows the FLD controller specifications.

Table KA-0. FLD controller specifications

Item		Specification
FLD controller port	High-breakdown-voltage output port	• 52 pins (20 pins can switch general purpose port)
	CMOS port	• 4 pins (4 pins can switch general purpose port) (A driver must be installed externally)
Display pixel number		• Used FLD output 28 segment X 28 digit (segment number + digit number ≤ 56) • Used digit output 40 segment X 16 digit (segment number ≤ 40, digit number ≤ 16) • Connected to M35501 56 segment X (connect number of M35501) digit (segment number ≤ 56, digit number ≤ number of M35501 X 16) • Used $P4_4$ to $P4_7$ expansion 52 segment X 16 digit (segment number ≤ 52, digit number ≤ 16)
Period		• 3.2 μs to 819.2 μs (count source X_{IN}/32,10MHz) • 12.8 μs to 3276.8 μs (count source X_{IN}/128,10MHz)
Dimmer time		• 3.2 μs to 819.2 μs (count source X_{IN}/32,10MHz) • 12.8 μs to 3276.8 μs (count source X_{IN}/128,10MHz)
Interrupt		• Digit interrupt • FLD blanking interrupt
Key-scan		• Key-scan used digit • Key-scan used segment
Expand function		• Digit pulse output function This function automatically outputs digit pulse. • M35501 connect function The number of digits can be increased easily by using the output of DIM_{OUT}($P9_7$) as CLK for the M35501. • Toff section generate / not generate function This function does not generate Toff1 section when the connected outputs are the same. • Gradation display function This function allows each segment to be set for dark or bright display. • $P4_4$ to $P4_7$ expansion function This function provides 16 lines of digit outputs from four ports by attaching a 4 → 16 decoder.

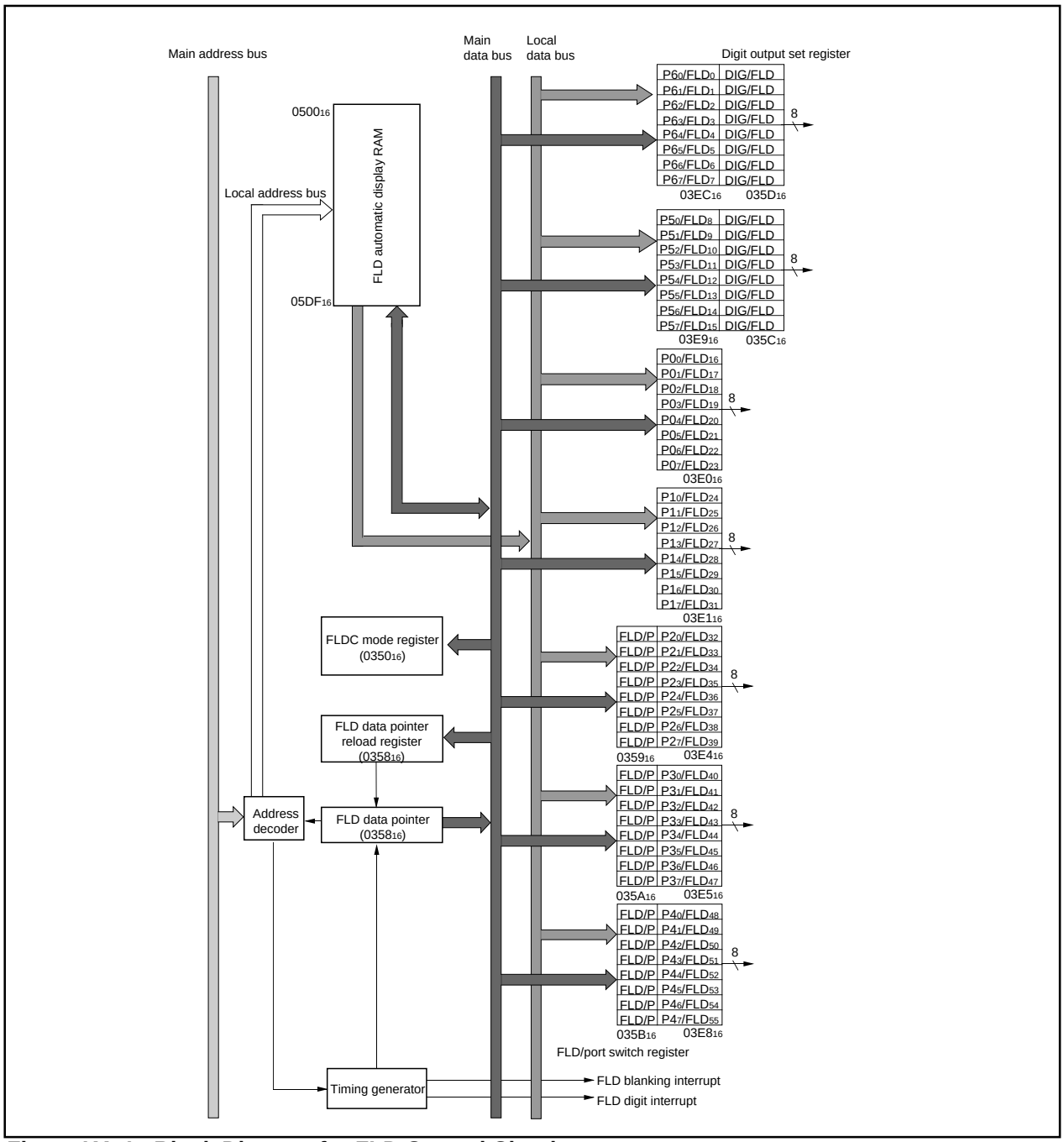
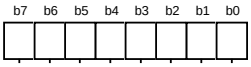


Figure KA-1. Block Diagram for FLD Control Circuit

FLDC mode register

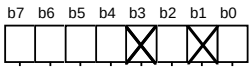


Symbol FLDM | Address 035016 | When reset 0016

Bit symbol	Bit name	Function	R	W
FLDM0	Automatic display control bit	0 : General-purpose mode 1 : Automatic display mode	O	O
FLDM1	Display start bit	0 : Stop display 1 : Display (start to display by switching "0" to "1")	O	O
FLDM2	Tscan control bits	b3b2 00 : FLD digit interrupt (at rising edge of each digit) 01 : 1 X Tdisp 10 : 2 X Tdisp 11 : 3 X Tdisp } FLD blanking interrupt (at falling edge of last digit)	O	O
FLDM3			O	O
FLDM4	Timing number control bit	0 : 16 timing mode 1 : 32 timing mode	O	O
FLDM5	Gradation display mode selection control bit	0 : Not selecting 1 : Selecting (Note)	O	O
FLDM6	Tdisp counter count source selection bit	0 : f(XIN)/32 1 : f(XIN)/128	O	O
FLDM7	High-breakdown voltage port drivability select bit	0 : Drivability strong 1 : Drivability weak	O	O

Note : When a gradation display mode is selected, a number of timing is max. 16 timing.
(Set the timing number control bit to "0".)

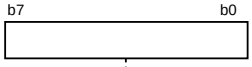
FLD output control register



Symbol FLDCON | Address 035116 | When reset 0016

Bit symbol	Bit name	Function	R	W
FLDCON0	P44 to P47 FLD output reverse bit	0 : Output normally 1 : Reverse output	O	O
Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be "0".			—	—
FLDCON2	P44 to P47 FLD Toff is invalid bit	0 : Perform normally 1 : Toff is invalid	O	O
Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be "0".			—	—
FLDCON4	P97 dimmer output control bit	0 : Output normally 1 : Dimmer output	O	O
FLDCON5	CMOS ports: section of Toff generate/not generate bit	0 : section of Toff does NOT generate 1 : section of Toff generates	O	O
FLDCON6	High-breakdown-voltage ports section of Toff generate/not generate bit	0 : section of Toff does NOT generate 1 : section of Toff generates	O	O
FLDCON7	Toff2 SET/RESET change bit	0 : gradation display data is reset at Toff2 (set at Toff1) 1 : gradation display data is set at Toff2 (reset at Toff1)	O	O

Tdisp time set register

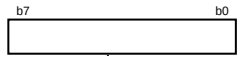


Symbol TDISP | Address 035216 | When reset 0016

Function	Values that can be set	R	W
Counts Tdisp time. Count source is selected by Tdisp counter count source select bit.	016 to FF16	O	O

Figure KA-2. FLDC-related Register(1)

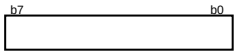
Toff1 time set register



Symbol	Address	When reset
TOFF1	0354_{16}	FF_{16}

Function	Values that can be set	R	W
Counts Toff1 time. Count source is selected by Tdisp counter count source select bit.	3 to FF_{16}	○	○

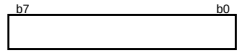
Toff2 time set register



Symbol	Address	When reset
TOFF2	0356_{16}	FF_{16}

Function	Values that can be set	R	W
Counts Toff2 time. Count source is selected by Tdisp counter count source select bit.	3 to FF_{16}	○	○

FLD data pointer



Symbol	Address	When reset
FLDDP	0358_{16}	indeterminate

Function	Values that can be set	R	W
Counts FLD output timing. Set this register to “FLD output data - 1 ”.	1 to $1F_{16}$	○	○

Note: Reading the FLD data pointer takes out the count at that moment.

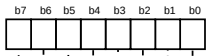
Port P2 FLD / port switch register

Symbol	Address	When reset
P2FPR	0359_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
P2FPR0	Port $P2_0$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P2FPR1	Port $P2_1$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P2FPR2	Port $P2_2$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P2FPR3	Port $P2_3$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P2FPR4	Port $P2_4$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P2FPR5	Port $P2_5$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P2FPR6	Port $P2_6$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P2FPR7	Port $P2_7$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○

Figure KA-2A. FLDC-related Register(2)

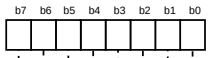
Port P3 FLD / port switch register



Symbol	Address	When reset
P3FPR	$035A_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
P3FPR0	Port $P3_0$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P3FPR1	Port $P3_1$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P3FPR2	Port $P3_2$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P3FPR3	Port $P3_3$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P3FPR4	Port $P3_4$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P3FPR5	Port $P3_5$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P3FPR6	Port $P3_6$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P3FPR7	Port $P3_7$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○

Port P4 FLD / port switch register



Symbol	Address	When reset
P4FPR	$035B_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
P4FPR0	Port $P4_0$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P4FPR1	Port $P4_1$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P4FPR2	Port $P4_2$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P4FPR3	Port $P4_3$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P4FPR4	Port $P4_4$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P4FPR5	Port $P4_5$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P4FPR6	Port $P4_6$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○
P4FPR7	Port $P4_7$ FLD/port switch bit	0 : Normal port 1 : FLD output port	○	○

Port P5 digit output set register

Symbol	Address	When reset
P5DOR	$035C_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
P5DOR0	Port $P5_0$ FLD/digit switch bit	0 : FLD output 1 : Digit output	○	○
P5DOR1	Port $P5_1$ FLD/digit switch bit	0 : FLD output 1 : Digit output	○	○
P5DOR2	Port $P5_2$ FLD/digit switch bit	0 : FLD output 1 : Digit output	○	○
P5DOR3	Port $P5_3$ FLD/digit switch bit	0 : FLD output 1 : Digit output	○	○
P5DOR4	Port $P5_4$ FLD/digit switch bit	0 : FLD output 1 : Digit output	○	○
P5DOR5	Port $P5_5$ FLD/digit switch bit	0 : FLD output 1 : Digit output	○	○
P5DOR6	Port $P5_6$ FLD/digit switch bit	0 : FLD output 1 : Digit output	○	○
P5DOR7	Port $P5_7$ FLD/digit switch bit	0 : FLD output 1 : Digit output	○	○

Figure KA-2B. FLDC-related Register(3)

Port P6 digit output set register

Symbol	Address	When reset
P6DOR	$035D_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
P6DOR0	Port $P6_0$ FLD/digit switch bit	0 : FLD output 1 : Digit output	○	○
P6DOR1	Port $P6_1$ FLD/digit switch bit	0 : FLD output 1 : Digit output	○	○
P6DOR2	Port $P6_2$ FLD/digit switch bit	0 : FLD output 1 : Digit output	○	○
P6DOR3	Port $P6_3$ FLD/digit switch bit	0 : FLD output 1 : Digit output	○	○
P6DOR4	Port $P6_4$ FLD/digit switch bit	0 : FLD output 1 : Digit output	○	○
P6DOR5	Port $P6_5$ FLD/digit switch bit	0 : FLD output 1 : Digit output	○	○
P6DOR6	Port $P6_6$ FLD/digit switch bit	0 : FLD output 1 : Digit output	○	○
P6DOR7	Port $P6_7$ FLD/digit switch bit	0 : FLD output 1 : Digit output	○	○

Figure KA-2C. FLDC-related Register(4)

FLD automatic display pins

P0 to P6 are the pins capable of automatic display output for the FLD. The FLD start operating by setting the automatic display control bit (bit 0 at address 0350_{16}) to “1”. There is the FLD output function that outputs RAM contents from the port every timing or the digit output function that drives the port high with digit timing. The FLD can be displayed using the FLD output for the segments and the digit or FLD output for the digits. When using the FLD output for the digits, be sure to write digit display patterns to the RAM in advance. The remaining segment and digit lines can be used as general-purpose ports. Settings of each port are shown below.

Table KA-1. Pins in FLD Automatic Display Mode

Port Name	Automatic Display Pins	Setting Method
P5, P6	FLD_0 to FLD_{15}	The individual bits of the digit output set register (address $035C_{16}$, $035D_{16}$) can set each pin either FLD port (“0”) or digit port (“1”). When the pins are set for the digit port, the digit pulse output function is enabled, so the digit pulses can always be output regardless the value of FLD automatic display RAM.
P0, P1	FLD_{16} to FLD_{31}	FLD exclusive use port (automatic display control bit (bit 0 of address 0350_{16})=“1”)
P2, P3, $P4_4$ to $P4_3$	FLD_{32} to FLD_{51}	The individual bits of the FLD/port switch register (addresses 0359_{16} to $035B_{16}$) can set each pin to either FLD port (“1”) or general-purpose port (“0”).
$P4_4$ to $P4_7$	FLD_{52} to FLD_{55}	The individual bits of the FLD/port switch register (address $035B_{16}$) can set each pin to either FLD port (“1”) or general-purpose port (“0”). The digit pulse output function turns to available, and the digit pulse can output by setting of the FLD output set register (address 0351_{16}). The port output format is the CMOS output. When using the port as a display pin, a driver must be installed externally.

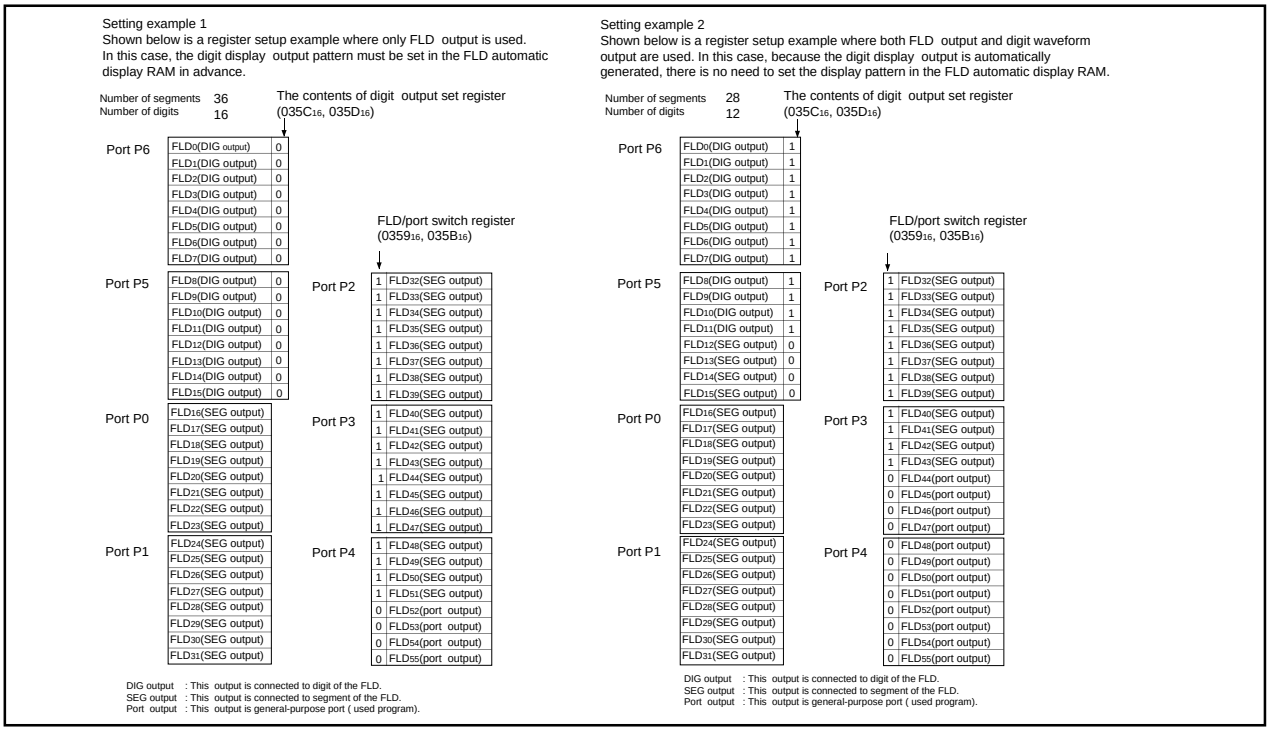


Figure KA-3. Segment/Digit Setting Example

FLD automatic display RAM

The FLD automatic display RAM uses the 224 bytes of addresses 0500_{16} to $05DF_{16}$. For FLD, the 3 modes of 16-timing ordinary mode, 16-timing•gradation display mode and 32-timing mode are available depending on the number of timings and the use/not use of gradation display.
The automatic display RAM in each mode is as follows:

(1) 16-timing•Ordinary Mode

This mode is used when the display timing is 16 or less. The 112 bytes of addresses 0570_{16} to $05DF_{16}$ are used as a FLD display data store area. Because addresses 0500_{16} to $056F_{16}$ are not used as the automatic display RAM, they can be the ordinary RAM.

(2) 16-timing•Gradation Display Mode

This mode is used when the display timing is 16 or less, in which mode each segment can be set for dark or bright display. The 224 bytes of addresses 0500_{16} to $05DF_{16}$ are used. The 112 bytes of addresses 0570_{16} to $05DF_{16}$ are used as an FLD display data store area, while the 112 bytes of addresses 0500_{16} to $056F_{16}$ are used as a gradation display control data store area.

(3) 32-timing Mode

This mode is used when the display timing is 16 or greater. This mode can be used for up to 32-timing. The 224 bytes of addresses 0500_{16} to $05DF_{16}$ are used as an FLD display data store area.

The FLD data pointer (address 0358_{16}) is a register to count display timings. This pointer has a reload register and when the terminal count is reached, it starts counting over again after being reloaded with the initial count. Make sure the timing count – 1 is set to the FLD data pointer. When writing data to this address, the data is written to the FLD data pointer reload register; when reading data from this address, the value in the FLD data pointer is read.

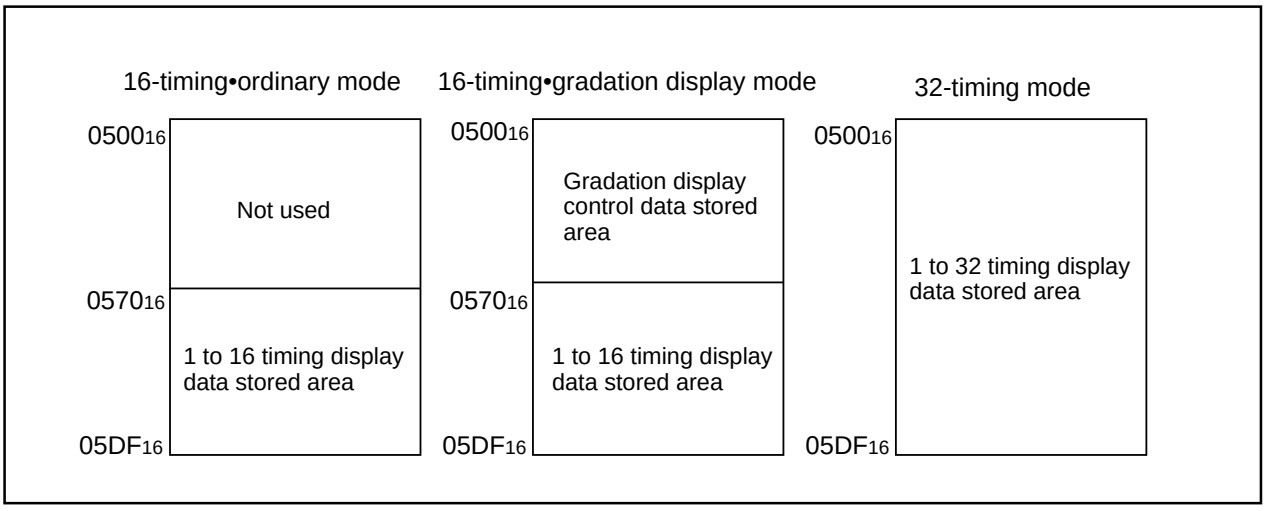


Figure KA-4. FLD Automatic Display RAM Assignment

Data setup

(1) 16-timing•Ordinary Mode

The area of addresses 0570_{16} to $05DF_{16}$ are used as a FLD automatic display RAM.

When data is stored in the FLD automatic display RAM, the last data of FLD port P4 is stored at address 0570_{16}, the last data of FLD port P3 is stored at address 0580_{16}, the last data of FLD port P2 is stored at address 0590_{16}, the last data of FLD port P1 is stored at address $05A0_{16}$, the last data of FLD port P0 is stored at address $05B0_{16}$, the last data of FLD port P5 is stored at address $05C0_{16}$, and the last data of FLD port P6 is stored at address $05D0_{16}$, to assign in sequence from the last data respectively.

The first data of the FLD port P4, P3, P2, P1, P0, P5, and P6 is stored at an address which adds the value of (the timing number – 1) to the corresponding address 0570_{16}, 0580_{16}, 0590_{16}, $05A0_{16}$, $05B0_{16}$, $05C0_{16}$ and $05DF_{16}$.

Set the FLD data pointer reload register to the value given by the number of digits – 1.

(2) 16-timing•Gradation Display Mode

Display data setting is performed in the same way as that of the 16-timing•ordinary mode. Gradation display control data is arranged at an address resulting from subtracting 0070_{16} from the display data store address of each timing and pin. Bright display is performed by setting “0”, and dark display is performed by setting “1” .

(3) 32-timing Mode

The area of addresses 0500_{16} to $05DF_{16}$ are used as a FLD automatic display RAM.

When data is stored in the FLD automatic display RAM, the last data of FLD port P4 is stored at address 0500_{16}, the last data of FLD port P3 is stored at address 0520_{16}, the last data of FLD port P2 is stored at address 0540_{16}, the last data of FLD port P1 is stored at address 0560_{16}, the last data of FLD port P0 is stored at address 0580_{16}, the last data of FLD port P5 is stored at address $05A0_{16}$, and the last data of FLD port P6 is stored at address $05C0_{16}$, to assign in sequence from the last data respectively.

The first data of the FLD port P4, P3, P2, P0, P1, P5, and P6 is stored at an address which adds the value of (the timing number – 1) to the corresponding address 0500_{16}, 0520_{16}, 0540_{16}, 0560_{16}, 0580_{16}, $05A0_{16}$ and $05C0_{16}$.

Set the FLD data pointer reload register to the value given by the number of digits - 1.

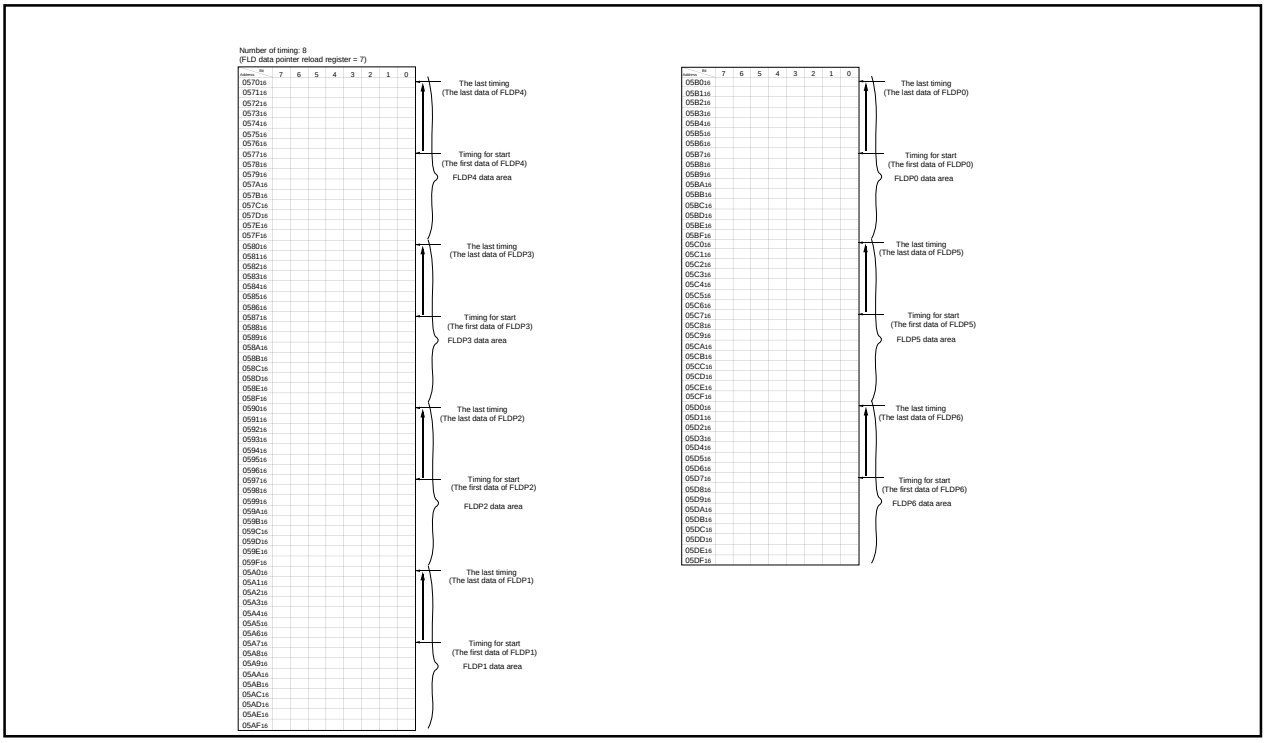


Figure KA-5. Example of Using the FLD Automatic Display RAM in 16-timing•Ordinary Mode

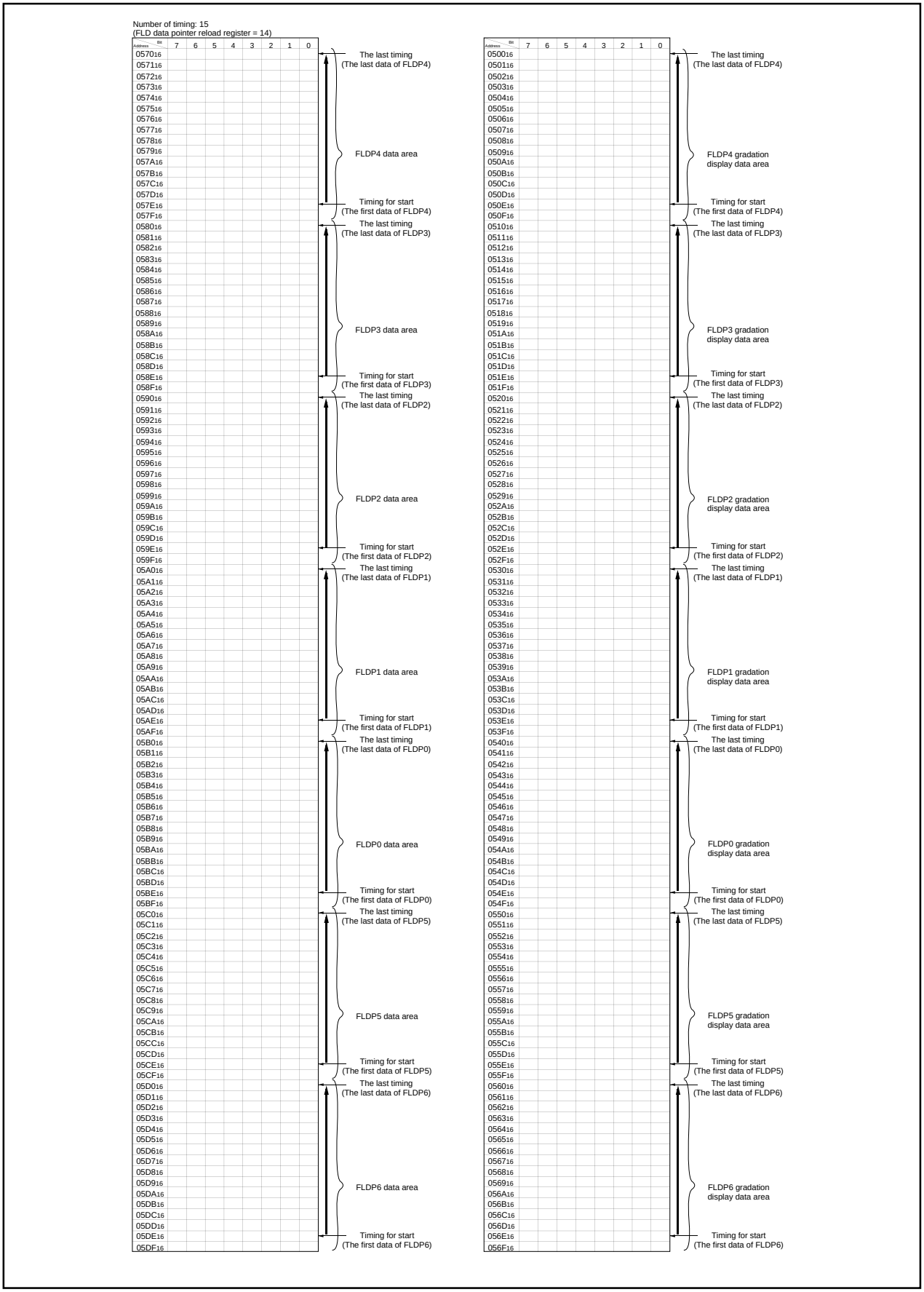


Figure KA-6. Example of Using the FLD Automatic Display RAM in 16-timing•Gradation Display Mode

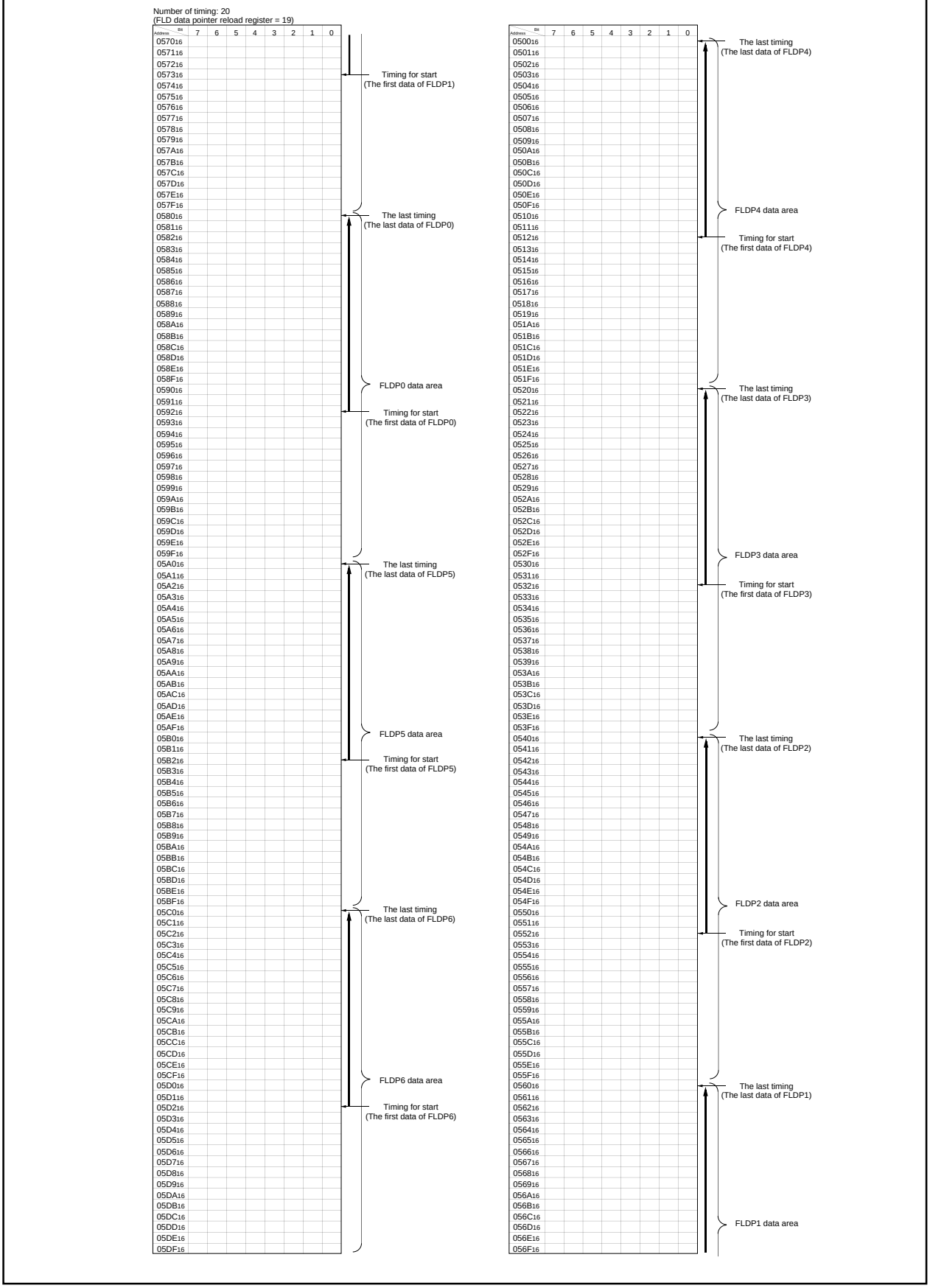


Figure KA-7. Example of Using the FLD Automatic Display RAM in 32-timing Mode

Timing setting

Each timing is set by the FLDC mode register, Tdisp time set register, Toff1 time set register, and Toff2 time set register.

•Tdisp time setting

The Tdisp time represents the length of display timing. In non-gradation display mode, it consists of a FLD display output period and a Toff1 time. In gradation display mode, it consists of the display output period and Toff1 time plus a low signal output period for dark display. Set the Tdisp time by the Tdisp counter count source select bit of the FLDC mode register and the Tdisp time set register. Supposing that the value of the Tdisp time set register is n, the Tdisp time is represented as Tdisp = (n+1) x t (t: count source). When the Tdisp counter count source select bit of the FLDC mode register is "0" and the value of the Tdisp time set register is 200 ($C8_{16}$), the Tdisp time is: Tdisp = (200+1) x 3.2 (at X_{IN}= 10 MHz) = 643 μs. When reading the Tdisp time set register, the value in the counter is read out.

•Toff1 time setting

The Toff1 time represents a non-output (low signal output) time to prevent blurring of FLD, and to dim the display. Use the Toff1 time set register to set this Toff1 time. Make sure the value set to Toff1 is smaller than Tdisp and Toff2. Supposing that the value of the Toff1 time set register is n1, the Toff1 time is represented as Toff1 = n1 x t. When the Tdisp counter count source select bit of the FLDC mode register is "0" and the value of the Toff1 time set register is 30 ($1E_{16}$), Toff1 = 30 x 3.2 (at X_{IN} = 10 MHz) = 96 μs.

•Toff2 time setting

The Toff2 time is provided for dark display. For bright display, the FLD display output remains effective until the counter that is counting Tdisp reaches the terminal count. For dark display, however, "L" (or "off") signal is output when the counter that is counting Toff2 reaches the terminal count. This Toff2 time setting is valid only for FLD ports which are in the gradation display mode and whose gradation display control RAM value is "1" .

Set the Toff2 time by the Toff2 time set register. Make sure the value set to Toff2 is smaller than Tdisp but larger than Toff1. Supposing that the value of the Toff2 time set register is n2, the Toff2 time is represented as Toff2 = n2 x t. When the Tdisp counter count source select bit of the FLDC mode register is "0" and the value of the Toff2 time set register is 180 ($B4_{16}$), Toff2 = 180 x 3.2 (at X_{IN} = 10 MHz) = 576 μs.

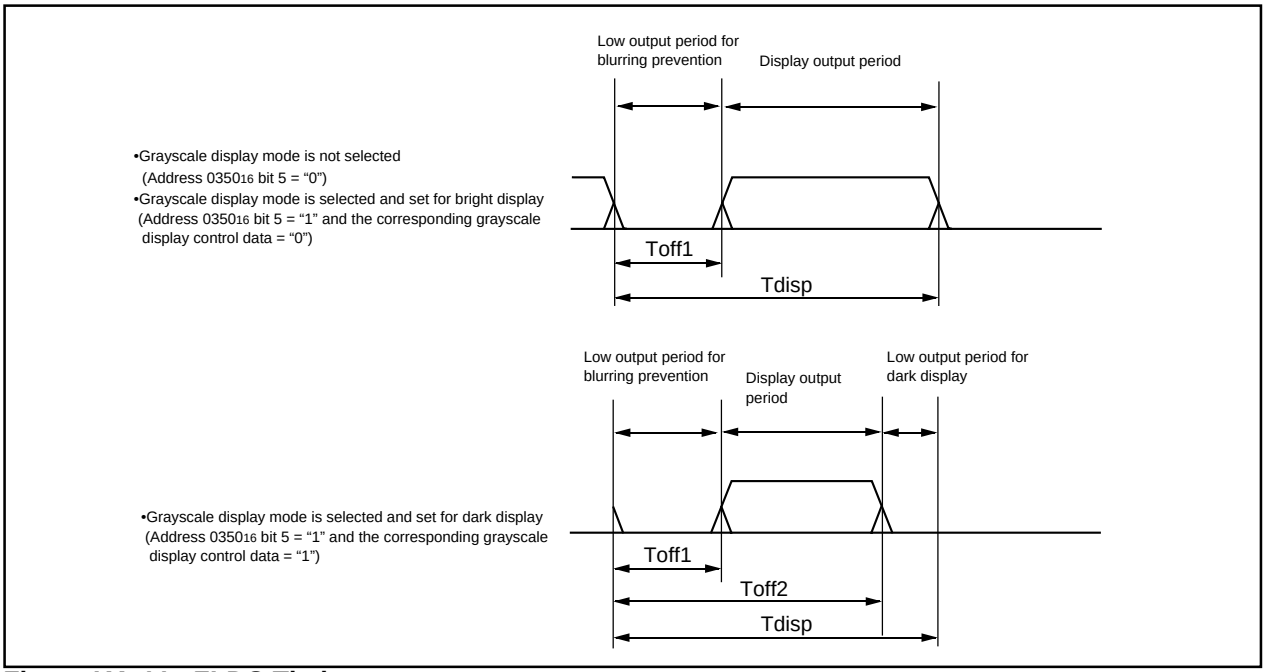


Figure KA-11. FLDC Timing

FLD automatic display start

Automatic display starts by setting both the automatic display control bit (bit 0 of address 0350_{16}) and the display start bit (bit 1 of address 0350_{16}) to "1". The RAM content at a location apart from the start address of the automatic display RAM for each port by (FLD data pointer (address 0358_{16}) – 1) is output to each port. The FLD data pointer (address 0358_{16}) counts down in the Tdisp interval. When the count "FF_{16}" is reached, the pointer is reloaded and starts counting over again. Before setting the display start bit (bit 1 of address 0350_{16}) to "1", be sure to set the FLD/port switch register, FLD/DIG switch register, FLDC mode register, Tdisp time set register, Toff1 time set register, Toff2 time set register, and FLD data pointer.

During FLD automatic display, bit 1 of the FLDC mode register (address 0350_{16}) always keeps "1", and FLD automatic display can be interrupted by writing "0" to bit 1.

Key-scan and interrupt

Either a FLD digit interrupt or FLD blanking interrupt can be selected using the Tscan control bits (bits 2, 3 of address 0350_{16}).

The FLD digit interrupt is generated when the Toff1 time in each timing expires (at rising edge of digit output). Key scanning that makes use of FLD digits can be achieved using each FLD digit interrupt. To use FLD digit interrupts for key scanning, follow the procedure described below.

(1) Read the port value each time the interrupt occurs.

(2) The key is fixed on the last digit interrupt.

The digit positions output can be determined by reading the FLD data pointer (address 0358_{16}).

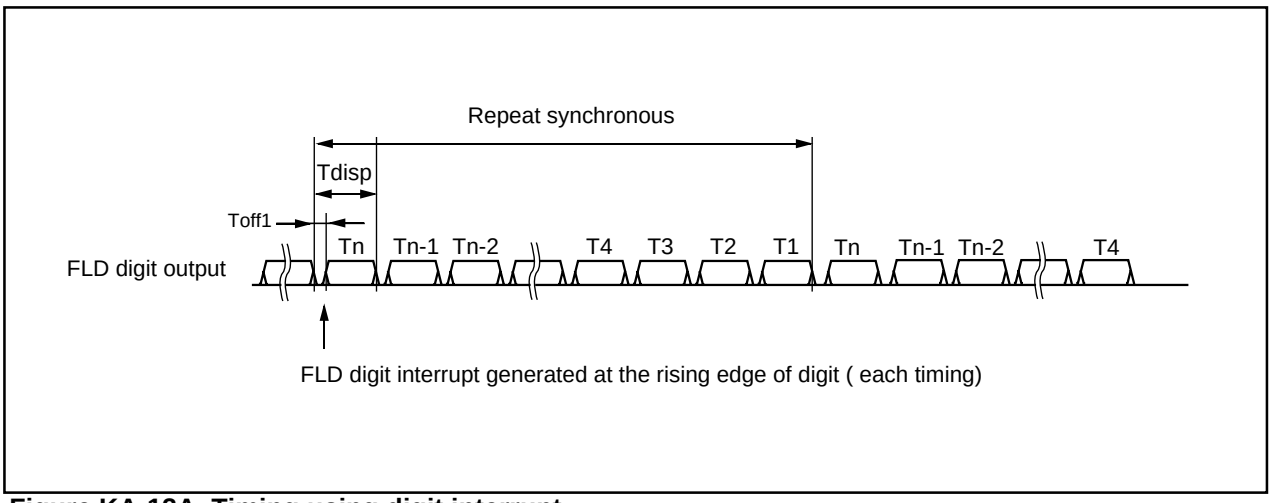


Figure KA-12A. Timing using digit interrupt

The FLD blanking interrupt is generated when the FLD data pointer (address 0358$_{16}$) reaches "FF$_{16}$". The FLD automatic display output is turned off for a duration of 1 x Tdisp, 2 x Tdisp, or 3 x Tdisp depending on post-interrupt settings. During this time, key scanning that makes use of FLD segments can be achieved.

When a key-scan is performed with the segment during key-scan blanking period Tscan, take the following sequence:

1. Write "0" to bit 0 of the FLDC mode register (address 0350$_{16}$).
2. Set the port corresponding to the segment for key-scan to the output port.
3. Perform the key-scan.
4. After the key-scan is performed, write "1" to bit 0 of FLDC mode register (address 0350$_{16}$).

•Note:

When performing a key-scan according to the above steps 1 to 4, take the following points into consideration.

1. Do not set "0" in bit 1 of the FLDC mode register (address 0350$_{16}$).
2. Do not set "1" in the ports corresponding to digits.

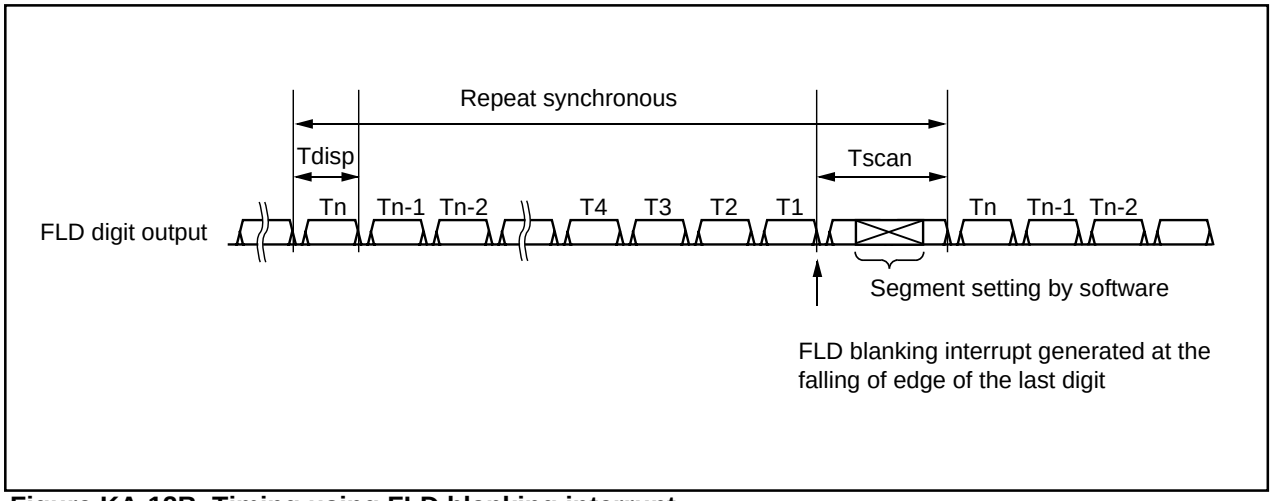


Figure KA-12B. Timing using FLD blanking interrupt

P44 to P47 Expansion Function

P44 to P47 are CMOS output-type ports. FLD digit outputs can be increased as many as 16 lines by connecting a 4-bit to 16-bit decoder to these ports. P44 to P47 have the function to allow for connection to a 4-bit to 16-bit decoder.

(1) P44 to P47 Toff invalid Function

This function disables the Toff1 time and Toff2 time and outputs display data for the duration of Tdisp. (See Figure KA-13.) This can be accomplished by setting the P44 to P47 Toff disable bit (address 035016 bit 2) to "1".

Unlike the Toff section generate/not generate function, this function disables all display data.

(2) Dimmer signal output Function

This function allows a dimmer signal creation signal to be output from DIMOUT (P97). The dimmer function can be materialized by controlling the decoder with this signal. (See Figure KA-13.) This function can be set by writing P97 dimmer output control bit (bit 4 of address 035116) to "1".

(3) P44 to P47 FLD Output Reverse Bit

P44 to P47 are provided with a function to reverse the polarity of the FLD output. This function is useful in adjusting the polarity when using an externally installed driver.

The output polarity can be reversed by setting bit 0 of the FLD output control register (address 035116) to "1" .

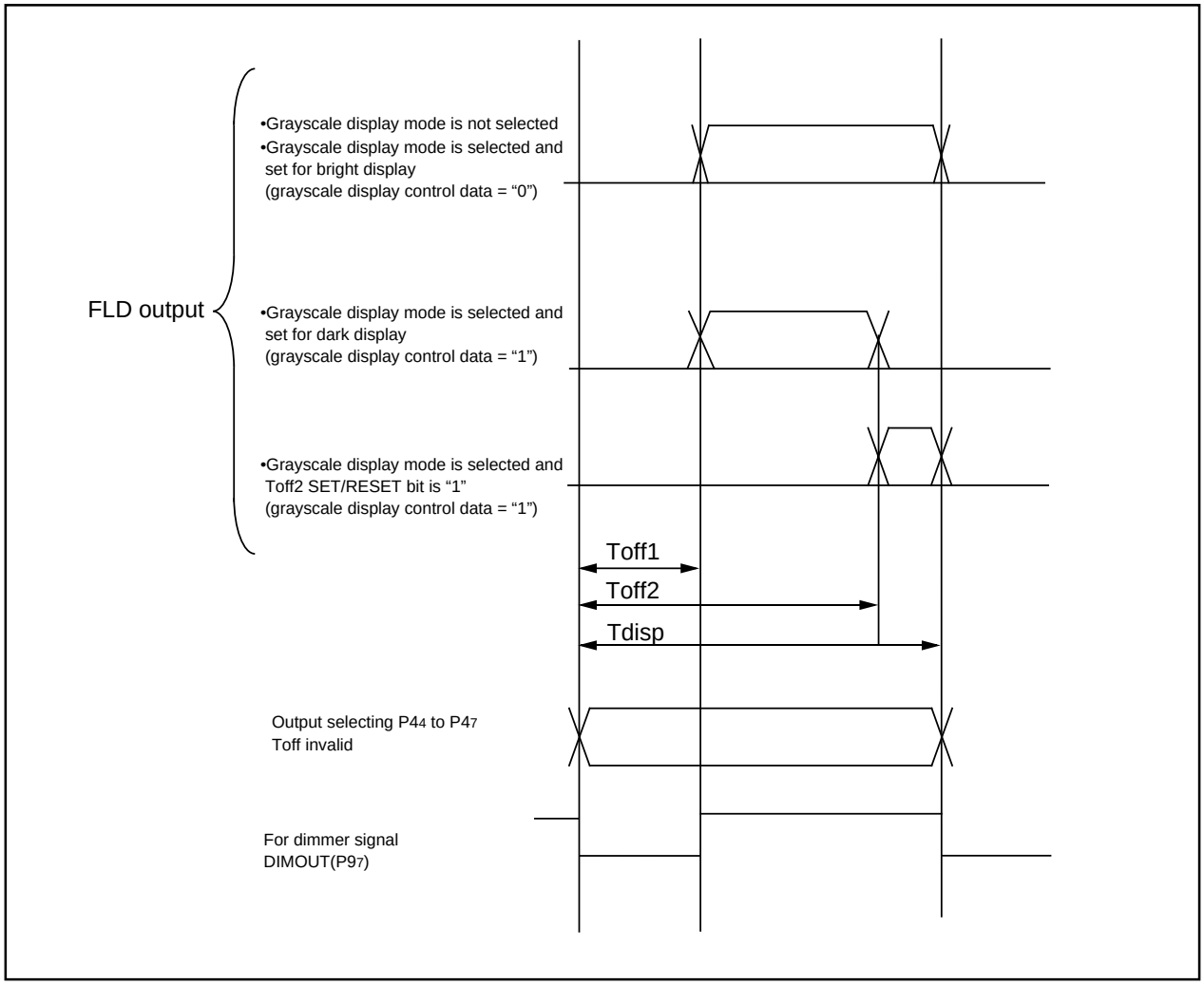


Figure KA-13. P4 to P47 FLD Output pulses

Toff section generate/not generate Function

The function is for reduction of useless noises which generated as every switching of ports, because of the combined capacity of among FLD ports. In case the continuous data output to each FLD ports, the Toff1 section of the continuous parts is not generated. (See Figure KA-15)

If it needs Toff1 section on FLD pulses, set “CMOS ports: section of Toff generate / not generate bit” to “1” and set “high-breakdown-voltage ports: section of Toff generate / not generate bit” to “1”. High-breakdown-voltage ports (P5, P6, P3, P2, P1, P0, $P4_0$ to $P4_3$, total 52 pins) generate Toff1 section, by setting “high-breakdown-voltage ports: section of Toff generate / not generate bit” to “1”.

The CMOS ports ($P4_4$ to $P4_7$, total 4 pins) generate Toff1 section, by setting “high-breakdown-voltage ports: section of Toff generate / not generate bit” to “1”.

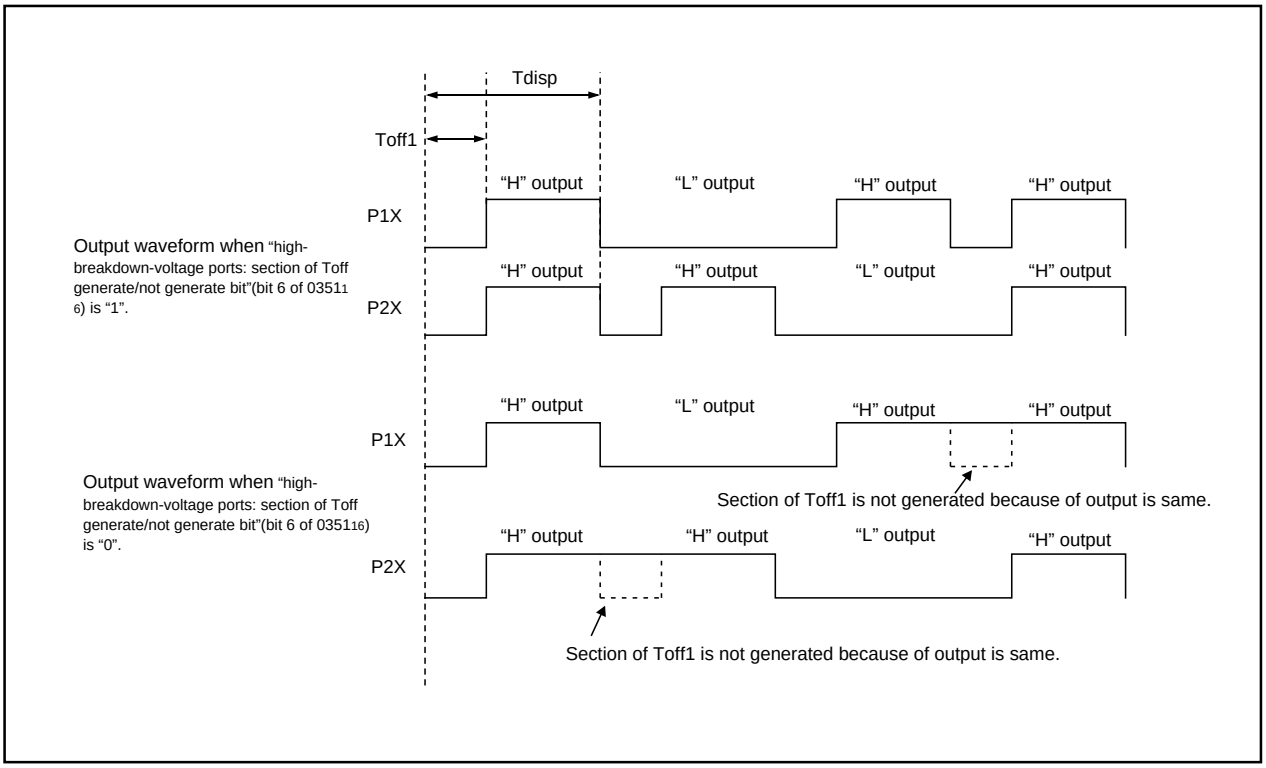


Fig. KA-15. Toff Section Generated/not generated Function

Toff2 SET/RESET change bit

In gradation display mode, the values set by the Toff2 time set register (TOFF2) are effective. When the FLD output control register (bit 7 of address 0351_{16}) in the initial state = “0”, RAM data is output to the FLD output ports (SET) at the time that is set by TOFF1 and is turned to “0” (RESET) at the time that is set by TOFF2. When bit 7 = “1”, RAM data is output (SET) at the time that is set by TOFF2 and is turned to “0” (RESET) when the Tdisp time expires.

Digit pulses output Function

P5$_0$ to P5$_7$ and P6$_0$ to P6$_7$ allow digit pulses to be output using the FLD/digit switch register. Set the digit output set register by writing as many consecutive 1s as the timing count from P6$_0$. The contents of FLD automatic display RAM for the ports that have been selected for digit output are disabled, and the pulse shown in Figure KA-16 is output automatically. In gradation display mode use, T$_{off2}$ time becomes effective for the port which selected digit output. Because the contents of FLD automatic display RAM are disabled, the segment data can be changed easily even when segment data and digit data coexist at the same address in the FLD automatic display RAM.

This function is effective in 16-timing normal mode and 16-timing gradation display mode. If a value is set exceeding the timing count (FLD data pointer reload register's set value + 1) for any port, the output of such port is “L”.

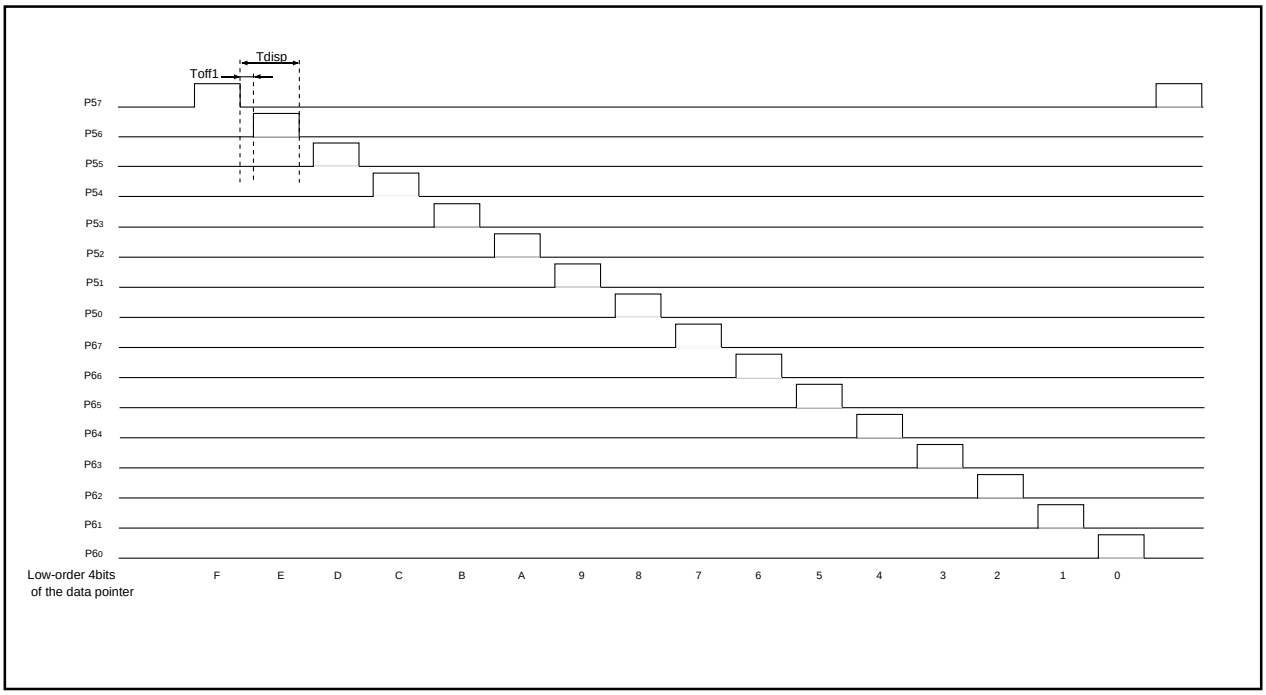


Fig. KA-16. Digit Pulses Output Function

Timer

There are eight 16-bit timers. These timers can be classified by function into timers A (five) and timers B (three). All these timers function independently. Figures FB-1 show the block diagram of timers.

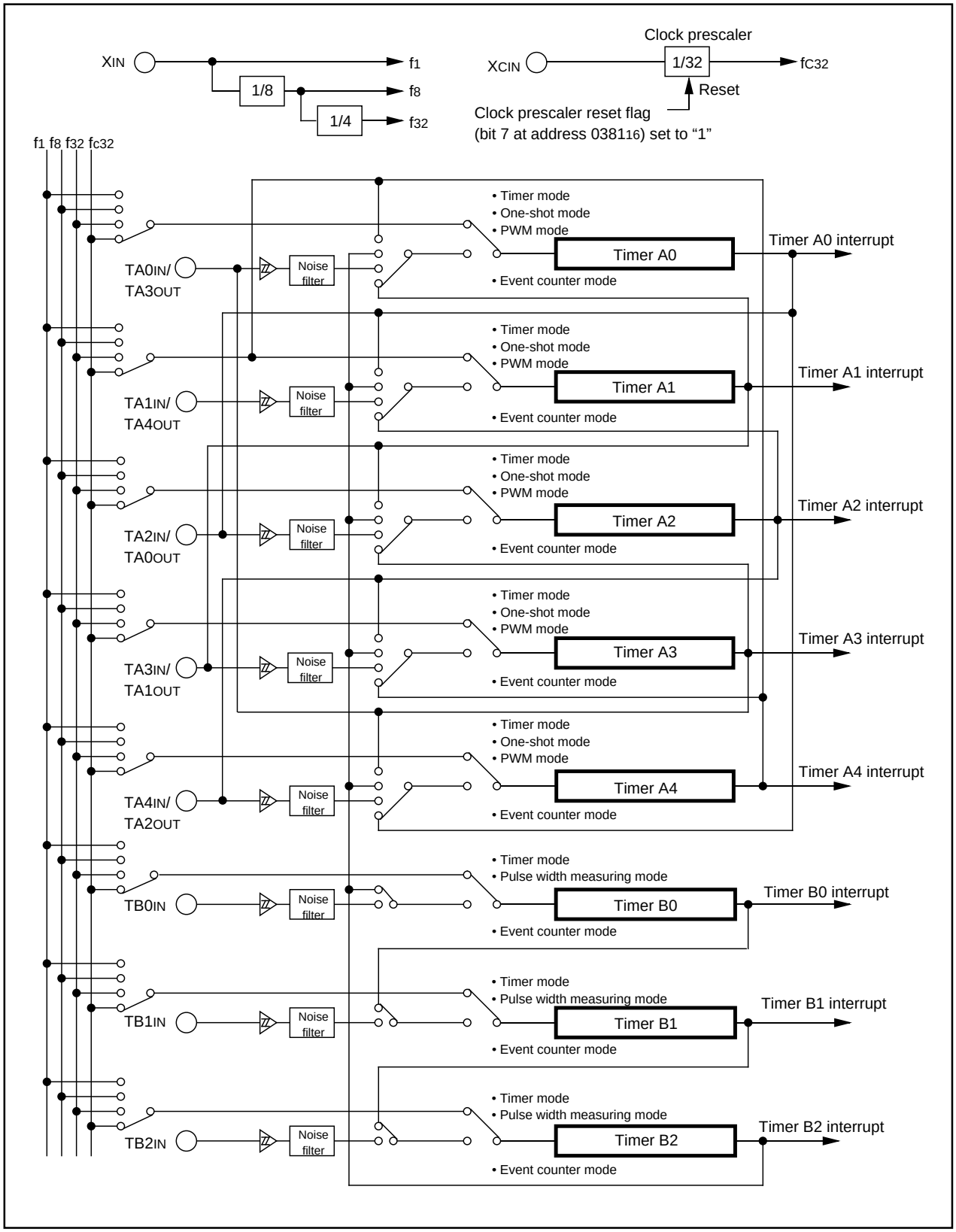


Figure FB-1. Timer block diagram

Timer A

Figure FB-2 shows the block diagram of timer A. Figures FB-3 to FB-5 show the timer A-related registers. Except in event counter mode, timers A0 through A4 all have the same function. Use the timer Ai mode register (i = 0 to 4) bits 0 and 1 to choose the desired mode.

Timer A has the four operation modes listed as follows:

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external source or a timer's over flow.
- One-shot timer mode: The timer stops counting when the count reaches "0000$_{16}$".
- Pulse width modulation (PWM) mode: The timer outputs pulses of a given width.

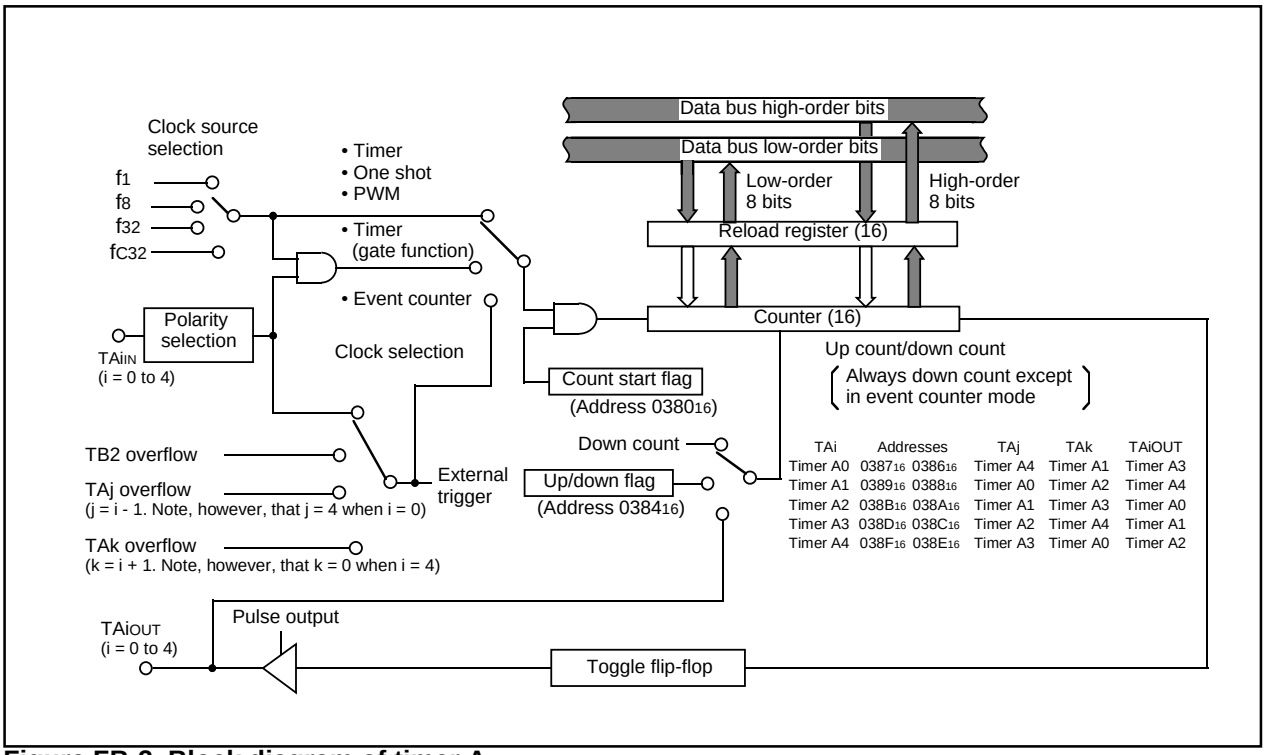


Figure FB-2. Block diagram of timer A

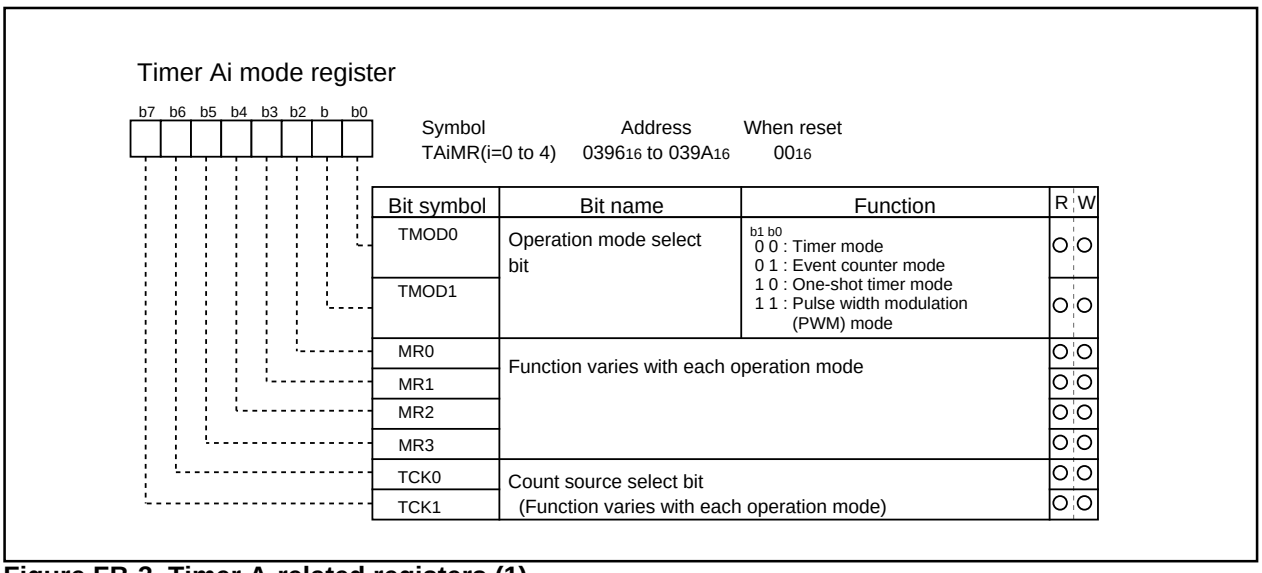
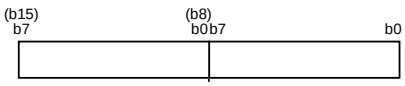


Figure FB-3. Timer A-related registers (1)

Timer Ai register (Note)

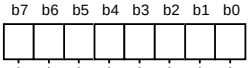


Symbol	Address	When reset
TA0	0387_{16},0386_{16}	Indeterminate
TA1	0389_{16},0388_{16}	Indeterminate
TA2	$038B_{16}$,$038A_{16}$	Indeterminate
TA3	$038D_{16}$,$038C_{16}$	Indeterminate
TA4	$038F_{16}$,$038E_{16}$	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts an internal count source	0000_{16} to FFFF	○	○
• Event counter mode Counts pulses from an external source or timer overflow	0000_{16} to $FFFF_{16}$	○	○
• One-shot timer mode Counts a one shot width	0000_{16} to $FFFF_{16}$	×	○
• Pulse width modulation mode (16-bit PWM) Functions as a 16-bit pulse width modulator	0000_{16} to $FFFE_{16}$	×	○
• Pulse width modulation mode (8-bit PWM) Timer low-order address functions as an 8-bit prescaler and high-order address functions as an 8-bit pulse width modulator	00_{16} to FE_{16} (Both high-order and low-order addresses)	×	○

Note: Read and write data is in 16-bit units.

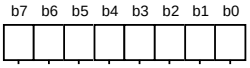
Count start flag



Symbol	Address	When reset
TABSR	0380_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TA1S	Timer A1 count start flag		○	○
TA2S	Timer A2 count start flag		○	○
TA3S	Timer A3 count start flag		○	○
TA4S	Timer A4 count start flag		○	○
TB0S	Timer B0 count start flag		○	○
TB1S	Timer B1 count start flag		○	○
TB2S	Timer B2 count start flag		○	○

Up/down flag

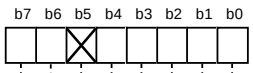


Symbol	Address	When reset
UDF	0384_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TA0UD	Timer A0 up/down flag	0 : Down count 1 : Up count This specification becomes valid when the up/down flag content is selected for up/down switching cause	○	○
TA1UD	Timer A1 up/down flag		○	○
TA2UD	Timer A2 up/down flag		○	○
TA3UD	Timer A3 up/down flag		○	○
TA4UD	Timer A4 up/down flag		○	○
TA2P	Timer A2 two-phase pulse signal processing select bit	0 : two-phase pulse signal processing disabled 1 : two-phase pulse signal processing enabled When not using the two-phase pulse signal processing function, set the select bit to “0”	×	○
TA3P	Timer A3 two-phase pulse signal processing select bit		×	○
TA4P	Timer A4 two-phase pulse signal processing select bit		×	○

Figure FB-4. Timer A-related registers (2)

One-shot start flag

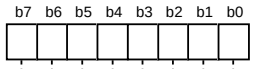


Symbol	Address	When reset
ONSF	0382_{16}	$00X00000_2$

Bit symbol	Bit name	Function	R	W
TA0OS	Timer A0 one-shot start flag	1: Timer start When read, the value is “0”	O	O
TA1OS	Timer A1 one-shot start flag		O	O
TA2OS	Timer A2 one-shot start flag		O	O
TA3OS	Timer A3 one-shot start flag		O	O
TA4OS	Timer A4 one-shot start flag		O	O
Nothing is assigned. This bit can neither be set nor reset. When read, the content is indeterminate.			—	—
TA0TGL	Timer A0 event/trigger select bit	b7 b6 0 0 : Input on TA0IN is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA4 overflow is selected 1 1 : TA1 overflow is selected	O	O
TA0TGH			O	O

Note: Set the corresponding port direction register to “0”.
When TAiIN is selected, TAiOUT assigned on same pin can not be used. (i=0 to 4)

Trigger select register

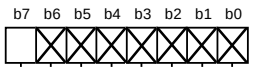


Symbol	Address	When reset
TRGSR	0383_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TA1TGL	Timer A1 event/trigger select bit	b1 b0 0 0 : Input on TA1IN is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA0 overflow is selected 1 1 : TA2 overflow is selected	O	O
TA1TGH			O	O
TA2TGL	Timer A2 event/trigger select bit	b3 b2 0 0 : Input on TA2IN is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA1 overflow is selected 1 1 : TA3 overflow is selected	O	O
TA2TGH			O	O
TA3TGL	Timer A3 event/trigger select bit	b5 b4 0 0 : Input on TA3IN is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA2 overflow is selected 1 1 : TA4 overflow is selected	O	O
TA3TGH			O	O
TA4TGL	Timer A4 event/trigger select bit	b7 b6 0 0 : Input on TA4IN is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA3 overflow is selected 1 1 : TA0 overflow is selected	O	O
TA4TGH			O	O

Note: Set the corresponding port direction register to “0”.
When TAiIN is selected, TAiOUT assigned on same pin can not be used. (i=0 to 4)

Clock prescaler reset flag



Symbol	Address	When reset
CPSRF	0381_{16}	$0XXXXXXX_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	O	O

Figure FB-5. Timer A-related registers (3)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table FB-1.) Figure FB-6 shows the timer Ai mode register in timer mode.

Table FB-1. Specifications of timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• Down count • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	When the timer underflows
TAiIN pin function	Programmable I/O port or gate input
TAiOUT pin function	Programmable I/O port or pulse output
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)
Select function	• Gate function Counting can be started and stopped by the TAiIN pin's input signal • Pulse output function Each time the timer underflows, the TAiOUT pin's polarity is reversed

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

Symbol: TAiMR(i=0 to 4) Address: 039_{16} to $039A_{16}$ When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TAiOUT pin is a normal port pin) 1 : Pulse is output (Note 1) (TAiOUT pin is a pulse output pin)	○	○
MR1	Gate function select bit	b4 b3 0 X (Note 2): Gate function not available (TAiIN pin is a normal port pin) 1 0 : Timer counts only when TAiIN pin is held "L" (Note 3)	○	○
MR2		1 1 : Timer counts only when TAiIN pin is held "H" (Note 3)	○	○
MR3	0 (Must always be fixed to "0" in timer mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: The settings of the corresponding port register and port direction register are invalid.
Note 2: The bit can be "0" or "1".
Note 3: Set the corresponding port direction register to "0".

Figure FB-6. Timer Ai mode register in timer mode

(2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. Timers A0 and A1 can count a single-phase external signal. Timers A2, A3, and A4 can count a single-phase and a two-phase external signal. Table FB-2 lists timer specifications when counting a single-phase external signal. Figure FB-7 shows the timer Ai mode register in event counter mode.

Table FB-3 lists timer specifications when counting a two-phase external signal. Figure FB-8 shows the timer Ai mode register in event counter mode.

Table FB-2. Timer specifications in event counter mode (when not processing two-phase pulse signal)

Item	Specification
Count source	•External signals input to TAiIN pin (effective edge can be selected by software) •TB2 overflow, TAj overflow
Count operation	•Up count or down count can be selected by external signal or software •When the timer overflows or underflows, the reload register's content is reloaded and the timer starts over again.(Note)
Divide ratio	1/ ($FFFF_{16}$ - n + 1) for up count 1/ (n + 1) for down count n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer overflows or underflows
TAiIN pin function	Programmable I/O port or count source input
TAiOUT pin function	Programmable I/O port, pulse output, or up/down count select input
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	•When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter •When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)
Select function	•Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it •Pulse output function Each time the timer overflows or underflows, the TAiOUT pin's polarity is reversed

Note: This does not apply when the free-run function is selected.

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

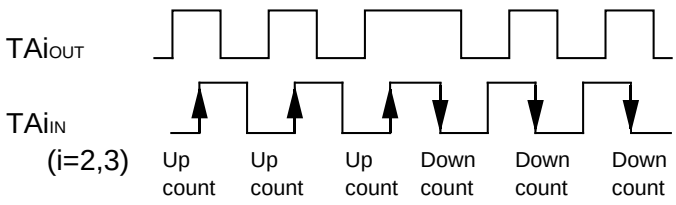
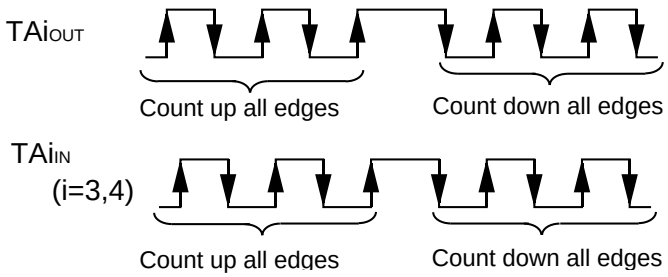
Symbol: TAiMR(i = 0, 1) — Address: 039_{16}, 039_{16} — When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode (Note 1)	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TAiOUT pin is a normal port pin) 1 : Pulse is output (Note 2) (TAiOUT pin is a pulse output pin)	○	○
MR1	Count polarity select bit (Note 3)	0 : Counts external signal's falling edge 1 : Counts external signal's rising edge	○	○
MR2	Up/down switching cause select bit	0 : Up/down flag's content 1 : TAiOUT pin's input signal (Note 4)	○	○
MR3	0 (Must always be fixed to "0" in event counter mode)		○	○
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Invalid in event counter mode Can be "0" or "1"		○	○

Note 1: In event counter mode, the count source is selected by the event / trigger select bit (addresses 0382_{16} and 0383_{16}).
Note 2: The settings of the corresponding port register and port direction register are invalid.
Note 3: Valid only when counting an external signal.
Note 4: When an "L" signal is input to the TAiOUT pin, the downcount is activated. When "H", the upcount is activated. Set the corresponding port direction register to "0".

Figure FB-7. Timer Ai mode register in event counter mode

Table FB-3. Timer specifications in event counter mode (when processing two-phase pulse signal with timer A2,A3 and A4

Item	Specification
Count source	•Two-phase pulse signals input to TAiIN or TAiOUT pin
Count operation	•Up count or down count can be selected by two-phase pulse signal •When the timer overflows or underflows, the reload register content is reloaded and the timer starts over again (Note)
Divide ratio	1/ (FFFF16 - n + 1) for up count 1/ (n + 1) for down count n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	Timer overflows or underflows
TAiIN pin function	Two-phase pulse input
TAiOUT pin function	Two-phase pulse input
Read from timer	Count value can be read out by reading timer A2, A3, or A4 register
Write to timer	•When counting stopped When a value is written to timer A2, A3, or A4 register, it is written to both reload register and counter •When counting in progress When a value is written to timer A2, A3, or A4 register, it is written to only reload register. (Transferred to counter at next reload time.)
Select function	•Normal processing operation The timer counts up rising edges or counts down falling edges on the TAiIN pin when input signal on the TAiOUT pin is "H"  •Multiply-by-4 processing operation If the phase relationship is such that the TAiIN pin goes "H" when the input signal on the TAiOUT pin is "H", the timer counts up rising and falling edges on the TAiOUT and TAiIN pins. If the phase relationship is such that the TAiIN pin goes "L" when the input signal on the TAiOUT pin is "H", the timer counts down rising and falling edges on the TAiOUT and TAiIN pins. 

Note: This does not apply when the free-run function is selected.

Timer Ai mode register
(When not using two-phase pulse signal processing)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

Symbol: TAiMR(i = 2 to 4) | Address: 0398$_{16}$ to 039A$_{16}$ | When reset: 00$_{16}$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	O	O
TMOD1			O	O
MR0	Pulse output function select bit	0 : Pulse is not output (TAiOUT pin is a normal port pin) 1 : Pulse is output (Note 1) (TAiOUT pin is a pulse output pin)	O	O
MR1	Count polarity select bit (Note 2)	0 : Counts external signal's falling edges 1 : Counts external signal's rising edges	O	O
MR2	Up/down switching cause select bit	0 : Up/down flag's content 1 : TAiOUT pin's input signal (Note 3)	O	O
MR3	0 (Must always be “0” in event counter mode)		O	O
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	O	O
TCK1	Two-phase pulse signal processing operation select bit (Note 4)(Note 5)	0 : Normal processing operation 1 : Multiply-by-4 processing operation	O	O

Note 1: The settings of the corresponding port register and port direction register are invalid
Note 2: This bit is valid when only counting an external signal.
Note 3: Set the corresponding port direction register to “0”.
Note 4: This bit is valid for timer A3 mode register.
For timer A2 and A4 mode registers, this bit can be “0 ”or “1”.
Note 5: When performing two-phase pulse signal processing, make sure the two-phase pulse signal processing operation select bit (address 0384$_{16}$) is set to “1”. Also, always be sure to set the event/trigger select bit (addresses 0382$_{16}$ and 0383$_{16}$) to “00”.

Timer Ai mode register
(When using two-phase pulse signal processing)

b7	b6	b5	b4	b3	b2	b1	b0
		0	1	0	0	0	1

Symbol: TAiMR(i = 2 to 4) | Address: 0398$_{16}$ to 039A$_{16}$ | When reset: 00$_{16}$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	O	O
TMOD1			O	O
MR0	0 (Must always be “0” when using two-phase pulse signal processing)		O	O
MR1	0 (Must always be “0” when using two-phase pulse signal processing)		O	O
MR2	1 (Must always be “1” when using two-phase pulse signal processing)		O	O
MR3	0 (Must always be “0” when using two-phase pulse signal processing)		O	O
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	O	O
TCK1	Two-phase pulse processing operation select bit (Note 1)(Note 2)	0 : Normal processing operation 1 : Multiply-by-4 processing operation	O	O

Note 1: This bit is valid for timer A3 mode register.
For timer A2 and A4 mode registers, this bit can be “0” or “1”.
Note 2: When performing two-phase pulse signal processing, make sure the two-phase pulse signal processing operation select bit (address 0384$_{16}$) is set to “1”. Also, always be sure to set the event/trigger select bit (addresses 0382$_{16}$ and 0383$_{16}$) to “00”.

Figure FB-8. Timer Ai mode register in event counter m

(3) One-shot timer mode

In this mode, the timer operates only once. (See Table FB-4.) When a trigger occurs, the timer starts up and continues operating for a given period. Figure FB-9 shows the timer Ai mode register in one-shot timer mode.

Table FB-4. Timer specifications in one-shot timer mode

Item	Specification
Count source	f1, f8, f32, fC32
Count operation	• The timer counts down • When the count reaches 0000_{16}, the timer stops counting after reloading a new count • If a trigger occurs when counting, the timer reloads a new count and restarts counting
Divide ratio	1/n n : Set value
Count start condition	• An external trigger is input • The timer overflows • The one-shot start flag is set (= 1)
Count stop condition	• A new count is reloaded after the count has reached 0000_{16} • The count start flag is reset (= 0)
Interrupt request generation timing	The count reaches 0000_{16}
TAiIN pin function	Programmable I/O port or trigger input
TAiOUT pin function	Programmable I/O port or pulse output
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	•When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				1	0

Symbol: TAiMR(i = 0 to 4) Address: 0396_{16} to $039A_{16}$ When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : One-shot timer mode	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TAiOUT pin is a normal port pin) 1 : Pulse is output (Note 1) (TAiOUT pin is a pulse output pin)	○	○
MR1	External trigger select bit (Note 2)	0 : Falling edge of TAiIN pin's input signal (Note 3) 1 : Rising edge of TAiIN pin's input signal (Note 3)	○	○
MR2	Trigger select bit	0 : One-shot start flag is valid 1 : Selected by event/trigger select register	○	○
MR3	0 (Must always be "0" in one-shot timer mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f1 0 1 : f8 1 0 : f32 1 1 : fC32	○	○
TCK1			○	○

Note 1: The settings of the corresponding port register and port direction register are invalid
Note 2: Valid only when the TAiIN pin is selected by the event/trigger select bit (addresses 0382_{16} and 0383_{16}). If timer overflow is selected, this bit can be "1" or "0".
Note 3: Set the corresponding port direction register to "0".

Figure FB-9. Timer Ai mode register in one-shot timer mode

(4) Pulse width modulation (PWM) mode

In this mode, the timer outputs pulses of a given width in succession. (See Table FB-5.) In this mode, the counter functions as either a 16-bit pulse width modulator or an 8-bit pulse width modulator. Figure FB-10 shows the timer Ai mode register in pulse width modulation mode. Figure FB-11 shows the example of how a 16-bit pulse width modulator operates. Figure FB-12 shows the example of how an 8-bit pulse width modulator operates.

Table FB-5. Timer specifications in pulse width modulation mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	•The timer counts down (operating as an 8-bit or a 16-bit pulse width modulator) •The timer reloads a new count at a rising edge of PWM pulse and continues counting •The timer is not affected by a trigger that occurs when counting
16-bit PWM	•High level width n / fi n : Set value •Cycle time $(2^{16}-1)$ / fi fixed
8-bit PWM	•High level width n X (m+1) / fi n : values set to timer Ai register's high-order address •Cycle time (2^8-1) X (m+1) / fi m : values set to timer Ai register's low-order address
Count start condition	•External trigger is input •The timer overflows •The count start flag is set (= 1)
Count stop condition	•The count start flag is reset (= 0)
Interrupt request generation timing	PWM pulse goes "L"
TAiIN pin function	Programmable I/O port or trigger input
TAiOUT pin function	Pulse output
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	•When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter •When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)

Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
					1	1	1

Symbol: TAiMR(i=0 to 4) Address: 039_{16} to $039A_{16}$ When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 1 : PWM mode	○	○
TMOD1			○	○
MR0	1 (Must always be fixed to "1" in PWM mode)		○	○
MR1	External trigger select bit (Note 1)	0: Falling edge of TAiIN pin's input signal (Note 2) 1: Rising edge of TAiIN pin's input signal (Note 2)	○	○
MR2	Trigger select bit	0: Count start flag is valid 1: Selected by event/trigger select register	○	○
MR3	16/8-bit PWM mode select bit	0: Functions as a 16-bit pulse width modulator 1: Functions as an 8-bit pulse width modulator	○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: Valid only when the TAiIN pin is selected by the event/trigger select bit (addresses 038_{216} and 038_{316}). If timer overflow is selected, this bit can be "1" or "0
Note 2: Set the corresponding port direction register to "0".

Figure FB-10. Timer Ai mode register in pulse width modulation mode

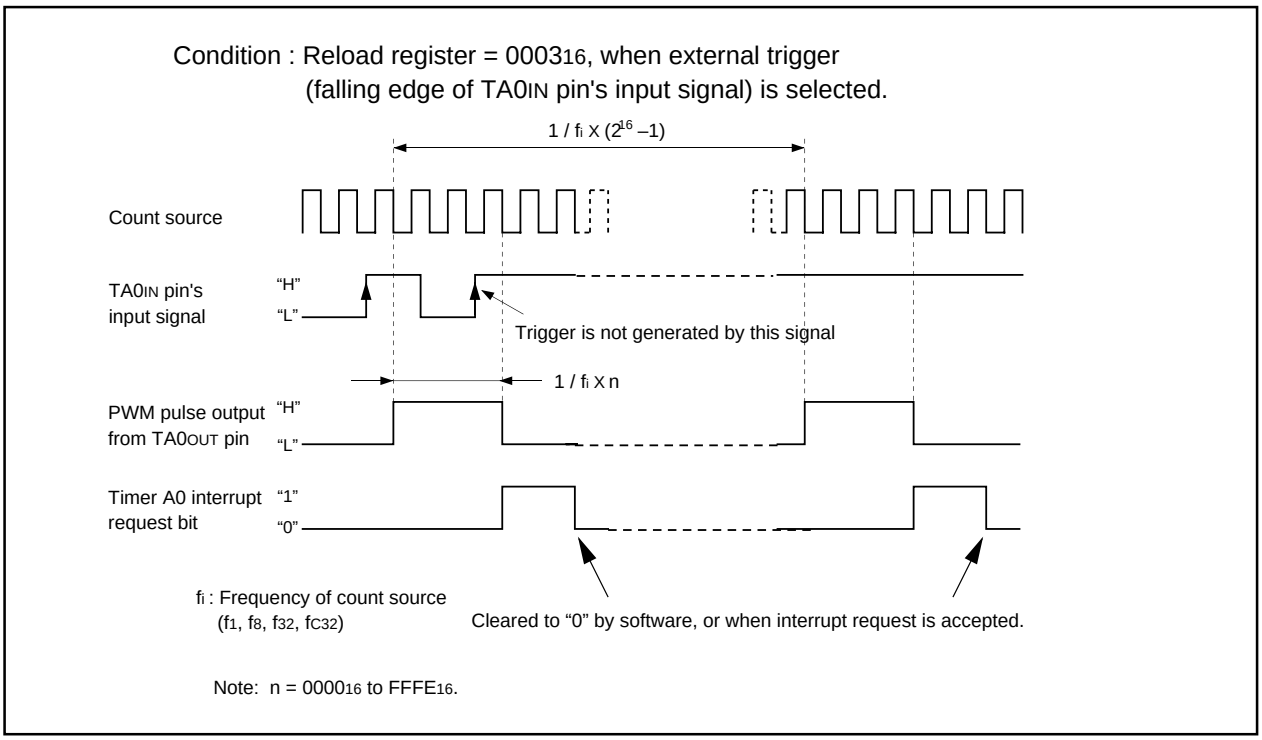


Figure FB-11. Example of how a 16-bit pulse width modulator operates

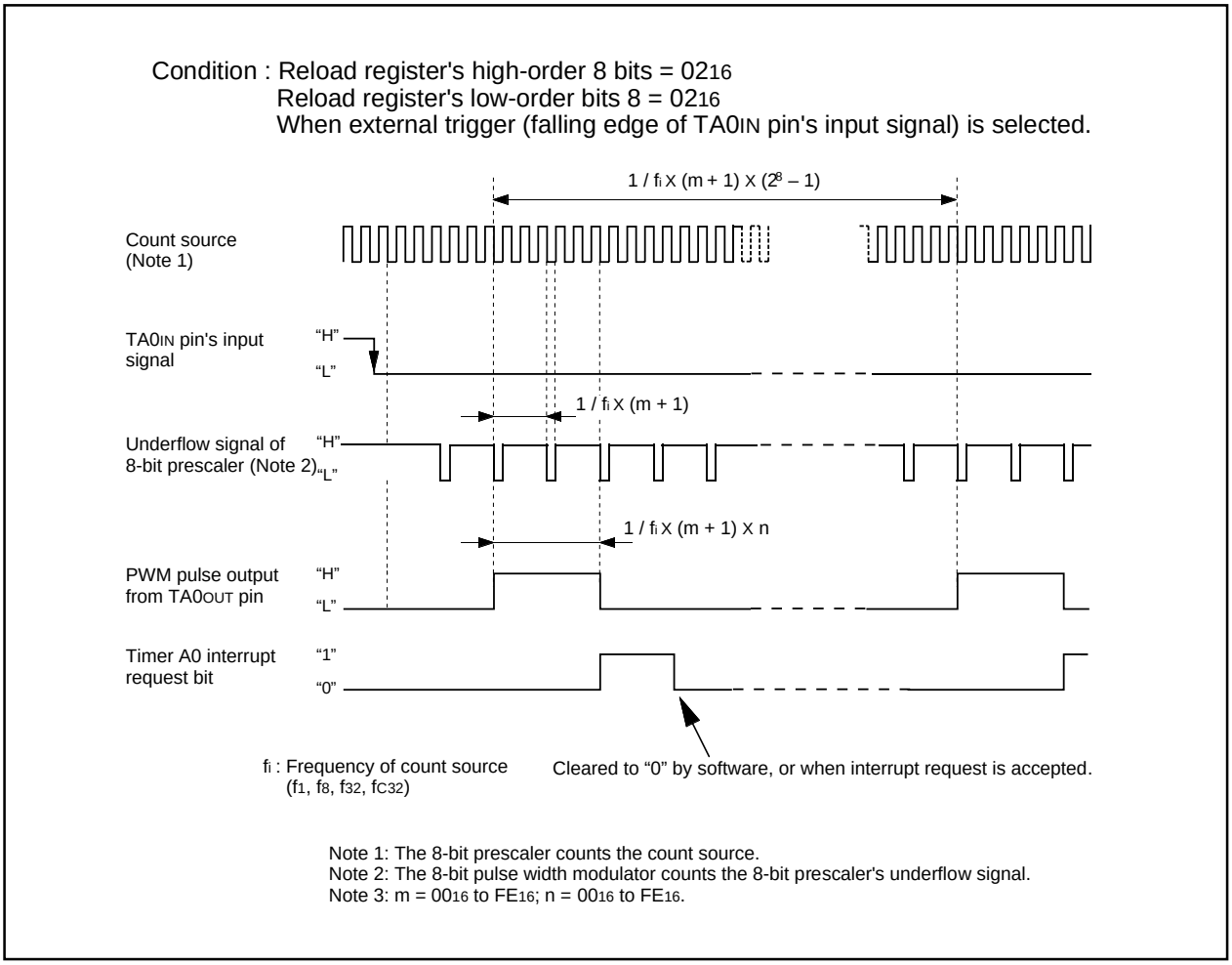


Figure FB-12. Example of how an 8-bit pulse width modulator operates

Timer B

Figure TA-1 shows the block diagram of timer B. Figures TA-2 and TA-3 show the timer B-related registers.

Use the timer Bi mode register (i = 0 to 2) bits 0 and 1 to choose the desired mode.

Timer B has three operation modes listed as follows:

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external source or a timer overflow.
- Pulse period/pulse width measuring mode: The timer measures an external signal's pulse period or pulse width.

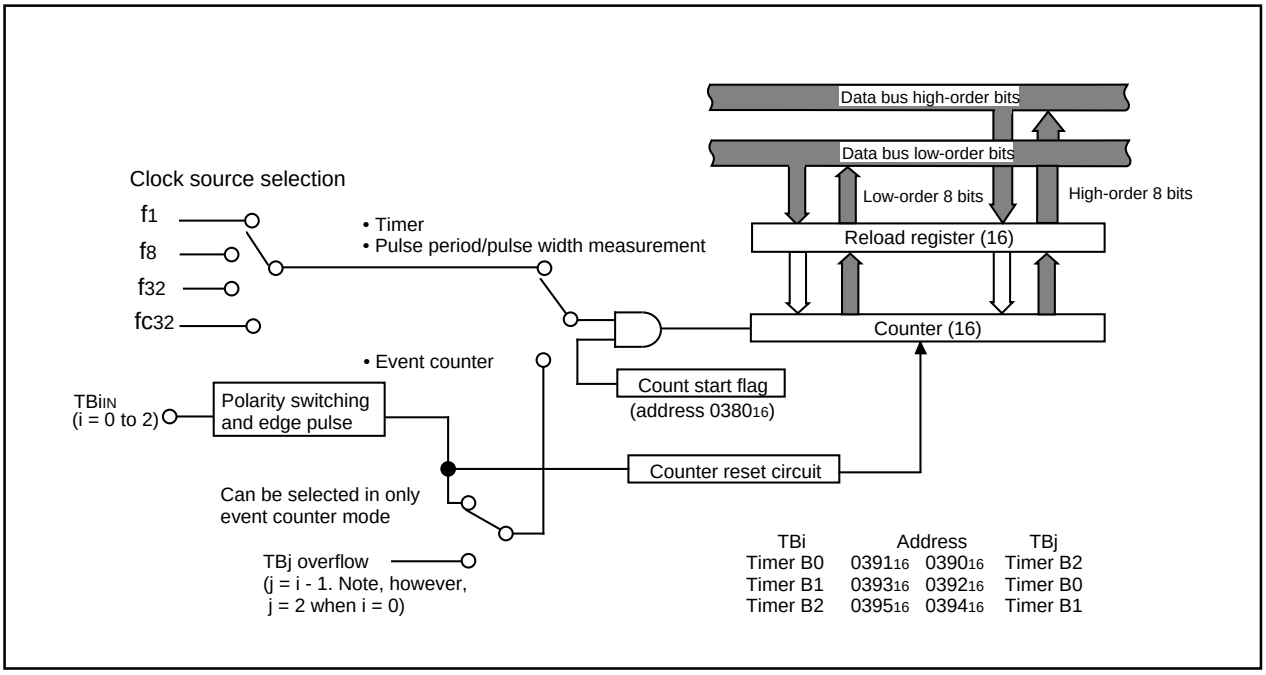


Figure TA-1. Block diagram of timer B

Timer Bi mode register

b7	b6	b5	b4	b3	b2	b1	b0

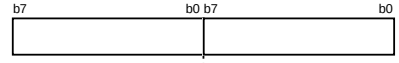
Symbol: TBiMR(i = 0 to 2) | Address: 039B16 to 039D16 | When reset: 00XX00002

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period/pulse width measurement mode 1 1 : Inhibited	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
MR2			○ (Note 1) × (Note 2)	○ ×
MR3			○	×
TCK0	Count source select bit (Function varies with each operation mode)		○	○
TCK1			○	○

Note 1: Timer B0.
Note 2: Timer B1, timer B2.

Figure TA-2. Timer B-related registers (1)

Timer Bi register (Note)



Symbol	Address	When reset
TB0	039116, 039016	Indeterminate
TB1	039316, 039216	Indeterminate
TB2	039516, 039416	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts the timer's period	0000_{16} to $FFFF_{16}$	○	○
• Event counter mode Counts external pulses input or a timer overflow	0000_{16} to $FFFF_{16}$	○	○
• Pulse period / pulse width measurement mode Measures a pulse period or width	——	○	×

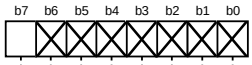
Note: Read and write data in 16-bit units.

Count start flag

Symbol	Address	When reset
TABSR	0380_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TA1S	Timer A1 count start flag		○	○
TA2S	Timer A2 count start flag		○	○
TA3S	Timer A3 count start flag		○	○
TA4S	Timer A4 count start flag		○	○
TB0S	Timer B0 count start flag		○	○
TB1S	Timer B1 count start flag		○	○
TB2S	Timer B2 count start flag		○	○

Clock prescaler reset flag



Symbol	Address	When reset
CPSRF	0381_{16}	$0XXXXXXX_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	○	○

Figure TA-3. Timer B-related registers (2)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table TA-1.) Figure TA-4 shows the timer Bi mode register in timer mode.

Table TA-1. Timer specifications in timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	•Counts down •When the timer underflows, the reload register's content is reloaded and the timer starts over again.
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBi_{IN} pin function	Programmable I/O port
Read from timer	Count value is read out by reading timer Bi register
Write to timer	•When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)

Timer Bi mode register

b7	b6	b5	b4	b3	b2	b1	b0
						0	0

Symbol: TBiMR(i=0 to 2) Address: $039B_{16}$ to $039D_{16}$ When reset: $00XX0000_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	○	○
TMOD1			○	○
MR0	Invalid in timer mode Can be "0" or "1"		○	○
MR1			○	○
MR2	0 (Fixed to "0" in timer mode ; i = 0)		○ (Note 1)	○
	Nothing is assigned (i = 1,2). In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.		× (Note 2)	×
MR3	Invalid in timer mode. In an attempt to write to these bits, write "0". The value, if read in timer mode, turns out to be indeterminate.		○	×
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: Timer B0.
Note 2: Timer B1, timer B2.

Figure TA-4. Timer Bi mode register in timer mode

(2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. (See Table TA-2.) Figure TA-5 shows the timer Bi mode register in event counter mode.

Table TA-2. Timer specifications in event counter mode

Item	Specification
Count source	•External signals input to TBiIN pin •Effective edge of count source can be a rising edge, a falling edge, or falling and rising edges as selected by software
Count operation	•Counts down •When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBiIN pin function	Count source input
Read from timer	Count value can be read out by reading timer Bi register
Write to timer	•When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter •When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)

Timer Bi mode register

b7	b6	b5	b4	b3	b2	b1	b0
						0	1

Symbol: TBiMR(i=0 to 2) Address: 039B16 to 039D16 When reset: 00XX00002

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	Count polarity select bit (Note 1)	b3 b2 0 0 : Counts external signal's falling edges 0 1 : Counts external signal's rising edges 1 0 : Counts external signal's falling and rising edges 1 1 : Inhibited	○	○
MR1			○	○
MR2	0 (Fixed to "0" in event counter mode; i = 0)		○ (Note 2)	○
	Nothing is assigned (i = 1, 2). In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.		× (Note 3)	×
MR3	Invalid in event counter mode. In an attempt to write to this bit, write "0". The value, if read in event counter mode, turns out to be indeterminate.		○	×
TCK0	Invalid in event counter mode. Can be "0" or "1".		○	○
TCK1	Event clock select	0: Input from TBiIN pin (Note 4) 1: TBj overflow (j = i-1; however, j = 2 when i = 0)	○	○

Note 1: Valid only when input from the TBiIN pin is selected as the event clock.
If timer's overflow is selected, this bit can be "0" or "1".
Note 2: Timer B0.
Note 3: Timer B1, timer B2.
Note 4: Set the corresponding port direction register to "0".

Figure TA-5. Timer Bi mode register in event counter mode

(3) Pulse period/pulse width measurement mode

In this mode, the timer measures the pulse period or pulse width of an external signal. (See Table TA-3.) Figure TA-6 shows the timer Bi mode register in pulse period/pulse width measurement mode. Figure TA-7 shows the operation timing when measuring a pulse period. Figure TA-8 shows the operation timing when measuring a pulse width.

Table TA-3. Timer specifications in pulse period/pulse width measurement mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	•Up count •Counter value "0000_{16}" is transferred to reload register at measurement pulse's effective edge and the timer continues counting
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	•When measurement pulse's effective edge is input (Note 1) •When an overflow occurs. (Simultaneously, the timer Bi overflow flag changes to "1". The timer Bi overflow flag changes to "0" when the count start flag is "1" and a value is written to the timer Bi mode register.)
TBi_{IN} pin function	Measurement pulse input
Read from timer	When timer Bi register is read, it indicates the reload register's content (measurement result) (Note 2)
Write to timer	Cannot be written to

Note 1: An interrupt request is not generated when the first effective edge is input after the timer has started counting.

Note 2: The value read out from the timer Bi register is indeterminate until the second effective edge is input after the timer.

Timer Bi mode register

b7	b6	b5	b4	b3	b2	b1	b0
						1	0

Symbol: TBiMR (i=0 to 2) — Address: $039B_{16}$ to $039D_{16}$ — When reset: $00XX0000_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : Pulse period / pulse width measurement mode	○	○
TMOD1			○	○
MR0	Measurement mode select bit	b3 b2 0 0 : Pulse period measurement (Interval between measurement pulse's falling edge to falling edge) 0 1 : Pulse period measurement (Interval between measurement pulse's rising edge to rising edge) 1 0 : Pulse width measurement (Interval between measurement pulse's falling edge to rising edge, and between rising edge to falling edge) 1 1 : Inhibited	○	○
MR1			○	○
MR2	0 (Fixed to "0" in pulse period/pulse width measurement mode; i = 0)		○ (Note 2)	○
	Nothing is assigned (i = 1, 2). In an attempt to write to this bit, write "0". The value, if read in event counter mode, turns out to be indeterminate.		× (Note 3)	×
MR3	Timer Bi overflow flag (Note 1)	0 : Timer did not overflow 1 : Timer has overflowed	○	×
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: The timer Bi overflow flag changes to "0" when the count start flag is "1" and a value is written to the timer Bi mode register. This flag cannot be set to "1" by software.
Note 2: Timer B0.
Note 3: Timer B1, timer B2.

Figure TA-6. Timer Bi mode register in pulse period/pulse width measurement mode

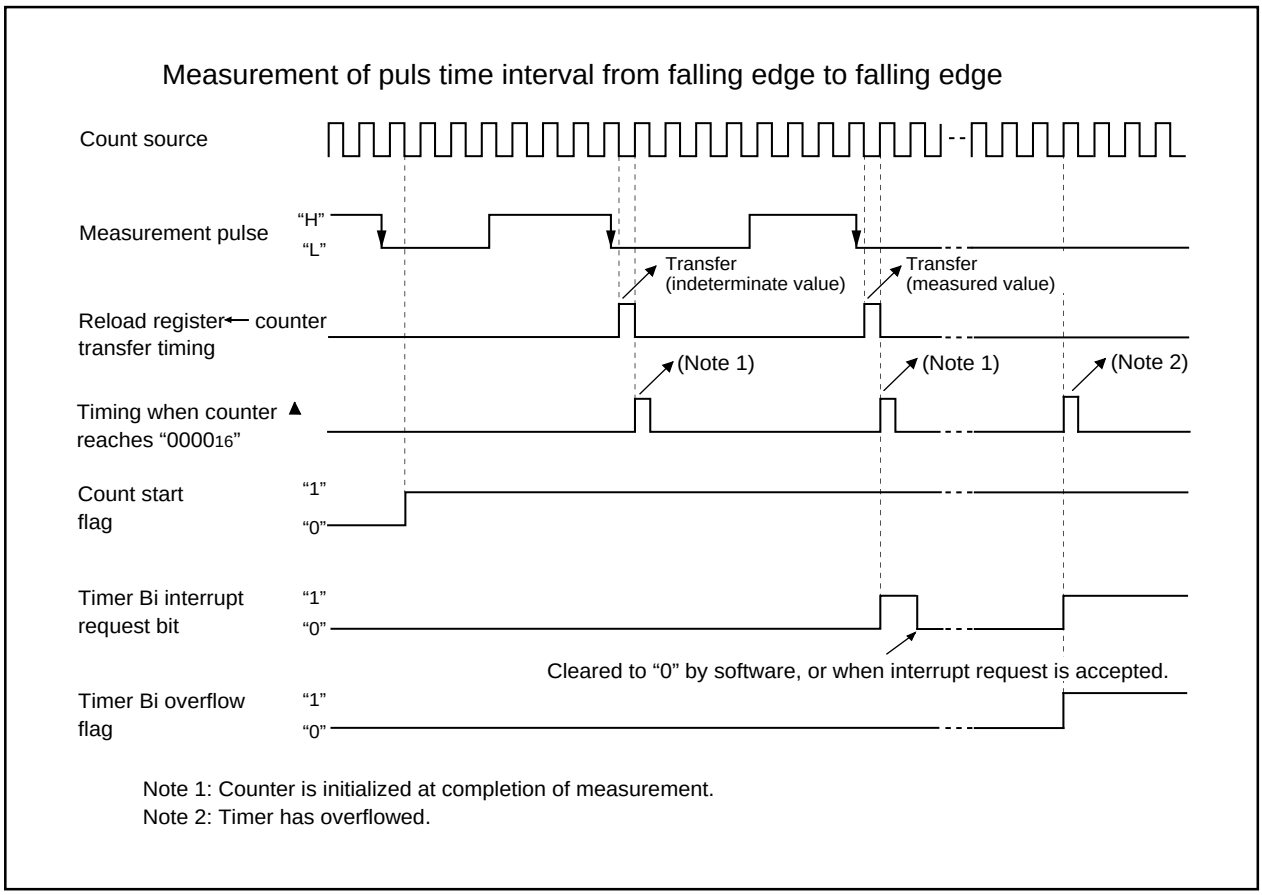


Figure TA-7. Operation timing when measuring a pulse period

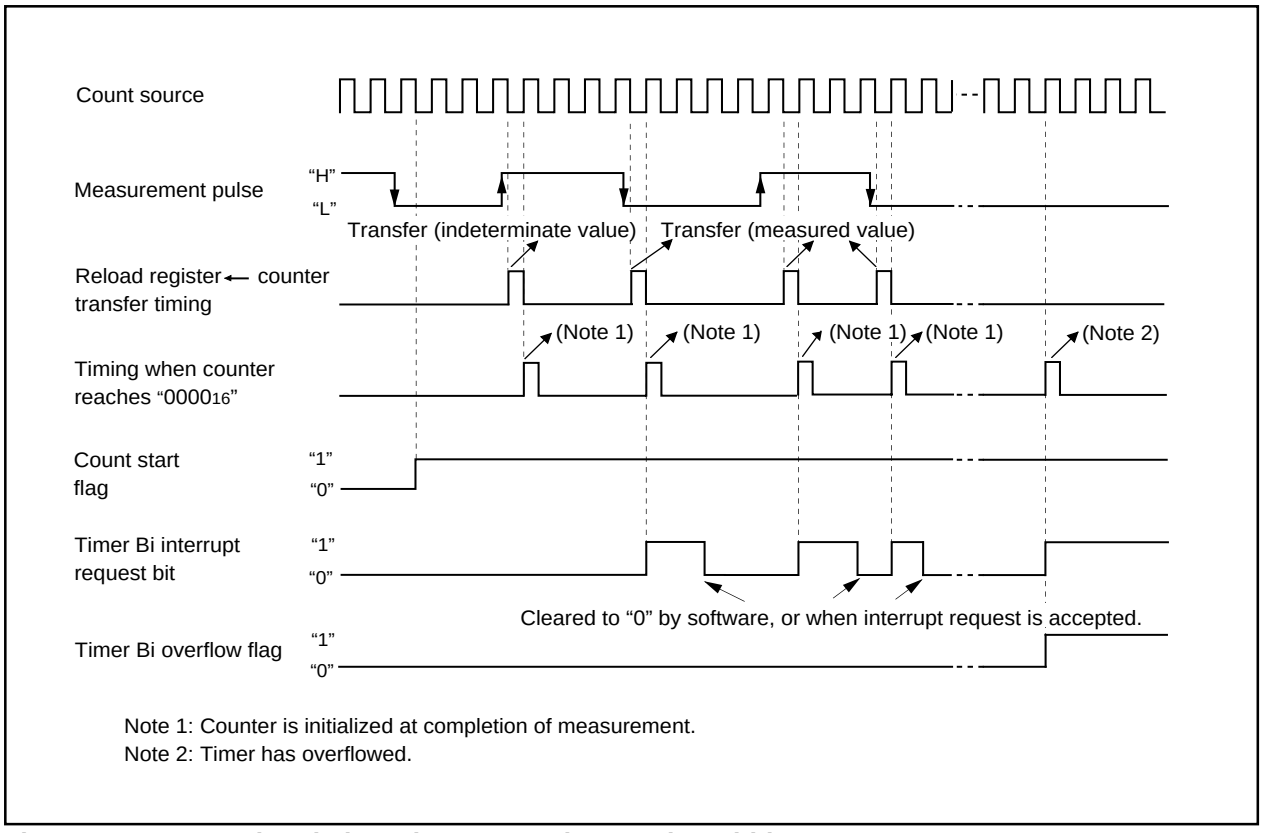


Figure TA-8. Operation timing when measuring a pulse width

Serial I/O

Serial I/O is configured as two channels: UART0 and UART1.

UART0 and UART1 each have an exclusive timer to generate a transfer clock, so they operate independently of each other.

Figure GA-1 shows the block diagram of UART0 and UART1. Figures GA-2 shows the block diagram of the transmit/receive unit.

UARTi (i=0, 1) has two operation modes: a clock synchronous serial I/O mode and a clock asynchronous serial I/O mode (UART mode). The contents of the serial I/O mode select bits (bits 0 to 2 at addresses $03A0_{16}$ and $03A8_{16}$) determine whether UARTi is used as a clock synchronous serial I/O or as a UART.

Although a few function are different, UART0 and UART1 have almost same functions.

Figures GA-3 through GA-5 show the registers related to UARTi.

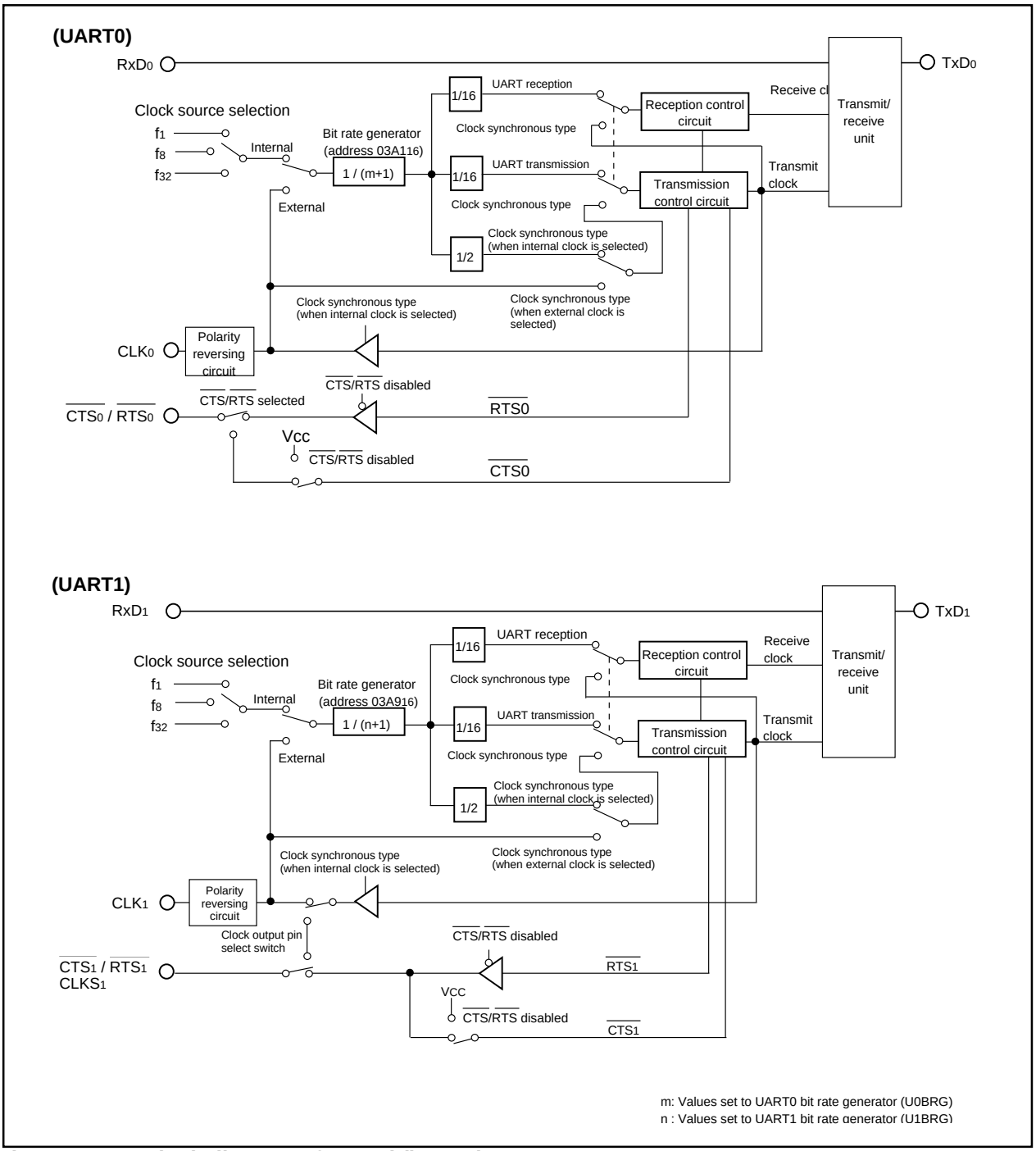


Figure GA-1. Block diagram of UARTi (i = 0, 1)

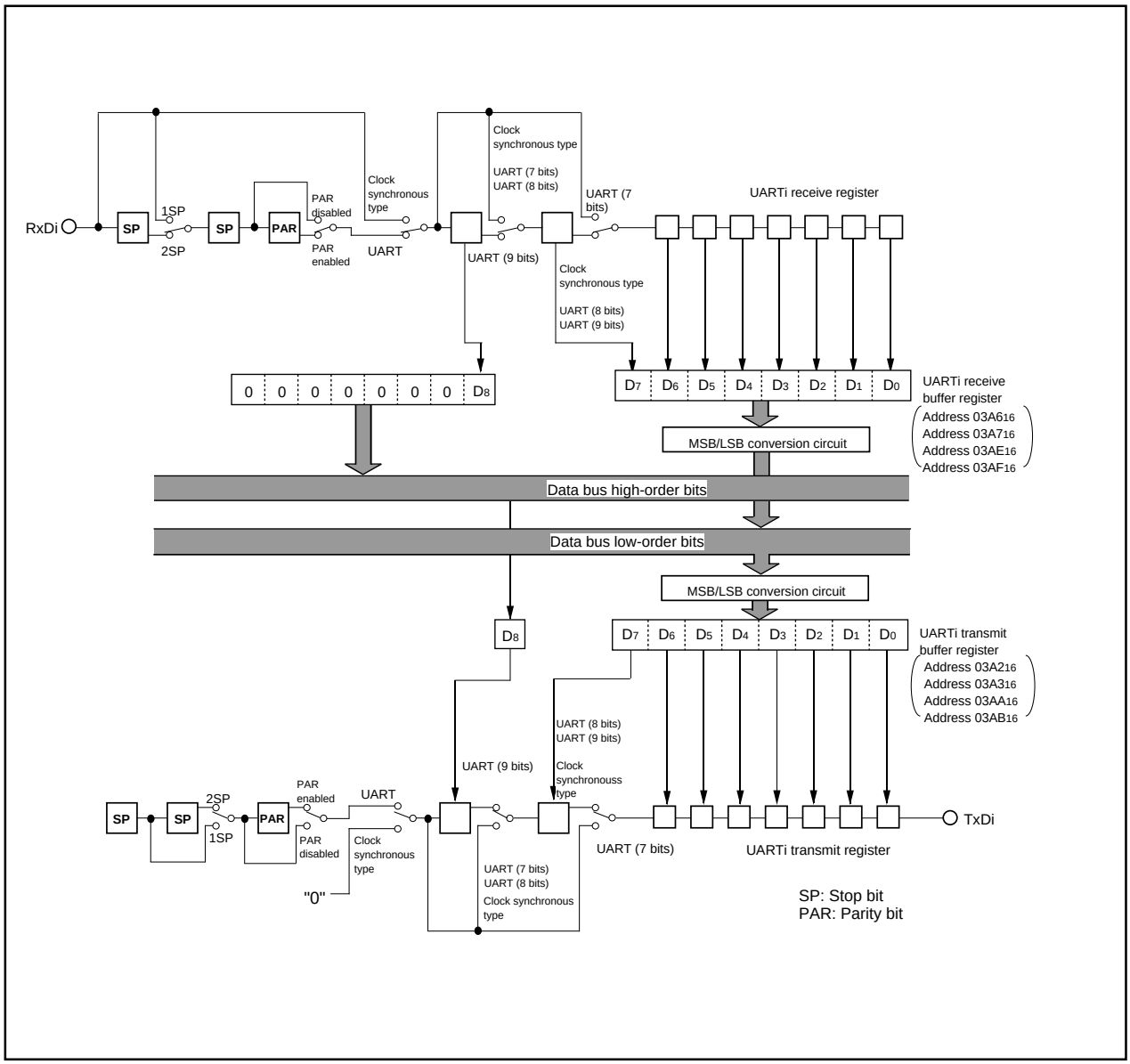
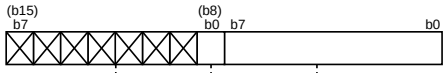


Figure GA-2. Block diagram of transmit/receive unit

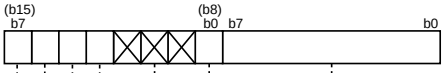
UARTi transmit buffer register



Symbol	Address	When reset
U0TB	$03A3_{16}$, $03A2_{16}$	Indeterminate
U1TB	$03AB_{16}$, $03AA_{16}$	Indeterminate

Function	R	W
Transmission data	×	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.	—	—

UARTi receive buffer register

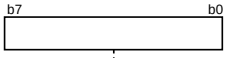


Symbol	Address	When reset
U0RB	$03A7_{16}$, $03A6_{16}$	Indeterminate
U1RB	$03AF_{16}$, $03AE_{16}$	Indeterminate

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
—	———	Reception data	Reception data	○	×
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.				—	—
OER	Overrun error flag (Note)	0 : No overrun error 1 : Overrun error found	0 : No overrun error 1 : Overrun error found	○	×
FER	Framing error flag (Note)	Invalid	0 : No framing error 1 : Framing error found	○	×
PER	Parity error flag (Note)	Invalid	0 : No parity error 1 : Parity error found	○	×
SUM	Error sum flag (Note)	Invalid	0 : No error 1 : Error found	○	×

Note: Bits 15 through 12 are set to “0” when the serial I/O mode select bit (bits 2 to 0 at addresses $03A0_{16}$ and $03A8_{16}$) are set to “000_2” or the receive enable bit is set to “0”.
(Bit 15 is set to “0” when bits 14 to 12 all are set to “0”.) Bits 14 and 13 are also set to “0” when the lower byte of the UARTi receive buffer register (addresses $03A6_{16}$ and $03AE_{16}$) is read out.

UARTi bit rate generator



Symbol	Address	When reset
U0BRG	$03A1_{16}$	Indeterminate
U1BRG	$03A9_{16}$	Indeterminate

Function	Values that can be set	R	W
Assuming that set value = n, BRGi divides the count source by (n + 1)	00_{16} to FF_{16}	×	○

Figure GA-3. Serial I/O-related registers (1)

UARTi transmit/receive mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: UiMR (i=0,1) | Address: 03A0$_{16}$, 03A8$_{16}$ | When reset: 00$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select bit	Must be fixed to 001	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	○	○
SMD1		b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited		○	○
SMD2				○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock	0 : Internal clock 1 : External clock	○	○
STPS	Stop bit length select bit	Invalid	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd/even parity select bit	Invalid	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	Invalid	0 : Parity disabled 1 : Parity enabled	○	○
SLEP	Sleep select bit	Must always be “0”	0 : Sleep mode deselected 1 : Sleep mode selected	○	○

UARTi transmit/receive control register 0

b7 b6 b5 b4 b3 b2 b1 b0

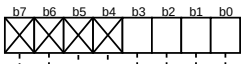
Symbol: UiC0 (i=0,1) | Address: 03A4$_{16}$, 03AC$_{16}$ | When reset: 08$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bit	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Inhibited	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Inhibited	○	○
CLK1				○	○
CRS	$\overline{CTS}$/$\overline{RTS}$ function select bit	Valid when bit 4 = “0” 0 : $\overline{CTS}$ function is selected (Note 1) 1 : $\overline{RTS}$ function is selected (Note 2)	Valid when bit 4 = “0” 0 : $\overline{CTS}$ function is selected (Note 1) 1 : $\overline{RTS}$ function is selected (Note 2)	○	○
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	○	×
CRD	$\overline{CTS}$/$\overline{RTS}$ disable bit	0 : $\overline{CTS}$/$\overline{RTS}$ function enabled 1 : $\overline{CTS}$/$\overline{RTS}$ function disabled ($P4_7$ and $P7_7$ function as programmable I/O port)	0 : $\overline{CTS}$/$\overline{RTS}$ function enabled 1 : $\overline{CTS}$/$\overline{RTS}$ function disabled ($P4_7$ and $P7_7$ function as programmable I/O port)	○	○
NCH	Data output select bit	0 : TXDi pin is CMOS output 1 : TXDi pin is N-channel open-drain output	0: TXDi pin is CMOS output 1: TXDi pin is N-channel open-drain output	○	○
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	Must always be “0”	○	○
UFORM	Transfer format select bit	0 : LSB first 1 : MSB first	Must always be “0”	○	○

Note 1: Set the corresponding port direction register to “0”.
Note 2: The settings of the corresponding port register and port direction register are invalid.

Figure GA-4. Serial I/O-related registers (2)

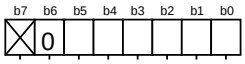
UARTi transmit/receive control register 1



Symbol: UiC1(i=0,1) — Address: 03A5$_{16}$, 03AD$_{16}$ — When reset: 02$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0 : Transmission disabled 1 : Transmission enabled	○	○
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	○	×
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	○	○
RI	Receive complete flag	0 : No data present in receive buffer register 1 : Data present in receive buffer register	0 : No data present in receive buffer register 1 : Data present in receive buffer register	○	×
Nothing is assigned. In an attempt to write to this bit, write “0”. The value, if read, turns out to be “0”.				–	–

UART transmit/receive control register 2



Symbol: UCON — Address: 03B0$_{16}$ — When reset: X0000000$_2$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
U0IRS	UART0 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U1IRS	UART1 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U0RRM	UART0 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enable	Invalid	○	○
U1RRM	UART1 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	Invalid	○	○
CLKMD0	CLK/CLKS select bit 0	Valid when bit 5 = “1” 0 : Clock output to CLK1 1 : Clock output to CLKS1	Invalid	○	○
CLKMD1	CLK/CLKS select bit 1 (Note)	0 : Normal mode (CLK output is CLK1 only) 1 : Transfer clock output from multiple pins function selected	Must always be “0”	○	○
Reserved bit		Must always be “0”	Must always be “0”	○	○
Nothing is assigned. In an attempt to write to this bit, write “0”. The value, if read, turns out to be indeterminate.				–	–

Note: When using multiple pins to output the transfer clock, the following requirement must be met:
• UART1 internal/external clock select bit (bit 3 at address 03A8$_{16}$) = “0”.

Figure GA-5. Serial I/O-related registers (3)

(1) Clock synchronous serial I/O mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Table GA-1 lists the specifications of the clock synchronous serial I/O mode. Figure GA-6 shows the UARTi transmit/ receive mode register.

Table GA-1. Specifications of clock synchronous serial I/O mode

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• When internal clock is selected (bit 3 at address $03A0_{16}$, $03A8_{16}$ = "0") : fi/ 2(n+1) (Note 1) fi = f_1, f_8, f_{32} • When external clock is selected (bit 3 at address $03A0_{16}$, $03A8_{16}$ ="1") : Input from CLKi pin (Note 2)
Transmission/reception control	• $\overline{CTS}$ function/ $\overline{RTS}$ function/ $\overline{CTS}$,$\overline{RTS}$ function chosen to be invalid
Transmission start condition	• To start transmission, the following requirements must be met: – Transmit enable bit (bit 0 at address $03A5_{16}$, $03AD_{16}$) = "1" – Transmit buffer empty flag (bit 1 at addresses $03A5_{16}$, $03AD_{16}$) = "0" – When $\overline{CTS}$ function is selected, $\overline{CTS}$ input level = "L" • Furthermore, if external clock is selected, the following requirements must also be met: – CLKi polarity select bit (bit 6 at address $03A4_{16}$, $03AC_{16}$) = "0": CLKi input level = "H" – CLKi polarity select bit (bit 6 at address $03A4_{16}$, $03AC_{16}$) = "1": CLKi input level = "L"
Reception start condition	• To start reception, the following requirements must be met: – Receive enable bit (bit 2 at address $03A5_{16}$, $03AD_{16}$) = "1" – Transmit enable bit (bit 0 at address $03A5_{16}$, $03AD_{16}$) = "1" – Transmit buffer empty flag (bit 1 at address $03A5_{16}$, $03AD_{16}$) = "0" • Furthermore, if external clock is selected, the following requirements must also be met: – CLKi polarity select bit (bit 6 at address $03A4_{16}$, $03AC_{16}$) = "0": CLKi input level = "H" – CLKi polarity select bit (bit 6 at address $03A4_{16}$, $03AC_{16}$) = "1": CLKi input level = "L"
Interrupt request generation timing	• When transmitting – Transmit interrupt cause select bit (bits 0,1 at address $03B0_{16}$) = "0": Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed – Transmit interrupt cause select bit (bits 0,1 at address $03B0_{16}$) = "1": Interrupts requested when data transmission from UARTi transfer register is completed • When receiving – Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed
Error detection	• Overrun error (Note 3) This error occurs when the next data is ready before contents of UARTi receive buffer register are read out
Select function	• CLK polarity selection Whether transmit data is output/input at the rising edge or falling edge of the transfer clock can be selected • LSB first/MSB first selection Whether transmission/reception begins with bit 0 or bit 7 can be selected • Continuous receive mode selection Reception is enabled simultaneously by a read from the receive buffer register • Transfer clock output from multiple pins selection UART1 transfer clock can be set 2 pins, and can be selected to output from which pin.

Note 1: "n" denotes the value 00_{16} to FF_{16} that is set to the UART bit rate generator.

Note 2: Maximum 5 Mbps.

Note 3: If an overrun error occurs, the UARTi receive buffer will have the next data written in. Note also that the UARTi receive interrupt request bit is not set to "1".

UARTi transmit/receive mode register

b7	b6	b5	b4	b3	b2	b1	b0
0					0	0	1

Symbol: UiMR(i=0,1) — Address: $03A0_{16}$, $03A8_{16}$ — When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 0 0 1 : Clock synchronous serial I/O mode	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock	○	○
STPS	Invalid in clock synchronous serial I/O mode		○	○
PRY			○	○
PRYE			○	○
SLEP	0 (Must always be "0" in clock synchronous serial I/O mode)		○	○

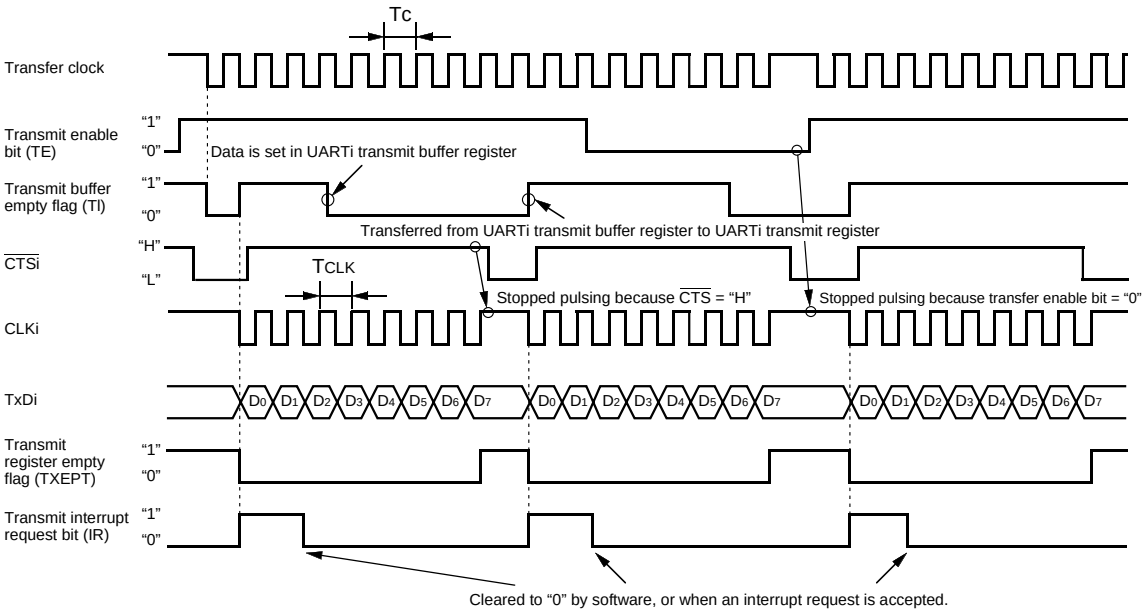
Figure GA-6. UARTi transmit/receive mode register in clock synchronous serial I/O mode (i=0,1)

Table GA-2 lists the functions of the input/output pins during clock synchronous serial I/O mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs a “H”. (If the N-channel open-drain is selected, this pin is in floating state.)

Table GA-2. Input/output pin functions in clock synchronous serial I/O mode (i=0,1)

Pin name	Function	Method of selection
TxDi ($P4_4$, $P7_4$)	Serial data output	(Outputs dummy data when performing reception only)
RxDi ($P4_5$, $P7_5$)	Serial data input	Port $P4_5$, $P7_5$ direction register (bits 5 at address $03EA_{16}$ and $03EF_{16}$)= “0” (Can be used as an input port when performing transmission only)
CLKi ($P4_6$, $P7_6$)	Transfer clock output	Internal/external clock select bit (bit 3 at address $03A0_{16}$, $03A8_{16}$) = “0”
	Transfer clock input	Internal/external clock select bit (bit 3 at address $03A0_{16}$, $03A8_{16}$) = “1” Port $P4_6$, $P7_6$ direction register (bits 6 at address $03EA_{16}$ and $03EF_{16}$) = “0”
$\overline{CTS}$i/$\overline{RTS}$i ($P4_7$, $P7_7$)	$\overline{CTS}$ input	$\overline{CTS}$/$\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$) =“0” CTS/RTS function select bit (bit 2 at address $03A4_{16}$, $03AC_{16}$) = “0” Port $P4_7$, $P7_7$ direction register (bits 7 address $03EA_{16}$ and $03EF_{16}$) = “0”
	$\overline{RTS}$ output	$\overline{CTS}$/$\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$) = “0” CTS/RTS function select bit (bit 2 at address $03A4_{16}$, $03AC_{16}$) = “1”
	Programmable I/O port	$\overline{CTS}$/$\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$) = “1”

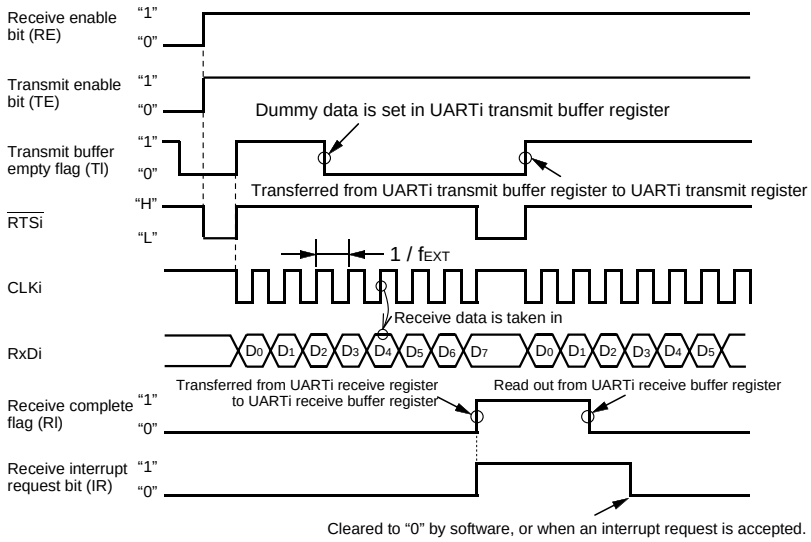
• Example of transmit timing (when internal clock is selected)



Shown in ( ) are bit symbols.
The above timing applies to the following settings:
- Internal clock is selected.
- CTS function is selected.
- CLK polarity select bit = "0".
- Transmit interrupt cause select bit = "0".

$Tc = T_{CLK} = 2(n + 1) / fi$
fi: frequency of BRGi's count source (f_1, f_8, f_{32})
n: value set to BRGi

• Example of receive timing (when external clock is selected)



f_{EXT}: frequency of external clock

Shown in ( ) are bit symbols.
The above timing applies to the following settings.
- External clock is selected.
- RTS function is selected.
- CLK polarity select bit = "0".

Meet the following conditions when the CLK input before data reception = "H"
- Transmit enable bit → "1"
- Receive enable bit → "1"
- Dummy data write to UARTi transmit buffer register

Figure GA-7. Typical transmit/receive timings in clock synchronous serial I/O mode

(a) Polarity select function

As shown in Figure GA-8, the CLK polarity select bit (bit 6 at addresses $03A4_{16}$, $03AC_{16}$) allows selection of the polarity of the transfer clock.

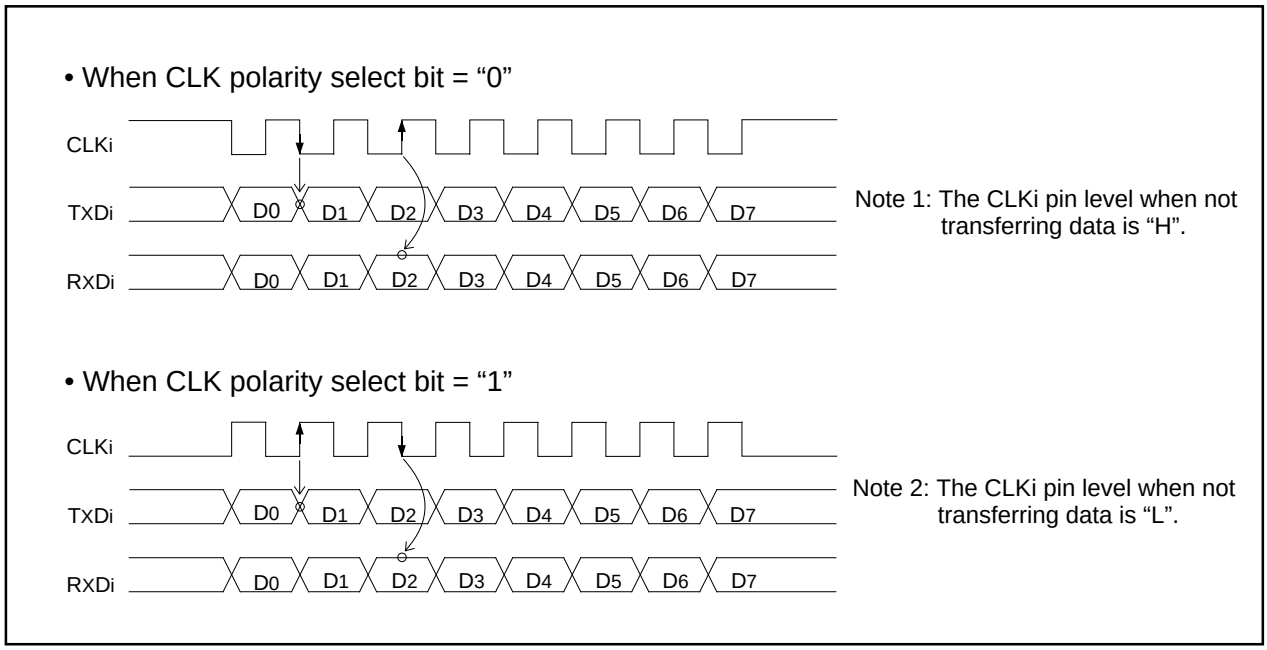


Figure GA-8. Polarity of transfer clock

(b) LSB first/MSB first select function

As shown in Figure GA-9, when the transfer format select bit (bit 7 at addresses $03A4_{16}$, $03AC_{16}$) = “0”, the transfer format is “LSB first”; when the bit = “1”, the transfer format is “MSB first”.

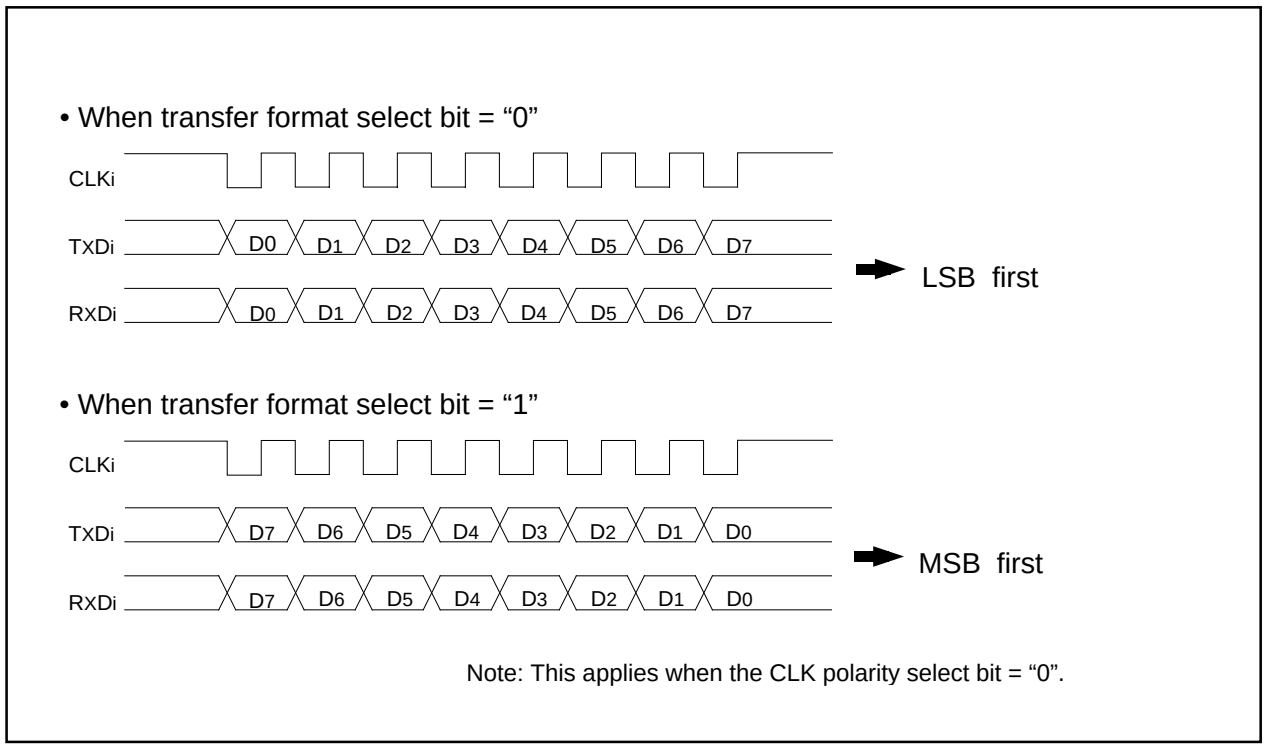


Figure GA-9. Transfer format

(c) Transfer clock output from multiple pins function

This function allows the setting two transfer clock output pins and choosing one of the two to output a clock by using the CLK and CLKS select bit (bits 4 and 5 at address 03B0$_{16}$). (See Figure GA-10.) The multiple pins function is valid only when the internal clock is selected for UART1. Note that when this function is selected, $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function of UART1 cannot be used.

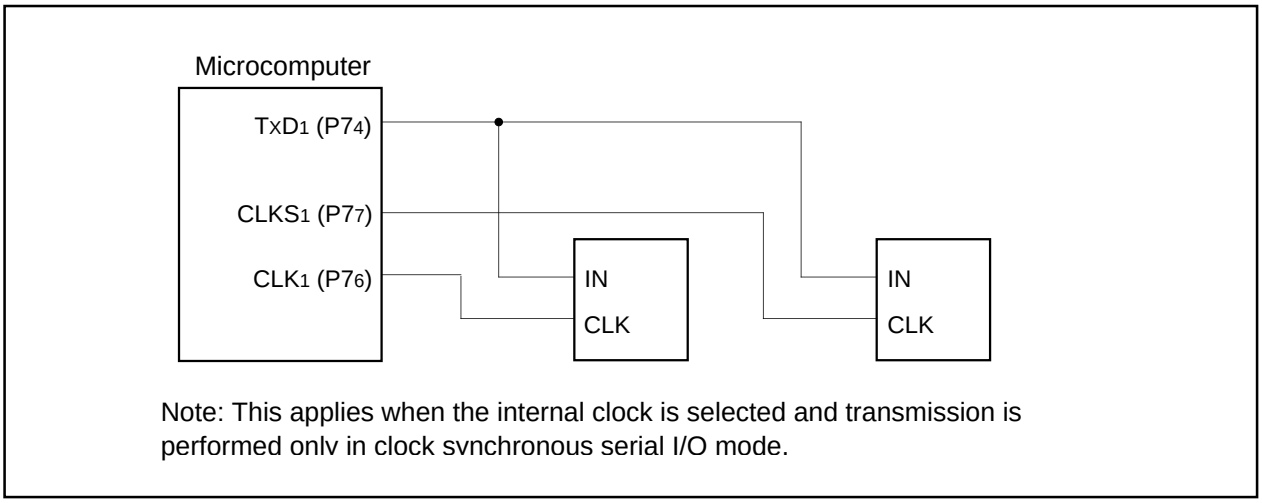


Figure GA-10. The transfer clock output from the multiple pins function usage

(d) Continuous receive mode

If the continuous receive mode enable bit (bits 2 and 3 at address 03B0$_{16}$) is set to “1”, the unit is placed in continuous receive mode. In this mode, when the receive buffer register is read out, the unit simultaneously goes to a receive enable state without having to set dummy data to the transmit buffer register back again.

(2) Clock asynchronous serial I/O (UART) mode

The UART allows transmitting and receiving data after setting the desired transfer rate and transfer data format. Tables GA-3 lists the specifications of the UART mode. Figure GA-11 shows the UARTi transmit/ receive mode register.

Table GA-3. Specifications of clock synchronous serial I/O mode

Item	Specification
Transfer data format	•Character bit (transfer data): 7 bits, 8 bits or 9 bits as selected •Start bit: 1 bit •Parity bit: Odd, even or nothing as selected •Stop bit: 1 bit or 2 bits as selected
Transfer clock	•When internal clock is selected (bit 3 at addresses $03A0_{16}$, $03A8_{16}$ = "0") : fi/16(n+1) (Note 1) fi = f_1, f_8, f_{32} •When external clock is selected (bit 3 at addresses $03A0_{16}$, $03A8_{16}$ ="1") : f_{EXT}/16(n+1) (Note 1) (Note 2)
Transmission/reception control	•$\overline{CTS}$ function/$\overline{RTS}$ function/$\overline{CTS}$, $\overline{RTS}$ function chosen to be invalid
Transmission start condition	•To start transmission, the following requirements must be met: - Transmit enable bit (bit 0 at addresses $03A5_{16}$, $03AD_{16}$) = "1" - Transmit buffer empty flag (bit 1 at addresses $03A5_{16}$, $03AD_{16}$) = "0" - When $\overline{CTS}$ function is selected, $\overline{CTS}$ input level = "L"
Reception start condition	•To start reception, the following requirements must be met: - Receive enable bit (bit 2 at addresses $03A5_{16}$, $03AD_{16}$) = "1" - Start bit detection
Interrupt request generation timing	•When transmitting - Transmit interrupt cause select bits (bits 0,1 at address $03B0_{16}$) = "0": Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed - Transmit interrupt cause select bits (bits 0, 1 at address $03B0_{16}$) = "1": Interrupts requested when data transmission from UARTi transfer register is completed •When receiving - Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed
Error detection	•Overrun error (Note 3) This error occurs when the next data is ready before contents of UARTi receive buffer register are read out •Framing error This error occurs when the number of stop bits set is not detected •Parity error This error occurs when if parity is enabled, the number of 1's in parity and character bits does not match the number of 1's set •Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered
select function	•Sleep mode selection This mode is used to transfer data to and from one of multiple slave microcomputers

Note 1: 'n' denotes the value 00_{16} to FF_{16} that is set to the UARTi bit rate generator.

Note 2: f_{EXT} is input from the CLKi pin.

Note 3: If an overrun error occurs, the UARTi receive buffer will have the next data written in. Note also that the UARTi receive interrupt request bit is not set to "1".

UARTi transmit/receive mode register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: UiMR (i=0,1) — Address: 03A0$_{16}$, 03A8$_{16}$ — When reset: 00$_{16}$

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock	○	○
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd/even parity select bit	Valid when bit 6 = "1" 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	○	○
SLEP	Sleep select bit	0 : Sleep mode deselected 1 : Sleep mode selected	○	○

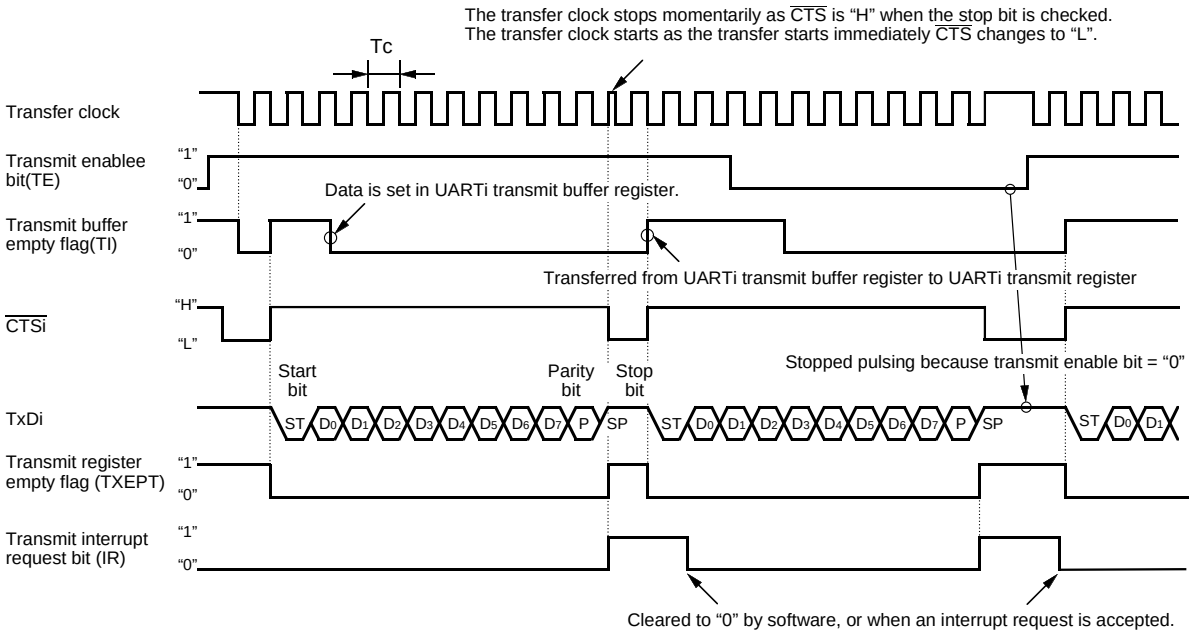
Figure GA-11. UARTi transmit/receive mode register in UART mode

Table GA-4 lists the functions of the input/output pins during UART mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs a "H". (If the N-channel open-drain is selected, this pin is in floating state.)

Table GA-4. Input/output pin functions in UART mode (i=0,1)

Pin name	Function	Method of selection
TxDi (P4$_4$, P7$_4$)	Serial data output	(Outputs dummy data when performing reception only)
RxDi (P4$_5$, P7$_5$)	Serial data input	Port P4$_5$, P7$_5$ direction register (bits 5 at address 03EA$_{16}$ and 03EF$_{16}$)= "0" (Can be used as an input port when performing transmission only)
CLKi (P4$_6$, P7$_6$)	Programmable I/O port	Internal/external clock select bit (bit 3 at address 03A0$_{16}$, 03A8$_{16}$) = "0"
	Transfer clock input	Internal/external clock select bit (bit 3 at address 03A0$_{16}$, 03A8$_{16}$) = "1"
$\overline{\text{CTSi}}$/$\overline{\text{RTSi}}$ (P4$_7$, P7$_7$)	$\overline{\text{CTS}}$ input	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4$_{16}$, 03AC$_{16}$) ="0" CTS/RTS function select bit (bit 2 at address 03A4$_{16}$, 03AC$_{16}$) = "0" Port P4$_7$, P7$_7$ direction register (bits 7 at address 03EA$_{16}$ and 03EF$_{16}$) = "0"
	$\overline{\text{RTS}}$ output	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4$_{16}$, 03AC$_{16}$) = "0" CTS/$\overline{\text{RTS}}$ function select bit (bit 2 at address 03A4$_{16}$, 03AC$_{16}$) = "1"
	Programmable I/O port	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4$_{16}$, 03AC$_{16}$) = "1"

• Example of transmit timing when transfer data is 8 bits long (parity enabled, one stop bit)

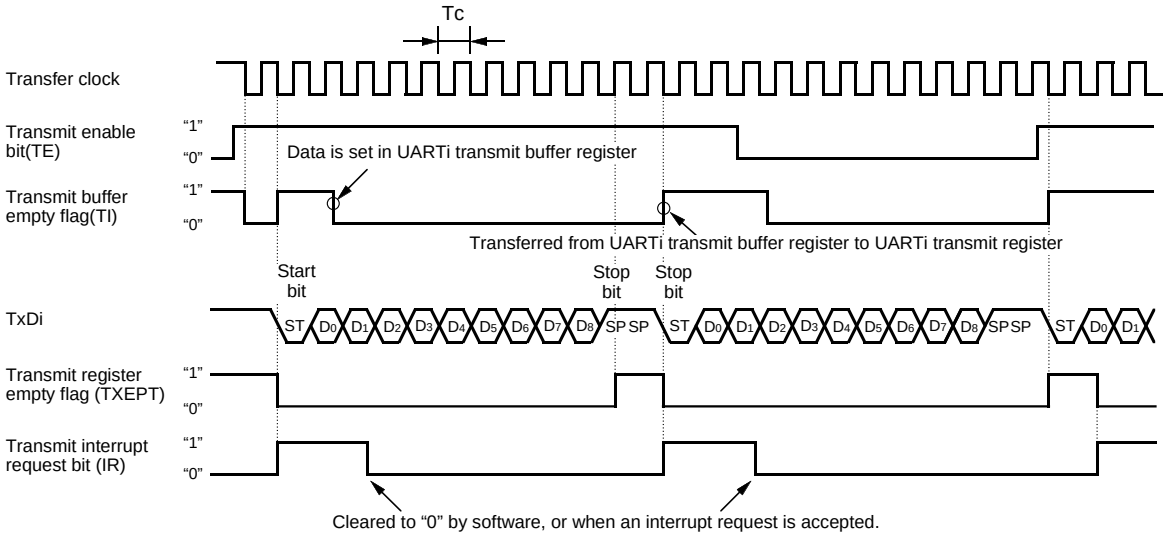


Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- CTS function is selected.
- Transmit interrupt cause select bit = "1".

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of BRGi's count source (f1, f8, f32)
fEXT : frequency of BRGi's count source (external clock)
n : value set to BRGi

• Example of transmit timing when transfer data is 9 bits long (parity disabled, two stop bits)



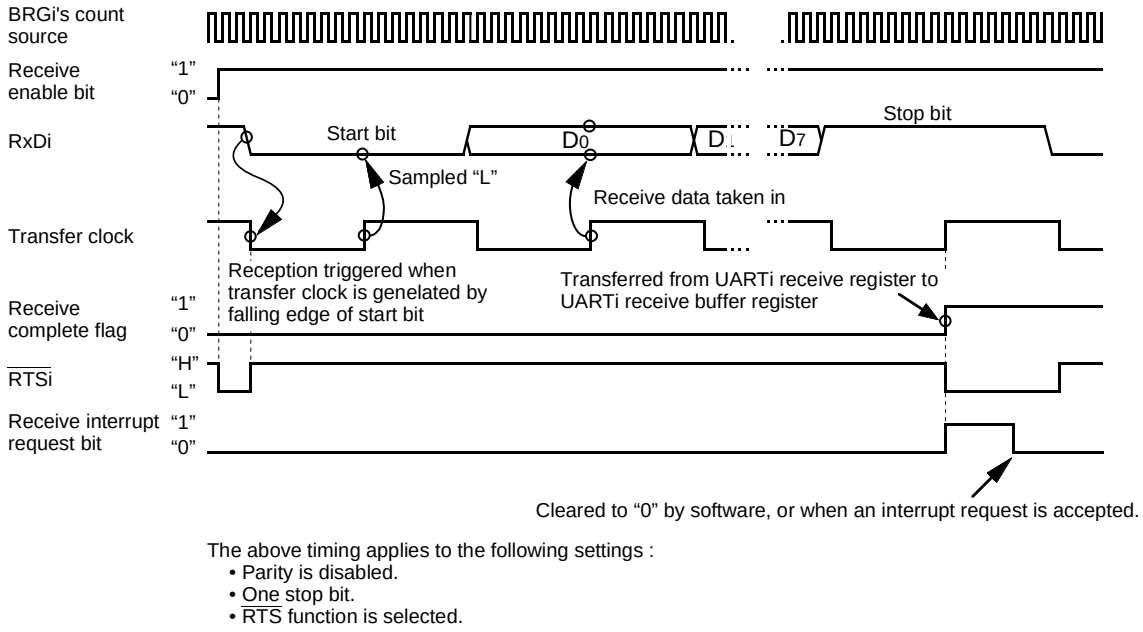
Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is disabled.
- Two stop bits.
- CTS function is disabled.
- Transmit interrupt causes select bit = "0".

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of BRGi's count source (f1, f8, f32)
fEXT : frequency of BRGi's count source (external clock)
n : value set to BRGi

Figure GA-12. Typical transmit timings in UART mode

• **Example of receive timing when transfer data is 8 bits long (parity disabled, one stop bit)**



The above timing applies to the following settings :
- Parity is disabled.
- One stop bit.
- $\overline{\text{RTS}}$ function is selected.

Figure GA-13. Typical receive timing in UART mode

(a) Sleep mode (UART0, UART1)

This mode is used to transfer data between specific microcomputers among multiple microcomputers connected using UARTi. The sleep mode is selected when the sleep select bit (bit 7 at addresses $03A0_{16}$, $03A8_{16}$) is set to "1" during reception. In this mode, the unit performs receive operation when the MSB of the received data = "1" and does not perform receive operation when the MSB = "0".

Serial I/O2

Serial I/O2 is used as the clock synchronous serial I/O and has an ordinary mode and an automatic transfer mode. In the automatic transfer mode, serial transfer is performed through the serial I/O automatic transfer RAM which has up to 256 bytes (addresses 0040_{16} to $004FF_{16}$).

The SRDY2, SBUSY2 and SSTB2 pins each have a handshake I/O signal function and can select either "H" active or "L" active for active logic.

Table GA-1. Specifications of clock synchronous serial I/O2

Item	Specification
Serial mode	• 8-bit serial I/O mode (non-automatic transfer) • Automatic transfer serial I/O mode
Transfer data format	• Transfer data length: 8 bits • Full duplex mode / transmit-only mode selected by bit 5 at address 0342_{16}
Transfer clock	• When internal clock is selected (bit 2 at address 0342_{16} = "0") : selected by bits 5 to 7 at address 0348_{16} • When external clock is selected (bit 2 at address 034216 = "1") : Input from SCLK21 pin, SCLK22 pin(Note 2)
Transfer rate	• When internal clock is selected : f(XIN)/4, f(XIN)/8, f(XIN)/16, f(XIN)/32, f(XIN)/64, f(XIN)/128, f(XIN)/256 • When external clock is selected : input cycle 0.95 µs or less
Transmission/reception control	• SSTB2 output / SBUSY2 input or output / SRDY2 input or output chosen
Transmission / reception start condition	• To start transmission / reception, the following requirements must be met: – Serial I/O initialization bit (bit 4 at address 0342_{16}) = "1" – When SBUSY2 input, or SRDY2 input is selected : selected input level = "H" – When $\overline{\text{SBUSY2}}$ input, or $\overline{\text{SRDY2}}$ input is selected : selected input level = "L" • Furthermore, if external clock is selected, the following requirements must also be met: – Input level of SCLK21 or SCLK22 = "H"
Transmission and reception stop condition	• To stop transmission and reception, set serial I/O initialization bit (bit 4 at address 0342_{16}) to "0" regardless internal clock and external clock.
Interrupt request generation timing	• 8-bit serial I/O mode : Interrupts requested when 8-bit data transfer is completed • Automatic transfer serial I/O mode :Interrupts requested when last receive data transfer to Automatic transfer RAM
Select function	• SOUT2 P-channel output disable function CMOS output or N-channel open-drain output can be selected • LSB first/MSB first selection Whether transmission/reception begins with bit 0 or bit 7 can be selected • Serial I/O2 clock pin select bit Serial clock input/output can be selected; SCLK21 or SCLK22 • SBUSY output, SSTB2 output select function (only automatic transfer serial mode) SBUSY output, SSTB2 output can be selected; 1-byte data transfer unit or all data transfer unit • SOUT2 pin control bit Either output active or high-impedance can be selected as a SOUT2 pin state at serial non-transfer .

Note 1: It is necessary to set the serial I/O clock pin select bit (bit 7 at address 0342_{16})

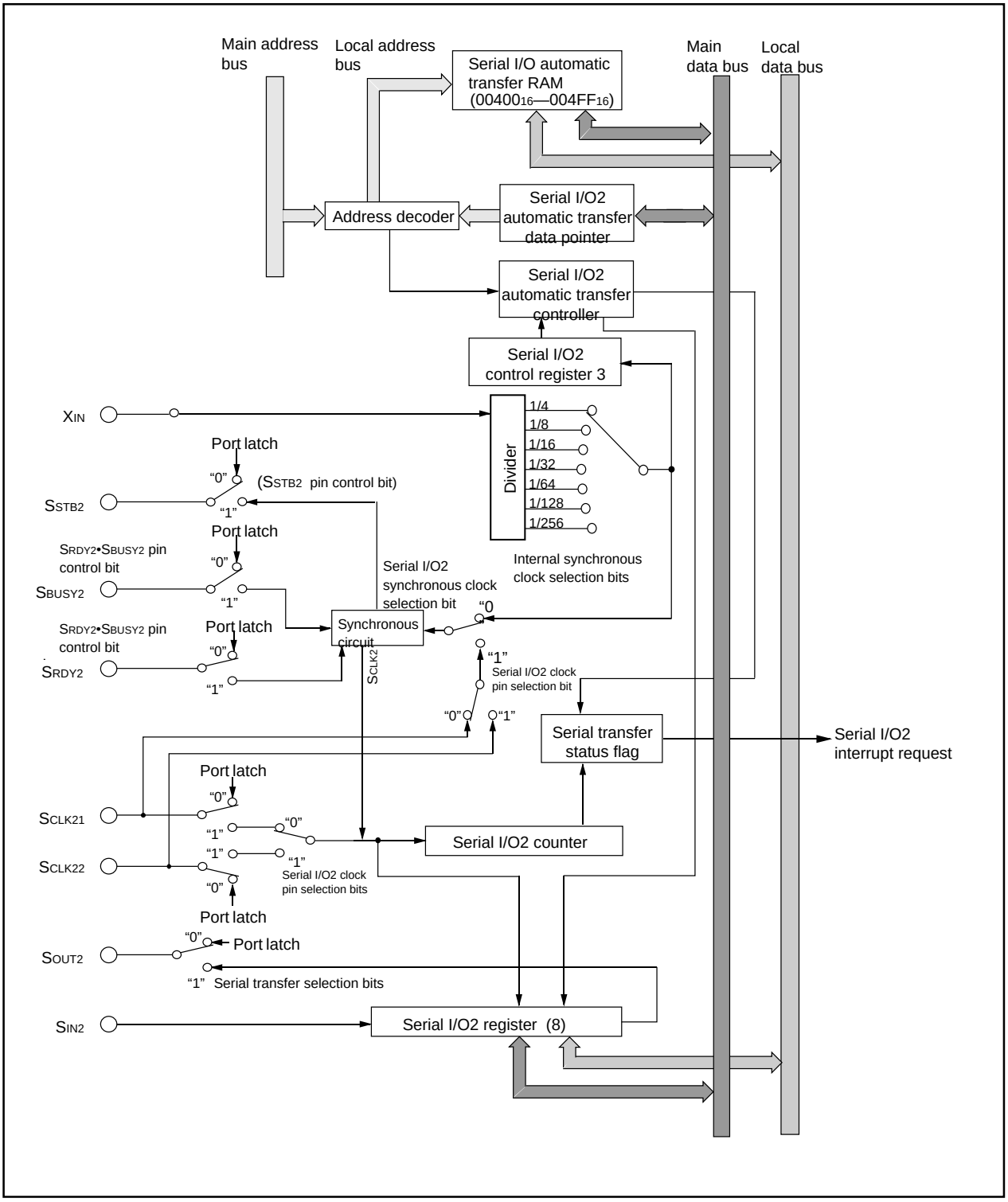


Figure GA-1. Block Diagram of Serial I/O2

Serial I/O2 control register 1

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: SIO2CON1 — Address: 0342_{16} — When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
SCON10	Serial transfer select bits	b1 b0 00: Serial I/O disabled (serial I/O pins are I/O ports) 01: 8-bits serial I/O 10: Inhibit 11: Automatic transfer serial I/O (8-bits)	○	○
SCON11			○	○
SCON12	Serial I/O2 synchronous clock select bits (S_{STB2} pin control bit)	b3 b2 00: Internal synchronous clock (S_{STB2} pin is an I/O port.) 01: External synchronous clock (S_{STB2} pin is an I/O port.) 10: Internal synchronous clock (S_{STB2} pin is an $\overline{S_{STB2}}$ output.) 11: Internal synchronous clock (S_{STB2} pin is an S_{STB2} output.)	○	○
SCON13			○	○
SCON14	Serial I/O initialization bit	0: Serial I/O initialization 1: Serial I/O enabled	○	○
SCON15	Transfer mode select bit	0: Full duplex (transmit and receive) mode (S_{IN2} pin is a S_{IN2} input.) 1: Transmit-only mode (S_{IN2} pin is an I/O port.)	○	○
SCON16	Transfer direction select bit	0: LSB first 1: MSB first	○	○
SCON17	Serial I/O2 clock pin select bit	0:S_{CLK21} (S_{CLK22} pin is an I/O port.) 1:S_{CLK22} (S_{CLK21} pin is an I/O port.)	○	○

Serial I/O2 control register 2

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: SIO2CON2 — Address: 0344_{16} — When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
SCON20	S_{RDY2} • S_{BUSY2} pin control bits	b3b2b1b0: S_{RDY2} pin / S_{BUSY2} pin 0000: I/O port / I/O port 0001: Not used 0010: S_{RDY2} output / I/O port 0011: S_{RDY2} output / I/O port 0100: I/O port / $\overline{S_{BUSY2}}$ input 0101: I/O port / S_{BUSY2} input 0110: I/O port / $\overline{S_{BUSY2}}$ output 0111: I/O port / S_{BUSY2} output 1000: S_{RDY2} input / $\overline{S_{BUSY2}}$ output 1001: S_{RDY2} input / S_{BUSY2} output 1010: S_{RDY2} input / $\overline{S_{BUSY2}}$ output 1011: S_{RDY2} input / S_{BUSY2} output 1100: $\overline{S_{RDY2}}$ output / $\overline{S_{BUSY2}}$ input 1101: $\overline{S_{RDY2}}$ output / S_{BUSY2} input 1110: S_{RDY2} output / $\overline{S_{BUSY2}}$ input 1111: S_{RDY2} output / S_{BUSY2} input	○	○
SCON21			○	○
SCON22			○	○
SCON23			○	○
SCON24	S_{BUSY2} output • S_{STB2} output function select bit (Valid in automatic transfer mode)	0: Functions as each 1-byte signal 1: Functions as signal for all transfer data	○	○
SCON25	Serial transfer status flag	0: Serial transfer completion 1: Serial transferring	○	○
SCON26	S_{OUT2} pin control bit (at no-transfer serial data)	0: Output active 1: Output high-impedance	○	○
SCON27	S_{OUT2} P-channel output disable bit	0: CMOS 3-state (P-channel output is valid.) 1: N-channel open-drain (P-channel output is invalid.)	○	○

Figure GA-2. Serial I/O2 Control Registers 1, 2

Serial I/O2 automatic transfer data pointer

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
SIO2DP	0340_{16}	00_{16}

Function	R	W
• Automatic transfer data pointer set Specify the low-order 8 bits of the first data store address on the serial I/O automatic transfer RAM. Data is written into the latch and read from the decrement counter.	○	○

Serial I/O2 register/transfer counter

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
SIO2	0346_{16}	00_{16}

Function	R	W
• Number of automatic transfer data set Set the number of automatic transfer data. Set a value one less than number of transfer data. Data is written into the latch and read from the decrement counter.	○	○

Serial I/O2 control register 3

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
SIO2CON3	0348_{16}	00000000_2

Bit symbol	Bit name	Function	R	W
TTRAN0	Automatic transfer interval set bits	b4b3b2b1b0 00000 :2 cycles of transfer clocks 00001 :3 cycles of transfer clocks : 11110 :32 cycles of transfer clocks 11111 :33 cycles of transfer clocks Data is written to a latch and read from a decrement counter.	○	○
TTRAN1			○	○
TTRAN2			○	○
TTRAN3			○	○
TTRAN4			○	○
TCLK0	Internal synchronous clock selection bits	b7b6b5 000:f(X_{IN})/4 001:f(X_{IN})/8 010:f(X_{IN})/16 011:f(X_{IN})/32 100:f(X_{IN})/64 101:f(X_{IN})/128 110:f(X_{IN})/256	○	○
TCLK1			○	○
TCLK2			○	○

Figure GA-3. Serial I/O2 automatic transfer data pointer

Table GA-2 lists the functions of the serial I/O2 input/output pins

Table GA-2. Functions of the serial I/O2 input/output pins

Pin name	Function	Method of selection
SOUT2 (P94)	Serial data output	Port P94 direction register (bit 4 at address 03F316)= "1" SOUT2 P-channel output disable bit (bit 7 at address 034416)= "0" , "1" SOUT2 pin control bit (bit 6 at address 034416)= "0" , "1" (Outputs dummy data when performing reception only)
SIN2 (P93)	Serial data input	Port P93 direction register (bit 4 at address 03F316)= "0" Transfer mode select bit (bit 5 at address 034216)= "0" (Input/output port when transfer mode select bit (bit 5 at address 034216)= "1")
SCLK21 (P95)	Transfer clock output	Serial I/O2 synchronous clock select bits (bits 2, 3 at address 034216) = "00" , "01" Serial I/O2 clock pin select bit (bit 7 at address 034216) = "0"
	Transfer clock input	Serial I/O2 synchronous clock select bits (bits 2, 3 at address 034216) = "01" , "11" Serial I/O2 clock pin select bit (bit 7 at address 034216) = "0" Port P95 direction register (bit 5 at address 03F316)= "0"
SCLK22 (P96)	Transfer clock output	Serial I/O2 synchronous clock select bits (bits 2, 3 at address 034216) = "00" , "01" Serial I/O2 clock pin select bit (bit 7 at address 034216) = "1"
	Transfer clock input	Serial I/O2 synchronous clock select bits (bits 2, 3 at address 034216) = "01" , "11" Serial I/O2 clock pin select bit (bit 7 at address 034216) = "1" Port P96 direction register (bit 6 at address 03F316)= "0"
SRDY2 (P90)	SRDY input / output	Set by SRDY2 • SBUSY2 pin control bits (bits 0 to 3 at address 034416)
SBUSY2 (P91)	SBUSY input / output	Set by SRDY2 • SBUSY2 pin control bits (bits 0 to 3 at address 034416) SBUSY2 output • SSTB2 output function select bit (bit 4 at address 034416)= "0" , "1"
SSTB2 (P92)	SSTB input / output	Serial I/O2 synchronous clock select bits (bits 2, 3 at address 034216) = "10" , "11" SBUSY2 output • SSTB2 output function select bit (bit 4 at address 034416)= "0" , "1"

SOUT2 Output

Either output active or high-impedance can be selected as a SOUT2 pin state at serial non-transfer by the SOUT2 pin control bit (bit 6 of address 034416).

However, when the external synchronous clock is selected, perform the following setup to put the SOUT2 pin into a high-impedance state.

When the SCLK2i (i = 1, 2) input is "H" after completion of transfer, set the SOUT2 pin control bit to "1". When the SCLK2i (i = 1, 2) input goes to "L" after the start of the next serial transfer, the SOUT2 pin control bit is automatically reset to "0" and put into an output active state.

Serial I/O2 Mode

There are two types of serial I/O2 modes: 8-bit serial I/O mode where automatic transfer RAM is not used, and an automatic transfer serial I/O mode.

(1) 8-bit Serial I/O Mode

Address 0346_{16} is assigned to the serial I/O2 register. When the internal synchronous clock is selected, a serial transfer of the 8-bit serial I/O is started by a write signal to the serial I/O2 register (address 0346_{16}).

The serial transfer status flag (bit 5 of address 0344_{16}) is set to "1" by writing into the serial I/O2 register and reset to "0" after completion of 8-bit transfer. At the same time, a serial I/O2 interrupt request occurs. If the transfer is completed, the receive data is read out from serial I/O2 register.

When the external synchronous clock is selected, the contents of the serial I/O2 register are continuously shifted while transfer clocks are input to SCLK21 or SCLK22. Therefore, the clock needs to be controlled externally.

(2) Automatic Transfer Serial I/O Mode

Address 0346_{16} is assigned to the transfer counter (1-byte units). The serial I/O2 automatic transfer controller controls the write and read operations of the serial I/O2 register. The serial I/O automatic transfer RAM is mapped to addresses 00400_{16} to $004FF_{16}$. Before starting transfer, make sure the 8 low-order bits of the address that contains the beginning data to be serially transferred is set to the automatic transfer data pointer (address 0340_{16}).

When the internal synchronous clock is selected, the transfer interval is inserted between one data and another in the following cases:

1. When using no handshake signal
2. When using the SRDY2 output, SBUSY2 output, and SSTB2 output of the handshake signal inde pendently
3. When using a combination of SRDY2 output and SSTB2 output or a combination of SBUSY2 output and SSTB2 output of the handshake signal

The transfer interval can be set in the range of 2 to 23 cycles using the automatic transfer interval set bit (bits 0–4 of address 0348_{16}).

Also, when using SBUSY2 output as a signal for each occurrence of the all transfer data, a transfer interval is inserted before the system starts sending or receiving the first data and after the system finished sending or receiving the last data, not just between one data and another.

Furthermore, when using SSTB2 output, the transfer interval between each 1-byte data is extended by 2 cycles from the set value no matter how the SBUSY2 output. SSTB2 output function select bit (bit 4 of address 0344_{16}) is set.

When using SBUSY2 output and SSTB2 output in combination as a signal for each occurrence of the all transfer data, the transfer interval after the system finished sending or receiving the last data is extended by 2 cycles from the set value.

When an external synchronous clock is selected, the automatic transfer interval is disabled.

When the internal synchronous clock is selected, automatic serial transfer starts by writing 1 less than the number of transfer bytes to the transfer counter (address 0346_{16}). When an external sync clock is selected, automatic serial transfer starts by writing 1 less than the number of transfer bytes to the transfer counter and the transfer clock is input. In this case, allow for at least 5 cycles of internal system clock before the transfer clock is input after writing to the transfer counter.

Also, for data to data transfer intervals, allow at least 5 cycles of internal system clock reckoning from a rise of clock at the last bit of one-byte data.

Regardless of whether the internal or external synchronous clock is selected, the automatic transfer data pointer and the transfer counter are decreased after each 1-byte data is received and then written into the automatic transfer RAM. The serial transfer status flag (bit5 of address 0344_{16}) is set to "1" by writing data into the transfer counter. The serial transfer status flag is reset to "0" after the last data is written into the automatic transfer RAM. At the same time, a serial I/O2 interrupt request occurs.

The values written in the automatic transfer data pointer (address 0340_{16}) and the automatic transfer interval set bits (bit 0 to bit 4 of address 0348_{16}) are held in the latch.

When data is written into the transfer counter, the values latched in the automatic transfer data pointer (address 0340_{16}) and the automatic transfer interval set bits (bit 0 to bit 4) are transferred to the decrement counter.

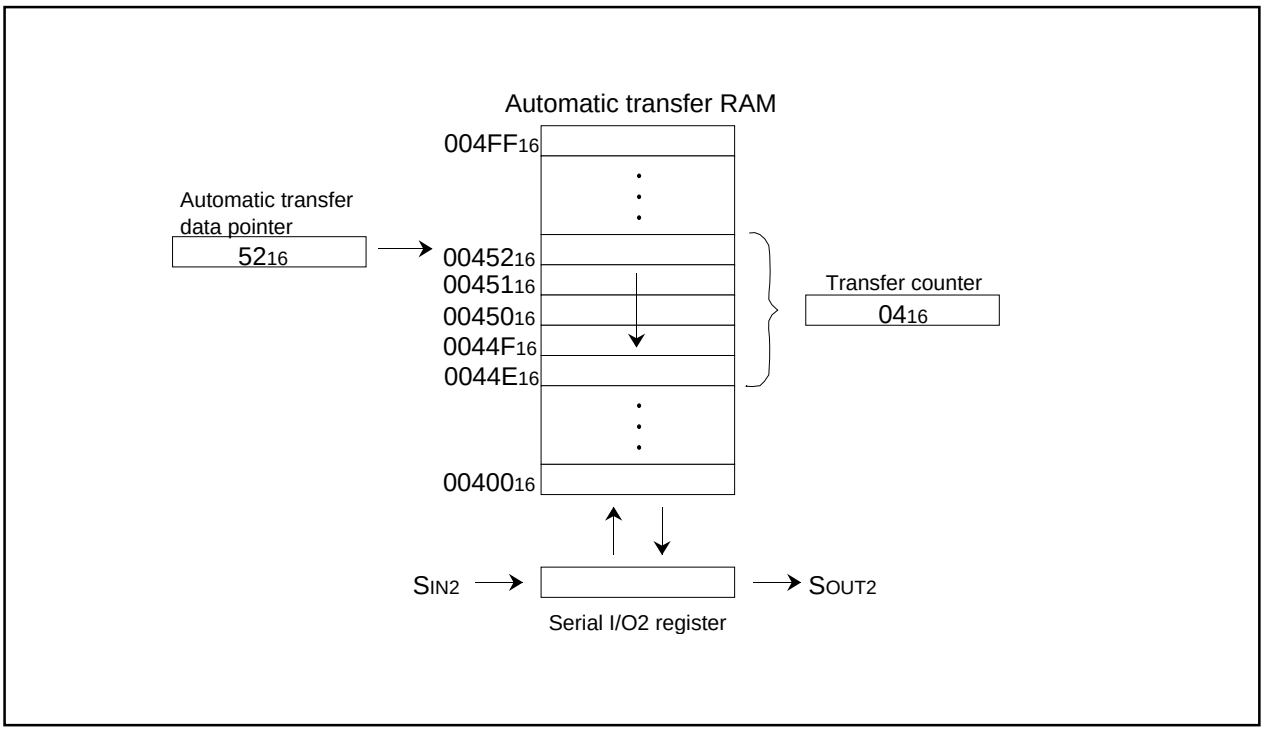


Figure GA-5. Automatic Transfer Serial I/O Operation

Handshake Signal

There are five types of handshake signal : SSTB2 output, SBUSY2 input/output, and SRDY2 input/output.

(1) SSTB2 output signal

The SSTB2 output is a signal to inform an end of transmission/reception to the serial transfer destination. The SSTB2 output signal can be used only when the internal synchronous clock is selected. In the initial status [serial I/O initialization bit (bit 4 of address 0342_{16}) = "0"], the SSTB2 output goes to "L" (bits 2, 3 of address 0342_{16}=11), or the $\overline{\text{SSTB2}}$ output goes to "H" (bits 2, 3 of address 0342_{16}=10).

At the end of transmit/receive operation, after the all data of the serial I/O2 register (address 0346_{16}) is output from SOUT2, SSTB2 output is "H" (or $\overline{\text{SSTB2}}$ output is "L") in the period of 1 cycle of the transfer clock. Furthermore, after 1 cycle, the serial transfer status flag (bit 5 of address 0344_{16}) is reset to "0".

In the automatic transfer serial I/O mode, whether the SSTB2 output is to be output at an end of each 1-byte data or after completion of transfer of all data can be selected by the SBUSY2 output • SSTB2 output function select bit (bit 4 of address 0344_{16}).

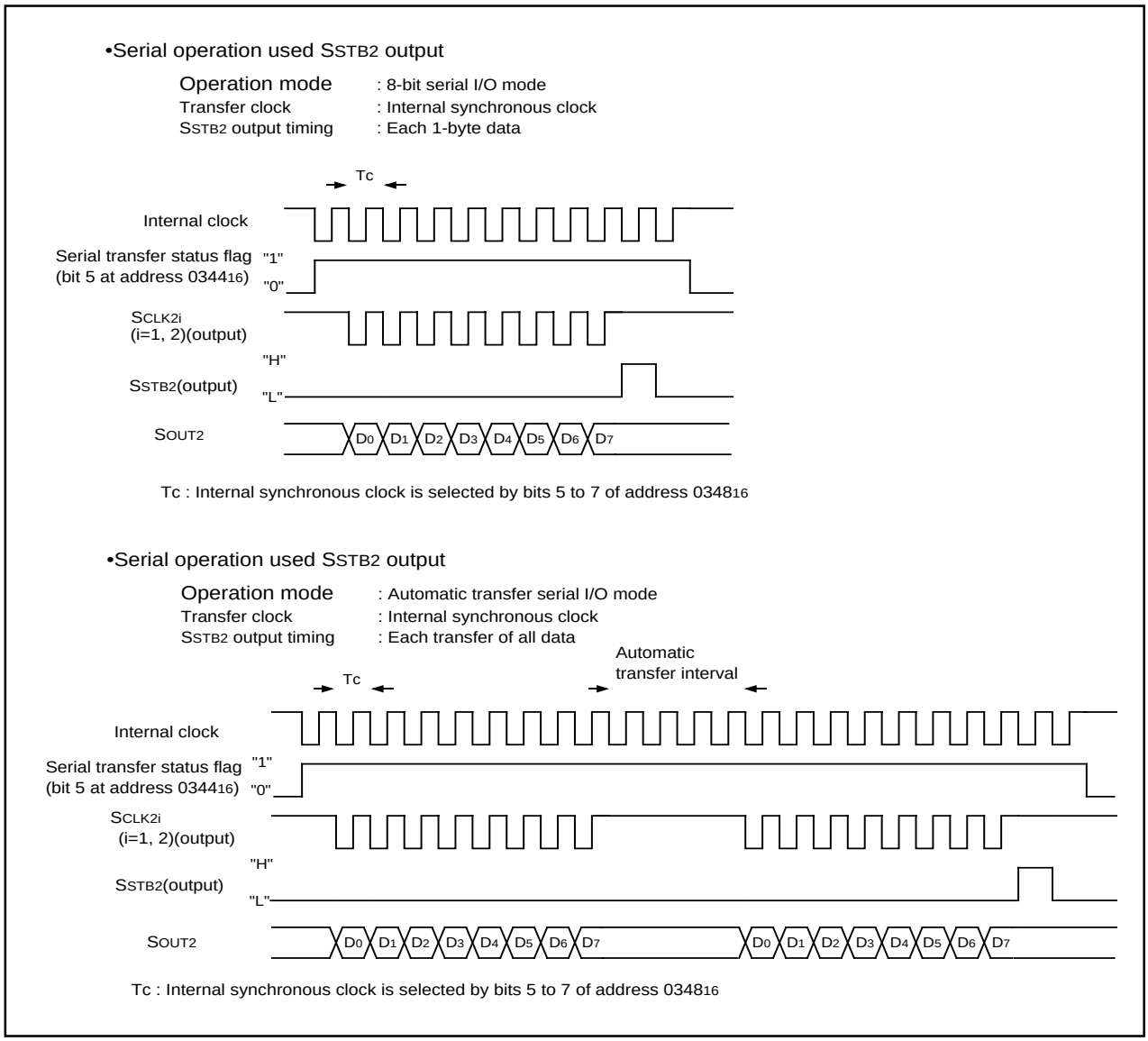


Figure GA-6. SSTB2 Output Operation

(2) SBUSY2 input signal

The SBUSY2 input is a signal requested to stop of transmission/reception from the serial transfer destination.

When the internal synchronous clock is selected, input a “H” level signal into the SBUSY2 input (or a “L” level signal into the $\overline{\text{SBUSY2}}$ input) in the initial status [serial I/O initialization bit (bit 4 of address 0342_{16}) = “0”]. When a “L” level signal into the SBUSY2 (or “H” on $\overline{\text{SBUSY2}}$) input for 1.5 cycles or more of transfer clock, transfer clocks are output from SCLK2i (i = 1, 2), and transmit/receive operation is started. When SBUSY2 input is driven “H” (or $\overline{\text{SBUSY2}}$ input is driven “L”) during transmit/receive operation, the transfer clock being output from SCLK2i (i = 1, 2) remains active until after the system finishes sending or receiving the designated number of bits, without stopping the transmit/receive operation immediately. The handshake unit of the 8-bit serial I/O is 8 bits, and that of the automatic transfer serial I/O is 8 bits.

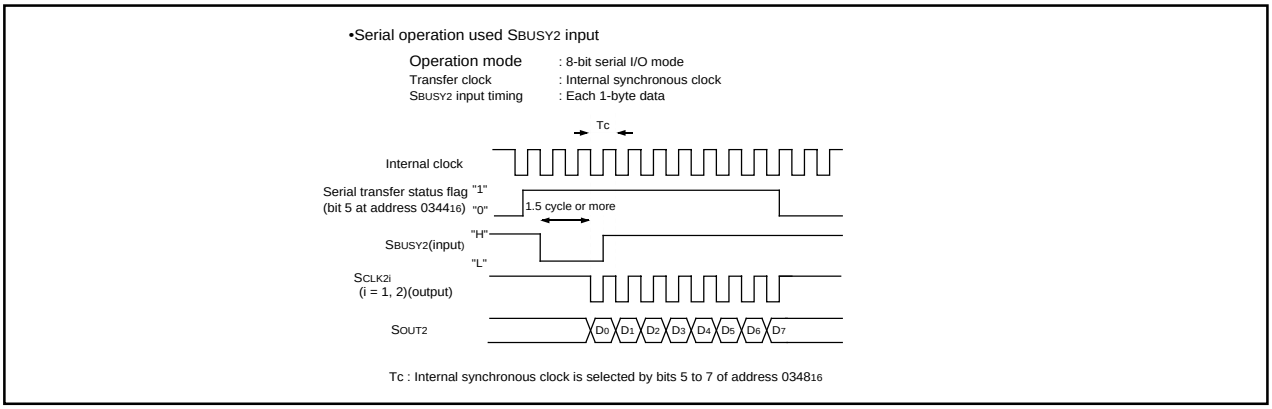


Figure GA-7. SBUSY2 Input Operation (1)

When the external synchronous clock is selected, input a “H” level signal into the SBUSY2 input (or a “L” level signal into the $\overline{\text{SBUSY2}}$ input) in the initial status[serial I/O initialization bit (bit 4 of address 0342_{16}) = “0”]. At this time, the transfer clock become invalid. The transfer clock become valid while a “L” level signal is input into the SBUSY2 input (or a “H” level signal into the $\overline{\text{SBUSY2}}$ input) and transmit/receive operation work.

When changing the input values into the SBUSY2 (or $\overline{\text{SBUSY2}}$) input at these operations, change them when the transfer clock input is in a “H” state. When the high-impedance of the SOUT2 output is selected by the SOUT2 pin control bit (bit 6 of address 0344_{16}), the SOUT2 becomes high-impedance, while a “H” level signal is input into the SBUSY2 input (or a “L” level signal into the $\overline{\text{SBUSY2}}$ input.)

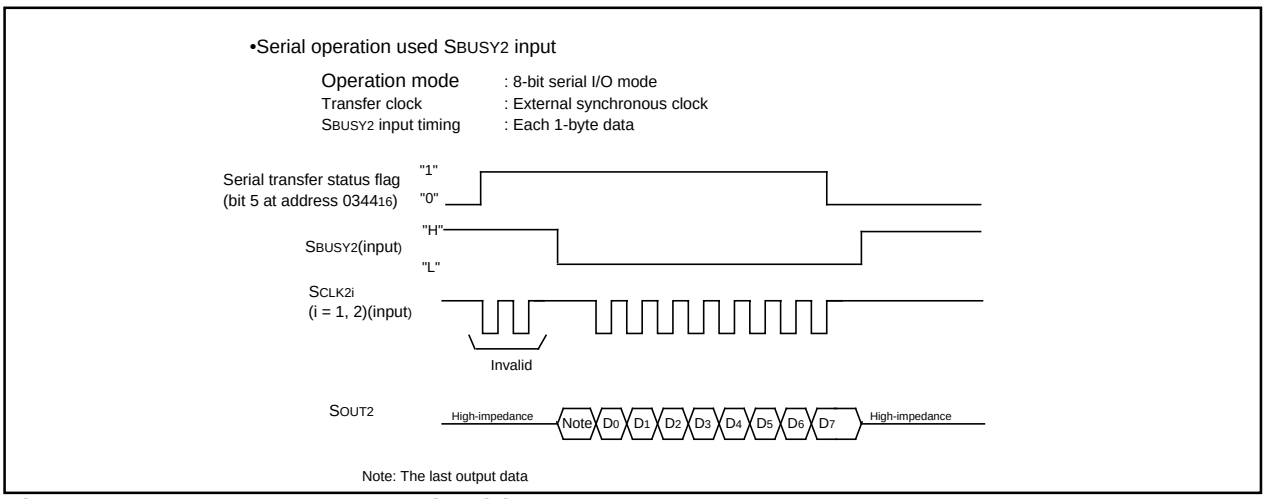


Figure GA-8. SBUSY2 Input Operation (2)

(3) SBUSY2 output signal

The SBUSY2 output is a signal which requests to stop of transmission/reception to the serial transfer destination. In the automatic transfer serial I/O mode, regardless of the internal or external synchronous clock, whether the SBUSY2 output is to be output at transfer of each 1-byte data or during transfer of all data can be selected by the SBUSY2 output • SSTB2 output function select bit (bit 4 of address 0344_{16}). In the initial status[serial I/O initialization bit (bit 4 of address 0342_{16}) = "0"], the status in which the SBUSY2 outputs "H" (or the $\overline{\text{SBUSY2}}$ outputs "L").

When the internal synchronous clock is selected, in the 8-bit serial I/O mode and the automatic transfer serial I/O mode (SBUSY2 output function: each 1-byte signal is selected), the SBUSY2 output goes to "L" (or the $\overline{\text{SBUSY2}}$ output goes to "H") before 0.5 cycle of the timing at which the transfer clock goes to "L" . In the automatic transfer serial I/O mode (the SBUSY2 output function: all transfer data is selected), the SBUSY2 output goes to "L" (or the $\overline{\text{SBUSY2}}$ output goes to "H") when the first transmit data is written into the serial I/O2 register (address 0346_{16}).

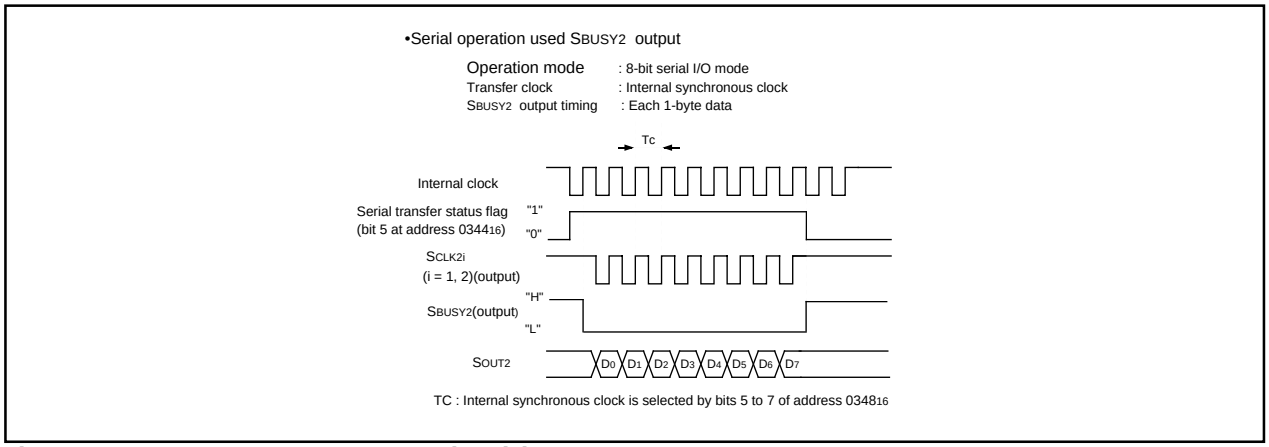


Figure GA-9. SBUSY2 Output Operation (1)

When the external synchronous clock is selected, the SBUSY2 output goes to "L" (or the $\overline{\text{SBUSY2}}$ output goes to "H") when transmit data is written into the serial I/O2 register(address 0346_{16}), regardless of the serial I/O transfer mode.

At termination of transmit/receive operation, in the 8-bit serial I/O mode, the SBUSY2 output goes to "H" (or the $\overline{\text{SBUSY2}}$ output returns to "L"), when the serial transfer status flag is set to "0", regardless of whether the internal or external synchronous clock is selected. Furthermore, in the automatic transfer serial I/O mode (SBUSY2 output function: each 1-byte signal is selected), the SBUSY2 output goes to "H" (or the $\overline{\text{SBUSY2}}$ output goes to "L") each time 1-byte of receive data is written into the automatic transfer RAM.

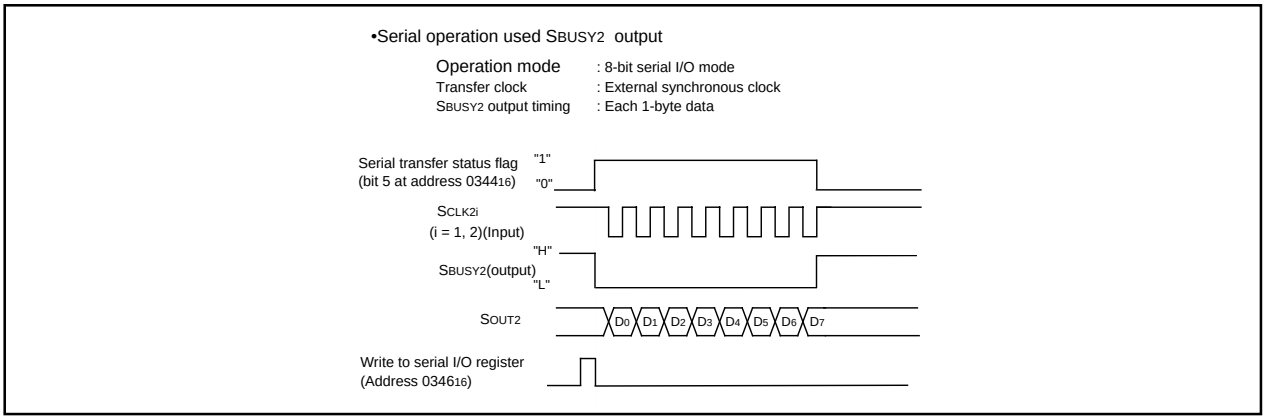


Figure GA-10. SBUSY2 Output Operation (2)

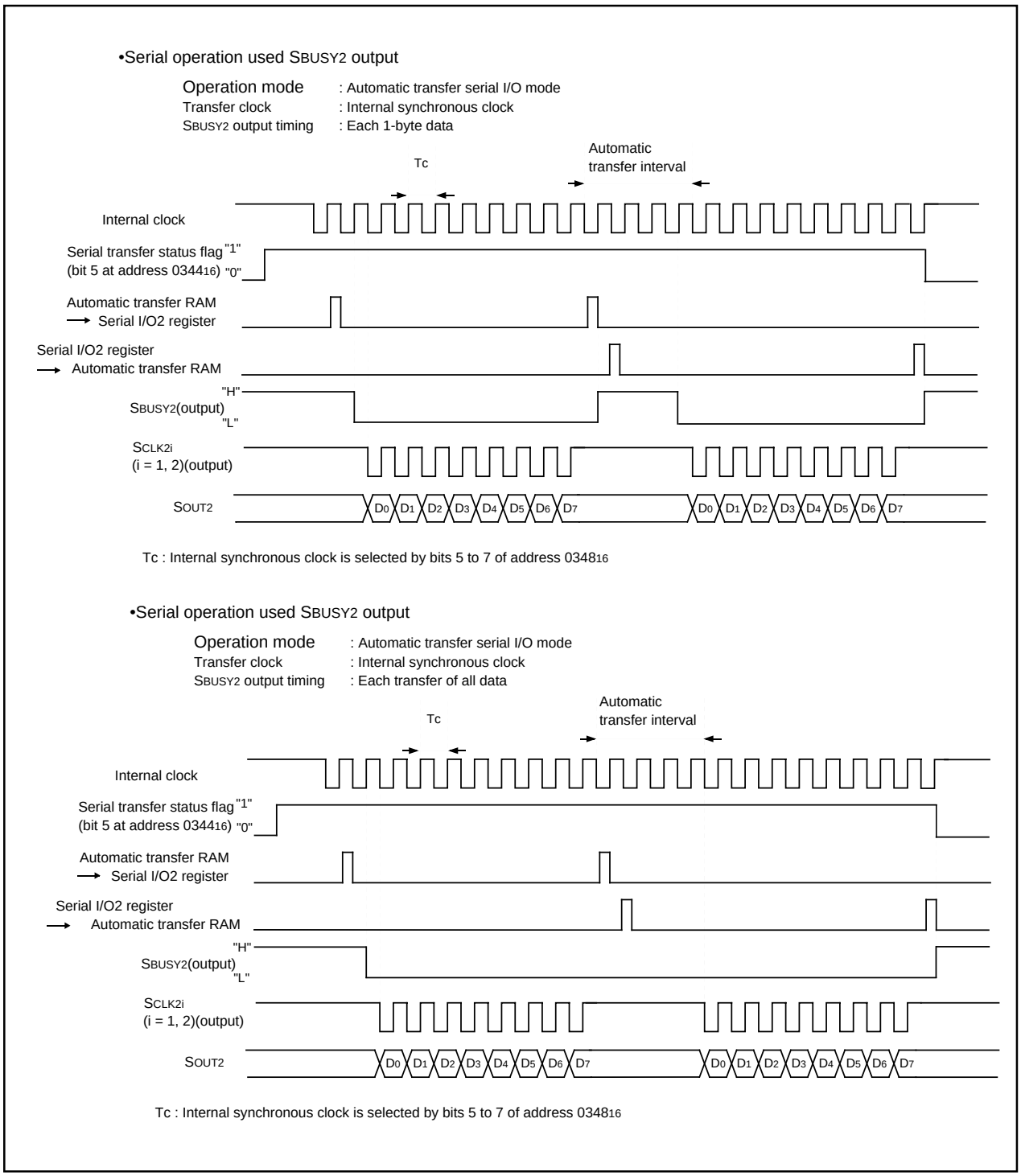


Figure GA-11. SBUSY2 Output Operation (3)

(4) SRDY2 output signal

The SRDY2 output is a transmit/receive enable signal which informs the serial transfer destination that transmit/receive is ready. In the initial status[serial I/O initialization bit (bit 4 of address 034216) = "0"], the SRDY2 output goes to "L" (or the $\overline{\text{SRDY2}}$ output goes to "H"). When the transmitted data is written to the serial I/O2 register (address 034616), the SRDY2 output goes to "H" (or the $\overline{\text{SRDY2}}$ output goes to "L"). When a transmit/receive operation is started and the transfer clock goes to "L", the SRDY2 output goes to "L" (or the $\overline{\text{SRDY2}}$ output goes to "H").

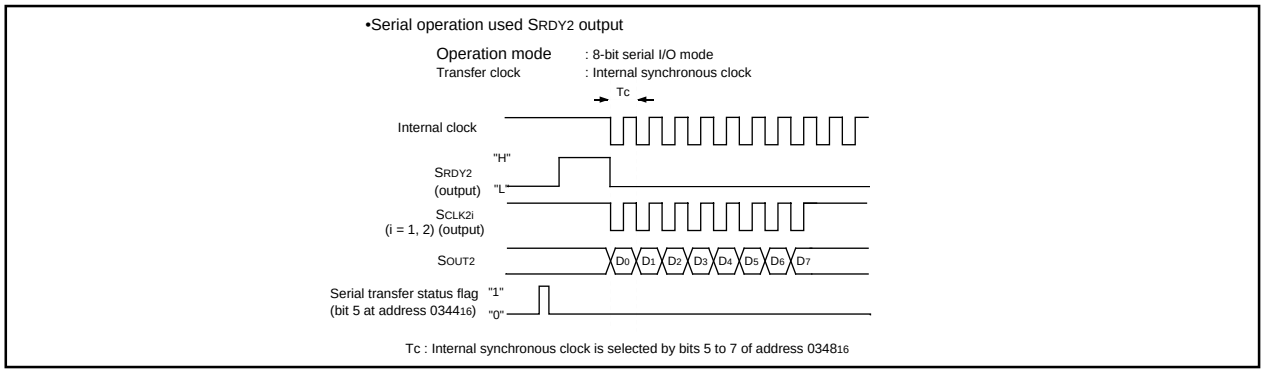


Figure GA-12. SRDY2 Output Operation

(5) SRDY2 input signal

The SRDY2 input is a signal for receiving a transmit/receive ready completion signal from the serial transfer destination. The SRDY2 input signal becomes valid only when the SRDY2 input and the SBUSY2 output are used.

When the internal synchronous clock is selected, input a "L" level signal into the SRDY2 input (or a "H" level signal into the $\overline{\text{SRDY2}}$ input) in the initial status[serial I/O initialization bit (bit 4 of address 034216) = "0"]. When a "H" level signal is input into the SRDY2 input (or a "L" level signal is input into the $\overline{\text{SRDY2}}$ input) for a period of 1.5 cycles or more of transfer clock, transfer clocks are output from the SCLK2i (i = 1, 2) output and a transmit/receive operation is started. When SRDY2 input is driven "L" (or $\overline{\text{SRDY2}}$ input is driven "H") during transmit/receive operation, the transfer clock being output from SCLK2i (i = 1, 2) remains active until after the system finishes sending or receiving the designated number of bits, without stopping the transmit/receive operation immediately.

The handshake unit of the 8-bit serial I/O is 8 bits, and that of the automatic transfer serial I/O is 8 bits.

When the external synchronous clock is selected, the SRDY2 input becomes one of the triggers to output the SBUSY2 signal. To start a transmit/receive operation (SBUSY2 output: "L", (or $\overline{\text{SBUSY2}}$ output: "H")), input a "H" level signal into the SRDY2 input (or a "L" level signal into the $\overline{\text{SRDY2}}$ input,) and also write transmit data into the serial I/O2 register (address 034616).

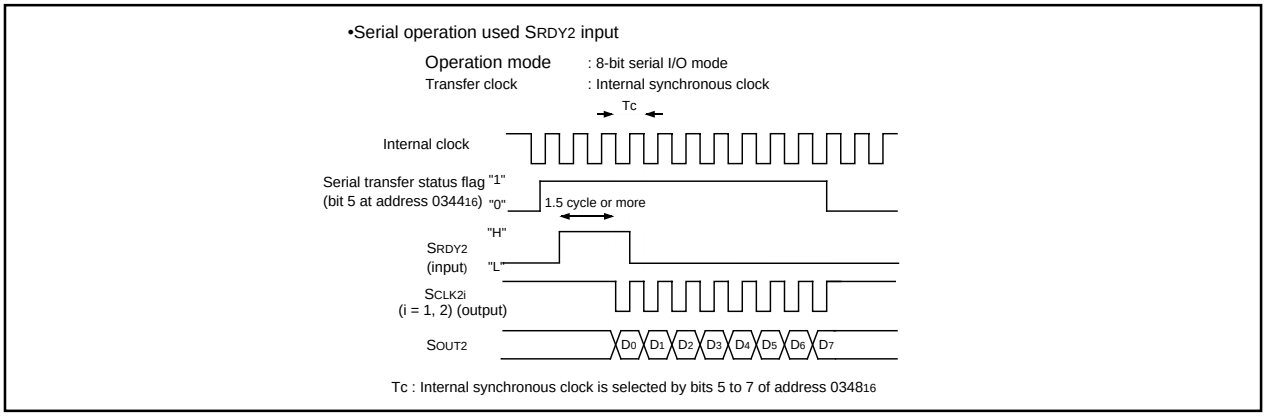


Figure GA-13. SRDY2 Input Operation

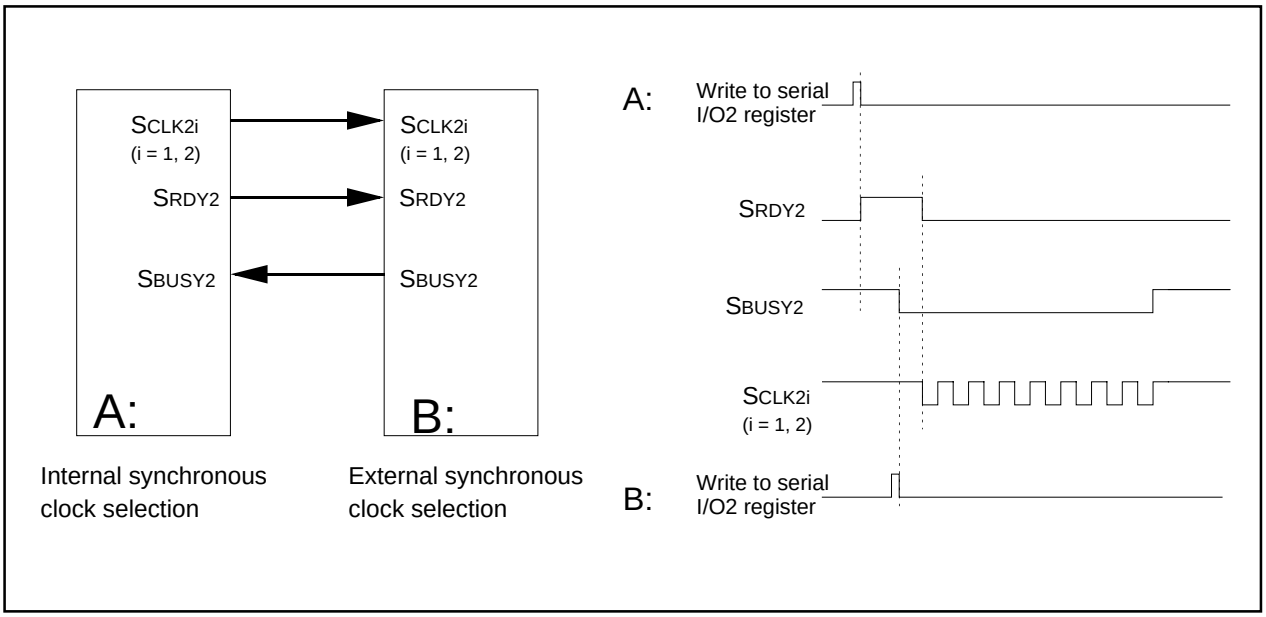


Figure GA-14. Handshake Operation at Serial I/O2 Mutual Connecting (1)

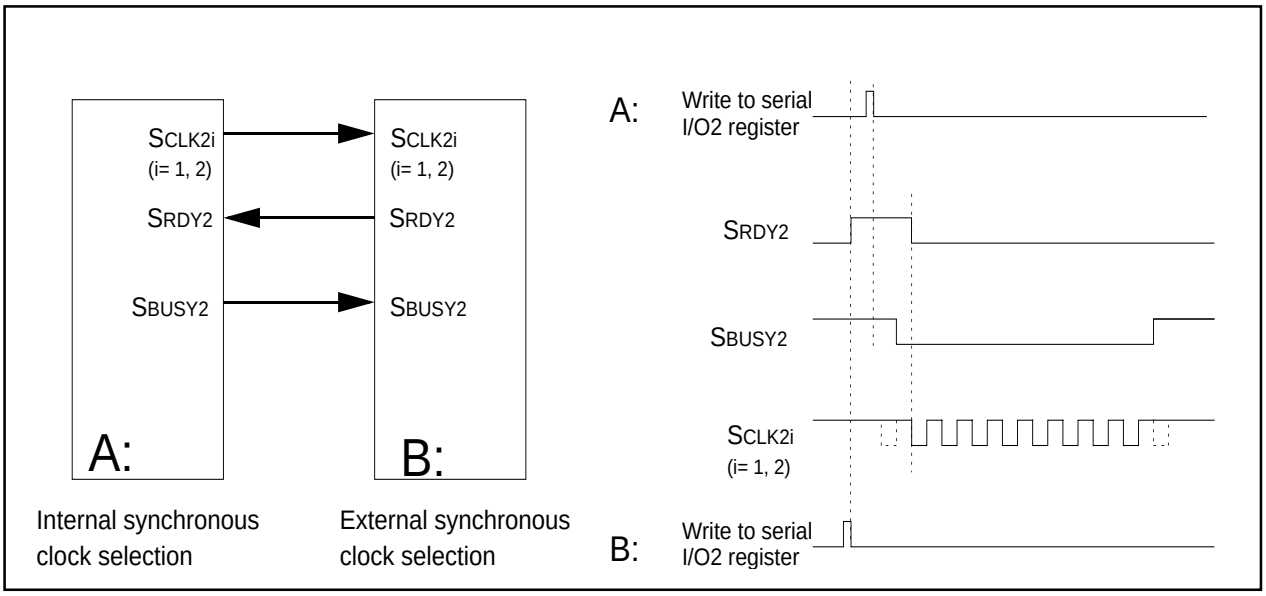


Figure GA-15. Handshake Operation at Serial I/O2 Mutual Connecting (2)

A-D Converter

The A-D converter consists of one 10-bit successive approximation A-D converter circuit with a capacitive coupling amplifier. Pins $P10_0$ to $P10_7$ also function as the analog signal input pins. The direction registers of these pins for A-D conversion must therefore be set to input. The Vref connect bit (bit 5 at address $03D7_{16}$) can be used to isolate the resistance ladder of the A-D converter from the reference voltage input pin (V_{REF}) when the A-D converter is not used. Doing so stops any current flowing into the resistance ladder from V_{REF}, reducing the power dissipation. When using the A-D converter, start A-D conversion only after setting bit 5 of $03D7_{16}$ to connect V_{REF}.

The result of A-D conversion is stored in the A-D registers of the selected pins. When set to 10-bit precision, the low 8 bits are stored in the even addresses and the high 2 bits in the odd addresses. When set to 8-bit precision, the low 8 bits are stored in the even addresses.

Table JA-1 shows the performance of the A-D converter. Figure JA-1 shows the block diagram of the A-D converter, and Figures JA-2 and JA-3 show the A-D converter-related registers.

Table JA-1. Performance of A-D converter

Item	Performance
Method of A-D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage (Note 1)	0V to AV_{CC} (V_{CC})
Operating clock ϕ_{AD} (Note 2)	V_{CC} = 5V f_{AD}/divide-by-2 of f_{AD}/divide-by-4 of f_{AD}, $f_{AD}=f(X_{IN})$ V_{CC} = 3V divide-by-2 of f_{AD}/divide-by-4 of f_{AD}, $f_{AD}=f(X_{IN})$
Resolution	8-bit or 10-bit (selectable)
Absolute precision	V_{CC} = 5V • Without sample and hold function ±3LSB • With sample and hold function (8-bit resolution) ±2LSB • Without sample and hold function (10-bit resolution) ±3LSB V_{CC} = 3V • Without sample and hold function (8-bit resolution)(Note 3) ±2LSB
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog input pins	8pins (AN_0 to AN_7)
A-D conversion start condition	•Software trigger A-D conversion starts when the A-D conversion start flag changes to “1”
Conversion speed per pin	•Without sample and hold function 8-bit resolution: 49 ϕ_{AD} cycles, 10-bit resolution: 59 ϕ_{AD} cycles • With sample and hold function 8-bit resolution: 28 ϕ_{AD} cycles, 10-bit resolution: 33 ϕ_{AD} cycles

Note 1: Does not depend on use of sample and hold function.

Note 2: Without sample and hold function, set the ϕ_{AD} frequency to 250kHz min.
With the sample and hold function, set the ϕ_{AD} frequency to 1MHz min.

Note 3: Only mask ROM version.

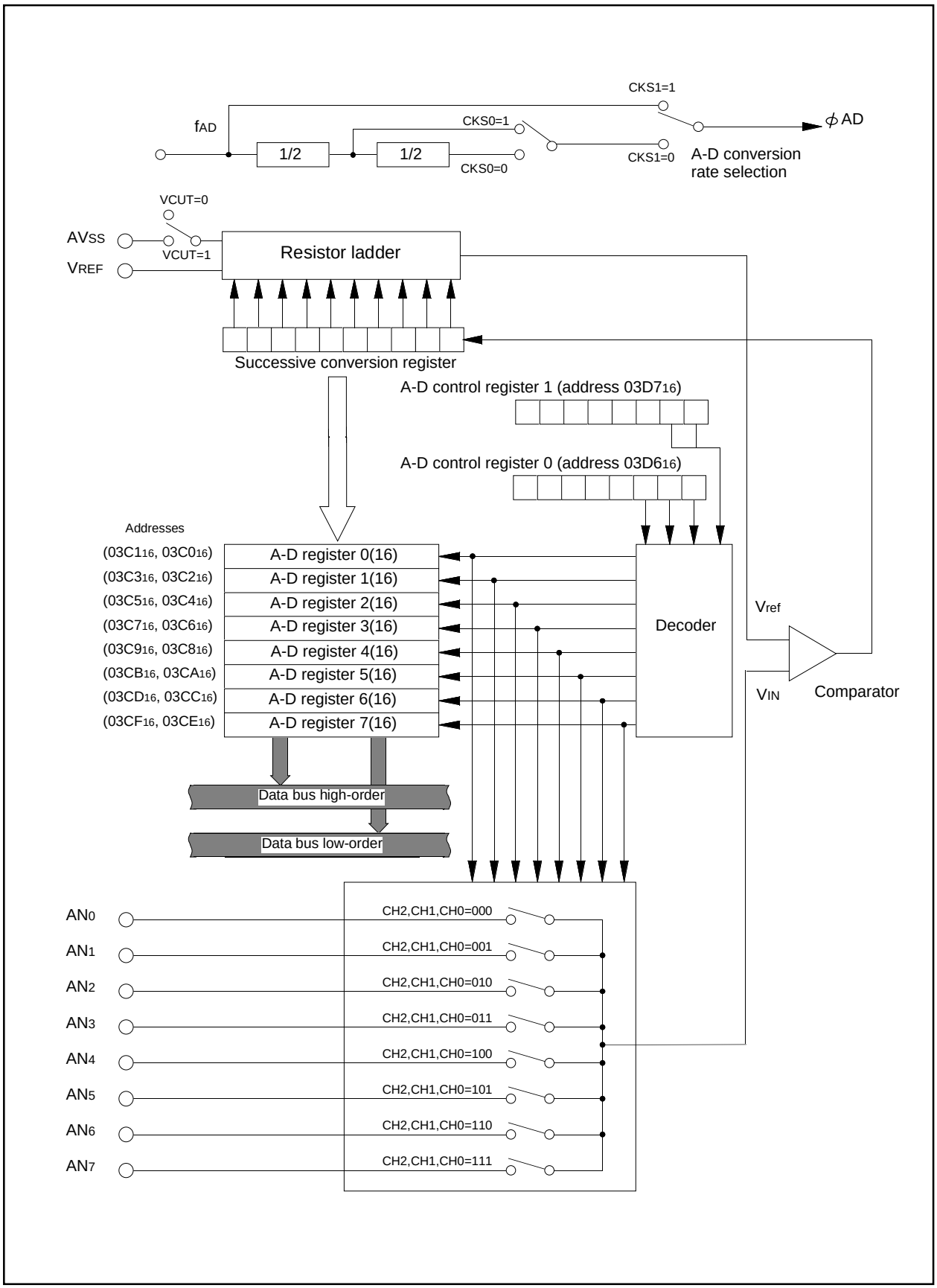


Figure JA-1. Block diagram of A-D converter

A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
		0					

Symbol	Address	When reset
ADCON0	03D6$_{16}$	00000XXX$_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN$_0$ is selected 0 0 1 : AN$_1$ is selected 0 1 0 : AN$_2$ is selected 0 1 1 : AN$_3$ is selected 1 0 0 : AN$_4$ is selected 1 0 1 : AN$_5$ is selected 1 1 0 : AN$_6$ is selected 1 1 1 : AN$_7$ is selected	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 Repeat sweep mode 1	○	○
MD1			○	○
Must always be “0”.			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : f$_{AD}$/4 is selected 1 : f$_{AD}$/2 is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
0	0						

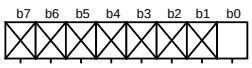
Symbol	Address	When reset
ADCON1	03D7$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN$_0$, AN$_1$ (2 pins) 0 1 : AN$_0$ to AN$_3$ (4 pins) 1 0 : AN$_0$ to AN$_5$ (6 pins) 1 1 : AN$_0$ to AN$_7$ (8 pins)	○	○
SCAN1		When repeat sweep mode 1 is selected b1 b0 0 0 : AN$_0$ (1 pin) 0 1 : AN$_0$, AN$_1$ (2 pins) 1 0 : AN$_0$ to AN$_2$ (3 pins) 1 1 : AN$_0$ to AN$_3$ (4 pins)	○	○
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1 1 : Repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : f$_{AD}$/2 or f$_{AD}$/4 is selected 1 : f$_{AD}$ is selected	○	○
VCUT	Vref connect bit	0 : Vref not connected 1 : Vref connected	○	○
Must always be “0”.			○	○
			○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

Figure JA-2. A-D converter-related registers (1)

A-D control register 2 (Note)



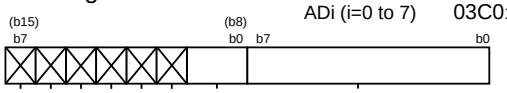
Symbol	Address	When reset
ADCON2	$03D4_{16}$	$XXXXXXX0_2$

Bit symbol	Bit name	Function	R	W
SMP	A-D conversion method select bit	0 Without sample and hold 1 With sample and hold	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.			—	—

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D register i

Symbol	Address	When reset
ADi (i=0 to 7)	$03C0_{16}$ to $03CF_{16}$	Indeterminate



Function	R	W
Eight low-order bits of A-D conversion result	○	×
• During 10-bit mode Two high-order bits of A-D conversion result	○	×
• During 8-bit mode When read, the content is indeterminate	×	×
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.	—	—

Figure JA-3. A-D converter-related registers (2)

(1) One-shot mode

In one-shot mode, the pin selected using the analog input pin select bit is used for one-shot A-D conversion. Table JA-2 shows the specifications of one-shot mode. Figure JA-4 shows the A-D control register in one-shot mode.

Table JA-2. One-shot mode specifications

Item	Specification
Function	The pin selected by the analog input pin select bit is used for one A-D conversion
Start condition	Writing "1" to A-D conversion start flag
Stop condition	•End of A-D conversion (A-D conversion start flag changes to "0") •Writing "0" to A-D conversion start flag
Interrupt request generation timing	End of A-D conversion
Input pin	One of AN_0 to AN_7, as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin

A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	0			

Symbol	Address	When reset
ADCON0	$03D6_{16}$	$00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN_0 is selected 0 0 1 : AN_1 is selected 0 1 0 : AN_2 is selected 0 1 1 : AN_3 is selected 1 0 0 : AN_4 is selected 1 0 1 : AN_5 is selected 1 1 0 : AN_6 is selected 1 1 1 : AN_7 is selected	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode	○	○
MD1			○	○
Must always be "0".			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0: $f_{AD}/4$ is selected 1: $f_{AD}/2$ is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
0	0	1			0		

Symbol	Address	When reset
ADCON1	$03D7_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	Invalid in one-shot mode	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Must always be "0".			○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

Figure JA-4. A-D conversion register in one-shot mode

(2) Repeat mode

In repeat mode, the pin selected using the analog input pin select bit is used for repeated A-D conversion. Table JA-3 shows the specifications of repeat mode. Figure JA-5 shows the A-D control register in repeat mode.

Table JA-3. Repeat mode specifications

Item	Specification
Function	The pin selected by the analog input pin select bit is used for repeated A-D conversion
Star condition	Writing "1" to A-D conversion start flag
Stop condition	Writing "0" to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	One of AN_0 to AN_7, as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin

A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	1			

Symbol ADCON0 — Address $03D6_{16}$ — When reset $00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN_0 is selected 0 0 1 : AN_1 is selected 0 1 0 : AN_2 is selected 0 1 1 : AN_3 is selected 1 0 0 : AN_4 is selected 1 0 1 : AN_5 is selected 1 1 0 : AN_6 is selected 1 1 1 : AN_7 is selected	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 1 : Repeat mode	○	○
MD1			○	○
Must always be "0".			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversin result is indeterminate.

A-D control register 1 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
0	0	1			0		

Symbol ADCON1 — Address $03D7_{16}$ — When reset 00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	Invalid in repeat mode	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Must always be "0".			○	○
			○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversn result is indeterminate.

Figure JA-5. A-D conversion register in repeat mode

(3) Single sweep mode

In single sweep mode, the pins selected using the A-D sweep pin select bit are used for one-by-one A-D conversion. Table JA-4 shows the specifications of single sweep mode. Figure JA-6 shows the A-D control register in single sweep mode.

Table JA-4. Single sweep mode specifications

Item	Specification
Function	The pins selected by the A-D sweep pin select bit are used for one-by-one A-D conversion
Start condition	Writing "1" to A-D converter start flag
Stop condition	•End of A-D conversion (A-D conversion start flag changes to "0", except when external trigger is selected) •Writing "0" to A-D conversion start flag
Interrupt request generation timing	End of A-D conversion
Input pin	AN_0 and AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins), or AN_0 to AN_7 (8 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin

A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
		0	1	0			

Symbol	Address	When reset
ADCON0	$03D6_{16}$	$00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in single sweep mode	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 1 0 : Single sweep mode	○	○
MD1			○	○
Must always be "0".			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
0	0	1			0		

Symbol	Address	When reset
ADCON1	$03D7_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN_0, AN_1 (2 pins) 0 1 : AN_0 to AN_3 (4 pins) 1 0 : AN_0 to AN_5 (6 pins) 1 1 : AN_0 to AN_7 (8 pins)	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Must always be "0".			○	○
			○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

Figure JA-6. A-D conversion register in single sweep mode

(4) Repeat sweep mode 0

In repeat sweep mode 0, the pins selected using the A-D sweep pin select bit are used for repeat sweep A-D conversion. Table JA-5 shows the specifications of repeat sweep mode 0. Figure JA-7 shows the A-D control register in repeat sweep mode 0.

Table JA-5. Repeat sweep mode 0 specifications

Item	Specification
Function	The pins selected by the A-D sweep pin select bit are used for repeat sweep A-D conversion
Start condition	Writing "1" to A-D conversion start flag
Stop condition	Writing "0" to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	AN_0 and AN_1 (2 pins), AN0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins), or AN_0 to AN_7 (8 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)

A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
		0	1	1			

Symbol: ADCON0 — Address: $03D6_{16}$ — When reset: $00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in repeat sweep mode 0	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 0	○	○
MD1			○	○
Must always be "0".			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
0	0	1			0		

Symbol: ADCON1 — Address: $03D7_{16}$ — When reset: 00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When repeat sweep mode 1 is selected b1 b0 0 0 : AN_0, AN_1 (2 pins) 0 1 : AN_0 to AN_3 (4 pins) 1 0 : AN_0 to AN_5 (6 pins) 1 1 : AN_0 to AN_7 (8 pins)	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	1 : Any mode other than repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Must always be "0".			○	○
			○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

Figure JA-7. A-D conversion register in repeat sweep mode 0

(5) Repeat sweep mode 1

In repeat sweep mode 1, all pins are used for A-D conversion with emphasis on the pin or pins selected using the A-D sweep pin select bit. Table JA-6 shows the specifications of repeat sweep mode 1. Figure JA-8 shows the A-D control register in repeat sweep mode 1.

Table JA-6. Repeat sweep mode 1 specifications

Item	Specification
Function	All pins perform repeat sweep A-D conversion, with emphasis on the pin or pins selected by the A-D sweep pin select bit Example : AN_0 selected -> AN_0 -> AN_1 -> AN_0 -> AN_2 -> AN_0 -> AN_3, etc
Start condition	Writing "1" to A-D conversion start flag
Stop condition	Writing "0" to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	Emphasis on the pin AN_0 (1 pin), AN_0 and AN_1 (2 pins), AN_0 to AN_2 (3 pins), AN_0 to AN_3 (4 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)

A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
		0	1	1			

Symbol	Address	When reset
ADCON0	$03D6_{16}$	$00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in repeat sweep mode 0	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 1	○	○
MD1			○	○
Must always be "0".			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
0	0	1			1		

Symbol	Address	When reset
ADCON1	$03D7_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When repeat sweep mode 1 is selected b1 b0 0 0 : AN_0 (1 pins) 0 1 : AN_0, AN_1 (2 pins) 1 0 : AN_0 to AN_2 (3 pins) 1 1 : AN_0 to AN_3 (4 pins)	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	1 : Repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Must always be "0".			○	○
			○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

Figure JA-8. A-D conversion register in repeat sweep mode 1

(a) Sample and hold

Sample and hold is selected by setting bit 0 of the A-D control register 2 (address 03D4$_{16}$) to “1”. When sample and hold is selected, the rate of conversion of each pin increases. As a result, 28 ϕ_{AD} cycles are achieved with 8-bit resolution and 33 ϕ_{AD} cycles with 10-bit resolution. Sample and hold can be selected in all modes. However, in all modes, be sure to specify before starting A-D conversion whether sample and hold is to be used.

D-A Converter

This is an 8-bit, R-2R type D-A converter. The microcomputer contains two independent D-A converters of this type. D-A conversion is performed when a value is written to the corresponding D-A register. Bits 0 and 1 (D-A output enable bits) of the D-A control register decide if the result of conversion is to be output. Do not set the target port to output mode if D-A conversion is to be performed.

Output analog voltage (V) is determined by a set value (n : decimal) in the D-A register.

$V = V_{REF} \times n / 256$ (n = 0 to 255)

V_{REF} : reference voltage

Table JB-1 lists the performance of the D-A converter. Figure JB-1 shows the block diagram of the D-A converter. Figure JB-2 shows the D-A control register. Figure JB-3 shows the D-A converter equivalent circuit.

Table JB-1. Performance of D-A converter

Item	Performance
Conversion method	R-2R method
Resolution	8 bits
Analog output pin	2 channels

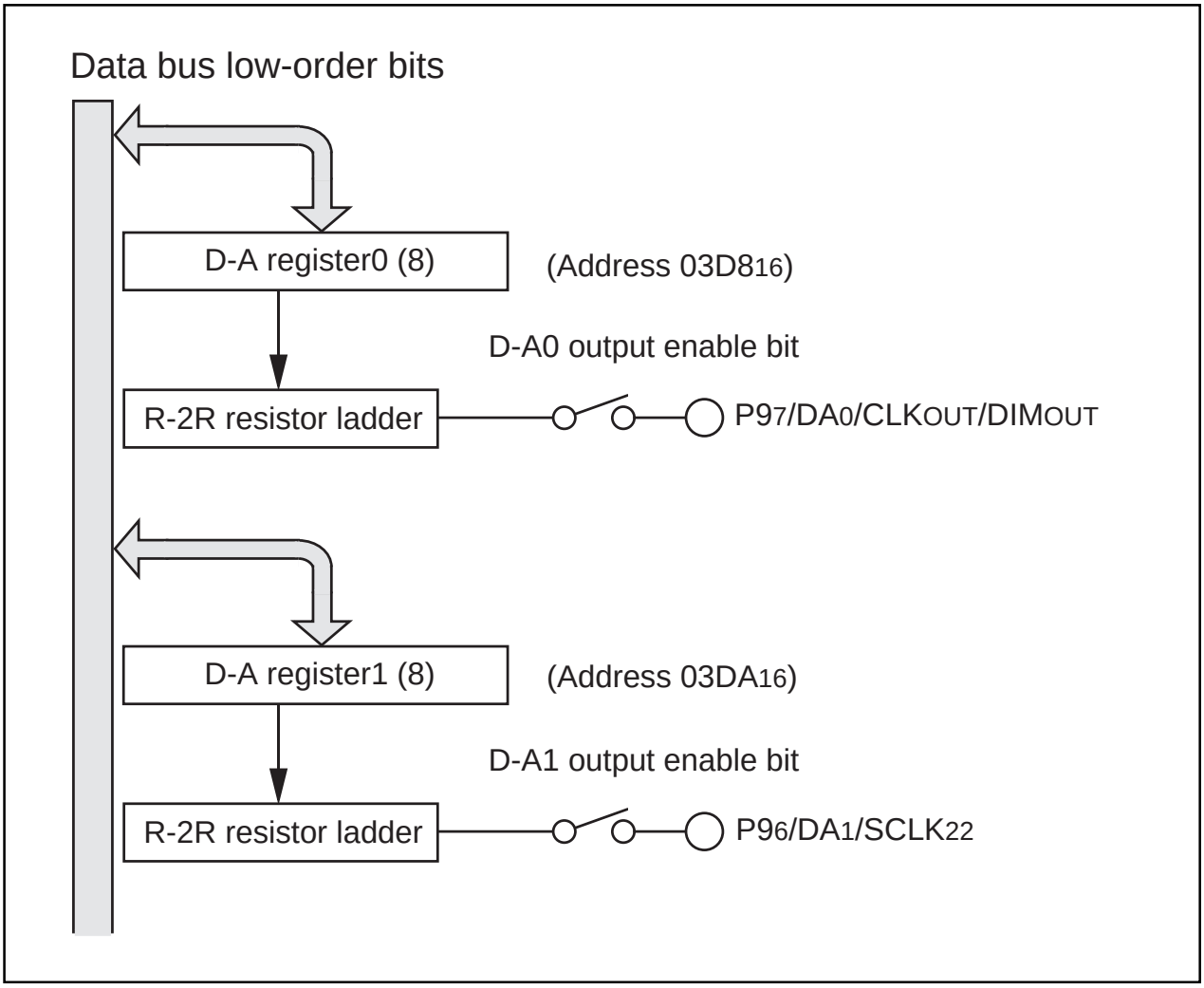


Figure JB-1. Block diagram of D-A converter

D-A control register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
DACON	$03DC_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
DA0E	D-A0 output enable bit	0 : Output disabled 1 : Output enabled	○	○
DA1E	D-A1 output enable bit	0 : Output disabled 1 : Output enabled	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.			–	–

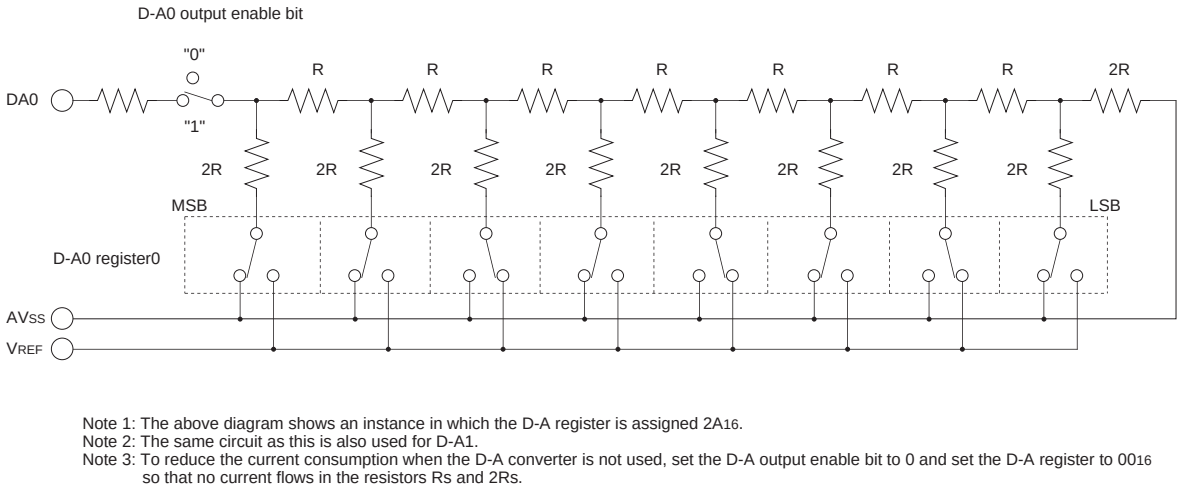
D-A register

b7	b0

Symbol	Address	When reset
DAi (i = 0,1)	$03D8_{16}$, $03DA_{16}$	Indeterminate

Function	R	W
Output value of D-A conversion	○	○

Figure JB-2. D-A control register



Note 1: The above diagram shows an instance in which the D-A register is assigned $2A_{16}$.
Note 2: The same circuit as this is also used for D-A1.
Note 3: To reduce the current consumption when the D-A converter is not used, set the D-A output enable bit to 0 and set the D-A register to 00_{16} so that no current flows in the resistors Rs and 2Rs.

Figure JB-3. D-A converter equivalent circuit

CRC Calculation Circuit

The Cyclic Redundancy Check (CRC) calculation circuit detects an error in data blocks. The microcomputer uses a generator polynomial of CRC_CCITT ($X^{16} + X^{12} + X^5 + 1$) to generate CRC code.

The CRC code is a 16-bit code generated for a block of a given data length in multiples of 8 bits. The CRC code is set in a CRC data register each time one byte of data is transferred to a CRC input register after writing an initial value into the CRC data register. Generation of CRC code for one byte of data is completed in two machine cycles.

Figure UC-1 shows the block diagram of the CRC circuit. Figure UC-2 shows the CRC-related registers. Figure UC-3 shows the calculation example using the CRC calculation circuit

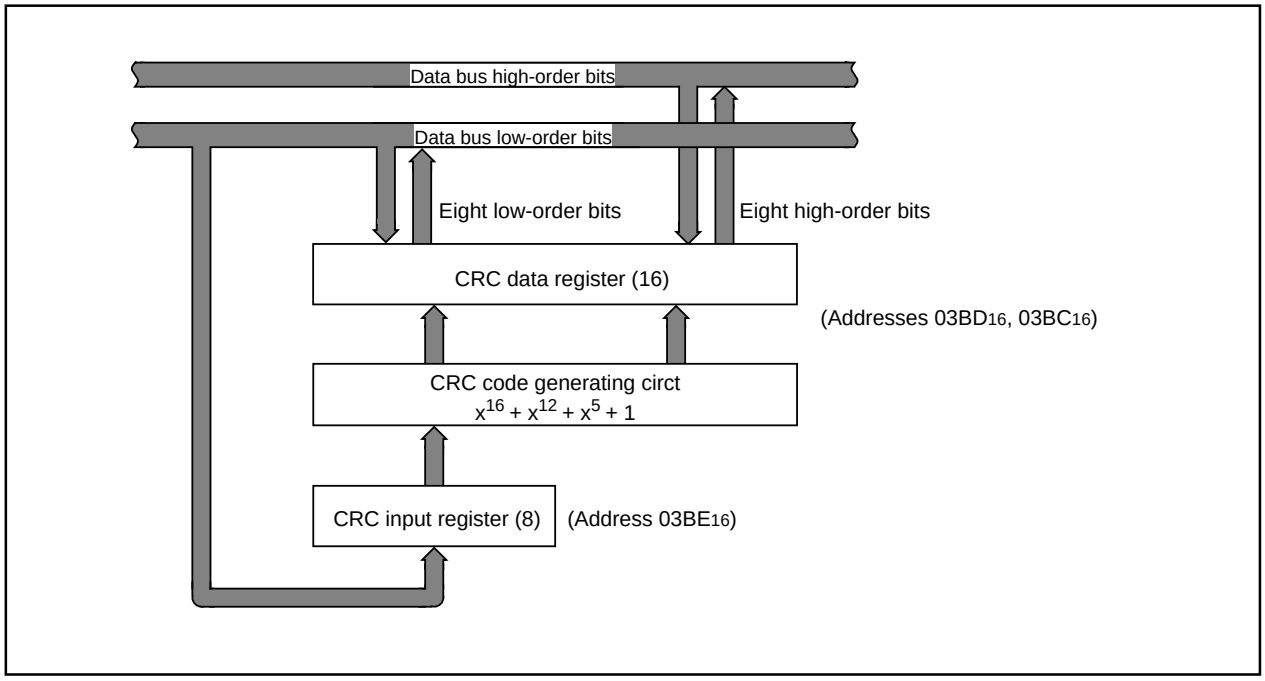


Figure UC-1. Block diagram of CRC circuit

CRC data register

(b15) (b8) b7 b0 b7 b0

Symbol	Address	When reset
CRCD	03BD16, 03BC16	Indeterminate

Function	Values that can be set	R	W
CRC calculation result output register	0000₁₆ to FFFF₁₆	○	○

CRC input register

b7 b0

Symbo	Address	When reset
CRCIN	03BE16	Indeterminate

Function	Values that can be set	R	W
Data input register	00₁₆ to FF₁₆	○	○

Figure UC-2. CRC-related registers

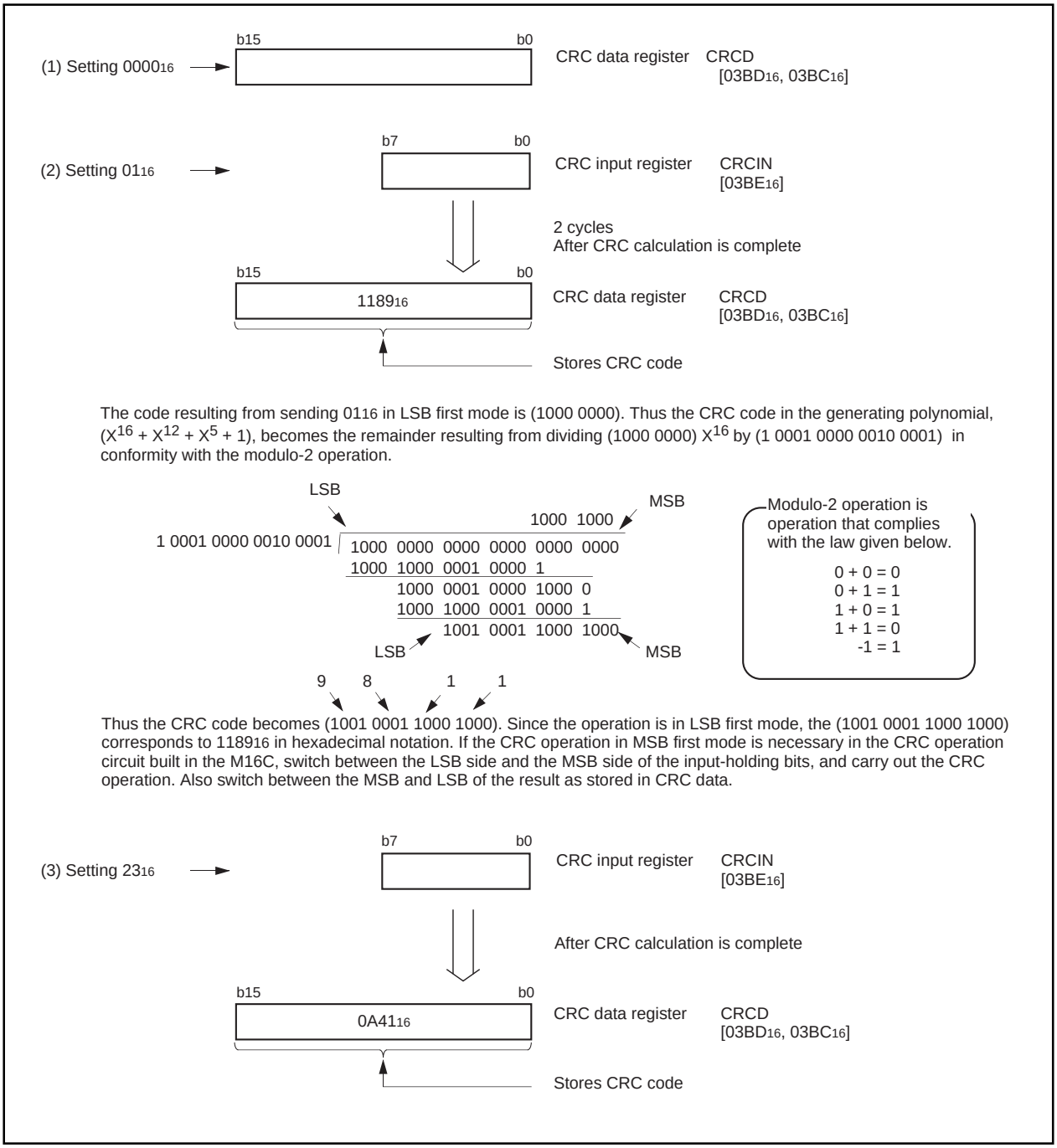


Figure UC-3. Calculation example using the CRC calculation circuit

Programmable I/O Ports

There are 48 programmable I/O ports: P3, P4 and P7 to P10. Each port can be set independently for input or output using the direction register. A pull-up resistance for each block of 4 ports can be set.

P3 and $P4_0$ to $P4_3$ are high-breakdown-voltage, P-channel open drain outputs, and have no built-in pull-down resistance (note).

Figures UA-1, UA-2 show the programmable I/O ports.

Each pin functions as a programmable I/O port and as the I/O for the built-in peripheral devices.

To use the pins as the inputs for the built-in peripheral devices, set the direction register of each pin to input mode. When the pins are used as the outputs for the built-in peripheral devices (other than the D-A converter), they function as outputs regardless of the contents of the direction registers. When pins are to be used as the outputs for the D-A converter, do not set the direction registers to output mode. See the descriptions of the respective functions for how to set up the built-in peripheral devices.

Note: These ports can be selected whether pull-down resistors are built-in or not by the option specify.

(1) Direction registers

Figure UA-3 shows the direction registers.

These registers are used to choose the direction of the programmable I/O ports. Each bit in these registers corresponds one for one to each I/O pin.

(2) Port registers

Figure UA-4 shows the port registers.

These registers are used to write and read data for input and output to and from an external device. A port register consists of a port latch to hold output data and a circuit to read the status of a pin. Each bit in port registers corresponds one for one to each I/O pin.

(3) Pull-up control registers

Figure UA-5 shows the pull-up control registers.

The pull-up control register can be set to apply a pull-up resistance to each block of 4 ports. When ports are set to have a pull-up resistance, the pull-up resistance is connected only when the direction register is set for input.

Note: P3, $P4_0$ to $P4_3$ have no built-in pull-up resistance, because of these pin's are high-breakdown-voltage, P-channel open drain outputs.

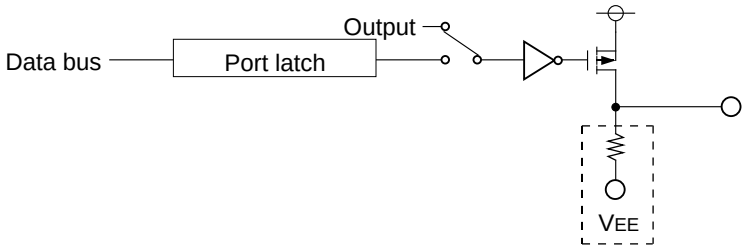
Exclusive High-breakdown-voltage Output Ports

There are 40 exclusive output Ports: P0 to P2, P5 and P6.

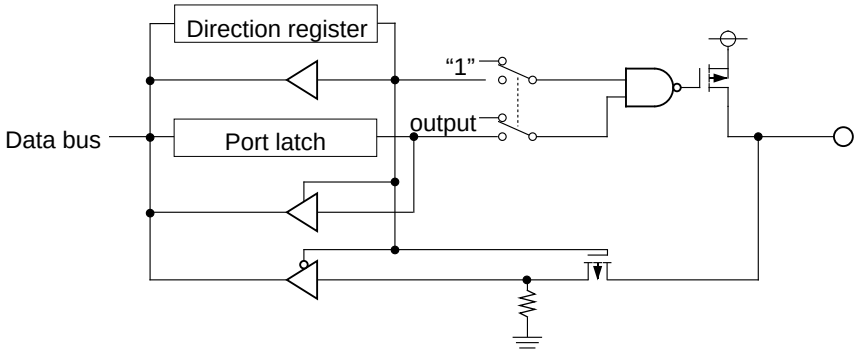
All ports have structure of high-breakdown-voltage P-channel open drain output. Exclusive output ports except P2 have built-in pull-down resistance.

Figure UA-1 shows the configuration of the exclusive high-breakdown-voltage output ports.

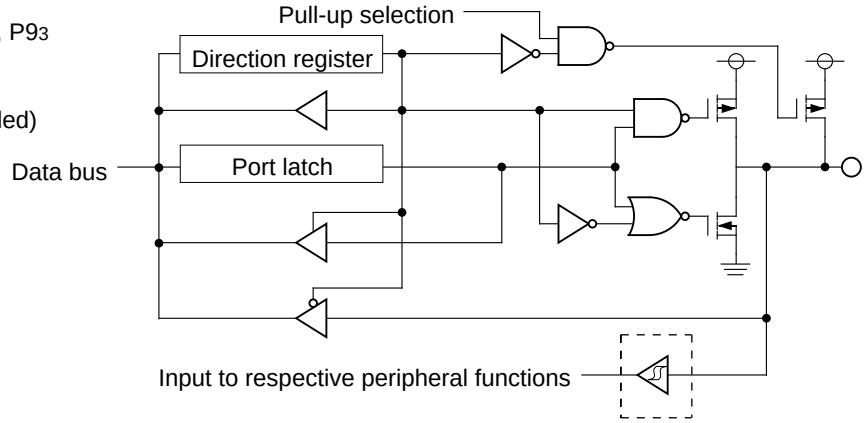
P0$_0$ to P0$_7$, P1$_0$ to P1$_7$,
P5$_0$ to P5$_7$, P6$_0$ to P6$_7$,
(inside dotted-line included)
P2$_0$ to P2$_7$
(inside dotted-line not included)



P3$_0$ to P3$_7$, P4$_0$ to P4$_3$



P7$_0$ to P7$_2$, P8$_0$ to P8$_5$, P8$_7$, P9$_3$
(inside dotted-line included)
P8$_6$
(inside dotted-line not included)



P4$_4$, P9$_2$,P9$_4$

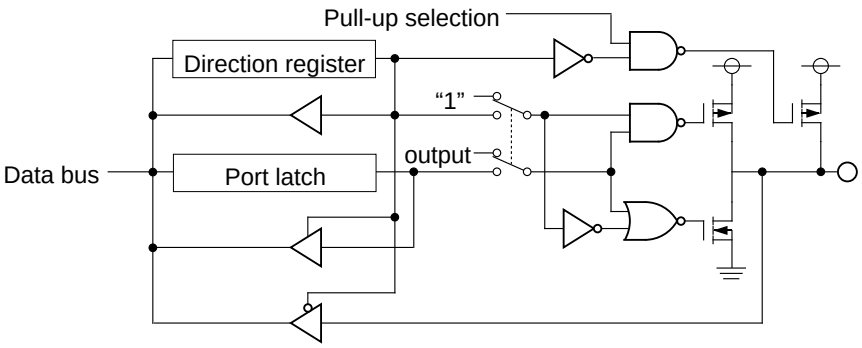


Figure UA-1. Programmable I/O ports (1)

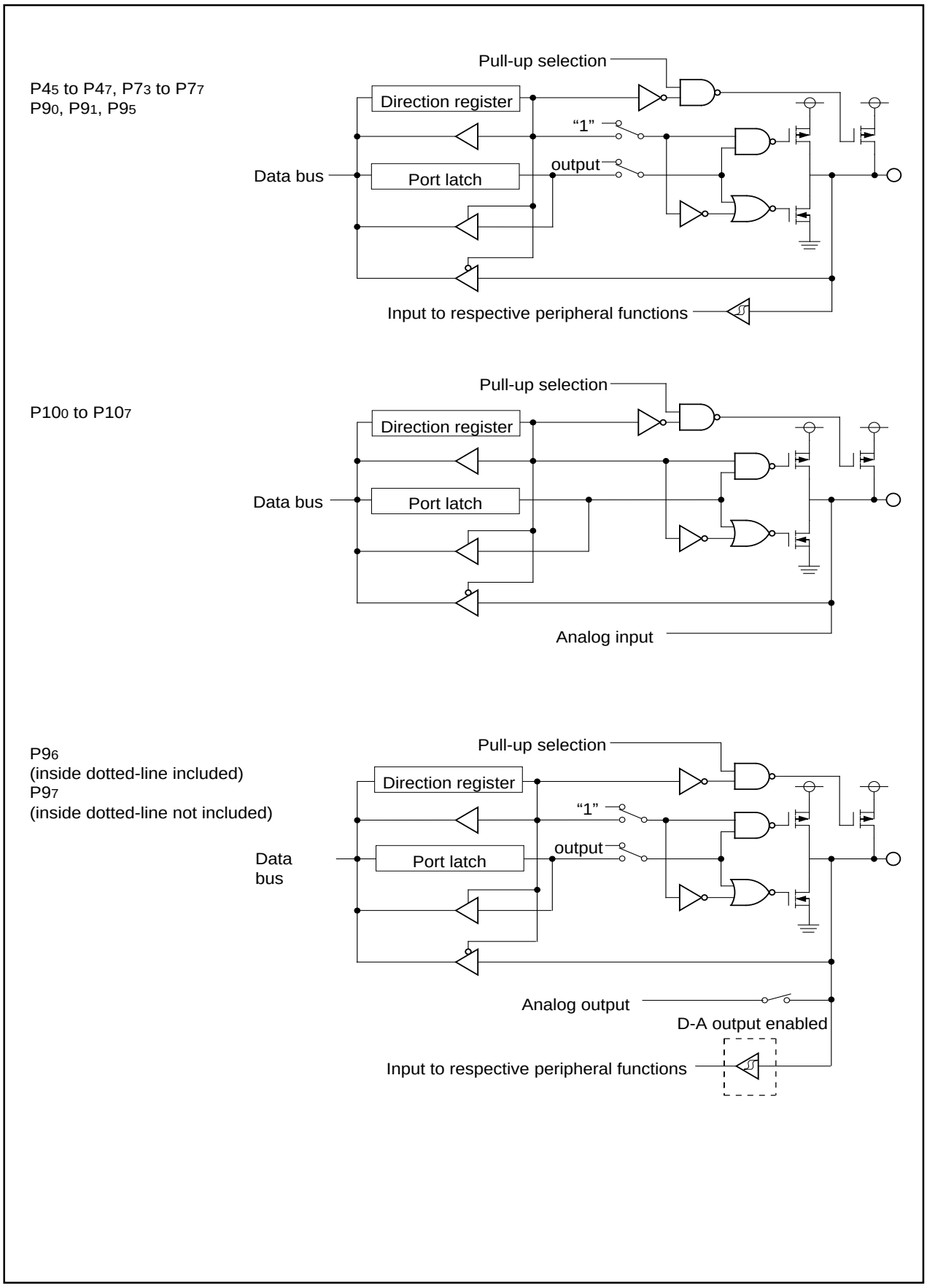


Figure UA-2. Programmable I/O ports (2)

Port Pi direction register

Symbol	Address	When reset
PDi (i = 3 to 10, except 5, 6)	$03E7_{16}$, $03EA_{16}$, $03EF_{16}$ $03F2_{16}$, $03F3_{16}$, $03F6_{16}$	00_{16} 00_{16}

Bit symbol	Bit name	Function	R	W
PDi_0	Port Pi_0 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (i = 3 to 10 except 5, 6)	○	○
PDi_1	Port Pi_1 direction register		○	○
PDi_2	Port Pi_2 direction register		○	○
PDi_3	Port Pi_3 direction register		○	○
PDi_4	Port Pi_4 direction register		○	○
PDi_5	Port Pi_5 direction register		○	○
PDi_6	Port Pi_6 direction register		○	○
PDi_7	Port Pi_7 direction register		○	○

Figure UA-3. Direction register

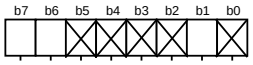
Port Pi register

Symbol	Addres	When reset
Pi (i = 0 to 10)	$03E0_{16}$, $03E1_{16}$, $03E4_{16}$, $03E5_{16}$, $03E8_{16}$	Indeterminate
	$03E9_{16}$, $03EC_{16}$, $03ED_{16}$, $03F0_{16}$, $03F1_{16}$, $03F4_{16}$	Indeterminate

Bit symbol	Bit name	Function	R	W
Pi_0	Port Pi_0 register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit 0 : "L" level data 1 : "H" level data (i = 0 to 10)	○	○
Pi_1	Port Pi_1 register		○	○
Pi_2	Port Pi_2 register		○	○
Pi_3	Port Pi_3 register		○	○
Pi_4	Port Pi_4 register		○	○
Pi_5	Port Pi_5 register		○	○
Pi_6	Port Pi_6 register		○	○
Pi_7	Port Pi_7 register		○	○

Figure UA-4. Port register

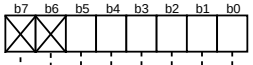
Pull-up control register 0



Symbol	Address	When reset
PUR0	03FD16	0016

Bit symbol	Bit name	Function	R	W
Nothing is assigned. In an attempt to write to this bit, write “0”. The value, if read, turns out to be indeterminate.			—	—
PU01	P44 to P47 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—
PU06	P70 to P73 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU07	P74 to P77 pull-up		○	○

Pull-up control register 1



Symbol	Address	When reset
PUR1	03FE16	0016

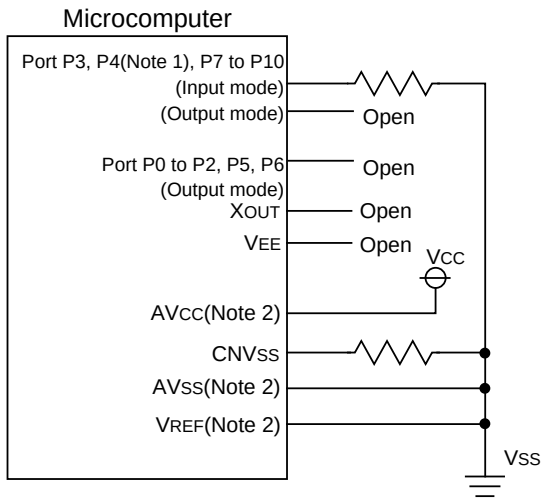
Bit symbol	Bit name	Function	R	W
PU10	P80 to P83 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU11	P84 to P87 pull-up		○	○
PU12	P90 to P93 pull-up		○	○
PU13	P94 to P97 pull-up		○	○
PU14	P100 to P103 pull-up		○	○
PU15	P104 to P107 pull-up		○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be indeterminate.			—	—
			—	—

Figure UA-5. Pull-up control register

Table UA-1. Example connection of unused pins

Pin name	Connection
Ports P3, P4(Note 2), P7 to P10	Specify output mode, and leave these pins open; or specify input mode, and connect to VSS via resistor (pull-down)
Ports P0 to P2, P5, P6	Leave these pins open
XOUT (Note 1), VEE	Open
AVCC	Connect to VCC (Note 3)
AVSS, VREF	Connect to VSS (Note 3)
CNVSS	Connect to VSS via resistor

Note 1: With external clock input to XIN pin.
Note 2: In case of pull-down option is specified, leave the specified ports open.
(Pull-down resistors are built-in the specified port)
Note 3: Connect a bypass capacitor.



Note 1: In case of pull-down option is specified, leave the specified ports (port P3, P4) open.
(Pull-down resistors are built-in the specified port)
Note 2: Connect a bypass capacitor.

Figure UA-6. Example connection of unused pins

Pull-down

MASK OPTION OF PULL-DOWN RESISTOR (object product: mask ROM version)

Whether built-in pull-down resistors are connected or not to high-breakdown voltage ports $P2_0$ to $P2_7$, $P3_0$ to $P3_7$,and $P4_0$ to $P4_3$ can be specified in ordering mask ROM. The option type can be specified from among 7 types; A to G.

	$P2_0$	$P2_1$	$P2_2$	$P2_3$	$P2_4$	$P2_5$	$P2_6$	$P2_7$	$P3_0$	$P3_1$	$P3_2$	$P3_3$	$P3_4$	$P3_5$	$P3_6$	$P3_7$	$P4_0$	$P4_1$	$P4_2$	$P4_3$
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
D	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
E	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
F	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
G	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note 1: The electrical characteristics of high-breakdown voltage ports $P2_0$ to $P2_7$, $P3_0$ to $P3_7$,and $P4_0$ to $P4_3$'s built-in pull-down resistors are the same as that of high-breakdown voltage ports $P0_0$ to $P0_7$.

Note 2: The absolute maximum ratings of power dissipation may be exceed owing to the number of built-in pull-down resistor. After calculating the power dissipation, specify the option type.

Note 3: The option types B to G cannot be specified because these types are currently under development.

Power Dissipation Calculating Method

(Fixed number depending on microcomputer's standard)

- V_{OH} output fall voltage of high-breakdown port
  2 V (max.); | Current value | = at 18 mA
- Resistor value = 68 kΩ (min.)
- Power dissipation of internal circuit (CPU, ROM, RAM etc.) = 5 V X 38 mA = 190 mW

(Fixed number depending on use condition)

- Apply voltage to V_{EE} pin: Vcc – 50 V
- Timing number a; digit number b; segment number c
- Ratio of Toff time corresponding Tdisp time: 1/16
- Turn ON segment number during repeat cycle: d
- All segment number during repeat cycle: e (= a X c)
- Total number of built-in resistor: for digit; f, for segment; g
- Digit pin current value h (mA)
- Segment pin current value i (mA)

(1) Digit pin power dissipation
{h X b X (1–Toff / Tdisp) X voltage} / a

(2) Segment pin power dissipation
{i X d X (1–Toff / Tdisp) X voltage} / a

(3) Pull-down resistor power dissipation (digit)
{power dissipation per 1 digit X (b X f / b) X (1–Toff / Tdisp) } / a

(4) Pull-down resistor power dissipation (segment)
{power dissipation per 1 segment X (d X g / c) X (1–Toff / Tdisp) } / a

(5) Internal circuit power dissipation (CPU, ROM, RAM etc.) = 190 mW

(1) + (2)+ (3) + (4) + (5) = X mW

Power Dissipation Calculating example 1

Fixed number depending on microcomputer's standard

- V_{OH} output fall voltage of high-breakdown port
  2 V (max.); | Current value | = at 18 mA
- Resistor value 68 kΩ (min.)
- Power dissipation of internal circuit (CPU, ROM, RAM etc.) = 5 V X 38 mA = 190 mW

Fixed number depending on use condition

- Apply voltage to V_{EE} pin: Vcc – 50 V
- Timing number 17; digit number 16; segment number 20
- Ratio of Toff time corresponding Tdisp time: 1/16
- Turn ON segment number during repeat cycle: 31
- All segment number during repeat cycle: 340 (= 17 X 20)
- Total number of built-in resistor: for digit; 16, for segment; 20
- Digit pin current value: 18 (mA)
- Segment pin current value: 3 (mA)

(1) Digit pin power dissipation
{18 X 16 X (1–1/16) X 2} / 17 = 31.77 mW
(2) Segment pin power dissipation
{3 X 31 X (1–1/16) X 2} / 17 = 10.26 mW
(3) Pull-down resistor power dissipation (digit)
$(50 – 2)^2$ /68 X (16 X 16/16) X (1 – 1/16) / 17 = 29.90 mW
(4) Pull-down resistor power dissipation (segment)
$(50 – 2)^2$ /68 X (31 X 20/20) X (1 – 1/16) / 17 = 57.93 mW
(5) Internal circuit power dissipation (CPU, ROM, RAM etc.) = 190.00 mW

(1) + (2)+ (3) + (4) + (5) = 319.86 mW

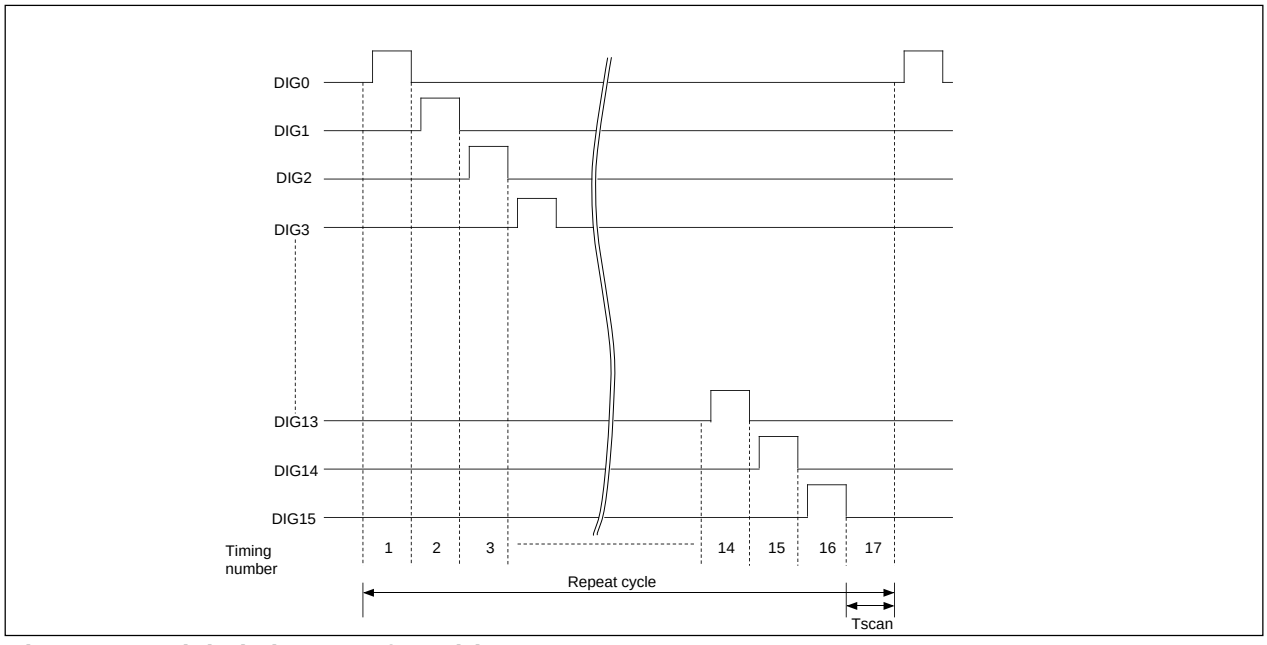


Figure S-1. Digit timing waveform (1)

Power Dissipation Calculating example 2(when 2 or more digit is turned ON at same time)

Fixed number depending on microcomputer's standard

- V_{OH} output fall voltage of high-breakdown port
  2 V (max.); | Current value | = at 18 mA
- Resistor value 68 kΩ (min.)
- Power dissipation of internal circuit (CPU, ROM, RAM etc.) = 5 V X 38 mA = 190 mW

Fixed number depending on use condition

- Apply voltage to V_{EE} pin: Vcc – 50 V
- Timing number 11; digit number 12; segment number 24
- Ratio of Toff time corresponding Tdisp time: 1/16
- Turn ON segment number during repeat cycle: 114
- All segment number during repeat cycle: 264 (= 11 X 24)
- Total number of built-in resistor: for digit; 10, for segment; 22
- Digit pin current value: 18 (mA)
- Segment pin current value: 3 (mA)

(1) Digit pin power dissipation
{18 X 12 X (1–1 / 16) X 2} / 11 = 36.82 mW
(2) Segment pin power dissipation
{3 X 114 X (1–1 / 16) X 2} / 11 = 58.30 mW
(3) Pull-down resistor power dissipation (digit)
$(50– 2)^2$ / 68 X (12 X 10 / 12) X (1 – 1 / 16) / 11 = 28.88 mW
(4) Pull-down resistor power dissipation (segment)
$(50 – 2)^2$ / 68 X (114 X 22 / 24) X (1 – 1 / 16) / 11 = 301.77 mW
(5) Internal circuit power dissipation (CPU, ROM, RAM etc.) = 190.00 mW

(1) + (2)+ (3) + (4) + (5) = 615.77 mW (There is a limit of use temperature)

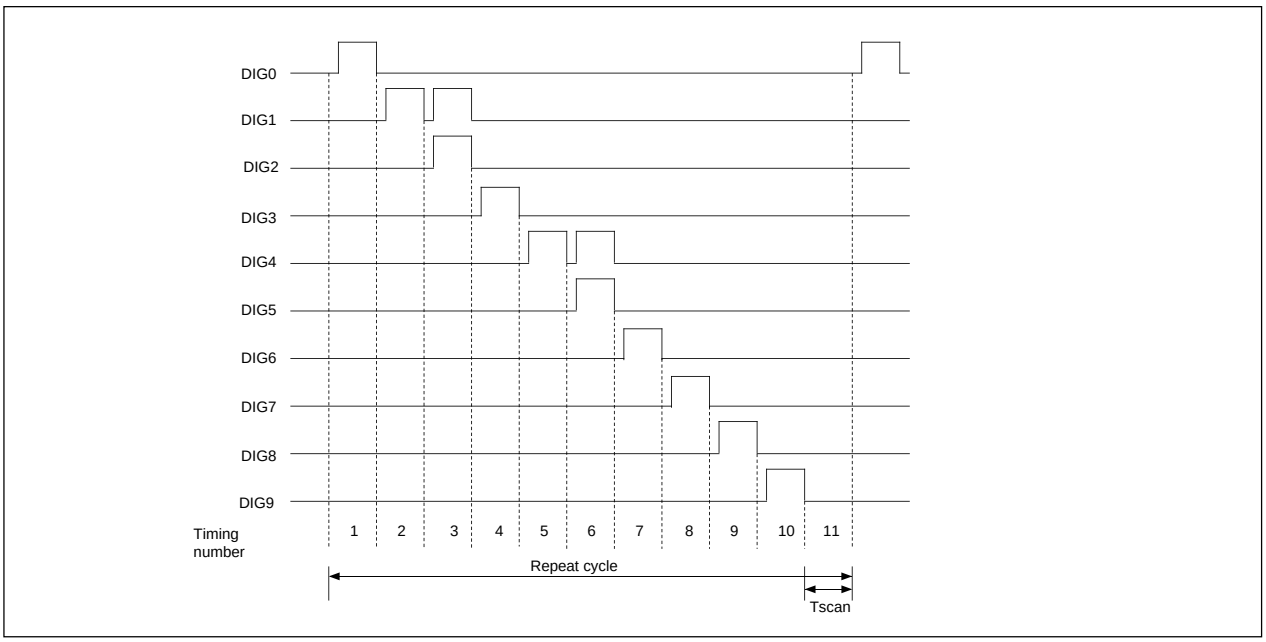


Figure S-2. Digit timing waveform (2)

Power Dissipation Calculating example 3
(when 2 or more digit is turned ON at same time, and used Toff invalid function)

Fixed number depending on microcomputer's standard

- V_{OH} output fall voltage of high-breakdown port 2 V (max.); | Current value | = at 18 mA
- Resistor value 68 kΩ (min.)
- Power dissipation of internal circuit (CPU, ROM, RAM etc.) = 5 V X 38 mA = 190 mW

Fixed number depending on use condition

- Apply voltage to V_{EE} pin: Vcc – 50 V
- Timing number 11; digit number 12; segment number 24
- Ratio of Toff time corresponding Tdisp time: 1/16
- Turn ON segment number during repeat cycle: 114 (for Toff invalid waveform;50)
- All segment number during repeat cycle: 264 (= 11 X 24)
- Total number of built-in resistor: for digit; 10, for segment; 22
- Digit pin current value: 18 (mA)
- Segment pin current value: 3 (mA)

(1) Digit pin power dissipation

[{18 X 10 X (1–1/16) X 2} + {18 X 2 X 2}] / 11 = 37.23 mW

(2) Segment pin power dissipation

[{3 X 64 X (1–1/16) X 2} + {3 X 50 X 2}] / 11 = 60.00 mW

(3) Pull-down resistor power dissipation (digit)

$[\{(50-2)^2 / 68 \text{ X } (10 \text{ X } 10 / 12) \text{ X } (1 - 1 / 16)\} + \{(50-2)^2 / 68 \text{ X } (2 \text{ X } 10 / 12)\ \}\ ] / 11 = 29.20$ mW

(4) Pull-down resistor power dissipation (segment)

$[\{(50-2)^2 / 68 \text{ X } (64 \text{ X } 22 / 24) \text{ X } (1 - 1 / 16)\} + \{(50-2)^2 / 68 \text{ X } (50 \text{ X } 22 / 24)\ \}\ ] / 11 = 310.59$ mW

(5) Internal circuit power dissipation (CPU, ROM, RAM etc.) = 190.00 mW

(1) + (2)+ (3) + (4) + (5) = 627.02 mW (There is a limit of use temperature)

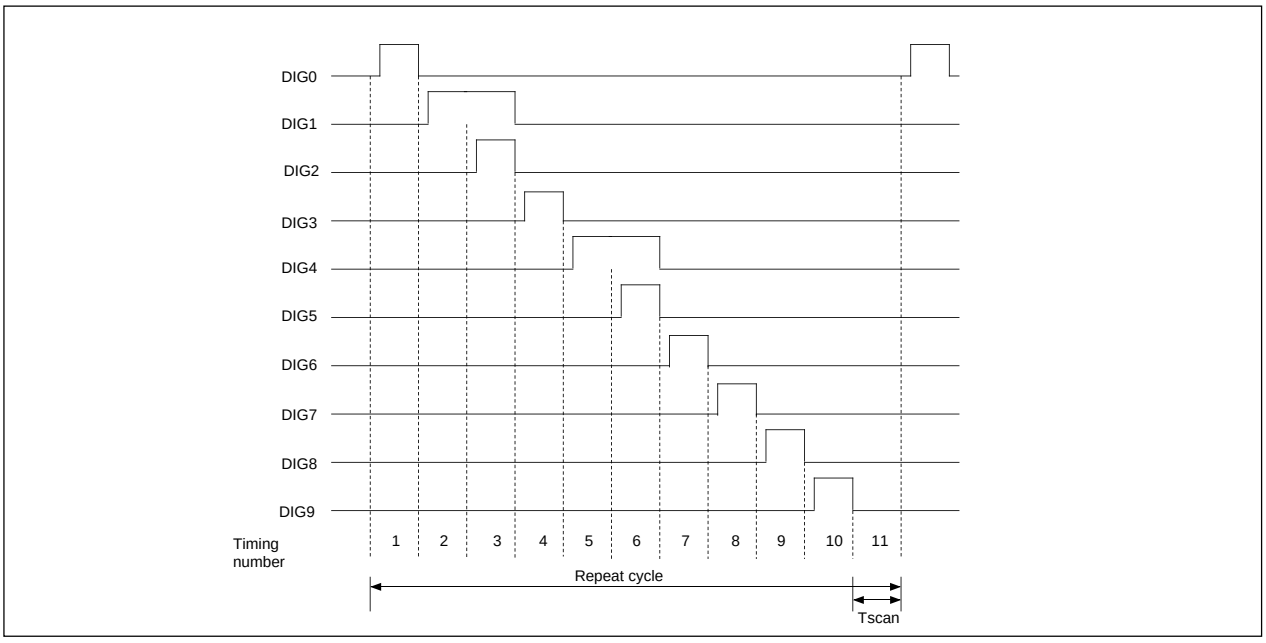


Figure S-3. Digit timing waveform (3)

Table Z-1. Absolute maximum ratings

Symbol	Parameter	Condition	Standard	Unit
Vcc	Supply voltage		- 0.3 to 6.5	V
AVcc	Analog supply voltage		- 0.3 to 6.5	V
V_{EE}	Pull-down supply voltage		Vcc - 50 to Vcc+0.3V	V
V_I	Input voltage $\overline{RESET}$, CNVss, $P4_4$ to $P4_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$, V_{REF}, X_{IN}		- 0.3 to Vcc+0.3 (Note)	V
V_I	Input voltage $P3_0$ to $P3_7$, $P4_0$ to $P4_3$		Vcc - 50 to Vcc+0.3	V
V_O	Output voltage $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_3$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$		Vcc - 50 to Vcc+0.3	V
V_O	Output voltage $P4_4$ to $P4_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$, X_{OUT}		-0.3 to Vcc+0.3	V
P_d	Power dissipation	Ta=-20 to 60 °C	750	mW
		Ta=60 to 85 °C	750-12 X (Ta-60)	mW
T_{opr}	Operating ambient temperature		-20 to 85	°C
T_{stg}	Storage temperature		-40 to 150	°C

Note 1: When writing to flash ,only CNVss is –0.3 to 13 (V) .

Table Z-2. Recommended operating conditions (referenced to V_{CC} = 2.7V to 5.5V at Ta = – 20 to 85°C unless otherwise specified) (Note)

Symbol	Parameter		Standard			Unit
			Min	Typ.	Max.	
Vcc	Supply voltage		2.7(Note1)	5.0	5.5	V
AVcc	Analog supply voltage			Vcc		V
Vss	Supply voltage			0		V
AVss	Analog supply voltage			0		V
V_{EE}	Pull-down supply voltage		Vcc-48		Vcc	V
V_{IH}	HIGH input voltage	$P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}	0.8Vcc		Vcc	V
V_{IH}	HIGH input voltage	$P4_4$ to $P4_7$	0.50Vcc		Vcc	V
V_{IH}	HIGH input voltage	$P3_0$ to $P3_7$, $P4_0$ to $P4_3$	0.52Vcc		Vcc	V
V_{IL}	LOW input voltage	$P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}	0		0.2Vcc	V
V_{IL}	LOW input voltage	$P3_0$ to $P3_7$, $P4_0$ to $P4_3$	0		0.16Vcc	V
V_{IL}	LOW input voltage	$P4_4$ to $P4_7$	0		0.16Vcc	V

Note: V_{CC} = 4.0V to 5.5V in flash memory version.

Table Z-3. Recommended operating conditions (referenced to V_{CC} = 2.7V to 5.5V at Ta = – 20 to 85°C unless otherwise specified) (Note 6)

Symbol	Parameter		Standard Min	Standard Typ.	Standard Max.	Unit
$I_{OH (peak)}$	HIGH total peak output current (Note 1)	$P0_0$ to $P0_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$			-240	mA
$I_{OH (peak)}$	HIGH total peak output current (Note 1)	$P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_3$			-240	mA
$I_{OH (peak)}$	HIGH total peak output current (Note 1)	$P4_4$ to $P4_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_5$			-80	mA
$I_{OH (peak)}$	HIGH total peak output current (Note 1)	$P8_6$, $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$			-80	mA
$I_{OL (peak)}$	LOW total peak output current (Note 1)	$P4_4$ to $P4_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_5$			80	mA
$I_{OL (peak)}$	LOW total peak output current (Note 1)	$P8_6$, $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$			80	mA
$I_{OH (avg)}$	HIGH total average output current (Note 1)	$P0_0$ to $P0_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$			-120	mA
$I_{OH (avg)}$	HIGH total average output current (Note 1)	$P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_3$			-120	mA
$I_{OH (avg)}$	HIGH total average output current (Note 1)	$P4_4$ to $P4_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_5$			-40	mA
$I_{OH (avg)}$	HIGH total average output current (Note 1)	$P8_6$, $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$			-40	mA
$I_{OL (avg)}$	LOW total average output current (Note 1)	$P4_4$ to $P4_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_5$			40	mA
$I_{OL (avg)}$	LOW total average output current (Note 1)	$P8_6$, $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$			40	mA
$I_{OH (peak)}$	HIGH peak output current (Note 2)	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$,$P3_0$ to $P3_7$, $P4_0$ to $P4_3$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$			-40	mA
$I_{OH (peak)}$	HIGH peak output current (Note 2)	$P4_4$ to $P4_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$ $P9_0$ to $P9_7$, $P10_0$ to $P10_7$			-10	mA
$I_{OL (peak)}$	LOW peak output current (Note 2)	$P4_4$ to $P4_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$ $P9_0$ to $P9_7$, $P10_0$ to $P10_7$			10	mA
$I_{OH (avg)}$	HIGH average output current (Note 3)	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$,$P3_0$ to $P3_7$, $P4_0$ to $P4_3$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$			-18	mA
$I_{OH (avg)}$	HIGH average output current (Note 3)	$P4_4$ to $P4_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$ $P9_0$ to $P9_7$, $P10_0$ to $P10_7$			-5	mA
$I_{OL (avg)}$	LOW average output current (Note 3)	$P4_4$ to $P4_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$ $P9_0$ to $P9_7$, $P10_0$ to $P10_7$			5	mA
$f(X_{IN})$	Main clock input oscillation frequency (Note 4, 7)	Vcc=4.0V to 5.5V	0		10	MHz
		Vcc=2.7V to 4.0V	0		5 X Vcc-10	MHz
$f(X_{CIN})$	Sub clock oscillation frequency (Note 4, 5)			32.768	50	kHz

Note 1: The total output current is the sum of all the currents through the applicable ports. The total average value measured over 100ms. The total peak current is the peak of all the currents.

Note 2: The peak output current is the peak current flowing in each port.

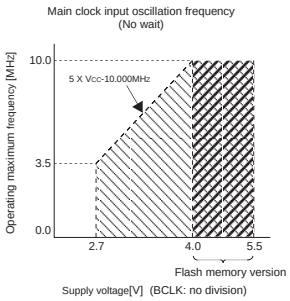
Note 3: The average output current in an average value measured over 100ms.

Note 4: When the oscillating frequency has a duty cycle of 50 %.

Note 5: When using the microcomputer in low-speed mode, set the sub-clock input oscillation frequency on condition that $f(X_{CIN}) < f(X_{IN}) / 3$.

Note 6: V_{CC}=4.0V to 5.5V in flash memory version.

Note 7: Relationship between main clock oscillation frequency and supply voltage.



V_{CC}=5V

Table Z-4. Electrical characteristics (referenced to V_{CC} = 5V, V_{SS} = 0V at Ta = 25°C, f(X_{IN}) =10MHz unless otherwise specified)

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
V_{OH}	HIGH output voltage	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P2_0$ to $P2_7$, $P3_0$ to $P3_7$,$P4_0$ to $P4_3$,$P5_0$ to $P5_7$, $P6_0$ to $P6_7$	I_{OH}= - 18mA	3.5			V
			I_{OH}= - 5mA	4.5			
V_{OH}	HIGH output voltage	$P4_4$ to $P4_7$,$P7_0$ to $P7_7$,$P8_0$ to $P8_7$, $P9_0$ to $P9_7$,$P10_0$ to $P10_7$	I_{OH}= - 5mA	3.0			V
V_{OH}	HIGH output voltage	X_{OUT} HIGH POWER	I_{OH}= - 1mA	3.0			V
		X_{OUT} LOW POWER	I_{OH}= - 0.5mA	3.0			
V_{OL}	LOW output voltage	$P4_4$ to $P4_7$,$P7_0$ to $P7_7$,$P8_0$ to $P8_7$, $P9_0$ to $P9_7$,$P10_0$ to $P10_7$	I_{OL}=5mA			2.0	V
V_{OL}	LOW output voltage	X_{OUT} HIGH POWER	I_{OL}=1mA			2.0	V
		X_{OUT} LOW POWER	I_{OL}=0.5mA			2.0	
V_{T+}-V_{T-}	Hysteresis	$TA0_{IN}$ to $TA4_{IN}$,$TB0_{IN}$ to $TB2_{IN}$, $\overline{INT0}$ to $\overline{INT5}$, $\overline{CTS0}$, $\overline{CTS1}$, $CLK0$,$CLK1$,$SRDY2_{IN}$,$SBSY2_{IN}$, SIN2,SCLK21,SCLK22,RxD0, RxD1		0.2		0.8	V
V_{T+}-V_{T-}	Hysteresis	$\overline{RESET}$		0.2		1.8	V
I_{IH}	HIGH input current	$P4_4$ to $P4_7$,$P7_0$ to $P7_7$,$P8_0$ to $P8_7$, $P9_0$ to $P9_7$,$P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNVss	V_I=5V			5.0	µA
I_{IH}		$P3_0$ to $P3_7$,$P4_0$ to $P4_3$(Note 1)	V_I=5V			5.0	µA
I_{IL}	LOW input current	$P4_4$ to $P4_7$,$P7_0$ to $P7_7$,$P8_0$ to $P8_7$, $P9_0$ to $P9_7$,$P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNVss	V_I=0V			- 5.0	µA
I_{IL}		$P3_0$ to $P3_7$,$P4_0$ to $P4_3$(Note1)	V_I=0V			- 5.0	µA
R_{PULLUP}	Pull-up resistance	$P4_4$ to $P4_7$,$P7_0$ to $P7_7$, $P8_0$ to $P8_7$,$P9_0$ to $P9_7$, $P10_0$ to $P10_7$	V_I=0V	30.0	50.0	167.0	kΩ
R_{PULLD}	Pull-down resistance	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$, $P5_0$ to $P5_7$,$P6_0$ to $P6_7$ ($P2_0$ to $P2_7$,$P3_0$ to $P3_7$, $P4_0$ to $P4_3$ in option specify)	V_{EE}=V_{CC} - 48V,V_{OL}=V_{CC} Output transistors “off”	68	80	120	kΩ
I_{LEAK}	Output leak current	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$, $P2_0$ to $P2_7$,$P3_0$ to $P3_7$, $P4_0$ to $P4_4$,$P5_0$ to $P5_7$, $P6_0$ to $P6_7$	V_{EE}=V_{CC} - 48V,V_{OL}=V_{CC} - 48V Output transistors “off”			- 10	µA
R_{fXIN}	Feedback resistance X_{IN}				1.0		MΩ
R_{fXCIN}	Feedback resistance X_{CIN}				6.0		MΩ
V_{RAM}	RAM retention voltage		When clock is stopped	2.0			V
Icc	Power supply current (Note 3)	The output pins are open and other pins are V_{SS}	f(X_{IN})=10MHz Square wave, no division		19.0	38.0	mA
			f(X_{IN})=10MHz Square wave, 8 division		4.2		mA
			f(X_{CIN})=32kHz Square wave (Note2)		90.0		µA
			f(X_{CIN})=32kHz When a WAIT instruction is executed (Note2)		4.0		µA
			Ta=25 °C when clock is stopped			1.0	µA
			Ta=85 °C when clock is stopped			20.0	

Note 1: Except when reading ports P3, $P4_0$ to $P4_3$.

Note 2: Fixed X_{CIN}-X_{COUT} drive capacity select bit to “HIGH” and X_{IN} pin to “H” level.

Note 3: This contains an electric current to flow into AV_{CC} pin.

V_{CC}=5V

Table Z-5. A-D conversion characteristics (referenced to V_{CC} = AV_{CC} = V_{REF} = 5V, Vss = AV_{SS} = 0V at Ta = 25°C, f(X_{IN}) = 10MHz unless otherwise specified)

Symbol	Parameter		Measuring condition		Standard			Unit
					Min.	Typ.	Max.	
–	Resolution		V_{REF} = V_{CC}				10	Bits
–	Absolute accuracy	Sample & hold function not available	V_{REF} = V_{CC} = 5V				±3	LSB
		Sample & hold function available(10bit)	V_{REF} =V_{CC} = 5V	AN_0 to AN_7 input			±3	LSB
		Sample & hold function available(8bit)	V_{REF} = V_{CC} = 5V				±2	LSB
R_{LADDER}	Ladder resistance		V_{REF} = V_{CC}		10		40	kΩ
t_{CONV}	Conversion time (10bit)				3.3			µs
t_{CONV}	Conversion time (8bit)				2.8			µs
t_{SAMP}	Sampling time				0.3			µs
V_{REF}	Reference voltage				2		V_{CC}	V
V_{IA}	Analog input voltage				0		V_{REF}	V

Table Z-6. D-A conversion characteristics (referenced to V_{CC} = 5V, V_{SS} = AV_{SS} = 0V, V_{REF} = 5V at Ta = 25°C, f(X_{IN}) = 10MHz unless otherwise specified)

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
–	Resolution				8	Bits
–	Absolute accuracy				1.0	%
t_{su}	Setup time				3	µs
R_O	Output resistance		4	10	20	kΩ
I_{VREF}	Reference power supply input current	(Note)			1.5	mA

Note: This applies when using one D-A converter, with the D-A register for the unused D-A converter set to "00_{16}".

The A-D converter's ladder resistance is not included.

Also, when the Vref is unconnected at the A-D control register, I_{VREF} is sent.

V_{CC}=5V

Timing requirements (referenced to V_{CC} = 5V, V_{SS} = 0V at Ta = 25°C unless otherwise specified)

Table Z-7. External clock input

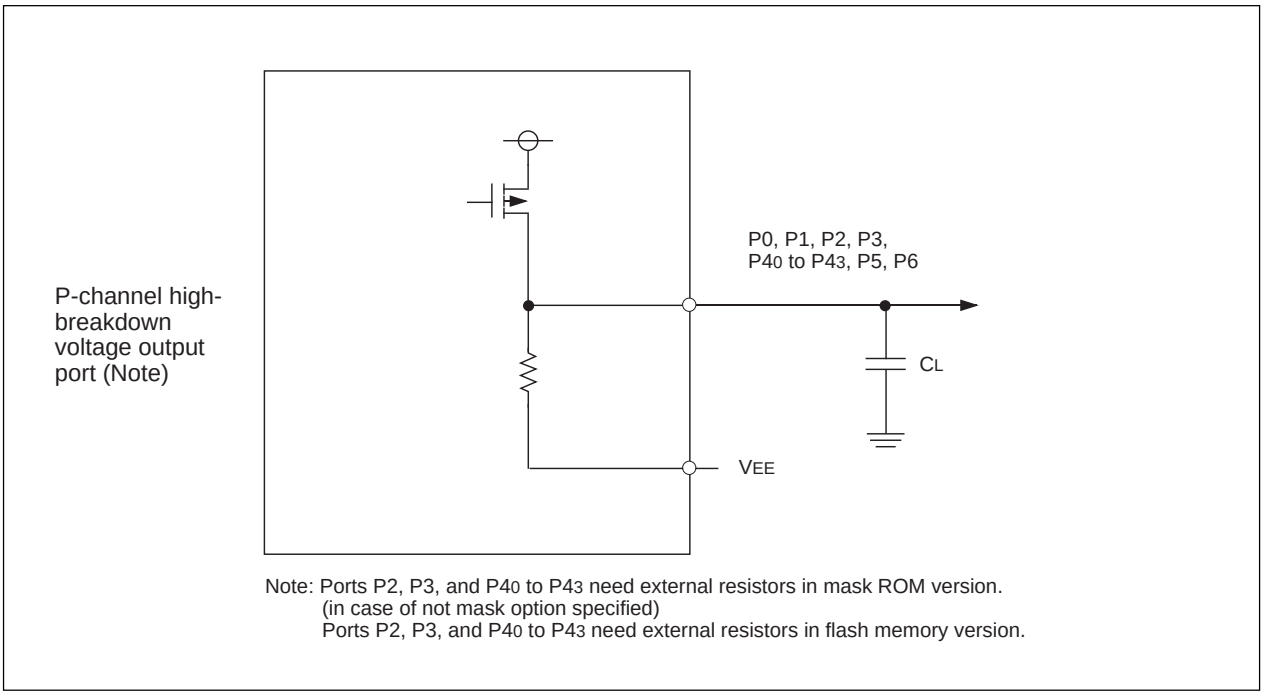
Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	100		ns
$t_{w(H)}$	External clock input HIGH pulse width	40		ns
$t_{w(L)}$	External clock input LOW pulse width	40		ns
t_r	External clock rise time		15	ns
t_f	External clock fall time		15	ns

Switching characteristics (referenced to V_{CC} = 5V, V_{SS} = 0V at Ta = 25°C unless otherwise specified)

Table Z-8. High-breakdown voltage p-channel open-drain output port

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
$t_{r(Pch\text{-}strg)}$	P-channel high-breakdown voltage output rising time (Note 1)	C_L=100pF V_{EE}=V_{CC} - 43V		55		ns
$t_{r(Pch\text{-}weak)}$	P-channel high-breakdown voltage output rising time (Note 2)	C_L=100pF V_{EE}=V_{CC} - 43V		1.8		µs

Note 1: When bit 7 of the FLDC mode register (address 0350_{16}) is at "0".
Note 2: When bit 7 of the FLDC mode register (address 0350_{16}) is at "1".



Note: Ports P2, P3, and $P4_0$ to $P4_3$ need external resistors in mask ROM version.
(in case of not mask option specified)
Ports P2, P3, and $P4_0$ to $P4_3$ need external resistors in flash memory version.

Figure Z-2. Circuit for measuring output switching characteristics

V_{CC}=5V

Timing requirements (referenced to V_{CC} = 5V, V_{SS} = 0V at Ta = 25°C unless otherwise specified)

Table Z-9. Timer A input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAi_{IN} input cycle time	100		ns
$t_{w(TAH)}$	TAi_{IN} input HIGH pulse width	40		ns
$t_{w(TAL)}$	TAi_{IN} input LOW pulse width	40		ns

Table Z-10. Timer A input (gating input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAi_{IN} input cycle time	400		ns
$t_{w(TAH)}$	TAi_{IN} input HIGH pulse width	200		ns
$t_{w(TAL)}$	TAi_{IN} input LOW pulse width	200		ns

Table Z-11. Timer A input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAi_{IN} input cycle time	200		ns
$t_{w(TAH)}$	TAi_{IN} input HIGH pulse width	100		ns
$t_{w(TAL)}$	TAi_{IN} input LOW pulse width	100		ns

Table Z-12. Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAi_{IN} input HIGH pulse width	100		ns
$t_{w(TAL)}$	TAi_{IN} input LOW pulse width	100		ns

Table Z-13. Timer A input (up/down input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAi_{OUT} input cycle time	2000		ns
$t_{w(UPH)}$	TAi_{OUT} input HIGH pulse width	1000		ns
$t_{w(UPL)}$	TAi_{OUT} input LOW pulse width	1000		ns
$t_{su(UP-TIN)}$	TAi_{OUT} input setup time	400		ns
$t_{h(TIN-UP)}$	TAi_{OUT} input hold time	400		ns

Timing (V_{CC}=5V)

V_{CC}=5V

Timing requirements (referenced to V_{CC} = 5V, V_{SS} = 0V at Ta = 25°C unless otherwise specified)

Table Z-14. Timer B input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time (counted on one edge)	100		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on one edge)	40		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on one edge)	40		ns
$t_{c(TB)}$	TBiIN input cycle time (counted on both edges)	200		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on both edges)	80		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on both edges)	80		ns

Table Z-15. Timer B input (pulse period measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	400		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	200		ns

Table Z-16. Timer B input (pulse width measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	400		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	200		ns

Table Z-17. Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	200		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	100		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	100		ns
$t_{d(C-Q)}$	TxDi output delay time		80	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	30		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

Table Z-18. External interrupt $\overline{INTi}$ inputs

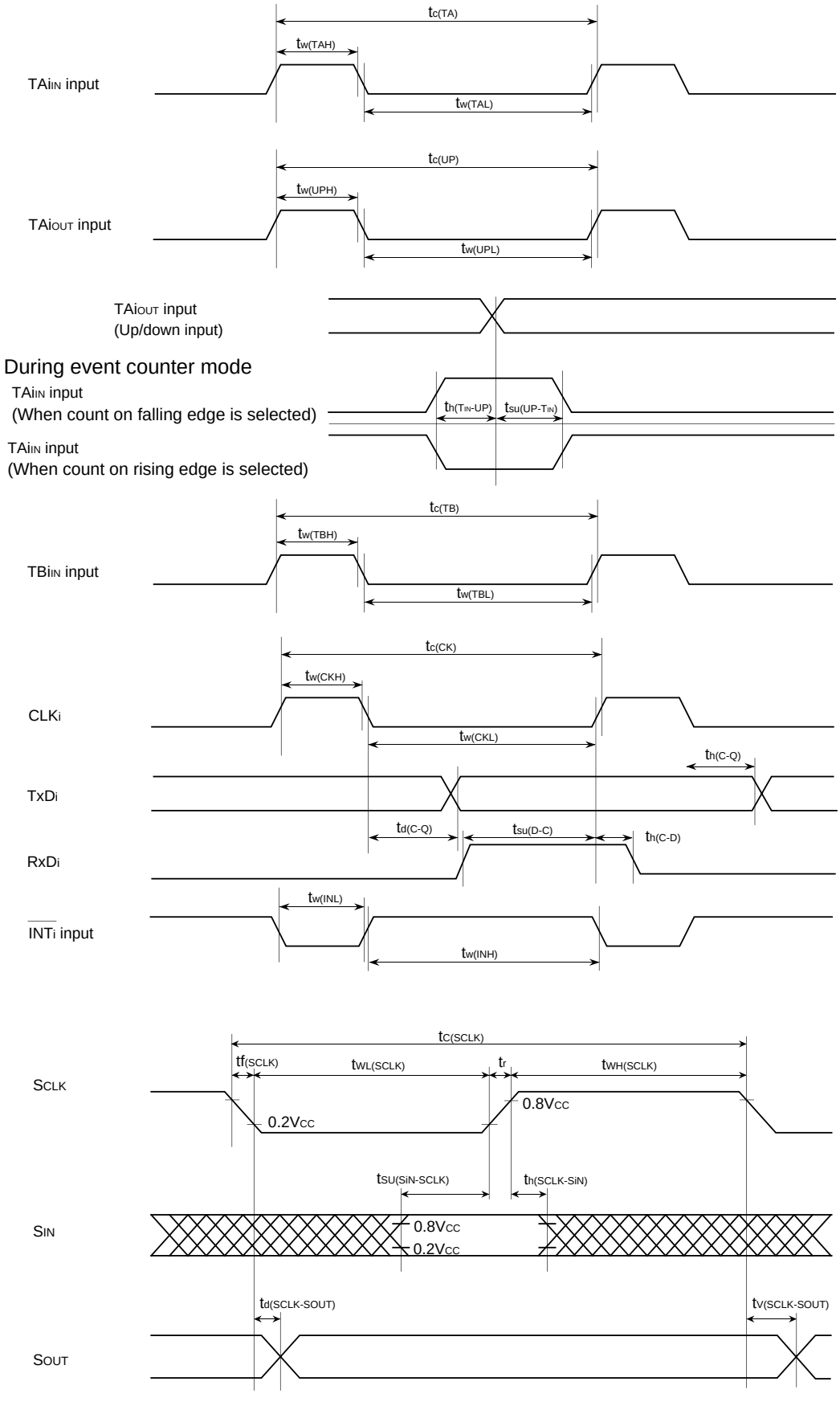
Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input HIGH pulse width	250		ns
$t_{w(INL)}$	$\overline{INTi}$ input LOW pulse width	250		ns

Table Z-19. Automatic transfer serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(SCLK)}$	Serial I/O clock input cycle time	0.95		µs
$t_{wH(SCLK)}$	Serial I/O clock input HIGH pulse width	400		ns
$t_{wL(SCLK)}$	Serial I/O clock input LOW pulse width	400		ns
$t_{su(SCLK-SIN)}$	Serial I/O input setup time	200		ns
$t_{h(SCLK-SIN)}$	Serial I/O input hold time	200		ns

Timing (V_{CC}=5V)

V_{CC}=5V



V_{CC}=3V

Table Z-20. Electrical characteristics (referenced to V_{CC} = 3V, V_{SS} = 0V at Ta = 25°C, f(X_{IN}) =5MHz unless otherwise specified)

Symbol	Parameter		Measuring condition	Standard Min.	Standard Typ.	Standard Max.	Unit
V_{OH}	HIGH output voltage $P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P2_0$ to $P2_7$, $P3_0$ to $P3_7$,$P4_0$ to $P4_3$,$P5_0$ to $P5_7$, $P6_0$ to $P6_7$		I_{OH}= - 18mA	1.5			V
			I_{OH}= - 5mA	2.5			
V_{OH}	HIGH output voltage $P4_4$ to $P4_7$,$P7_0$ to $P7_7$,$P8_0$ to $P8_7$, $P9_0$ to $P9_7$,$P10_0$ to $P10_7$		I_{OH}= - 1mA	2.5			V
V_{OH}	HIGH output voltage X_{OUT}	HIGH POWER	I_{OH}= - 0.1mA	2.5			V
		LOW POWER	I_{OH}= - 50µA	2.5			
V_{OL}	LOW output voltage $P4_4$ to $P4_7$,$P7_0$ to $P7_7$,$P8_0$ to $P8_7$, $P9_0$ to $P9_7$,$P10_0$ to $P10_7$		I_{OL}=1mA			0.5	V
V_{OL}	LOW output voltage X_{OUT}	HIGH POWER	I_{OL}=0.1mA			0.5	V
		LOW POWER	I_{OL}=50µA			0.5	
V_{T+}-V_{T-}	Hysteresis TA0$_{IN}$ to TA4$_{IN}$,TB0$_{IN}$ to TB2$_{IN}$, $\overline{INT0}$ to $\overline{INT5}$, $\overline{CTS0}$, $\overline{CTS1}$, CLK0,CLK1,SRDY2$_{IN}$,SBSY2$_{IN}$, SIN2,SCLK21,SCLK22 RTS0,RTS1			0.2		0.8	V
V_{T+}-V_{T-}	Hysteresis $\overline{RESET}$			0.2		1.8	V
I_{IH}	HIGH input current $P4_4$ to $P4_7$,$P7_0$ to $P7_7$,$P8_0$ to $P8_7$, $P9_0$ to $P9_7$,$P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNVss		V_I=3V			4.0	µA
I_{IH}	$P3_0$ to $P3_7$,$P4_0$ to $P4_3$(Note 1)		V_I=3V			4.0	µA
I_{IL}	LOW input current $P4_4$ to $P4_7$,$P7_0$ to $P7_7$,$P8_0$ to $P8_7$, $P9_0$ to $P9_7$,$P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNVss		V_I=0V			- 4.0	µA
I_{IL}	$P3_0$ to $P3_7$,$P4_0$ to $P4_3$(Note 1)		V_I=0V			- 4.0	µA
R_{PULLUP}	Pull-up resistance $P4_4$ to $P4_7$,$P7_0$ to $P7_7$, $P8_0$ to $P8_7$,$P9_0$ to $P9_7$, $P10_0$ to $P10_7$		V_I=0V	66.0	120.0	500.0	kΩ
R_{PULLD}	Pull-down resistance $P0_0$ to $P0_7$,$P1_0$ to $P1_7$, $P5_0$ to $P5_7$,$P6_0$ to $P6_7$ ($P2_0$ to $P2_7$,$P3_0$ to $P3_7$, $P4_0$ to $P4_3$ in option specify)		V_{EE}=V_{CC} - 48V,V_{OL}=V_{CC} Output transistors “off”	68	80	120	kΩ
I_{LEAK}	Output leak current $P0_0$ to $P0_7$,$P1_0$ to $P1_7$, $P2_0$ to $P2_7$,$P3_0$ to $P3_7$, $P4_0$ to $P4_4$,$P5_0$ to $P5_7$, $P6_0$ to $P6_7$		V_{EE}=V_{CC} - 48V,V_{OL}=V_{CC} - 48V Output transistors “off”			- 10	µA
R_{fXIN}	Feedback resistance X_{IN}				3.0		MΩ
R_{fXCIN}	Feedback resistance X_{CIN}				10.0		MΩ
V_{RAM}	RAM retention voltage		When clock is stopped	2.0			V
Icc	Power supply current (Note 3)	The output pins are open and other pins are V_{SS}	f(X_{IN})=5MHz Square wave, no division		6.0	15.0	mA
			f(X_{IN})=5MHz Square wave, 8 division		1.6		mA
			f(X_{CIN})=32kHz Square wave		50.0		µA
			f(X_{CIN})=32kHz When a WAITinstruction is executed. Oscillation capacity High (Note2)		2.8		µA
			f(X_{CIN})=32kHz When a WAIT instruction is executed. Oscillation capacity Low (Note2)		0.9		µA
			Ta=25 °C when clock is stopped			1.0	µA
			Ta=85 °C when clock is stopped			20.0	

Note 1: Except when reading ports P3, $P4_0$ to $P4_3$.

Note 2: With one timer operated using f_{C32}.

Note 3: This contains an electric current to flow into AV_{CC} pin.

Electrical characteristics(V_{CC}=3V, only mask ROM version)

V_{CC}=3V

Table Z-21. A-D conversion characteristics (referenced to V_{CC} = AV_{CC} = V_{REF} = 3V, Vss = AV_{SS} = 0V at Ta = 25°C, f(X_{IN}) = 5MHz unless otherwise specified)

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max	
–	Resolution		V_{REF} = V_{CC}			10	Bits
–	Absolute accuracy	Sample & hold function not available (8 bit)	VREF = V_{CC} = 3V, ϕ_{AD} = f(X_{IN})/2			±2	LSB
R_{LADDER}	Ladder resistance		V_{REF} = V_{CC}	10		40	kΩ
t_{CONV}	Conversion time (8bit)			14.0			µs
V_{REF}	Reference voltage			2.7		V_{CC}	V
V_{IA}	Analog input voltage			0		V_{REF}	V

Table Z-22. D-A conversion characteristics (referenced to V_{CC} = 3V, V_{SS} = AV_{SS} = 0V, V_{REF} = 3V at Ta = 25°C, f(X_{IN}) = 5MHz unless otherwise specified)

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max	
–	Resolution				8	Bits
–	Absolute accuracy				1.0	%
t_{su}	Setup time				3	µs
R_O	Output resistance		4	10	20	kΩ
I_{VREF}	Reference power supply input current	(Note)			1.0	mA

Note: This applies when using one D-A converter, with the D-A register for the unused D-A converter set to "00$_{16}$".
The A-D converter's ladder resistance is not included.
Also, when the Vref is unconnected at the A-D control register, I_{VREF} is sent.

V_{CC}=3V

Timing requirements (referenced to V_{CC} = 3V, V_{SS} = 0V at Ta = 25°C unless otherwise specified)

Table Z-23. External clock input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	200		ns
$t_{w(H)}$	External clock input HIGH pulse width	85		ns
$t_{w(L)}$	External clock input LOW pulse width	85		ns
t_r	External clock rise time		18	ns
t_f	External clock fall time		18	ns

V_{CC}=3V

Timing requirements (referenced to V_{CC} = 3V, V_{SS} = 0V at Ta = 25°C unless otherwise specified)

Table Z-24. Timer A input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAi$_{IN}$ input cycle time	150		ns
$t_{w(TAH)}$	TAi$_{IN}$ input HIGH pulse width	60		ns
$t_{w(TAL)}$	TAi$_{IN}$ input LOW pulse width	60		ns

Table Z-25. Timer A input (gating input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAi$_{IN}$ input cycle time	600		ns
$t_{w(TAH)}$	TAi$_{IN}$ input HIGH pulse width	300		ns
$t_{w(TAL)}$	TAi$_{IN}$ input LOW pulse width	300		ns

Table Z-26. Timer A input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAi$_{IN}$ input cycle time	300		ns
$t_{w(TAH)}$	TAi$_{IN}$ input HIGH pulse width	150		ns
$t_{w(TAL)}$	TAi$_{IN}$ input LOW pulse width	150		ns

Table Z-27. Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAi$_{IN}$ input HIGH pulse width	150		ns
$t_{w(TAL)}$	TAi$_{IN}$ input LOW pulse width	150		ns

Table Z-28. Timer A input (up/down input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAi$_{OUT}$ input cycle time	3000		ns
$t_{w(UPH)}$	TAi$_{OUT}$ input HIGH pulse width	1500		ns
$t_{w(UPL)}$	TAi$_{OUT}$ input LOW pulse width	1500		ns
$t_{su(UP-TIN)}$	TAi$_{OUT}$ input setup time	600		ns
$t_{h(TIN-UP)}$	TAi$_{OUT}$ input hold time	600		ns

V_{CC}=3V

Timing requirements (referenced to V_{CC} = 3V, V_{SS} = 0V at Ta = 25°C unless otherwise specified)

Table Z-29. Timer B input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time (counted on one edge)	150		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on one edge)	60		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on one edge)	60		ns
$t_{c(TB)}$	TBiIN input cycle time (counted on both edges)	300		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on both edges)	160		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on both edges)	160		ns

Table Z-30. Timer B input (pulse period measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	600		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	300		ns

Table Z-31. Timer B input (pulse width measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	600		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	300		ns

Table Z-32. Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	300		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	150		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	150		ns
$t_{d(C-Q)}$	TxDi output delay time		160	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	50		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

Table Z-33. External interrupt $\overline{INTi}$ inputs

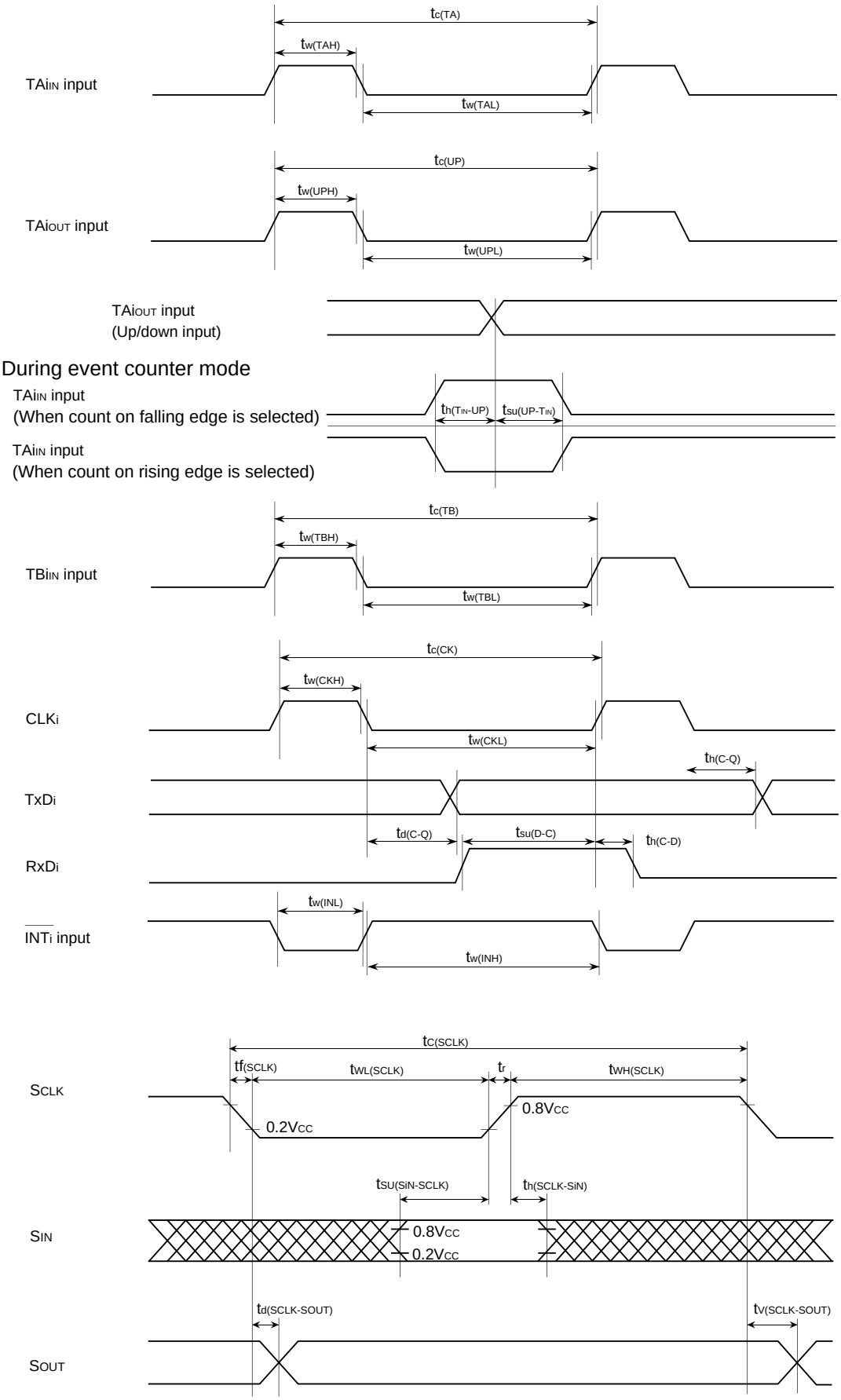
Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input HIGH pulse width	380		ns
$t_{w(INL)}$	$\overline{INTi}$ input LOW pulse width	380		ns

Table Z-34. Automatic transfer serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(SCLK)}$	Serial I/O clock input cycle time	TBD		µs
$t_{wH(SCLK)}$	Serial I/O clock input HIGH pulse width	TBD		ns
$t_{wL(SCLK)}$	Serial I/O clock input LOW pulse width	TBD		ns
$t_{su(SCLK-SIN)}$	Serial I/O input setup time	TBD		ns
$t_{h(SCLK-SIN)}$	Serial I/O input hold time	TBD		ns

Timing(V_{CC}=3V, only mask ROM version)

V_{CC}=3V



Outline Performance

Table AA-1 shows the outline performance of the M30218 group (flash memory version).

Table AA-1. Outline Performance of the M30218 group (flash memory version)

Item		Performance
Power supply voltage		4.0V to 5.5 V (f(X_{IN})=10MHz)
Program/erase voltage		V_{PP}=12V ± 5% (f(X_{IN})=10MHz)
		V_{CC}=5V ± 10% (f(X_{IN})=10MHz)
Flash memory operation mode		Three modes (parallel I/O, standard serial I/O, CPU rewrite)
Erase block division	User ROM area	See Figure 1.AA.3.
	Boot ROM area	One division (3.5 K bytes) (Note)
Program method		In units of byte
Erase method		Collective erase / block erase
Program/erase control method		Program/erase control by software command
Number of commands		6 commands
Program/erase count		100 times
ROM code protect		Standard serial I/O mode is supported.

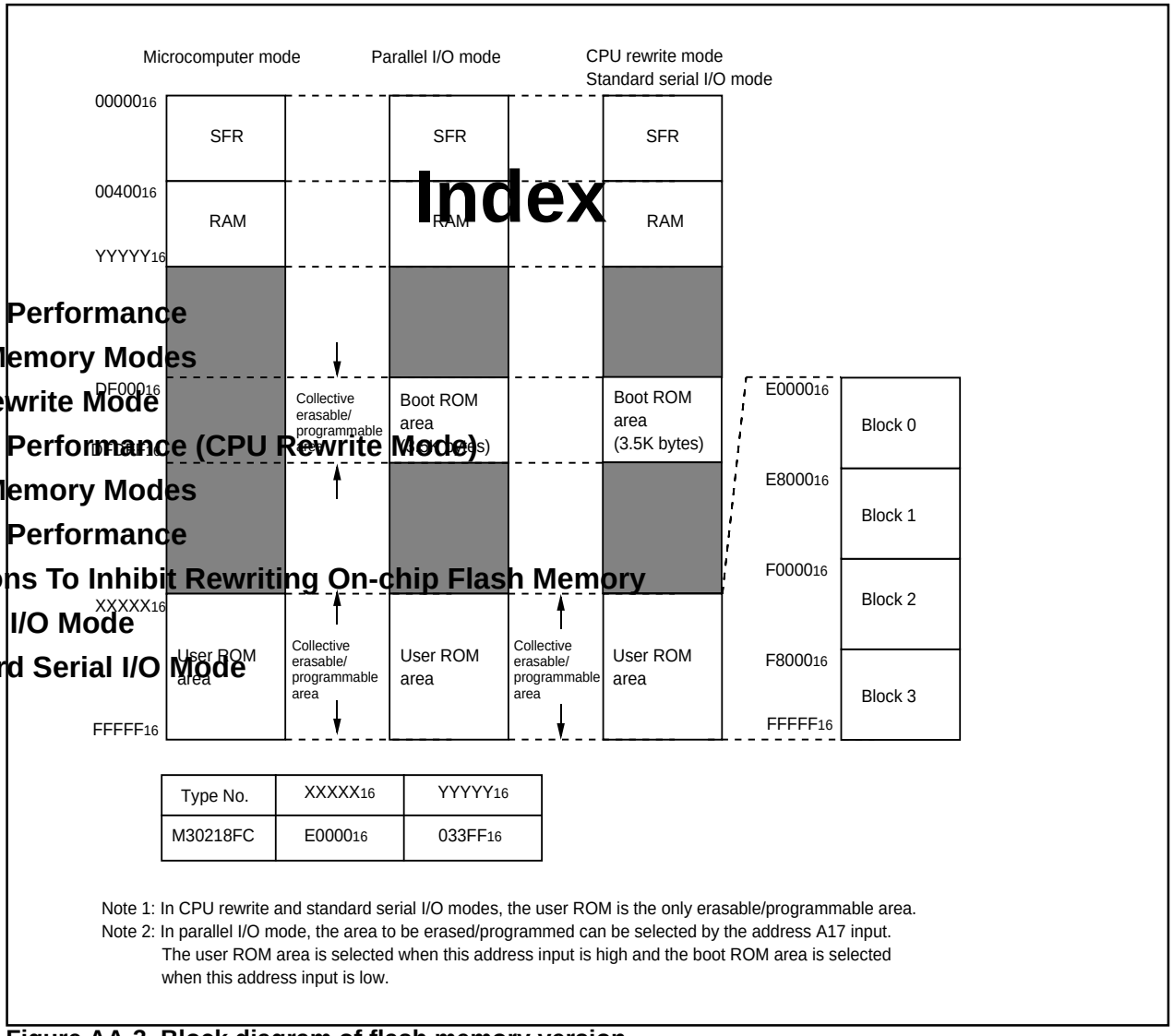
Note: The boot ROM area contains a standard serial I/O mode control program which is stored in it when shipped from the factory. This area can be erased and programmed in only parallel I/O mode.

Flash Memory

The M30218 group (flash memory version) contains the NOR type of flash memory that requires a high-voltage V_{PP} power supply for program/erase operations, in addition to the V_{CC} power supply for device operation. For this flash memory, three flash memory modes are available in which to read, program, and erase: parallel I/O and standard serial I/O modes in which the flash memory can be manipulated using a programmer and a CPU rewrite mode in which the flash memory can be manipulated by the Central Processing Unit (CPU). Each mode is detailed in the pages to follow.

In addition to the ordinary user ROM area to store a microcomputer operation control program, the flash memory has a boot ROM area that is used to store a program to control rewriting in CPU rewrite and standard serial I/O modes. This boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the factory. However, the user can write a rewrite control program in this area that suits the user's application system. This boot ROM area can be rewritten in only parallel I/O mode.

Index
Outline Performance
Flash Memory Modes
CPU Rewrite Mode
Outline Performance (CPU Rewrite Mode)
Flash Memory Modes
Outline Performance
Functions To Inhibit Rewriting On-chip Flash Memory
Parallel I/O Mode
Standard Serial I/O Mode



Type No.	XXXXX16	YYYYY16
M30218FC	E000016	033FF16

Note 1: In CPU rewrite and standard serial I/O modes, the user ROM is the only erasable/programmable area.
Note 2: In parallel I/O mode, the area to be erased/programmed can be selected by the address A17 input. The user ROM area is selected when this address input is high and the boot ROM area is selected when this address input is low.

Figure AA-3. Block diagram of flash memory version

CPU Rewrite Mode

In CPU rewrite mode, the on-chip flash memory can be operated on (read, program, or erase) under control of the Central Processing Unit (CPU). In CPU rewrite mode, the flash memory can be operated on by reading or writing to the flash memory control register and flash command register. Figure BB-1, Figure BB-2 show the flash memory control register, and flash command register respectively.

Also, in CPU rewrite mode, the CNVSS pin is used as the VPP power supply pin. Apply the power supply voltage, VPPH, from an external source to this pin.

In CPU rewrite mode, only the user ROM area shown in Figure AA-3 can be rewritten; the boot ROM area cannot be rewritten. Make sure the program and block commands are issued for only the user ROM area.

The control program for CPU rewrite mode can be stored in either user ROM or boot ROM area. In the CPU rewrite mode, because the flash memory cannot be read from the CPU, the rewrite control program must be transferred to internal RAM before it can be executed.

Flash memory control register 0

b7	b6	b5	b4	b3	b2	b1	b0
0				0			

Symbol: FCON0 — Address: $03B4_{16}$ — When reset: 00100000_2

Bit symbol	Bit name	Function	R	W
FCON00	CPU rewrite mode select bit	0: CPU rewrite mode is invalid 1: CPU rewrite mode is valid	○	○
Reserved bit		This bit can not write. The value, if read, turns out to be indeterminate.	—	—
FCON02	CPU rewrite mode monitor flag	0: CPU rewrite mode is invalid 1: CPU rewrite mode is valid	○	—
Reserved bit		Must always be set to "0".	○	○
FCON04	Erase / program area select bit	b6b5b4 000: Block 0 program/erase 001: Block 1 program/erase 010: Block 2 program/erase 011: Block 3 program/erase 110: Block 0 to 3 erase 111: Inhibit	○	○
FCON05			○	○
FCON06			○	○
Reserved bit		Must always be set to "0".	○	○

Flash memory control register 1

b7	b6	b5	b4	b3	b2	b1	b0
						0	0

Symbol: FCON1 — Address: $03B5_{16}$ — When reset: XXXXXX00$_2$

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to "0".	○	○
Nothing is assigned. In an attempt to write these bits, write "0". The value, if read, turns out to be indeterminate.			—	—

Figure BB-1. Flash memory control register

Flash command register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: FCMD — Address: $03B6_{16}$ — When reset: 00_{16}

Function	R	W
Writing of software command <Software command name> <Command code> •Read command "00_{16}" •Program command "40_{16}" •Program verify command "$C0_{16}$" •Erase command "20_{16}" +"20_{16}" •Erase verify command "$A0_{16}$" •Reset command "FF_{16}" +"FF_6"	×	○

Figure BB-2. Flash command register

Microcomputer Mode and Boot Mode

The control program for CPU rewrite mode must be written into the user ROM or boot ROM area in parallel I/O mode beforehand. (If the control program is written into the boot ROM area, the standard serial I/O mode becomes unusable.)

See Figure AA-3 for details about the boot ROM area.

Normal microcomputer mode is entered when the microcomputer is reset with pulling CNVSS pin low (VSS). In this case, the CPU starts operating using the control program in the user ROM area.

When the microcomputer is reset by pulling the P52 pin high (VCC), the CNVSS pin high(VPPH), the CPU starts operating using the control program in the boot ROM area. This mode is called the “boot” mode. The control program in the boot ROM area can also be used to rewrite the user ROM area.

CPU rewrite mode operation procedure

The internal flash memory can be operated on to program, read, verify, or erase it while being placed on-board by writing commands from the CPU to the flash memory control register (addresses $03B4_{16}$, $03B5_{16}$) and flash command register (address $03B6_{16}$). Note that when in CPU rewrite mode, the boot ROM area cannot be accessed for program, read, verify, or erase operations. Before this can be accomplished, a CPU write control program must be written into the boot ROM area in parallel input/output mode. The following shows a CPU rewrite mode operation procedure.

<Start procedure (Note 1)>

(1) Apply VPPH to the CNVSS/VPP pin and VCC to the port P46 pin for reset release. Or the user can jump from the user ROM area to the boot ROM area using the JMP instruction and execute the CPU write control program. In this case, set the CPU write mode select bit of the flash memory control register to “1” before applying VPPH to the CNVSS/VPP pin.

(2) After transferring the CPU write control program from the boot ROM area to the internal RAM, jump to this control program in RAM. (The operations described below are controlled by this program.)

(3) Set the CPU rewrite mode select bit to “1”.

(4) Read the CPU rewrite mode monitor flag to see that the CPU rewrite mode is enabled.

(5) Execute operation on the flash memory by writing software commands to the flash command register.

Note 1: In addition to the above, various other operations need to be performed, such as for entering the data to be written to flash memory from an external source (e.g., serial I/O), initializing the ports, and writing to the watchdog timer.

<Clearing procedure>

(1) Apply VSS to the CNVSS/VPP pin.

(2) Set the CPU rewrite mode select bit to “0”.

Precautions on CPU Rewrite Mode

Described below are the precautions to be observed when rewriting the flash memory in CPU rewrite mode.

(1) Operation speed

During erase/program mode, set BCLK to one of the following frequencies by changing the divide ratio:

5 MHz or less when wait bit (bit 7 at address 0005_{16}) = 0 (without internal access wait state)

10 MHz or less when wait bit (bit 7 at address 0005_{16}) = 1 (with internal access wait state)(Note 1)

(2) Instructions inhibited against use

The instructions listed below cannot be used during CPU rewrite mode because they refer to the internal data of the flash memory:

UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

(3) Interrupts inhibited against use

No interrupts can be used that look up the fixed vector table in the flash memory area. Maskable interrupts may be used by setting the interrupt vector table in a location outside the flash memory area.

Note 1: Internal access wait state can be set in CPU rewrite mode. In this time, the following function is only used.

- CPU, ROM, RAM, timer, UART, SI/O2(non-automatic transfer), port

In case of setting internal access wait state, refer to the following explain (software wait).

Software wait

A software wait can be inserted by setting the wait bit (bit 7) of the processor mode register 1 (address 0005_{16}) (Note 2).

A software wait is inserted in the internal ROM/RAM area by setting the wait bit of the processor mode register 1. When set to “0”, each bus cycle is executed in one BCLK cycle. When set to “1”, each bus cycle is executed in two BCLK cycles. After the microcomputer has been reset, this bit defaults to “0”.

The SFR area is always accessed in two BCLK cycles regardless of the setting of this control bit.

Table DA-1 shows the software wait and bus cycles. Figure DA-6 shows example bus timing when using software waits.

Note 2: Before attempting to change the contents of the processor mode register 1, set bit 1 of the protect register (address $000A_{16}$) to “1”.

Table DA-1. Software waits and bus cycles

Area	Wait bit	Bus cycle
SFR	Invalid	2 BCLK cycles
Internal ROM/RAM	0	1 BCLK cycle
	1	2 BCLK cycles

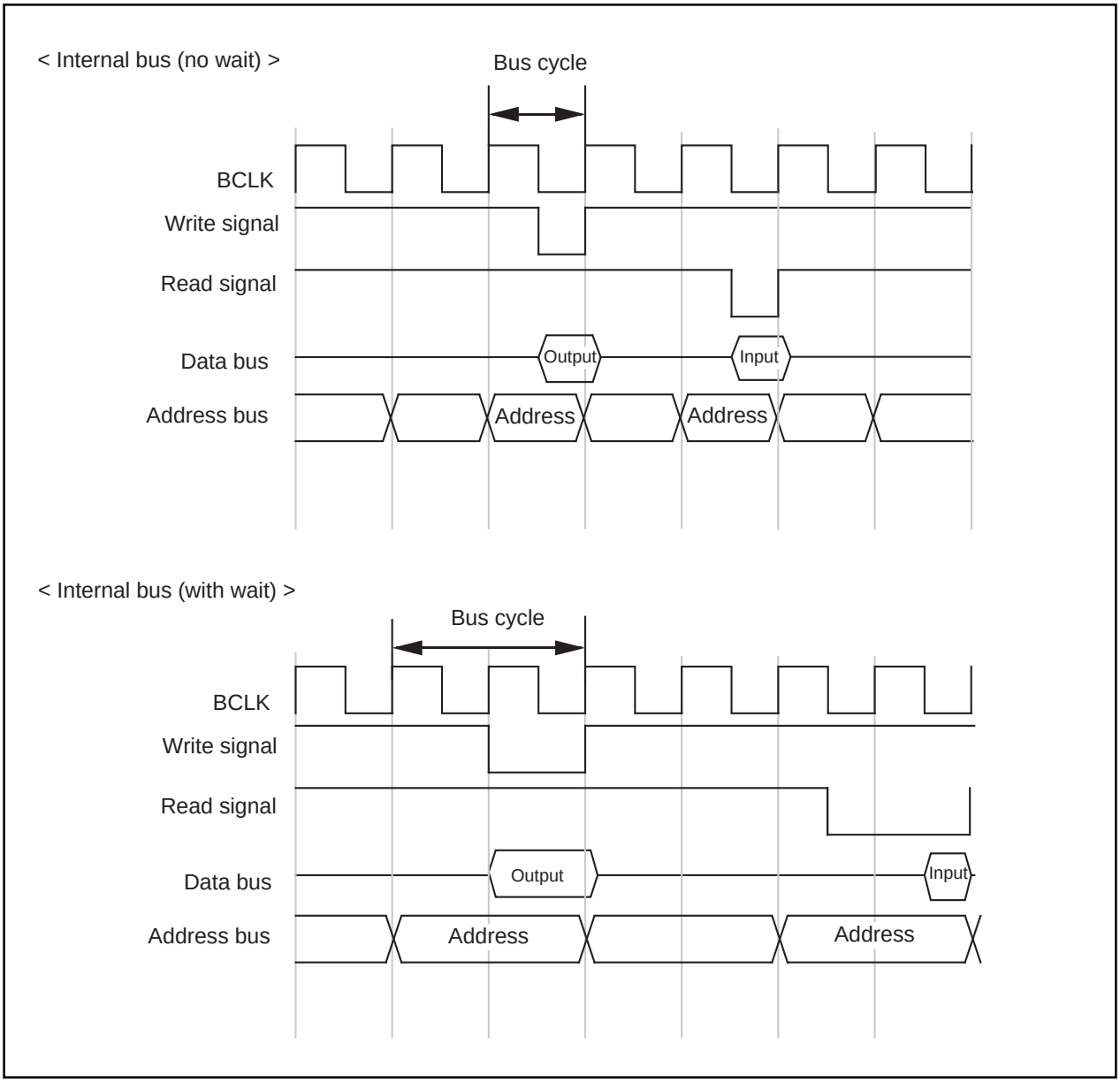


Figure DA-6. Typical bus timings using software wait

Software Commands

Table BB-1 lists the software commands available with the M30218 group (flash memory version).
When CPU rewrite mode is enabled, write software commands to the flash command register to specify the operation to erase or program.
The content of each software command is explained below.

Table BB-1. List of Software Commands (CPU Rewrite Mode)

Command	First bus cycle			Second bus cycle		
	Mode	Address	Data (D_0 to D_7)	Mode	Address	Data (D_0 to D_7)
Read	Write	$03B6_{16}$	00_{16}			
Program	Write	$03B6_{16}$	40_{16}	Write	Program address	Program data
Program verify	Write	$03B6_{16}$	$C0_{16}$	Read	Verify address	Verify data
Erase	Write	$03B6_{16}$	20_{16}	Write	$03B6_{16}$	20_{16}
Erase verify	Write	$03B6_{16}$	$A0_{16}$	Read	Verify address	Verify data
Reset	Write	$03B6_{16}$	FF_{16}	Write	$03B6_{16}$	FF_{16}

Read Command (00_{16})

The read mode is entered by writing the command code "00_{16}" to the flash command register in the first bus cycle. When an address to be read is input in one of the bus cycles that follow, the content of the specified address is read out at the data bus (D_0–D_7), 8 bits at a time.
The read mode is retained intact until another command is written.
After reset and after the reset command is executed, the read mode is set.

Program Command (40_{16})

The program mode is entered by writing the command code "40_{16}" to the flash command register in the first bus cycle. When the user execute an instruction to write byte data to the desired address (e.g., STE instruction) in the second bus cycle, the flash memory control circuit executes the program operation. The program operation requires approximately 20 μs. Wait for 20 μs or more before the user go to the next processing.
During program operation, the watchdog timer remains idle, with the value "$7FFF_{16}$" set in it.

Note 1: The write operation is not completed immediately by writing a program command once. The user must always execute a program-verify command after each program command executed. And if verification fails, the user need to execute the program command repeatedly until the verification passes. See Figure BB.3 for an example of a programming flowchart.

Program-verify command (C0$_{16}$)

The program-verify mode is entered by writing the command code "C0$_{16}$" to the flash command register in the first bus cycle. When the user execute an instruction (e.g., LDE instruction) to read byte data from the address to be verified (the previously programmed address) in the second bus cycle, the content that has actually been written to the address is read out from the memory.

The CPU compares this read data with the data that it previously wrote to the address using the program command. If the compared data do not match, the user need to execute the program and program-verify operations one more time.

Erase command (20$_{16}$ + 20$_{16}$)

The flash memory control circuit executes an erase operation by writing command code "20$_{16}$" to the flash command register in the first bus cycle and the same command code to the flash command register again in the second bus cycle. The erase operation requires approximately 20 ms. Wait for 20 ms or more before the user go to the next processing.

Before this erase command can be performed, all memory locations to be erased must have had data "00$_{16}$" written to by using the program and program-verify commands. During erase operation, the watchdog timer remains idle, with the value "7FFF$_{16}$ set in it.

Note 1: The erase operation is not completed immediately by writing an erase command once. The user must always execute an erase-verify command after each erase command executed. And if verification fails, the user need to execute the erase command repeatedly until the verification passes. See Figure BB-3 for an example of an erase flowchart.

Erase-verify command (A0$_{16}$)

The erase-verify mode is entered by writing the command code "A0$_{16}$" to the flash command register in the first bus cycle. When the user execute an instruction to read byte data from the address to be verified (e.g., LDE instruction) in the second bus cycle, the content of the address is read out.

The CPU must sequentially erase-verify memory contents one address at a time, over the entire area erased. If any address is encountered whose content is not "FF$_{16}$" (not erased), the CPU must stop erase-verify at that point and execute erase and erase-verify operations one more time.

Note 1: If any unerased memory location is encountered during erase-verify operation, be sure to execute erase and erase-verify operations one more time. In this case, however, the user does not need to write data "00$_{16}$" to memory before erasing.

Reset command (FF16 + FF16)

The reset command is used to stop the program command or the erase command in the middle of operation. After writing command code "4016" or "2016" twice to the flash command register, write command code "FF16" to the flash command register in the first bus cycle and the same command code to the flash command register again in the second bus cycle. The program command or erase command is disabled, with the flash memory placed in read mode.

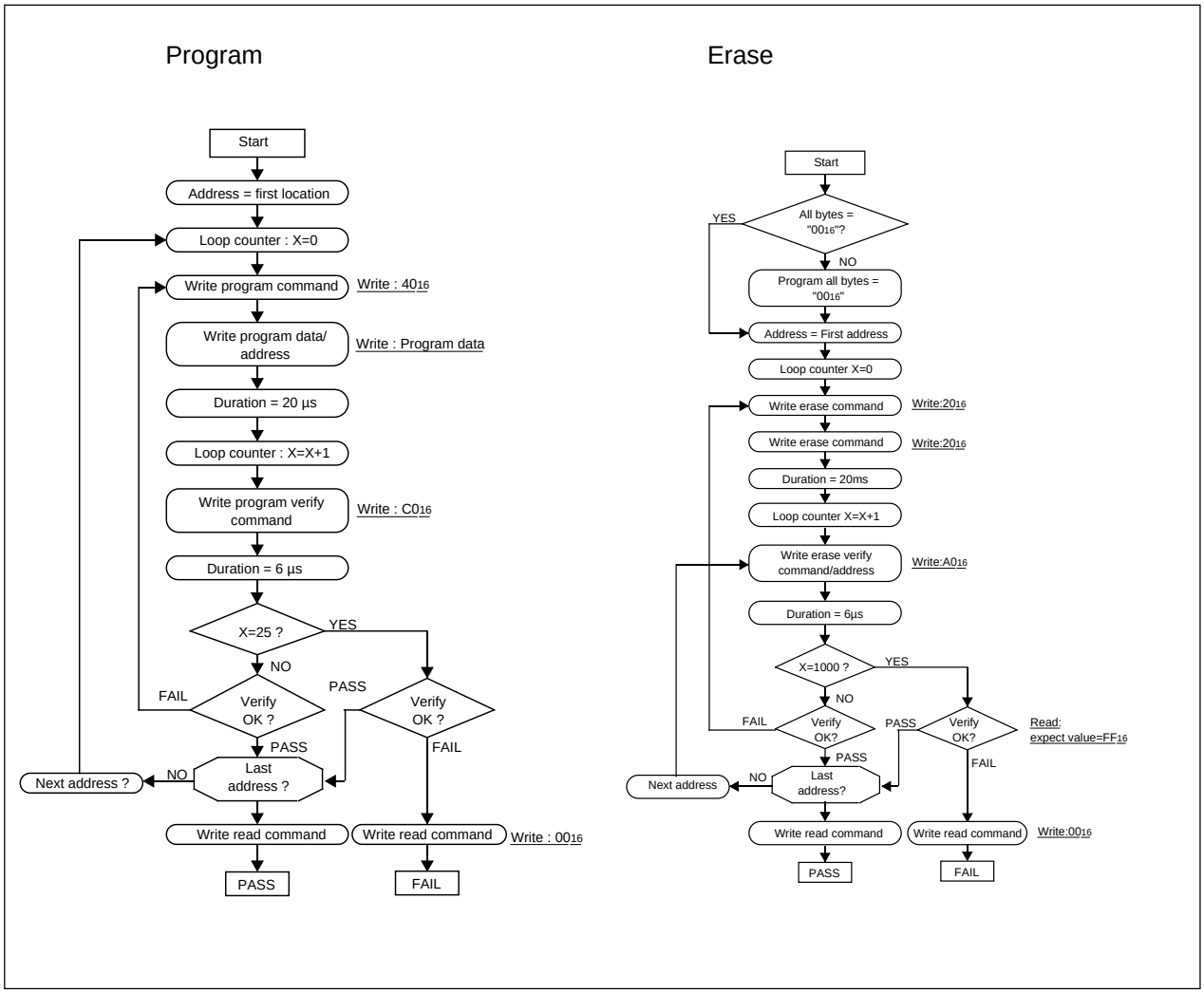


Figure BB-3. Program and erase execution flowchart in the CPU rewrite mode

Pin functions (Flash memory standard serial I/O mode)

Pin	Name	I/O	Description
V_{CC},V_{SS}	Power input		Apply 5V ± 10 % to Vcc pin and 0 V to Vss pin.
CNV_{SS}	CNV_{SS}	I	Apply 12V ± 5 % to this pin.
$\overline{RESET}$	Reset input	I	Reset input pin. While reset is "L" level, a 20 cycle or longer clock must be input to XIN pin.
X_{IN}	Clock input	I	Connect a ceramic resonator or crystal oscillator between X_{IN} and X_{OUT} pins. To input an externally generated clock, input it to X_{IN} pin and open X_{OUT} pin.
X_{OUT}	Clock output	O	
AV_{CC}, AV_{SS}	Analog power supply input		Connect AV_{SS} to Vss and AVcc to Vcc, respectively.
V_{REF}	Reference voltage input	I	Enter the reference voltage for AD from this pin.
$P0_0$ to $P0_7$	Output port P0	O	Output exclusive use pin.
$P1_0$ to $P1_7$	Output port P1	O	Output exclusive use pin.
$P2_0$ to $P2_7$	Output port P2	O	Output exclusive use pin.
$P3_0$ to $P3_7$	Input port P3	I	Input "H" or "L" level signal or open.
$P4_0$ to $P4_3$	Input port P4	I	Input "H" or "L" level signal or open.
$P4_4$	TxD output	O	Serial data output pin.
$P4_5$	RxD input	I	Serial data input pin.
$P4_6$	SCLK input	I	Serial clock input pin.
$P4_7$	BUSY output	O	BUSY signal output pin.
$P5_0$ to $P5_7$	Output port P5	O	Output exclusive use pin.
$P6_0$ to $P6_7$	Output port P6	O	Output exclusive use pin.
$P7_0$ to $P7_7$	Input port P7	I	Input "H" or "L" level signal or open.
$P8_0$ to $P8_7$	Input port P8	I	Input "H" or "L" level signal or open.
$P9_0$ to $P9_7$	Input port P9	I	Input "H" or "L" level signal or open.
$P10_0$ to $P10_7$	Input port P10	I	Input "H" or "L" level signal or open.

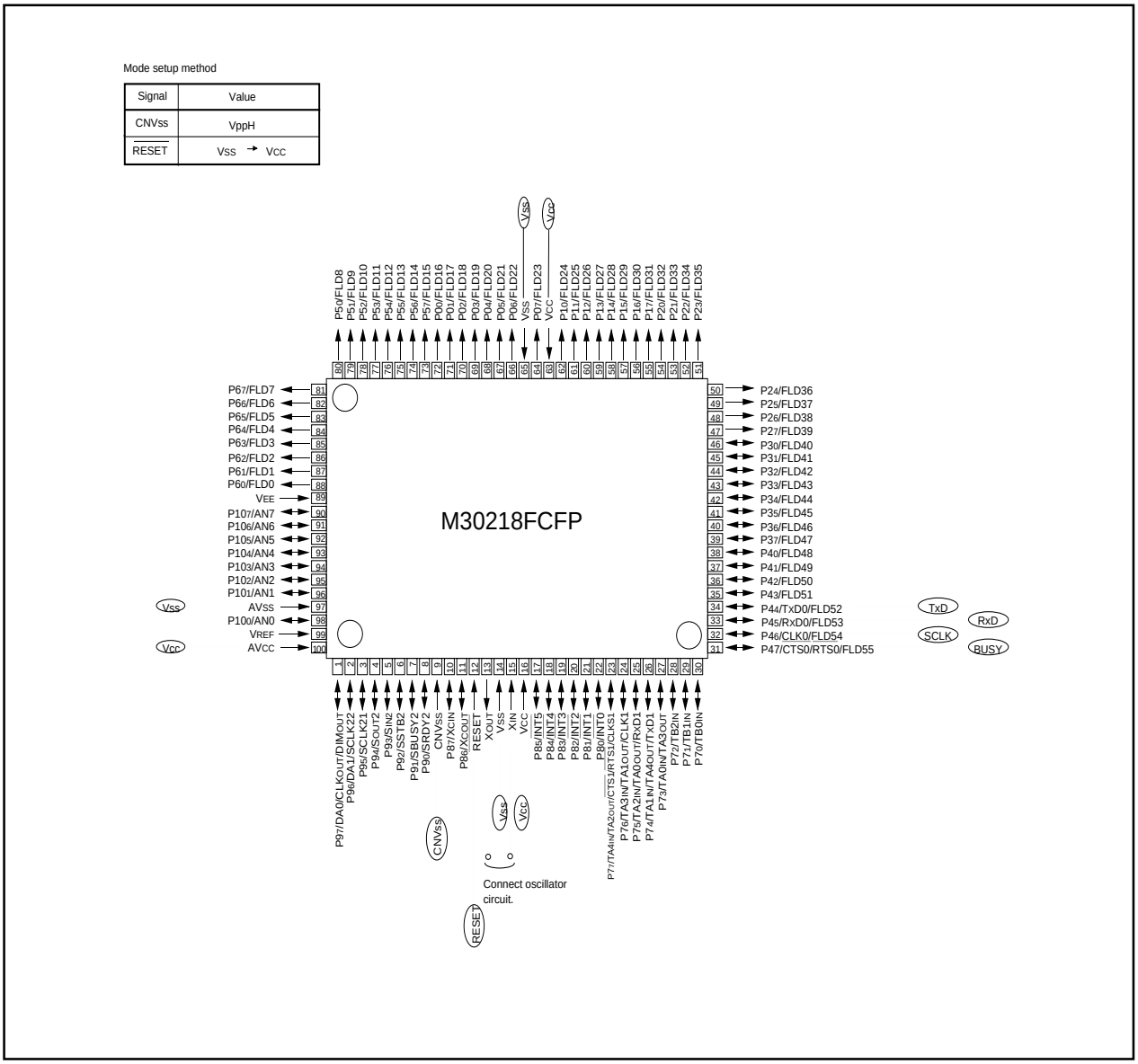


Figure DD-1. Pin connections for serial I/O mode (1)

Standard Serial I/O Mode

The standard serial I/O mode serially inputs and outputs the software commands, addresses and data necessary for operating (read, program, erase, etc.) the internal flash memory. It uses a purpose-specific serial programmer.

The standard serial I/O mode differs from the parallel I/O mode in that the CPU controls operations like rewriting (uses the CPU rewrite mode) in the flash memory or serial input for rewriting data. The standard serial I/O mode is started by clearing the reset with V_{PPH} at the CNVss pin. (For the normal microprocessor mode, set CNVss to “L”.)

This control program is written in the boot ROM area when shipped from Mitsubishi Electric. Therefore, if the boot ROM area is rewritten in the parallel I/O mode, the standard serial I/O mode cannot be used. Figures DD-1 shows the pin connections for the standard serial I/O mode. Serial data I/O uses three UART0 pins: CLK_0, RxD_0, TxD_0, and RTS_0 (BUSY).

The CLK_0 pin is the transfer clock input pin and it transfers the external transfer clock. The TxD_0 pin outputs the CMOS signal. The RTS_0 (BUSY) pin outputs an “L” level when reception setup ends and an “H” level when the reception operation starts. Transmission and reception data is transferred serially in 8-byte blocks.

In the standard serial I/O mode, only the user ROM area shown in Figure AA-3 can be rewritten, the boot ROM area cannot.

The standard serial I/O mode has a 7-byte ID code. When the flash memory is not blank and the ID code does not match the content of the flash memory, the command sent from the programmer is not accepted.

Function Overview (Standard Serial I/O Mode)

In the standard serial I/O mode, software commands, addresses and data are input and output between the flash memory and an external device (serial programmer, etc.) using a clock synchronized serial I/O (UART0) . In reception, the software commands, addresses and program data are synchronized with the rise of the transfer clock input to the CLK_0 pin and input into the flash memory via the RxD_0 pin.

In transmission, the read data and status are synchronized with the fall of the transfer clock and output to the outside from the TxD_0 pin.

The TxD_0 pin is CMOS output. Transmission is in 8-bit blocks and LSB first.

When busy, either during transmission or reception, or while executing an erase operation or program, the RTS_0 (BUSY) pin is “H” level. Accordingly, do not start the next transmission until the RTS_0 (BUSY) pin is “L” level.

Also, data in memory and the status register can be read after inputting a software command. It is possible to check flash memory operating status or whether a program or erase operation ended successfully or in error by reading the status register.

Software commands and the status register are explained here following.

Software Commands

Table DD-1 lists software commands. In the standard serial I/O mode, erase operations, programs and reading are controlled by transferring software commands via the RxD pin. Software commands are explained here below.

Table DD-1. Software commands (Standard serial I/O mode)

	Control command	1st byte transfer	2nd byte	3rd byte	4th byte	5th byte	6th byte		When ID is not verificate
1	Page read	FF_{16}	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
2	Page program	41_{16}	Address (middle)	Address (high)	Data input	Data input	Data input	Data input to 259th byte	Not acceptable
3	Bclock ease	20_{16}	Address (middle)	Address (high)	$D0_{16}$				Not acceptable
4	Erase all unlocked blocks	$A7_{16}$	$D0_{16}$						Not acceptable
5	Read status register	70_{16}	SRD output	SRD1 output					Acceptable
6	Clear status register	50_{16}							Not acceptable
7	Read lockbit status	71_{16}	Address (middle)	Address (high)	Lock bit data output				Not acceptable
8	ID check function	$F5_{16}$	Address (low)	Address (middle)	Address (high)	ID size	ID1	To ID7	Acceptable
9	Download function	FA_{16}	Size (low)	Size (high)	Check-sum	Data input	To required number of times		Not acceptable
10	Version data output function	FB_{16}	Version data output	Version data output	Version data output	Version data output	Version data output	Version data output to 9th byte	Acceptable
11	Boot area output function	FC_{16}	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable

Note1: Shading indicates transfer from flash memory microcomputer to serial programmer. All other data is transferred from the serial programmer to the flash memory microcomputer.

Note2: SRD refers to status register data. SRD1 refers to status register 1 data.

Note3: All commands can be accepted when the flash memory is totally blank.

Page Read Command

This command reads the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page read command as explained here following.

(1) Send the "FF_{16}" command code in the 1st byte of the transmission.

(2) Send addresses A_8 to A_{15} and A_{16} to A_{23} in the 2nd and 3rd bytes of the transmission respectively.

(3) From the 4th byte onward, data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} will be output sequentially from the smallest address first in sync with the rise of the clock.

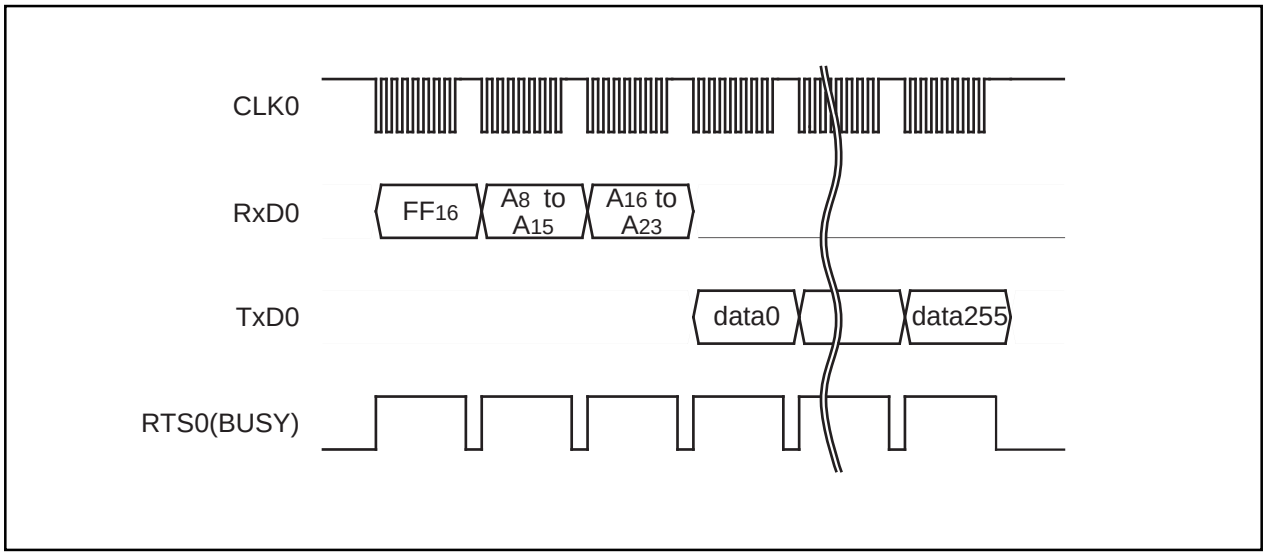


Figure DD-2. Timing for page read

Read Status Register Command

This command reads status information. When the "70_{16}" command code is sent in the 1st byte of the transmission, the contents of the status register (SRD) specified in the 2nd byte of the transmission and the contents of status register 1 (SRD1) specified in the 3rd byte of the transmission are read.

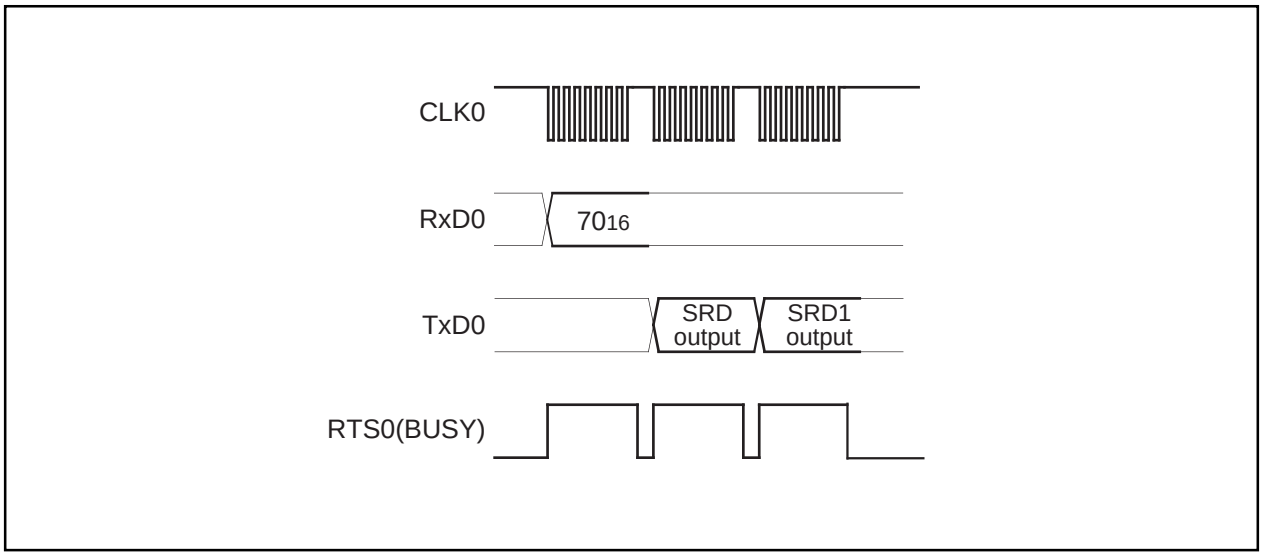


Figure DD-3. Timing for reading the status register

Clear Status Register Command

This command clears the bits (SR3–SR4) which are set when the status register operation ends in error. When the "50_{16}" command code is sent in the 1st byte of the transmission, the aforementioned bits are cleared. When the clear status register operation ends, the RTS_0 (BUSY) signal changes from the "H" to the "L" level.

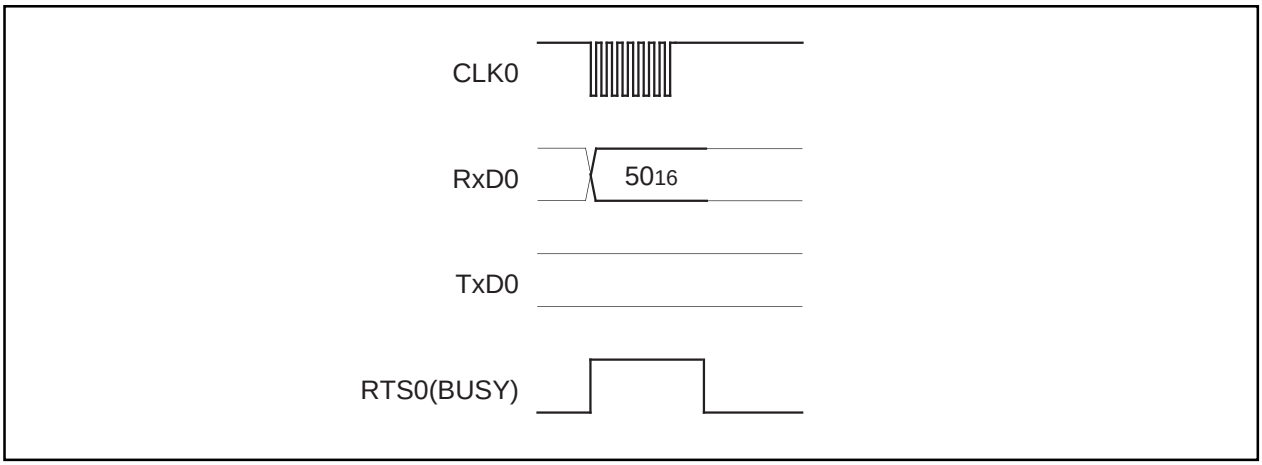


Figure DD-4. Timing for clearing the status register

Page Program Command

This command writes the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page program command as explained here following.

(1) Send the "41_{16}" command code in the 1st byte of the transmission.
(2) Send addresses A_8 to A_{15} and A_{16} to A_{23} in the 2nd and 3rd bytes of the transmission respectively.
(3) From the 4th byte onward, as write data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} is input sequentially from the smallest address first, that page is automatically written.

When reception setup for the next 256 bytes ends, the RTS_0 (BUSY) signal changes from the "H" to the "L" level. The result of the page program can be known by reading the status register. For more information, see the section on the status register.

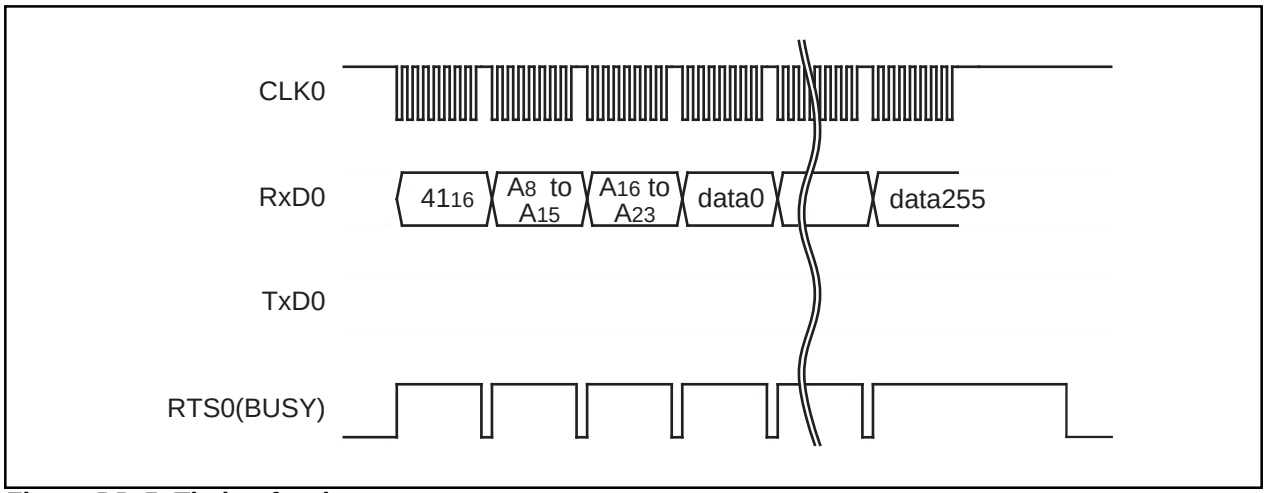


Figure DD-5. Timing for the page program

Block Erase Command

This command erases the data in the specified block. Execute the block erase command as explained here following.

(1) Send the "20_{16}" command code in the 1st byte of the transmission.

(2) Send addresses A_8 to A_{15} and A_{16} to A_{23} in the 2nd and 3rd bytes of the transmission respectively.

(3) Send the verify command code "$D0_{16}$" in the 4th byte of the transmission. With the verify command code, the erase operation will start for the specified block in the flash memory. Write the highest address of the specified block for addresses A_{16} to A_{23}.

When block erasing ends, the RTS0 (BUSY) signal changes from the "H" to the "L" level. After block erase ends, the result of the block erase operation can be known by reading the status register. For more information, see the section on the status register.

Each block can be erase-protected with the lock bit. For more information, see the section on the data protection function.

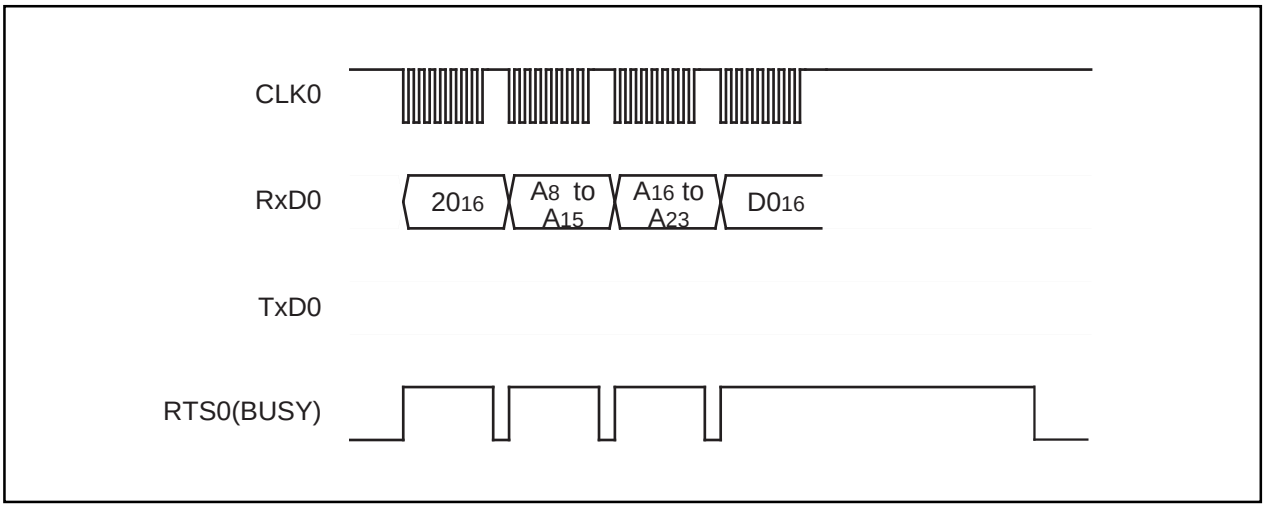


Figure DD-6.Timing for block erasing

Erase All Unlocked Blocks Command

This command erases the content of all blocks. Execute the erase all unlocked blocks command as explained here following.

(1) Send the "A7$_{16}$" command code in the 1st byte of the transmission.

(2) Send the verify command code "D0$_{16}$" in the 2nd byte of the transmission. With the verify command code, the erase operation will start and continue for all blocks in the flash memory.

When block erasing ends, the RTS$_0$ (BUSY) signal changes from the "H" to the "L" level. The result of the erase operation can be known by reading the status register.

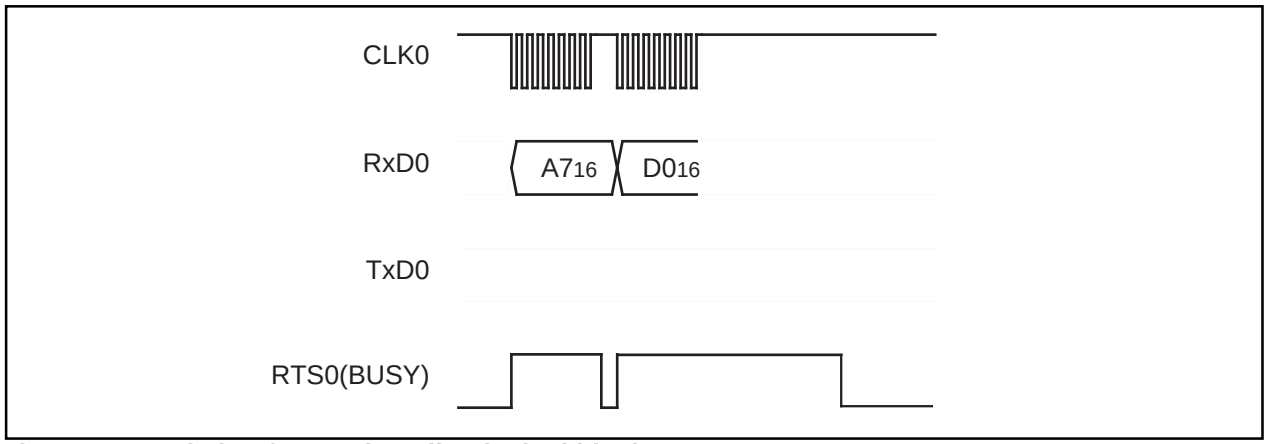


Figure DD-7. Timing for erasing all unlocked blocks

Read Lock Bit Status Command

This command reads the lock bit status of the specified block. Execute the read lock bit status command as explained here following.

(1) Send the "71$_{16}$" command code in the 1st byte of the transmission.

(2) Send addresses A$_8$ to A$_{15}$ and A$_{16}$ to A$_{23}$ in the 2nd and 3rd bytes of the transmission respectively.

(3) The lock bit data of the specified block is output in the 4th byte of the transmission. Write the highest address of the specified block for addresses A$_8$ to A$_{23}$.

The M30218 group (flash memory version) does not have the lock bit, so the read value is always "1" (block unlock).

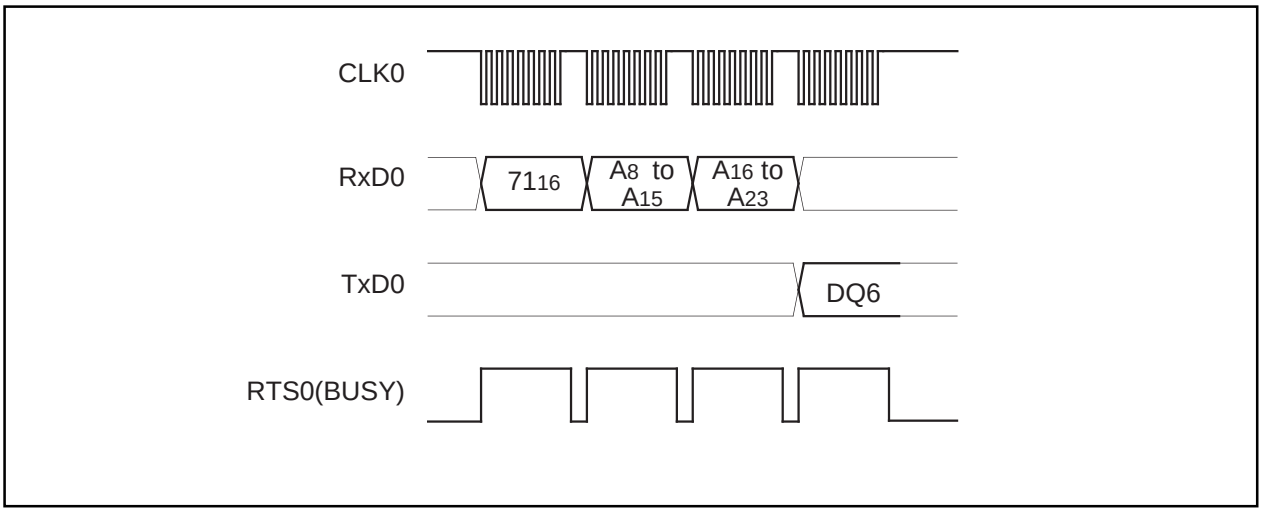


Figure DD-8. Timing for reading lock bit status

Download Command

This command downloads a program to the RAM for execution. Execute the download command as explained here following.

(1) Send the "FA16" command code in the 1st byte of the transmission.
(2) Send the program size in the 2nd and 3rd bytes of the transmission.
(3) Send the check sum in the 4th byte of the transmission. The check sum is added to all data sent in the 5th byte onward.
(4) The program to execute is sent in the 5th byte onward.

When all data has been transmitted, if the check sum matches, the downloaded program is executed. The size of the program will vary according to the internal RAM.

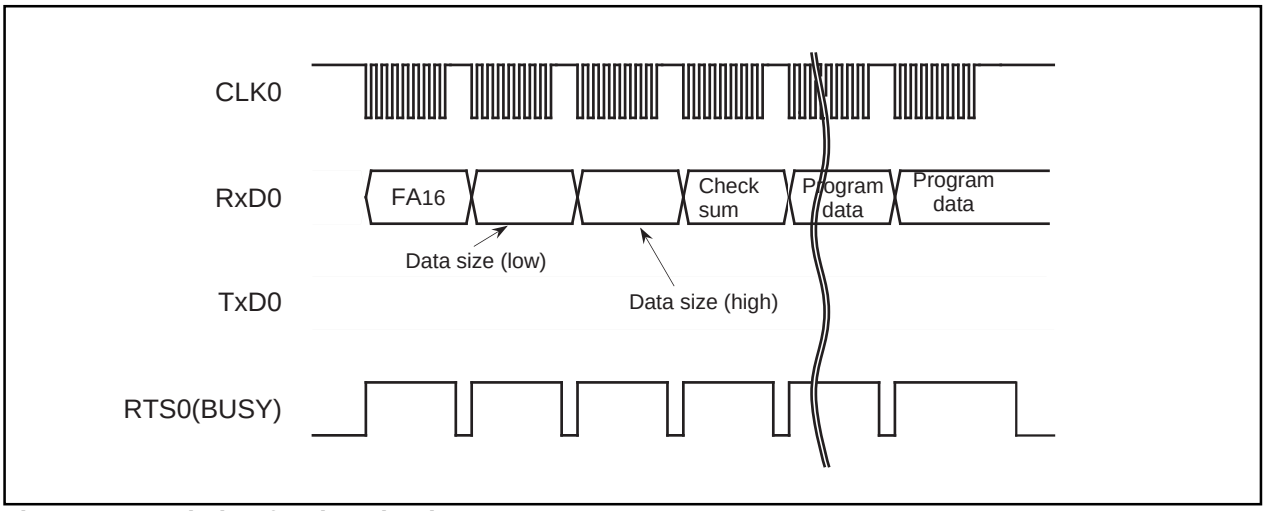


Figure DD-9. Timing for download

Version Information Output Command

This command outputs the version information of the control program stored in the boot area. Execute the version information output command as explained here following.

(1) Send the "FB16" command code in the 1st byte of the transmission.

(2) The version information will be output from the 2nd byte onward. This data is composed of 8 ASCII code characters.

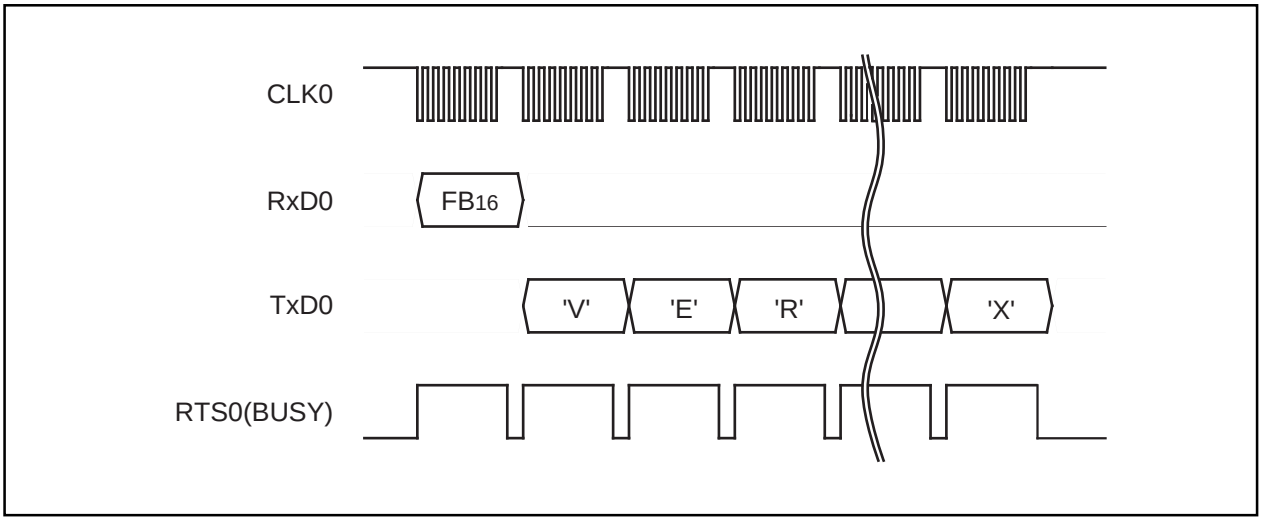


Figure DD-10. Timing for version information output

Boot Area Output Command

This command outputs the control program stored in the boot area in one page blocks (256 bytes). Execute the boot area output command as explained here following.

(1) Send the "FC16" command code in the 1st byte of the transmission.

(2) Send addresses A8 to A15 and A16 to A23 in the 2nd and 3rd bytes of the transmission respectively.

(3) From the 4th byte onward, data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first, in sync with the rise of the clock.

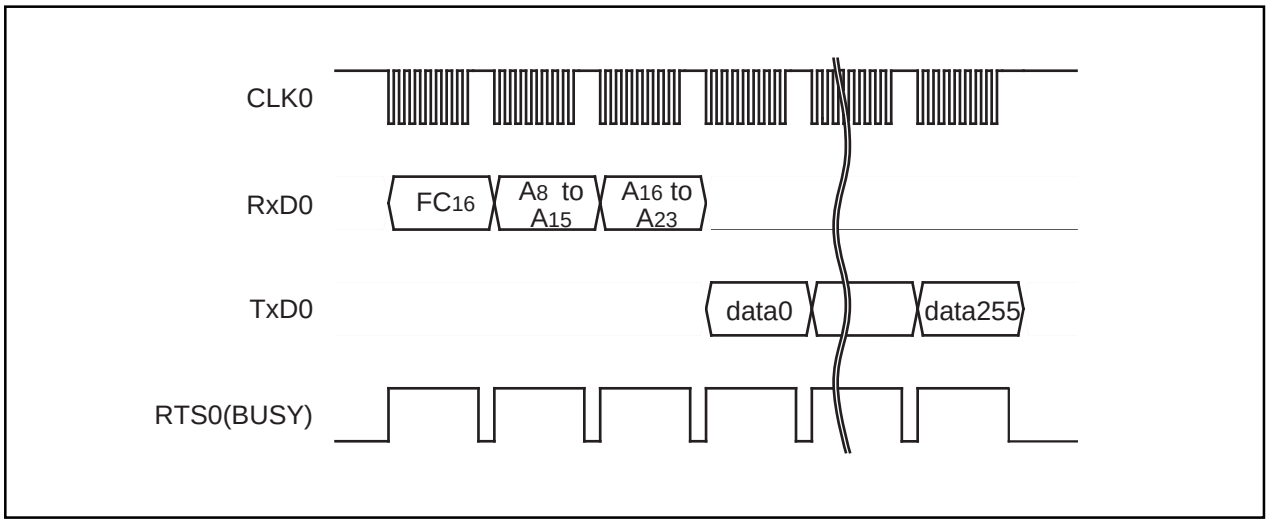


Figure DD-11. Timing for boot area output

ID Check

This command checks the ID code. Execute the boot ID check command as explained here following.

(1) Send the "F5$_{16}$" command code in the 1st byte of the transmission.

(2) Send addresses A$_0$ to A$_7$, A$_8$ to A$_{15}$ and A$_{16}$ to A$_{23}$ of the 1st byte of the ID code in the 2nd, 3rd and 4th bytes of the transmission respectively.

(3) Send the number of data sets of the ID code in the 5th byte.

(4) The ID code is sent in the 6th byte onward, starting with the 1st byte of the code.

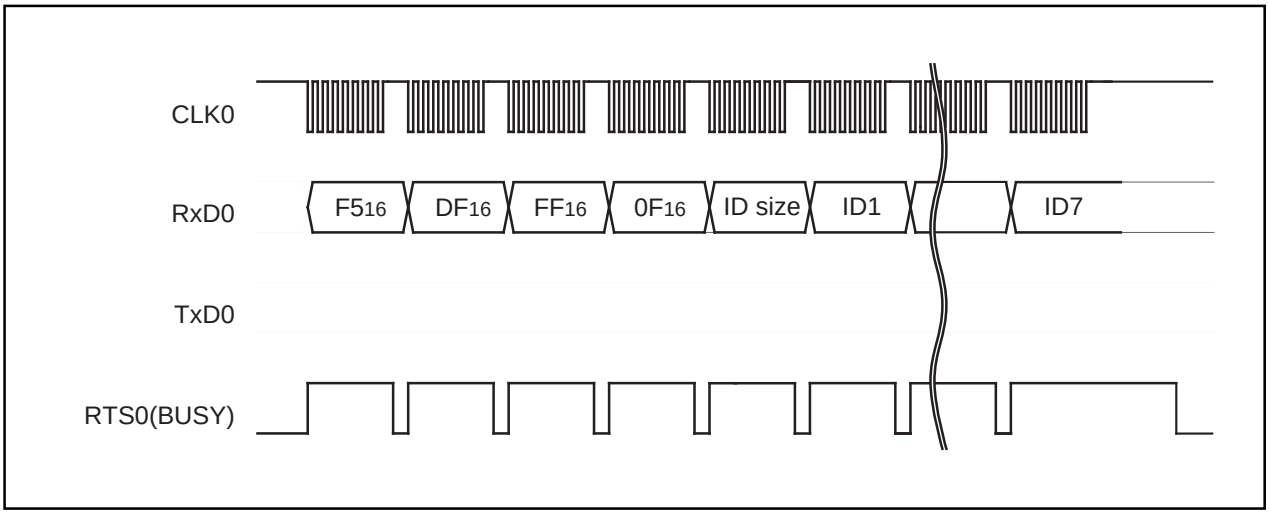


Figure DD-12. Timing for the ID check

ID Code

When the flash memory is not blank, the ID code sent from the serial programmer and the ID code written in the flash memory are compared to see if they match. If the codes do not match, the command sent from the serial programmer is not accepted. An ID code contains 8 bits of data. Area is, from the 1st byte, addresses 0FFFDF$_{16}$, 0FFFE3$_{16}$, 0FFFEB$_{16}$, 0FFFEF$_{16}$, 0FFFF3$_{16}$, and 0FFFF7$_{16}$. Write a program into the flash memory, which already has the ID code set for these addresses.

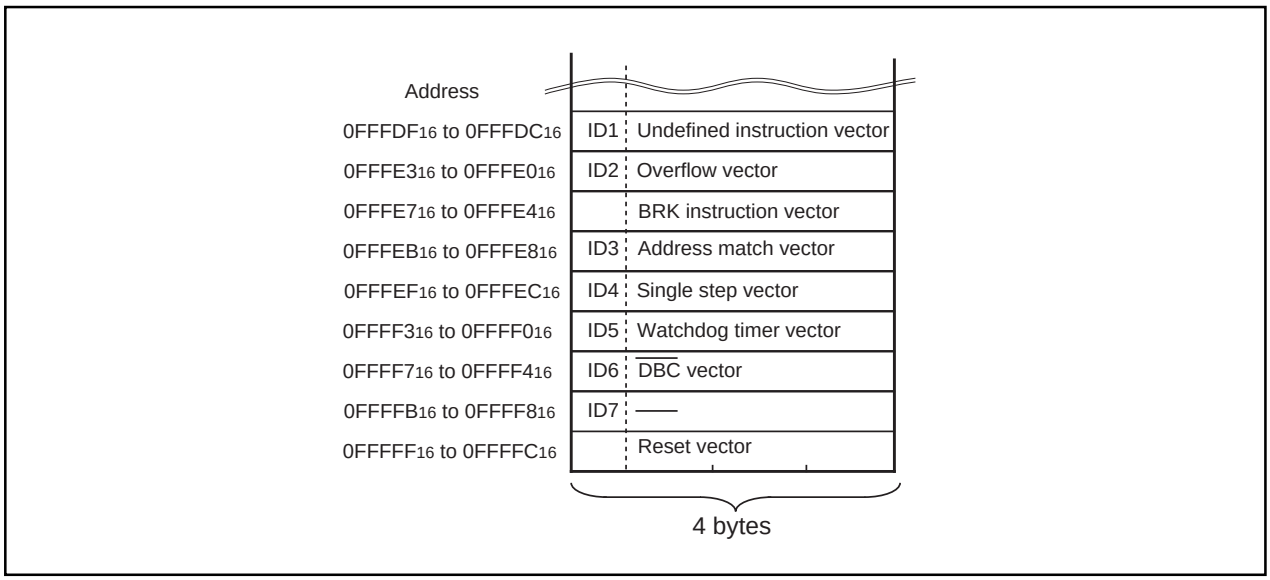


Figure DD-13. ID code storage addresses

Status Register (SRD)

The status register indicates operating status of the flash memory and status such as whether an erase operation or a program ended successfully or in error. It can be read by writing the read status register command (70_{16}). Also, the status register is cleared by writing the clear status register command (50_{16}). Table DD-2 gives the definition of each status register bit. After clearing the reset, the status register outputs "80_{16}".

Table DD-2. Status register (SRD)

SRD0 bits	Status name	Definition	
		"1"	"0"
SR7 (bit7)	Status bit	Ready	Busy
SR6 (bit6)	Reserved	-	-
SR5 (bit5)	Erase bit	Terminated in error	Terminated normally
SR4 (bit4)	Program bit	Terminated in error	Terminated normally
SR3 (bit3)	Reserved	-	-
SR2 (bit2)	Reserved	-	-
SR1 (bit1)	Reserved	-	-
SR0 (bit0)	Reserved	-	-

Status Bit (SR7)

The status bit indicates the operating status of the flash memory. When power is turned on, "1" (ready) is set for it. The bit is set to "0" (busy) during an auto write or auto erase operation, but it is set back to "1" when the operation ends.

Erase Bit (SR5)

The erase bit reports the operating status of the auto erase operation. If an erase error occurs, it is set to "1". When the erase status is cleared, it is set to "0".

Program Bit (SR4)

The program bit reports the operating status of the auto write operation. If a write error occurs, it is set to "1". When the program status is cleared, it is set to "0".

Status Register 1 (SRD1)

Status register 1 indicates the status of serial communications, results from ID checks and results from check sum comparisons. It can be read after the SRD by writing the read status register command (70_{16}). Also, status register 1 is cleared by writing the clear status register command (50_{16}).

Table DD-3 gives the definition of each status register 1 bit. "00_{16}" is output when power is turned ON and the flag status is maintained even after the reset.

Table DD-3. Status register 1 (SRD1)

SRD1 bits	Status name	Definition	
		"1"	"0"
SR15 (bit7)	Boot update completed bit	Update completed	Not update
SR14 (bit6)	Reserved	-	-
SR13 (bit5)	Reserved	-	-
SR12 (bit4)	Checksum match bit	Match	Mismatch
SR11 (bit3) SR10 (bit2)	ID check completed bits	00 Not verified 01 Verification mismatch 10 Reserved 11 Verified	
SR9 (bit1)	Data receive time out	Time out	Normal operation
SR8 (bit0)	Reserved	-	-

Boot Update Completed Bit (SR15)

This flag indicates whether the control program was downloaded to the RAM or not, using the download function.

Check Sum Consistency Bit (SR12)

This flag indicates whether the check sum matches or not when a program, is downloaded for execution using the download function.

ID Check Completed Bits (SR11 and SR10)

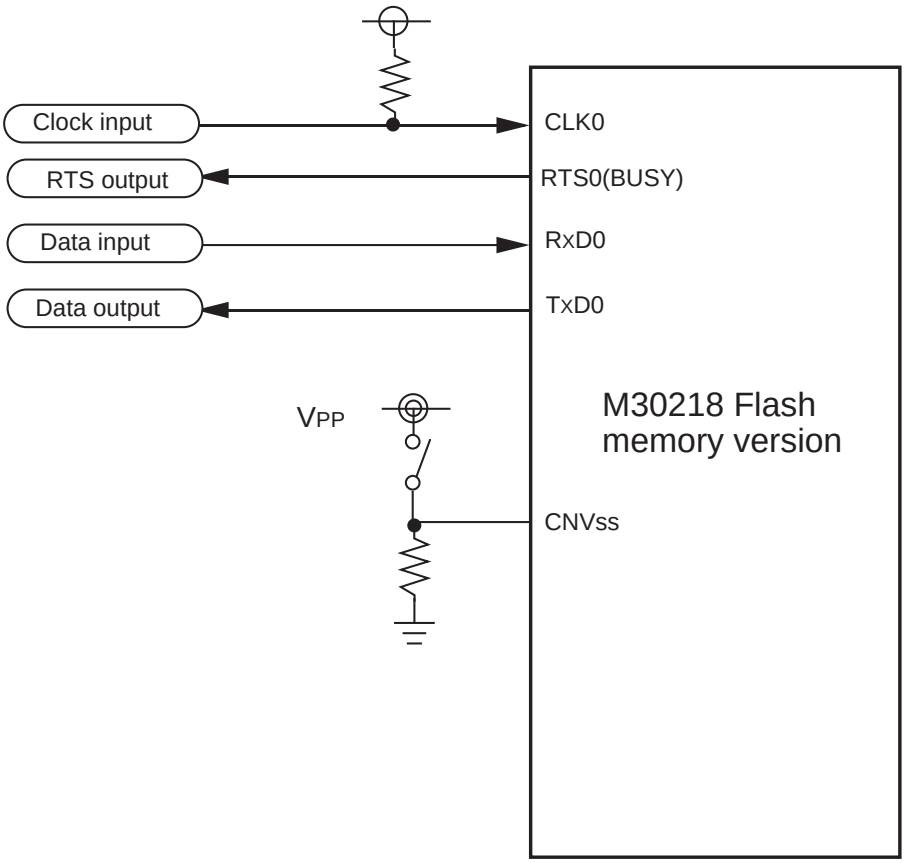
These flags indicate the result of ID checks. Some commands cannot be accepted without an ID check.

Data Reception Time Out (SR9)

This flag indicates when a time out error is generated during data reception. If this flag is attached during data reception, the received data is discarded and the microcomputer returns to the command wait state.

Example Circuit Application for The Standard Serial I/O Mode

The below figure shows a circuit application for the standard serial I/O mode. Control pins will vary according to programmer, therefore see the programmer manual for more information.



(1) Control pins and external circuitry will vary according to programmer. For more information, see the programmer manual.
(2) In this example, the microprocessor mode and standard serial I/O mode are switched via a switch.

Figure DD-14. Example circuit application for the standard serial I/O mode

Revision History

Version	Contents for change	Revision date
REV.A1	Page 2 Figure AA-1 M30218-XXXFP ---> M30218-XXXXFP Page 10 Figure BA-3 $03B8_{16}$ DMA0 cause select register ---> DMA0 request cause select register $03BA_{16}$ DMA1 cause select register ---> DMA1 request cause select register Page 55 Figure KA-2 FLDC mode register bit3, bit2 (at rising edge of each edge) ---> (at rising edge of each digit) 11: ---> 10: Page 90 Figure GA-4 UARTi transmit/receive control register 0 bit4 ($P4_7$ and $P7_4$ function as) ---> ($P4_7$ and $P7_7$ function as) Page 128 Exclusive High-breakdown]voltage Output Ports Line 2 All ports have structure of high-breakdown-voltage P-channel open drain output and pull-down resistance. ---> All ports have structure of high-breakdown-voltage P-channel open drain output. Exclusive output ports except P2 have built-in pull-down resistance. Page 134 Add to Note 3.	99.12.21

Version	Contents for change	Revision date

Revision history	M30218 Data sheet	

MITSUBISHI SEMICONDUCTORS
M30218 Group Specification REV.A1

Dec. First Edition 1999

Editioned by
Committee of editing of Mitsubishi Semiconductor

Published by
Mitsubishi Electric Corp., Kitaitami Works