To all our customers

Regarding the change of names mentioned in the document, such as Mitsubishi Electric and Mitsubishi XX, to Renesas Technology Corp.

The semiconductor operations of Hitachi and Mitsubishi Electric were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Mitsubishi Electric, Mitsubishi Electric Corporation, Mitsubishi Semiconductors, and other Mitsubishi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Note : Mitsubishi Electric will continue the business operations of high frequency & optical devices and power devices.

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

Description

Description

The M16C/6N 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 office, communications, industrial equipment, and other high-speed processing applications.

The M16C/6N group is consisted of two sub-groups, M16C/6N0 group and M16C/6N1 group. The M16C/6N0 group has two CAN (Controller Area Network) modules and the M16C/6N1 group has one CAN module (See Figure 1.1.4 Memory Expansion). The CAN modules comply with the 2.0B specification. The M16C/6N group is suited to drive automotive and industrial control systems.

Features

- Memory capacity ..... ROM 128K/256K bytes
  RAM 5K/10K bytes
- Shortest instruction execution time ..... 62.5 ns ($f(X_{IN})$ = 16MHz, 1/1 prescaler, without software wait)
- Supply voltage ..... 4.2 to 5.5V ($f(X_{IN})$ = 16MHz, 1/1 prescaler, without software wait)
- Low power dissipation ..... 60mA M16C/6N0 group Mask products
  65mA M16C/6N0 group Flash products
  50mA M16C/6N1 group Mask products
  55mA M16C/6N1 group Flash products
  ($f(X_{IN})$ = 16MHz, 1/1 prescaler, without software wait)
- Interrupts ..... 29 internal and 9 external interrupt sources, 4 software interrupt sources, 7 priority levels (including key input interrupt)
- Multifunction 16-bit timer ..... 5 output timers + 6 input timers
- Serial I/O ..... 3 channels (UART/clock synchronous, 1 channel clock synchronous)
- DMAC ..... 2 channels (trigger: 24 sources)
- CAN module ..... 2 channels for M16C/6N0 group
  1 channel for M16C/6N1 group
- A-D converter ..... 10 bits X (8X3+2) channels
- D-A converter ..... 8 bits X 2 channels
- CRC calculation circuit ..... 1 circuit
- Watchdog timer ..... 1 15-bit timer
- Programmable I/O ..... 87 lines
- Input port ..... 1 line ($P8_5$ shared with $\overline{NMI}$ pin)
- Chip select output ..... 4 lines
- Memory expansion ..... Available (to a maximum of 1M bytes)
- Clock generating circuit ..... 3 built-in circuits
  Main clock generating circuit, Sub clock generating circuit,
  (built-in feedback resistor, and external ceramic or quartz crystal oscillator)
  Ring oscillation circuit (with an oscillation stop detection circuit)

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.

Applications

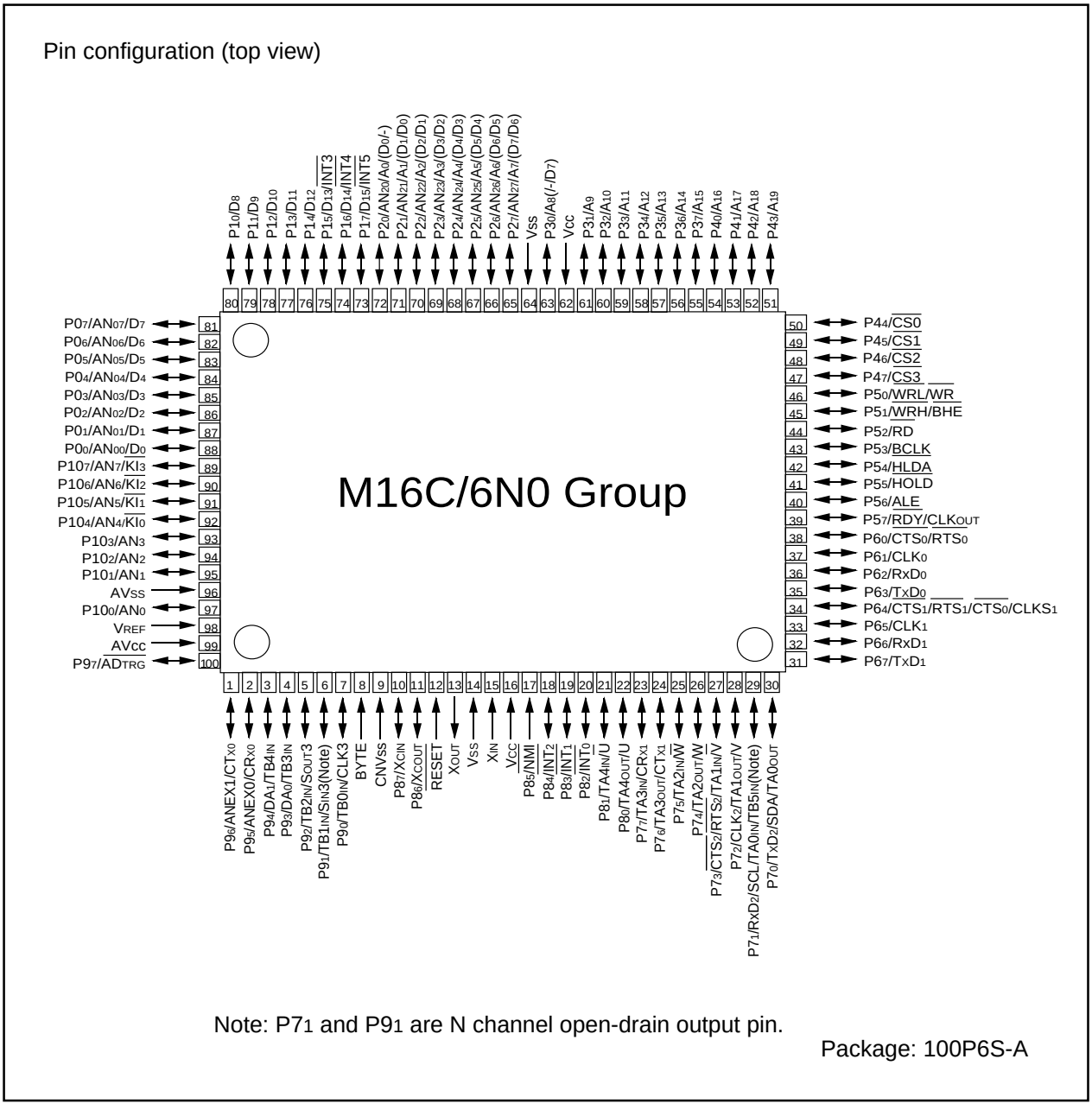
Automotive and industrial control systems

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

Central Processing Unit (CPU) 11
Reset 14
Processor Mode 22
Clock Generating Circuit 36
Protection 52
Interrupt 53
Watchdog Timer 75
DMAC 77
Timer 87
Serial I/O 117
A-D Converter 157
D-A Converter 167
CRC Calculation Circuit 169
CAN Module 171
Programmable I/O Port 196
Electrical Characteristics 207
Flash Memory Version 224

Pin Configuration

Figures 1.1.1 and 1.1.2 show the pin configuration (top view).



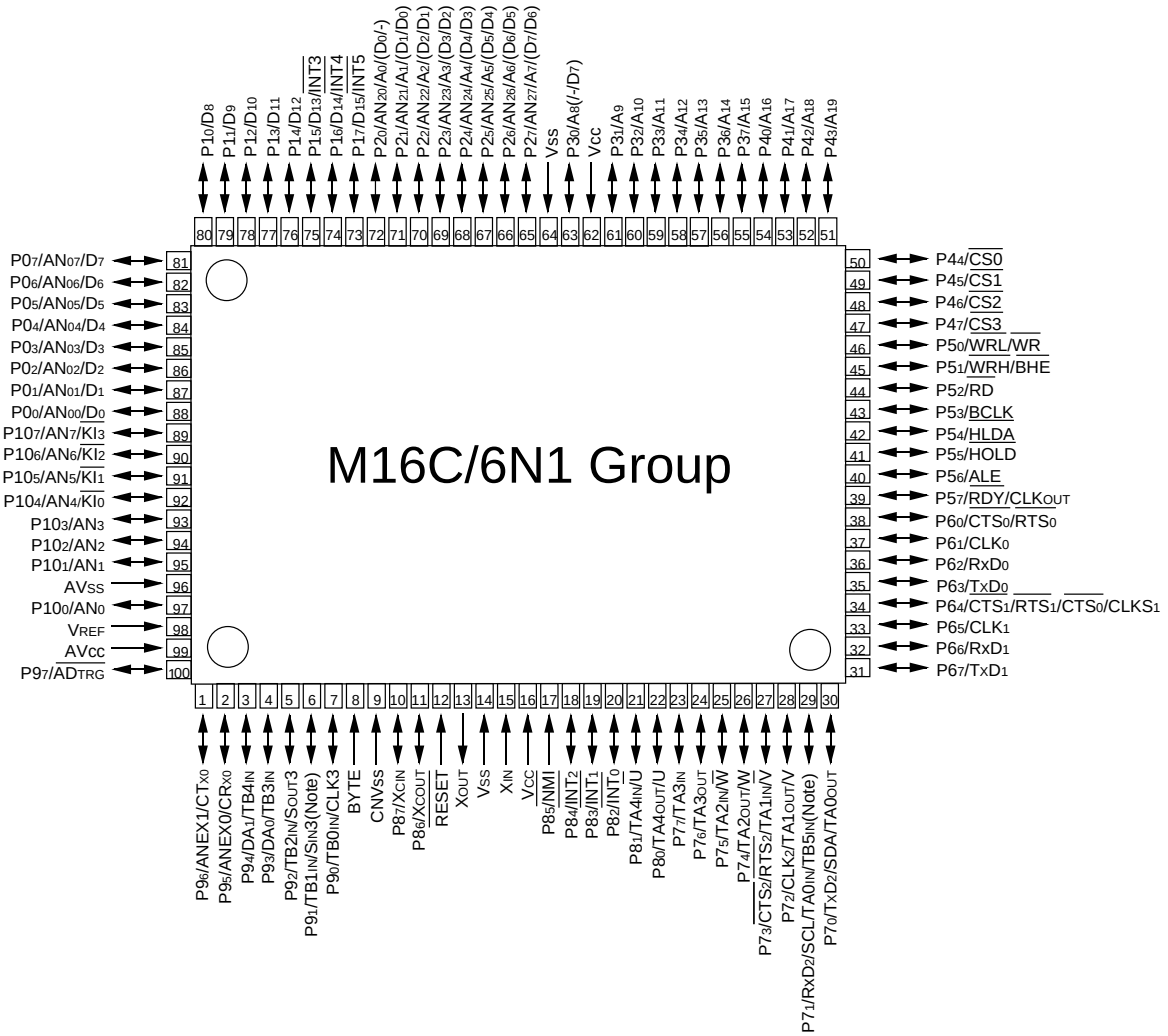
Note: P71 and P91 are N channel open-drain output pin.

Package: 100P6S-A

Figure 1.1.1. Pin configuration for M16C/6N0 group (top view)

Description

Pin configuration (top view)



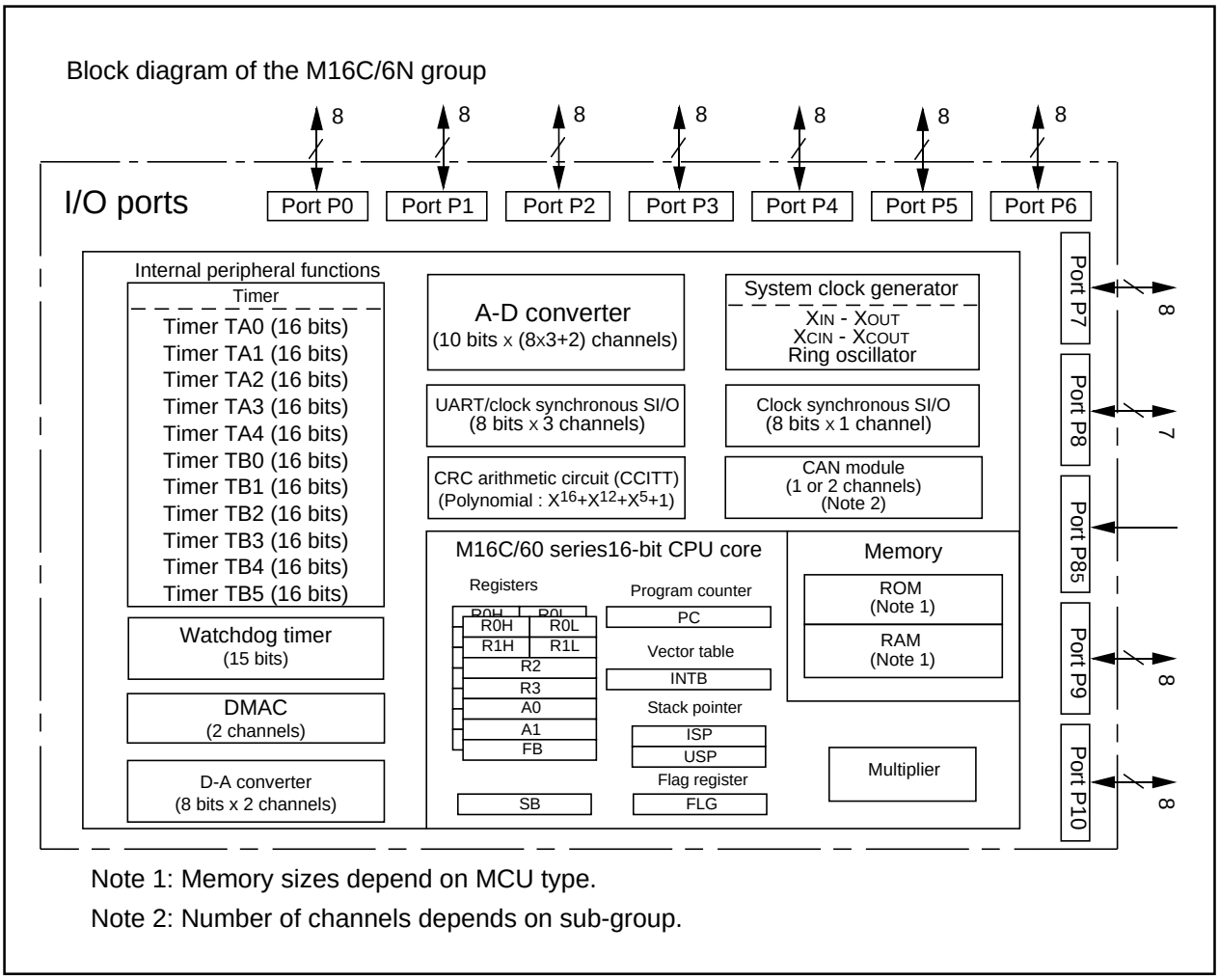
Note: P71 and P91 are N channel open-drain output pin.

Package: 100P6S-A

Figure 1.1.2. Pin configuration for M16C/6N1 group (top view)

Block Diagram

Figure 1.1.3 is a block diagram of the M16C/6N group.



Note 1: Memory sizes depend on MCU type.

Note 2: Number of channels depends on sub-group.

Figure 1.1.3. Block diagram of M16C/6N group

Performance Outline

Table 1.1.1 is a performance outline of the M16C/6N group.

Table 1.1.1. Performance outline of M16C/6N group

Item		Performance
Number of basic instructions		91 instructions
The shortest instruction execution time		62.5 ns (f(X_{IN}) = 16 MHz, 1/1 prescaler, without software wait)
Memory capacity	ROM	128K / 256K bytes
	RAM	5K / 10 K bytes
I/O ports	P0 to P10 (except $P8_5$)	8-bit x 10, 7-bit x 1
Input port	$P8_5$	1-bit x 1
Multifunction timer	TA0, TA1, TA2, TA3, TA4	16-bit x 5
	TB0, TB1, TB2, TB3, TB4, TB5	16-bit x 6
Serial I/O	UART0, UART1, UART2	(UART or clock synchronous) x 3
	S I/O3	Clock synchronous
A-D converter		10 bits x (8 x 3 + 2) channels
D-A converter		8 bits x 2 channels
CRC calculation circuit		1 circuit CRC-CCITT
DMAC		2 channels (trigger: 24 sources)
CAN Module		2 channels for M16C/6N0 group 1 channel for M16C/6N1 group
Watch-dog timer		15-bit x 1 (with prescaler)
Interrupt		29 internal and 9 external sources, 4 software sources, 7 priority levels
Clock generating circuit		3 built-in clock generating circuits (built-in feedback resistor, external ceramic or quartz crystal oscillator, and internal ring oscillator)
Supply voltage		4.2 to 5.5 V (f(X_{IN}) = 16 MHz, 1/1 prescaler, without software wait)
Power dissipation		60 mA M16C/6N0 group Mask products 65 mA M16C/6N0 group Flash products 50 mA M16C/6N1 group Mask products 55 mA M16C/6N1 group Flash products (f(X_{IN}) = 16 MHz, 1/1 prescaler, without software wait)
I/O characteristics	I/O withstand voltage	5 V
	Output current	5 mA
Operating ambient temperature		-40 to 125 ˚C
Device configuration		CMOS high performance silicon gate

M16C/6N Group Expansion

Mitsubishi releases the following products in the M16C/6N group.

(1) Supports to mask ROM versions and flash memory versions

(2) ROM size

(3) Package

100P6S-A ..... Plastic mold QFP (Mask ROM and Flash memory versions)

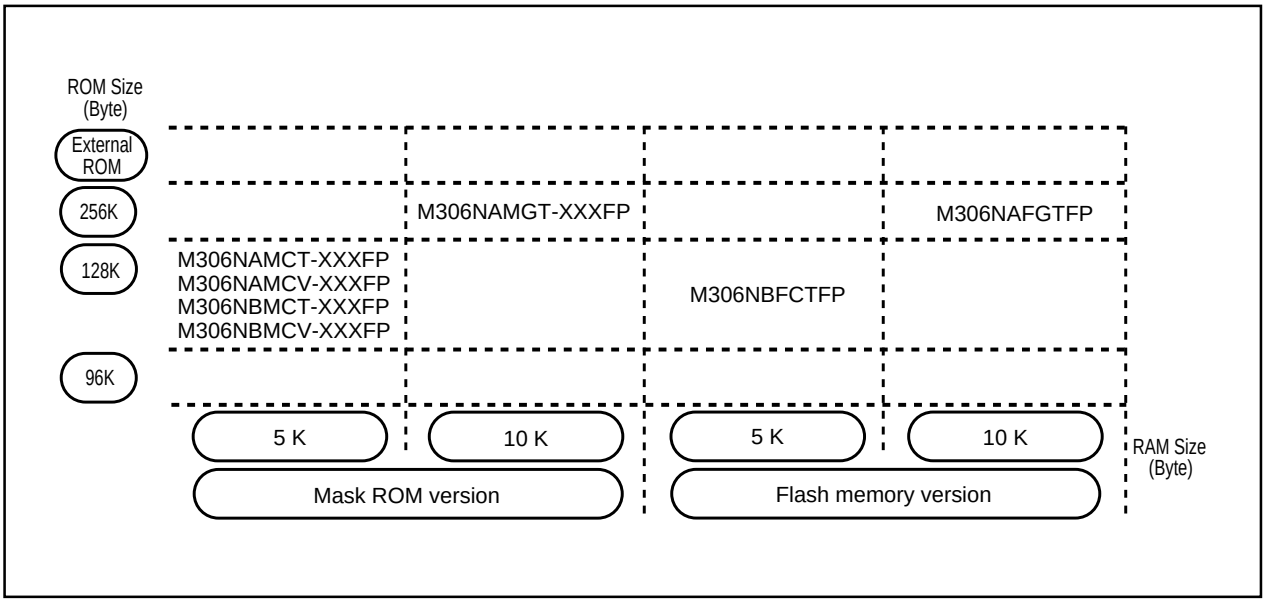


Figure 1.1.4. Memory expansion

Mitsubishi supports the types shown in the list below.

Table 1.1.2. M16C/6N group

October 2002

Type No.		ROM size (Byte)	RAM size (Byte)	CAN module (Channel)	Characteristic	Package type	Remarks
M16C/6N0 group	M306NAMGT-XXXFP	256K	10K	2	85˚C guaranteed version	100P6S-A	Mask ROM version
	M306NAMCT-XXXFP	128K	5K				
	M306NAMCV-XXXFP	128K	5K		125˚C guaranteed version(Note 2)		
	M306NAFGTFP	256K	10K		85˚C guaranteed version		Flash memory 5V version
M16C/6N1 group	M306NBMCT-XXXFP	128K	5K	1	85˚C guaranteed version	100P6S-A	Mask ROM version
	M306NBMCV-XXXFP	128K	5K		125˚C guaranteed version(Note 2)		
	M306NBFCTFP	128K	5K		85˚C guaranteed version		Flash memory 5V version

Note 1: It may change in the future.

Note 2: It is difficult from 85˚C guranteed version on operating ambient temperature and terms of the use, please inguire.

Description

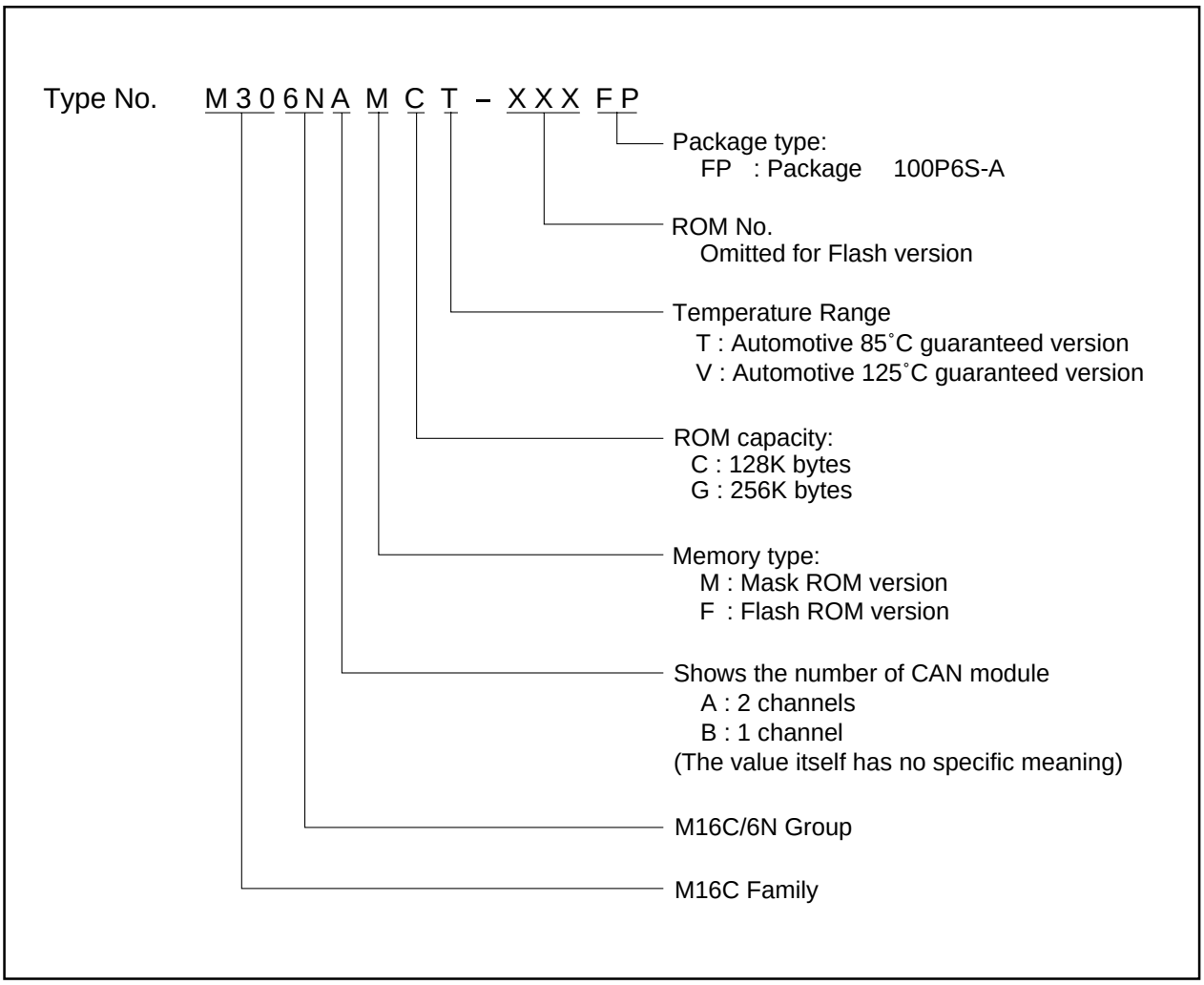


Figure 1.1.5. Type No., memory size, and package

Pin Description

Table 1.2.1. Pin description of M16C/6N group (1)

Pin name	Signal name	I/O type	Function
V_{CC}, V_{SS}	Power supply input		Supply 4.2 to 5.5 V to the V_{CC} pin. Supply 0 V to the V_{SS} pin.
CNV$_{SS}$	CNV$_{SS}$	Input	This pin switches between processor modes. Connect it to the V_{SS} pin to operate in single-chip or memory expansion mode. Connect it to the V_{CC} pin to operate in microprocessor mode.
$\overline{\text{RESET}}$	Reset input	Input	A "L" on this input resets the microcomputer.
X_{IN}	Clock input	Input	These pins are provided for the main clock generating circuit. Connect a ceramic resonator or quartz 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.
X_{OUT}	Clock output	Output	
BYTE	External data bus width select input	Input	This pin selects the width of an external data bus. A 16-bit width is selected when this input is "L"; an 8-bit width is selected when this input is "H". This input must be fixed to either "H" or "L". When operating in single-chip mode, connect this pin to V_{SS}.
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.
$P0_0$ to $P0_7$	I/O port P0	Input/output	This is an 8-bit CMOS I/O port. It has an input/output port direction register that allows the user to set each pin for input or output individually. When set for input, the user can specify in units of four bits via software whether or not they are tied to a pull-up resistor. Pins in this port also function as A-D converter input pins.
D_0 to D_7		Input/output	When set as a separate bus, these pins input and output data (D_0 to D_7).
$P1_0$ to $P1_7$	I/O port P1	Input/output	This is an 8-bit I/O port equivalent to P0. Pins in this port also function as external interrupt pins as selected by software.
D_8 to D_{15}		Input/output	When set as a separate bus, these pins input and output data (D_8 to D_{15}).
$P2_0$ to $P2_7$	I/O port P2	Input/output	This is an 8-bit I/O port equivalent to P0. Pins in this port also function as A-D converter input pins.
A_0 to A_7		Output	These pins output 8 low-order address bits (A_0 to A_7).
A_0/D_0 to A_7/D_7		Input/output	If the external bus is set as an 8-bit wide multiplexed bus, these pins input and output data (D_0 to D_7) and output 8 low-order address bits (A_0 to A_7) separated in time by multiplexing.
A_0, A_1/D_0 to A_7/D_6		Output Input/output	If the external bus is set as a 16-bit wide multiplexed bus, these pins input and output data (D_0 to D_6) and output address (A_1 to A_7) separated in time by multiplexing. They also output address (A_0).
$P3_0$ to $P3_7$	I/O port P3	Input/output	This is an 8-bit I/O port equivalent to P0.
A_8 to A_{15}		Output	These pins output 8 middle-order address bits (A_8 to A_{15}).
A_8/D_7, A_9 to A_{15}		Input/output Output	If the external bus is set as a 16-bit wide multiplexed bus, these pins input and output data (D_7) and output address (A_8) separated in time by multiplexing. They also output address (A_9 to A_{15}).
$P4_0$ to $P4_7$	I/O port P4	Input/output	This is an 8-bit I/O port equivalent to P0.
$\overline{CS_0}$ to $\overline{CS_3}$, A_{16} to A_{19}		Output Output	These pins output $\overline{CS_0}$ to $\overline{CS_3}$ signals and A_{16} to A_{19}. $\overline{CS_0}$ to $\overline{CS_3}$ are chip select signals used to specify an access space. A_{16} to A_{19} are 4 high-order address bits.

Table 1.2.2. Pin description of M16C/6N group (2)

Pin name	Signal name	I/O type	Function
P5$_0$ to P5$_7$	I/O port P5	Input/output	This is an 8-bit I/O port equivalent to P0. In single-chip mode, P5$_7$ in this port outputs a divide-by-8 or divide-by-32 clock of XIN or a clock of the same frequency as XCIN as selected by software.
$\overline{WRL}$ / $\overline{WR}$, $\overline{WRH}$ / $\overline{BHE}$, $\overline{RD}$, BCLK, $\overline{HLDA}$, $\overline{HOLD}$, ALE, $\overline{RDY}$		Output Output Output Output Output Input Output Input	Output $\overline{WRL}$, $\overline{WRH}$ ($\overline{WR}$ and $\overline{BHE}$), $\overline{RD}$, BCLK, $\overline{HLDA}$, and ALE signals. $\overline{WRL}$ and $\overline{WRH}$, and $\overline{BHE}$ and $\overline{WR}$ can be switched using software control. ■ $\overline{WRL}$, $\overline{WRH}$, and $\overline{RD}$ selected With a 16-bit external data bus, data is written to even addresses when the $\overline{WRL}$ signal is "L" and to the odd addresses when the $\overline{WRH}$ signal is "L". Data is read when $\overline{RD}$ is "L". ■ $\overline{WR}$, $\overline{BHE}$, and $\overline{RD}$ selected Data is written when $\overline{WR}$ is "L". Data is read when $\overline{RD}$ is "L". Odd addresses are accessed when $\overline{BHE}$ is "L". Use this mode when using an 8-bit external data bus. While the input level at the $\overline{HOLD}$ pin is "L", the microcomputer is placed in the hold state. While in the hold state, $\overline{HLDA}$ outputs a "L" level. ALE is used to latch the address. While the input level of the $\overline{RDY}$ pin is "L", the microcomputer is in the ready state. BCLK outputs a clock with the same cycle as the internal clock ø.
P6$_0$ to P6$_7$	I/O port P6	Input/output	This is an 8-bit I/O port equivalent to P0. When used for input in single-chip, memory expansion, and microprocessor modes, the port can be set to have or not have a pull-up resistor in units of four bits by software. Pins in this port also function as UART0 and UART1 I/O pins as selected by software.
P7$_0$ to P7$_7$	I/O port P7	Input/output	This is an 8-bit I/O port equivalent to P6. Pins in this port also function as timer A0 to A3, timer B5, UART2 I/O or CAN1 transmit/receive data pins as selected by software. (Note) Port 7$_1$ has an n-channel open drain output.
P8$_0$ to P8$_4$, P8$_6$, P8$_7$,	I/O port P8	Input/output Input/output Input/output	P8$_0$ to P8$_4$, P8$_6$ and P8$_7$ are I/O ports with the same functions as P6. Using software, they can be made to function as the I/O pins for timer A4 and the input pins for external interrupts. P8$_6$ and P8$_7$ can be set using software to function as the I/O pins for a sub clock generation circuit.
P8$_5$	Input port P8$_5$	Input	In this case, connect a quartz crystal oscillator between P8$_6$ (XCOUT pin) and P8$_7$ (XCIN pin). P8$_5$ is an input-only port that also functions for $\overline{NMI}$. The $\overline{NMI}$ interrupt is generated when the input at this pin changes from "H" to "L". The $\overline{NMI}$ function cannot be cancelled using software. P85 is not equipped with a pull-up transistor.
P9$_0$ to P9$_7$	I/O port P9	Input/output	This is an 8-bit I/O port equivalent to P6. Pins in this port also function as S I/O3 I/O pins, Timer B0 to B4 input pins, D-A converter output pins, A-D converter extended input pins, A-D trigger input pins or CAN0 transmit/receive data pins as selected by software. Port 9$_1$ has an n-channel open drain output.
P10$_0$ to P10$_7$	I/O port P10	Input/output	This is an 8-bit I/O port equivalent to P6. Pins in this port also function as A-D converter input pins. Furthermore, P10$_4$ to P10$_7$ also function as input pins for the key input interrupt function.

Note: Channel CAN1 is not available for M16C/6N1 group.

Operation of Functional Blocks

The M16C/6N group accommodates several 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 CAN module, timers, serial I/O, D-A converter, DMAC, CRC calculation circuit, A-D converter, and I/O ports.

Each unit is explained in the following.

Memory

Figure 1.3.1 shows the memory map of the M16C/6N group. The address space extends the 1M bytes from address 00000$_{16}$ to FFFFF$_{16}$. The ROM area is mapped top-aligned up to FFFFF$_{16}$. The start address depends on the memory capacity of the device; e.g. 128Kbytes ROM are mapped E0000$_{16}$ up to FFFFF$_{16}$. The vector table for fixed interrupts such as the reset and $\overline{\text{NMI}}$ are mapped to FFFDC$_{16}$ to FFFFF$_{16}$. The starting addresses of the interrupt routines are 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.

The RAM area is mapped bottom-aligned starting from 00400$_{16}$. The end address depends on the RAM capacity of the device; e.g. 5Kbytes RAM are mapped to 00400$_{16}$ to 017FF$_{16}$. In addition to storing data, the RAM also stores the stack used when calling subroutines and when interrupts are generated.

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, CAN modules and timers, etc. Figure 1.6.1 to 1.6.3 show the locations of peripheral unit control registers. 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 implemented as 2-byte instructions, reducing the number of program steps.

Depending on processor mode setting, a part of the space is reserved and cannot be used. For example, in the M306NAMCT-XXXFP, the following space cannot be used.

- The space between 01800$_{16}$ and 03FFF$_{16}$ (All modes)
- The space between 04000$_{16}$ and CFFFF$_{16}$ (Single-chip mode)
- The space between D0000$_{16}$ and DFFFF$_{16}$ (Single-chip mode and memory expansion mode)

For details on how to enable usage of specific memory areas, see the section "Processor Mode".

Note 1: Cannot be used in any mode.
Note 2: Cannot be used in single-chip mode.
Note 3: Cannot be used in single-chip mode or memory expansion mode.

Type No.	Address XXXXX$_{16}$	Address YYYYY$_{16}$
M306NBMCT/FCT	017FF$_{16}$	E0000$_{16}$
M306NAMCT	017FF$_{16}$	E0000$_{16}$
M306NAMGT	02BFF$_{16}$	C0000$_{16}$
M306NAFGT	02BFF$_{16}$	C0000$_{16}$

PM13 (Note)	Address MMMMM$_{16}$	Address NNNNN$_{16}$
0	04000$_{16}$	D0000$_{16}$
1	06000$_{16}$	C0000$_{16}$

Note: PM13 (Internal reserved area expansion bit)

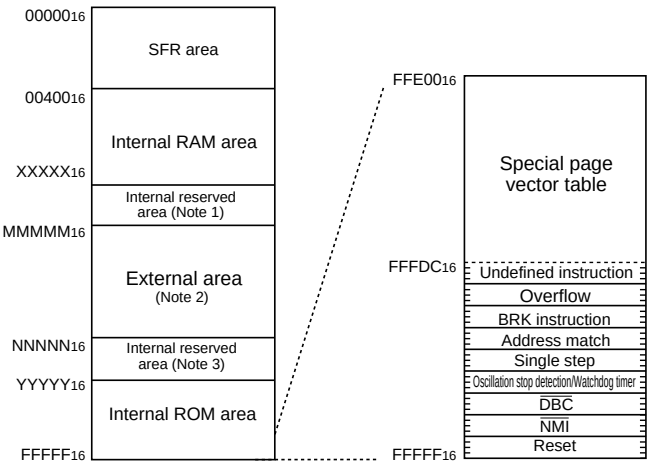


Figure 1.3.1. Memory map

Central Processing Unit (CPU)

The CPU has a total of 13 registers shown in Figure 1.4.1. Seven of these registers (R0, R1, R2, R3, A0, A1, and FB) come in two sets; therefore, these registers have two register banks.

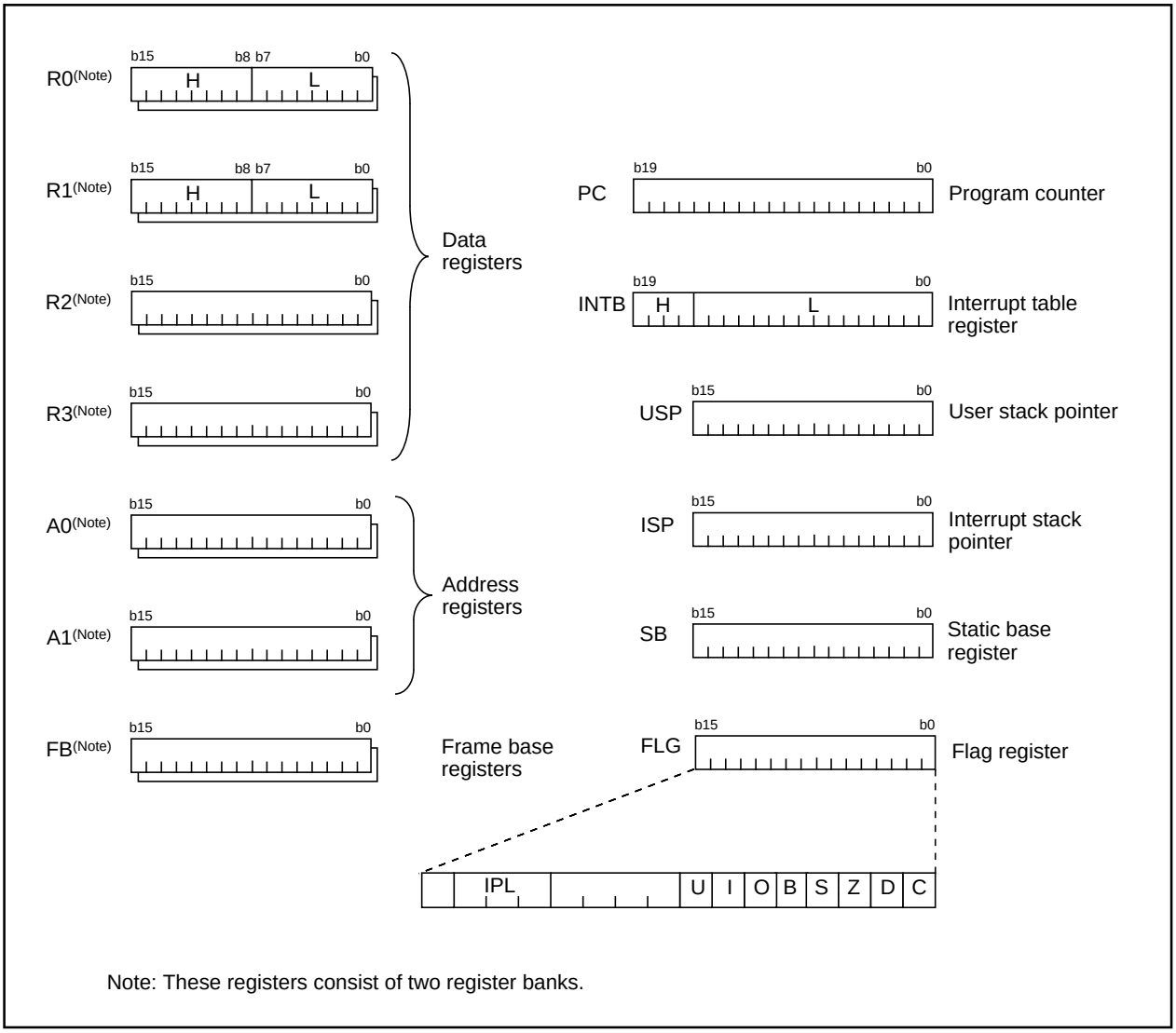


Figure 1.4.1. Central processing unit register

(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 be used as 32-bit data registers (R2R0/R3R1).

(2) Address rgisters (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)

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

(4) Program counter (PC)

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

(5) Interrupt table register (INTB)

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

(6) Stack pointer (USP/ISP)

Stack pointers come in two types: the user stack pointer (USP) and the 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)

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

(8) Flag register (FLG)

The flag register (FLG) is configured with 11 bits, each bit is used as a flag. Figure 1.4.2 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 results in 0; otherwise, cleared to "0".
- **Bit 3: Sign flag (S flag)**
  This flag is set to "1" when an arithmetic operation results 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 results 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)**
  The 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 a 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.

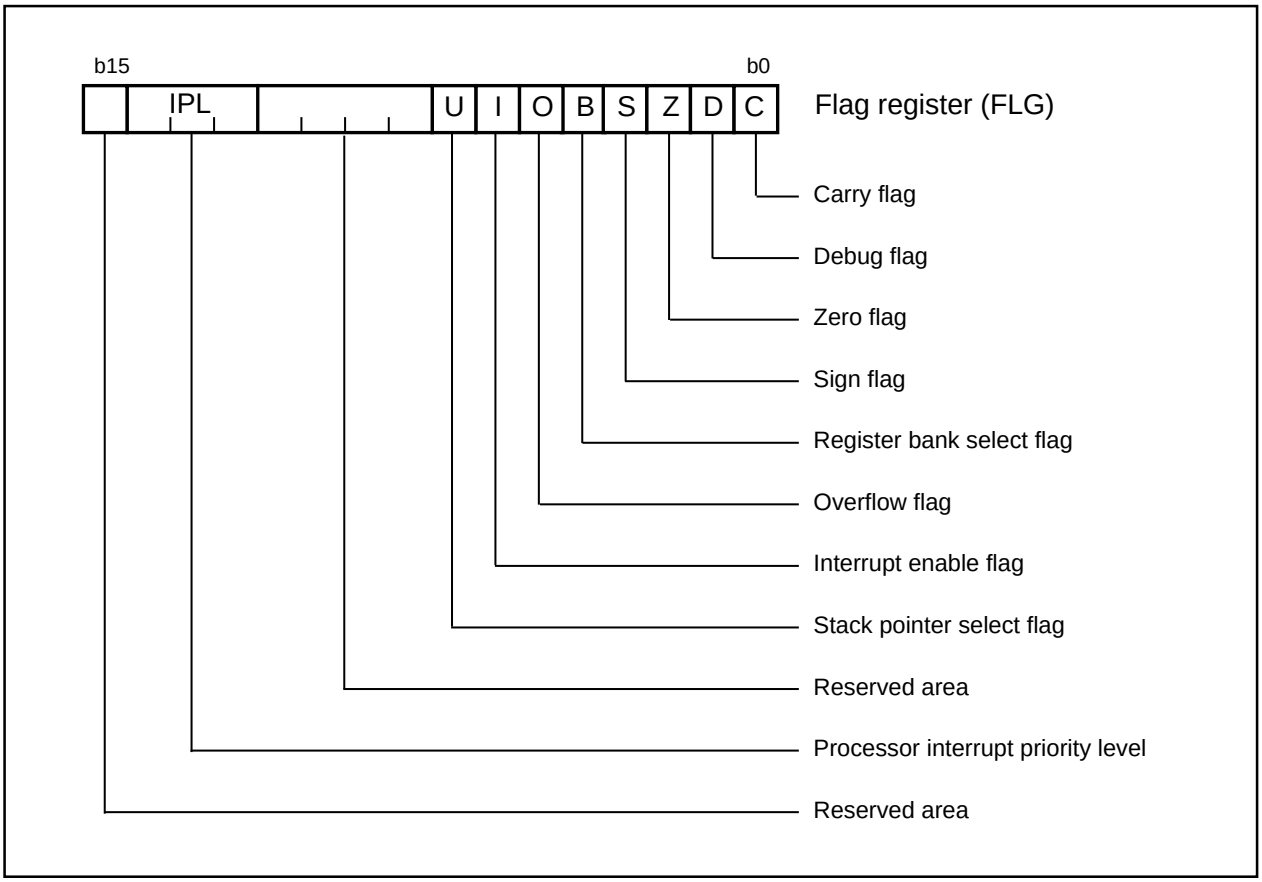


Figure 1.4.2. Flag register (FLG)

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 of f(XIN). When the reset pin level is then returned to the "H" level while main clock is stable, the reset status is released and program execution resumes from the address in the reset vector table.

Figure 1.5.1 shows the example reset circuit. Figure 1.5.2 shows the reset sequence.

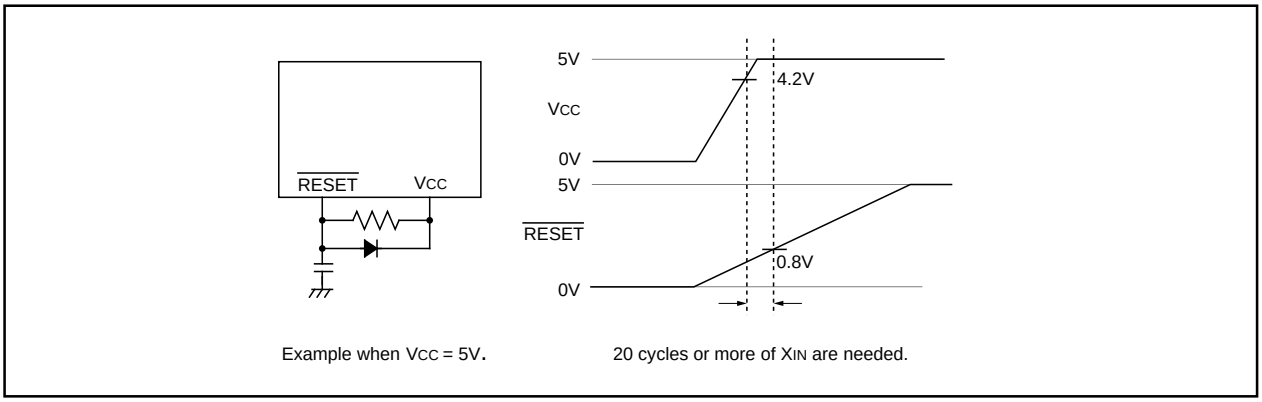


Figure 1.5.1. Example reset circuit

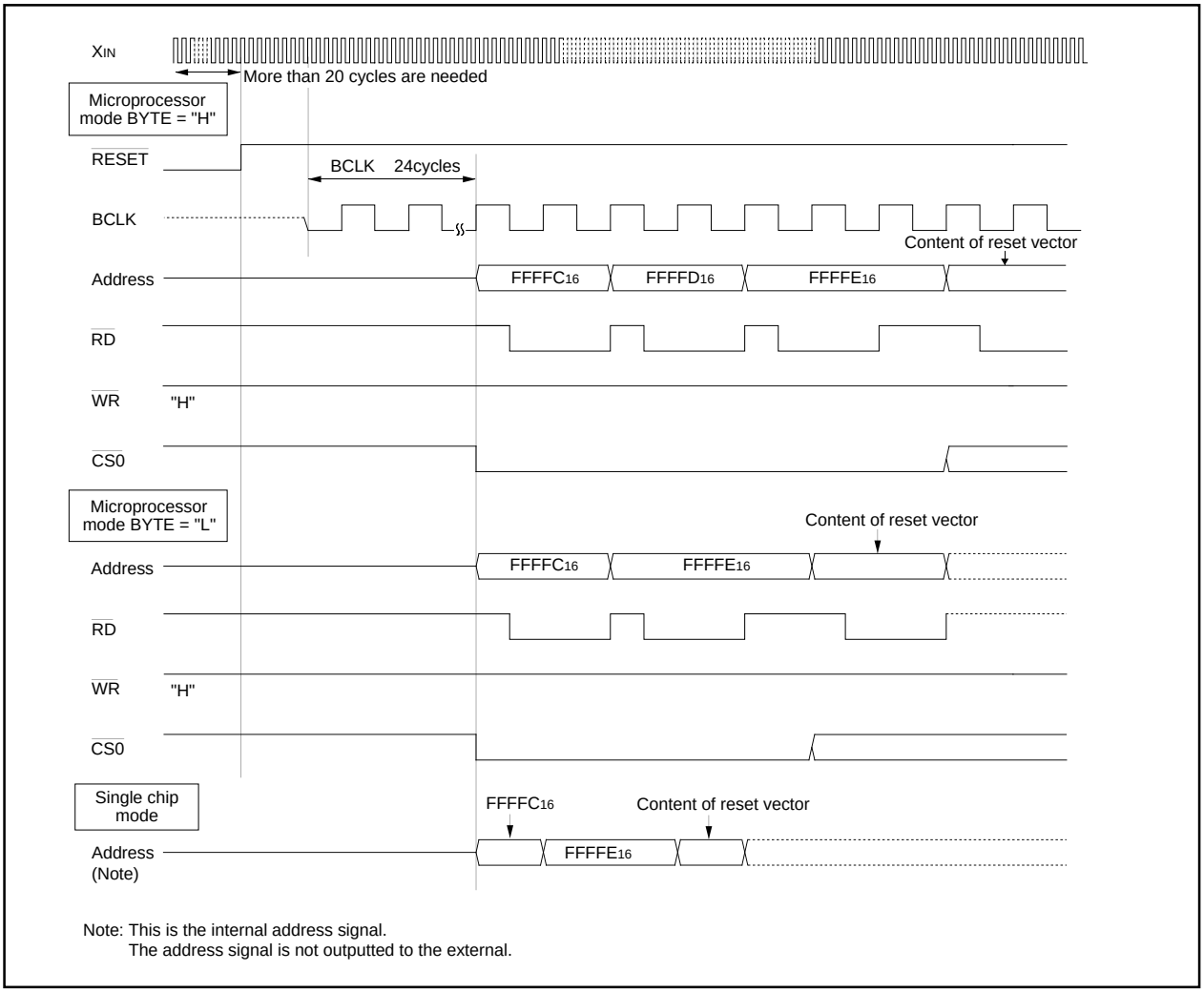
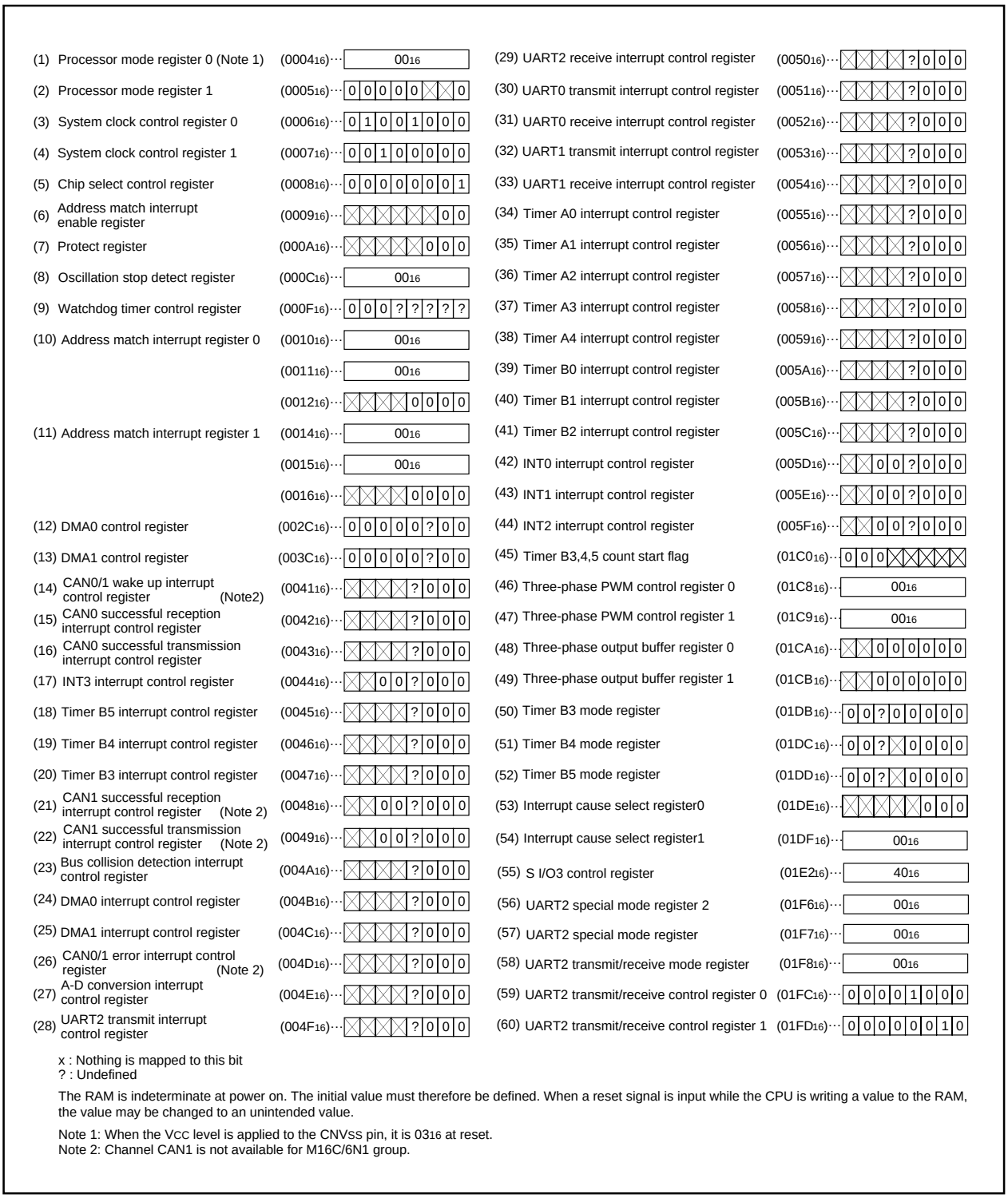


Figure 1.5.2. Reset sequence

Table 1.5.1 shows the statuses of the other pins while the $\overline{\text{RESET}}$ pin level is "L". Figures 1.5.3 to 1.5.5 show the internal status of the microcomputer immediately after the reset ha been released.

Table 1.5.1. Pin status when $\overline{\text{RESET}}$ pin level is "L"

Pin name	Status		
	CNVSS = VSS	CNVSS = VCC	
		BYTE = VSS	BYTE = VCC
P0	Input port (floating)	Data input (floating)	Data input (floating)
P1	Input port (floating)	Data input (floating)	Input port (floating)
P2, P3, P40 to P43	Input port (floating)	Address output (undefined)	Address output (undefined)
P44	Input port (floating)	$\overline{\text{CS0}}$ output ("H" level is output)	$\overline{\text{CS0}}$ output ("H" level is output)
P45 to P47	Input port (floating)	Input port (floating) (pull-up resistor is on)	Input port (floating) (pull-up resistor is on)
P50	Input port (floating)	$\overline{\text{WR}}$ output ("H" level is output)	$\overline{\text{WR}}$ output ("H" level is output)
P51	Input port (floating)	$\overline{\text{BHE}}$ output (undefined)	$\overline{\text{BHE}}$ output (undefined)
P52	Input port (floating)	$\overline{\text{RD}}$ output ("H" level is output)	$\overline{\text{RD}}$ output ("H" level is output)
P53	Input port (floating)	BCLK output	BCLK output
P54	Input port (floating)	$\overline{\text{HLDA}}$ output (The output value depends on the input to the $\overline{\text{HOLD}}$ pin)	$\overline{\text{HLDA}}$ output (The output value depends on the input to the $\overline{\text{HOLD}}$ pin)
P55	Input port (floating)	$\overline{\text{HOLD}}$ input (floating)	$\overline{\text{HOLD}}$ input (floating)
P56	Input port (floating)	ALE output ("L" level is output)	ALE output ("L" level is output)
P57	Input port (floating)	$\overline{\text{RDY}}$ input (floating)	$\overline{\text{RDY}}$ input (floating)
P6, P7, P80 to P84, P86, P87, P9, P10	Input port (floating)	Input port (floating)	Input port (floating)



x : Nothing is mapped to this bit
? : Undefined

The RAM is indeterminate at power on. The initial value must therefore be defined. When a reset signal is input while the CPU is writing a value to the RAM, the value may be changed to an unintended value.

Note 1: When the V_{CC} level is applied to the CNV_{SS} pin, it is 03_{16} at reset.
Note 2: Channel CAN1 is not available for M16C/6N1 group.

Figure 1.5.3. Device's internal status after a reset is cleared

(61) CAN0 message control register 0	(0200$_{16}$)···	00$_{16}$	(86) CAN1 message control register 0	(Note) (0220$_{16}$)···	00$_{16}$
(62) CAN0 message control register 1	(0201$_{16}$)···	00$_{16}$	(87) CAN1 message control register 1	(Note) (0221$_{16}$)···	00$_{16}$
(63) CAN0 message control register 2	(0202$_{16}$)···	00$_{16}$	(88) CAN1 message control register 2	(Note) (0222$_{16}$)···	00$_{16}$
(64) CAN0 message control register 3	(0203$_{16}$)···	00$_{16}$	(89) CAN1 message control register 3	(Note) (0223$_{16}$)···	00$_{16}$
(65) CAN0 message control register 4	(0204$_{16}$)···	00$_{16}$	(90) CAN1 message control register 4	(Note) (0224$_{16}$)···	00$_{16}$
(66) CAN0 message control register 5	(0205$_{16}$)···	00$_{16}$	(91) CAN1 message control register 5	(Note) (0225$_{16}$)···	00$_{16}$
(67) CAN0 message control register 6	(0206$_{16}$)···	00$_{16}$	(92) CAN1 message control register 6	(Note) (0226$_{16}$)···	00$_{16}$
(68) CAN0 message control register 7	(0207$_{16}$)···	00$_{16}$	(93) CAN1 message control register 7	(Note) (0227$_{16}$)···	00$_{16}$
(69) CAN0 message control register 8	(0208$_{16}$)···	00$_{16}$	(94) CAN1 message control register 8	(Note) (0228$_{16}$)···	00$_{16}$
(70) CAN0 message control register 9	(0209$_{16}$)···	00$_{16}$	(95) CAN1 message control register 9	(Note) (0229$_{16}$)···	00$_{16}$
(71) CAN0 message control register 10	(020A$_{16}$)···	00$_{16}$	(96) CAN1 message control register 10	(Note) (022A$_{16}$)···	00$_{16}$
(72) CAN0 message control register 11	(020B$_{16}$)···	00$_{16}$	(97) CAN1 message control register 11	(Note) (022B$_{16}$)···	00$_{16}$
(73) CAN0 message control register 12	(020C$_{16}$)···	00$_{16}$	(98) CAN1 message control register 12	(Note) (022C$_{16}$)···	00$_{16}$
(74) CAN0 message control register 13	(020D$_{16}$)···	00$_{16}$	(99) CAN1 message control register 13	(Note) (022D$_{16}$)···	00$_{16}$
(75) CAN0 message control register 14	(020E$_{16}$)···	00$_{16}$	(100) CAN1 message control register 14	(Note) (022E$_{16}$)···	00$_{16}$
(76) CAN0 message control register 15	(020F$_{16}$)···	00$_{16}$	(101) CAN1 message control register 15	(Note) (022F$_{16}$)···	00$_{16}$
(77) CAN0 control register	(0210$_{16}$)···	x 0 0 0 0 0 0 1	(102) CAN1 control register	(Note) (0230$_{16}$)···	x 0 0 0 0 0 0 1
	(0211$_{16}$)···	x x 0 x 0 0 0 0		(0231$_{16}$)···	x x 0 x 0 0 0 0
(78) CAN0 status register	(0212$_{16}$)···	00$_{16}$	(103) CAN1 status register	(Note) (0232$_{16}$)···	00$_{16}$
	(0213$_{16}$)···	x 0 0 0 0 0 0 1		(0233$_{16}$)···	x 0 0 0 0 0 0 1
(79) CAN0 slot status register	(0214$_{16}$)···	00$_{16}$	(104) CAN1 slot status register	(Note) (0234$_{16}$)···	00$_{16}$
	(0215$_{16}$)···	00$_{16}$		(0235$_{16}$)···	00$_{16}$
(80) CAN0 interrupt control register	(0216$_{16}$)···	00$_{16}$	(105) CAN1 interrupt control register	(Note) (0236$_{16}$)···	00$_{16}$
	(0217$_{16}$)···	00$_{16}$		(0237$_{16}$)···	00$_{16}$
(81) CAN0 extended register	(0218$_{16}$)···	00$_{16}$	(106) CAN1 extended register	(Note) (0238$_{16}$)···	00$_{16}$
	(0219$_{16}$)···	00$_{16}$		(0239$_{16}$)···	00$_{16}$
(82) CAN0 configuration register	(021A$_{16}$)···	? ? ? ? ? ? ? ?	(107) CAN1 configuration register	(Note) (023A$_{16}$)···	? ? ? ? ? ? ? ?
	(021B$_{16}$)···	? ? ? ? ? ? ? ?		(023B$_{16}$)···	? ? ? ? ? ? ? ?
(83) CAN0 receive error count register	(021C$_{16}$)···	00$_{16}$	(108) CAN1 receive error count register	(Note) (023C$_{16}$)···	00$_{16}$
(84) CAN0 transmit error count register	(021D$_{16}$)···	00$_{16}$	(109) CAN1 transmit error count register	(Note) (023D$_{16}$)···	00$_{16}$
(85) CAN0 time stamp register	(021E$_{16}$)···	00$_{16}$	(110) CAN1 time stamp register	(Note) (023E$_{16}$)···	00$_{16}$
	(021F$_{16}$)···	00$_{16}$		(023F$_{16}$)···	00$_{16}$

x : Nothing is mapped to this bit
? : Undefined

The RAM is indeterminate at power on. The initial value must therefore be defined. When a reset signal is input while the CPU is writing a value to the RAM, the value may be changed to an unintended value.

Note: Channel CAN1 is not available for M16C/6N1 group.

Figure 1.5.4. Device's internal status after a reset is cleared

(111) Peripheral function clock select register	(025E16)···	0016	(138) A-D control register 0	(03D616)···	0 0 0 0 0 ? ? ?
(112) CAN0/1 clock select register (Note 3)	(025F16)···	0016	(139) A-D control register 1	(03D716)···	0016
(113) Count start flag	(038016)···	0016	(140) D-A control register	(03DC16)···	0016
(114) Clock prescaler reset flag	(038116)···	0 x x x x x x x	(141) Port P0 direction register	(03E216)···	0016
(115) One-shot start flag	(038216)···	0 0 x 0 0 0 0 0	(142) Port P1 direction register	(03E316)···	0016
(116) Trigger select flag	(038316)···	0016	(143) Port P2 direction register	(03E616)···	0016
(117) Up-down flag	(038416)···	0016	(144) Port P3 direction register	(03E716)···	0016
(118) Timer A0 mode register	(039616)···	0016	(145) Port P4 direction register	(03EA16)···	0016
(119) Timer A1 mode register	(039716)···	0016	(146) Port P5 direction register	(03EB16)···	0016
(120) Timer A2 mode register	(039816)···	0016	(147) Port P6 direction register	(03EE16)···	0016
(121) Timer A3 mode register	(039916)···	0016	(148) Port P7 direction register	(03EF16)···	0016
(122) Timer A4 mode register	(039A16)···	0016	(149) Port P8 direction register	(03F216)···	0 0 x 0 0 0 0 0
(123) Timer B0 mode register	(039B16)···	0 0 ? 0 0 0 0 0	(150) Port P9 direction register	(03F316)···	0016
(124) Timer B1 mode register	(039C16)···	0 0 ? x 0 0 0 0	(151) Port P10 direction register	(03F616)···	0016
(125) Timer B2 mode register	(039D16)···	0 0 ? x 0 0 0 0	(152) Pull-up control register 0	(03FC16)···	0016
(126) UART0 transmit/receive mode register	(03A016)···	0016	(153) Pull-up control register 1 (Note1)	(03FD16)···	0016
(127) UART0 transmit/receive control register 0	(03A416)···	0 0 0 0 1 0 0 0	(154) Pull-up control register 2	(03FE16)···	0016
(128) UART0 transmit/receive control register 1	(03A516)···	0 0 0 0 0 0 1 0	(155) Port control register	(03FF16)···	0016
(129) UART1 transmit/receive mode register	(03A816)···	0016	(156) Data registers (R0/R1/R2/R3)		000016
(130) UART1 transmit/receive control register 0	(03AC16)···	0 0 0 0 1 0 0 0	(157) Address registers (A0/A1)		000016
(131) UART1 transmit/receive control register 1	(03AD16)···	0 0 0 0 0 0 1 0	(158) Frame base register (FB)		000016
(132) UART transmit/receive control register 2	(03B016)···	x 0 0 0 0 0 0 0	(159) Interrupt table register (INTB)		0000016
(133) Flash memory control register 1 (Note2)	(03B616)···	? ? ? ? ? 0 ? ?	(160) User stack pointer (USP)		000016
(134) Flash memory control register 0 (Note2)	(03B716)···	x x 0 0 0 0 0 1	(161) Interrupt stack pointer (ISP)		000016
(135) DMA0 cause select register	(03B816)···	0016	(162) Static base register (SB)		000016
(136) DMA1 cause select register	(03BA16)···	0016	(163) Flag register (FLG)		000016
(137) A-D control register 2	(03D416)···	x x x x 0 0 0 0			

x : Nothing is mapped to this bit
? : Undefined

The RAM is indeterminate at power on. The initial value must therefore be defined. When a reset signal is input while the CPU is writing a value to the RAM, the value may be changed to an unintended value.

Note 1: When the VCC level is applied to the CNVSS pin, it is 0216 at a reset.
Note 2: These registers are available on the flash version only.
Note 3: Channel CAN1 is not available for M16C/6N1 group.

Figure 1.5.5. Device's internal status after a reset is cleared

0000–0003	
0004	Processor mode register 0 (PM0)
0005	Processor mode register 1 (PM1)
0006	System clock control register 0 (CM0)
0007	System clock control register 1 (CM1)
0008	Chip select control register (CSR)
0009	Address match interrupt enable register (AIER)
000A	Protect register (PRCR)
000B	
000C	Oscillation stop detection register (CM2)
000D	
000E	Watchdog timer start register (WDTS)
000F	Watchdog timer control register (WDC)
0010–0012	Address match interrupt register 0 (RMAD0)
0013	
0014–0016	Address match interrupt register 1 (RMAD1)
0017–001F	
0020–0022	DMA0 source pointer (SAR0)
0023	
0024–0026	DMA0 destination pointer (DAR0)
0027	
0028–0029	DMA0 transfer counter (TCR0)
002A–002B	
002C	DMA0 control register (DM0CON)
002D–002F	
0030–0032	DMA1 source pointer (SAR1)
0033	
0034–0036	DMA1 destination pointer (DAR1)
0037	
0038–0039	DMA1 transfer counter (TCR1)
003A–003B	
003C	DMA1 control register (DM1CON)
003D–0040	
0041	CAN0/1 Wake Up interrupt control register (C01WKIC) (Note1)
0042	CAN0 successful reception interrupt control register (C0RECIC)
0043	CAN0 successful transmission interrupt control register (C0TRMIC)
0044	INT3 interrupt control register (INT3IC)
0045	Timer B5 interrupt control register (TB5IC)
0046	Timer B4 interrupt control register (TB4IC)
0047	Timer B3 interrupt control register (TB3IC)
0048	CAN1 successful reception interrupt control register (C1RECIC) INT5 interrupt control register (INT5IC) (Note1)
0049	CAN1 successful transmission interrupt control register (C1TRMIC) SIO3 interrupt control register (S3IC) INT4 interrupt control register (INT4IC) (Note1)
004A	Bus collision detection interrupt control register (BCNIC)
004B	DMA0 interrupt control register (DM0IC)
004C	DMA1 interrupt control register (DM1IC)
004D	CAN0/1 error interrupt control register (C01ERRIC) (Note1)
004E	A-D conversion interrupt control register (ADIC) Key input interrupt control register (KUPIC)
004F	UART2 transmit interrupt control register (S2TIC)
0050	UART2 receive interrupt control register (S2RIC)
0051	UART0 transmit interrupt control register (S0TIC)
0052	UART0 receive interrupt control register (S0RIC)
0053	UART1 transmit interrupt control register (S1TIC)
0054	UART1 receive interrupt control register (S1RIC)
0055	Timer A0 interrupt control register (TA0IC)
0056	Timer A1 interrupt control register (TA1IC)
0057	Timer A2 interrupt control register (TA2IC)
0058	Timer A3 interrupt control register (TA3IC)
0059	Timer A4 interrupt control register (TA4IC)
005A	Timer B0 interrupt control register (TB0IC)
005B	Timer B1 interrupt control register (TB1IC)
005C	Timer B2 interrupt control register (TB2IC)
005D	INT0 interrupt control register (INT0IC)
005E	INT1 interrupt control register (INT1IC)
005F	INT2 interrupt control register (INT2IC)
0060–006F	CAN0 message box 0: Identifier / DLC, Data Field, Time Stamp
0070–007F	CAN0 message box 1: Identifier / DLC, Data Field, Time Stamp
0080–008F	CAN0 message box 2: Identifier / DLC, Data Field, Time Stamp
0090–009F	CAN0 message box 3: Identifier / DLC, Data Field, Time Stamp
00A0–00AF	CAN0 message box 4: Identifier / DLC, Data Field, Time Stamp
00B0–00BF	CAN0 message box 5: Identifier / DLC, Data Field, Time Stamp
00C0–00CF	CAN0 message box 6: Identifier / DLC, Data Field, Time Stamp
00D0–00DF	CAN0 message box 7: Identifier / DLC, Data Field, Time Stamp
00E0–00EF	CAN0 message box 8: Identifier / DLC, Data Field, Time Stamp
00F0–00FF	CAN0 message box 9: Identifier / DLC, Data Field, Time Stamp
0100–010F	CAN0 message box 10: Identifier / DLC, Data Field, Time Stamp
0110–011F	CAN0 message box 11: Identifier / DLC, Data Field, Time Stamp
0120–012F	CAN0 message box 12: Identifier / DLC, Data Field, Time Stamp
0130–013F	CAN0 message box 13: Identifier / DLC, Data Field, Time Stamp
0140–014F	CAN0 message box 14: Identifier / DLC, Data Field, Time Stamp
0150–015F	CAN0 message box 15: Identifier / DLC, Data Field, Time Stamp
0160–0165	CAN0 global mask (C0GMR)
0166–016B	CAN0 local mask A (C0LMAR)
016C–0171	CAN0 local mask B (C0LMBR)
0172–01BF	
01C0	Timer B3, 4, 5 count start flag (TBSR)
01C1	
01C2–01C3	Timer A1-1 register (TA11)
01C4–01C5	Timer A2-1 register (TA21)
01C6–01C7	Timer A4-1 register (TA41)
01C8	Three-phase PWM control register 0 (INVC0)
01C9	Three-phase PWM control register 1 (INVC1)
01CA	Three-phase output buffer register 0 (IDB0)
01CB	Three-phase output buffer register 1 (IDB1)
01CC	Dead time timer (DTT)
01CD	Timer B2 interrupt occurrence frequency set counter (ICTB2)

Note 1: Channel CAN1 is not available for M16C/6N1 group.
Note 2: Locations in the SFR area where nothing is allocated are reserved areas. Do not access these areas for read or write.

Figure 1.6.1. Location of peripheral unit control registers (1)

01CE16		
01CF16		
01D016	Timer B3 register (TB3)	
01D116		
01D216	Timer B4 register (TB4)	
01D316		
01D416	Timer B5 register (TB5)	
01D516		
01D616		
01DA16		
01DB16	Timer B3 mode register (TB3MR)	
01DC16	Timer B4 mode register (TB4MR)	
01DD16	Timer B5 mode register (TB5MR)	
01DE16	Interrupt cause select register 0 (IFSR0)	
01DF16	Interrupt cause select register 1 (IFSR1)	
01E016	S I/O3 transmit/receive register (S3TRR)	
01E116		
01E216	S I/O3 control register (S3C)	
01E316	S I/O3 bit rate generator (S3BRG)	
01E416		
01F516		
01F616	UART2 special mode register 2 (U2SMR2)	
01F716	UART2 special mode register (U2SMR)	
01F816	UART2 transmit/receive mode register (U2MR)	
01F916	UART2 bit rate generator (U2BRG)	
01FA16	UART2 transmit buffer register (U2TB)	
01FB16		
01FC16	UART2 transmit/receive mode register 0 (U2C0)	
01FD16	UART2 transmit/receive mode register 1 (U2C1)	
01FE16	UART2 receive buffer register (U2RB)	
01FF16		
020016	CAN0 message control register 0 (C0MCTL0)	
020116	CAN0 message control register 1 (C0MCTL1)	
020216	CAN0 message control register 2 (C0MCTL2)	
020316	CAN0 message control register 3 (C0MCTL3)	
020416	CAN0 message control register 4 (C0MCTL4)	
020516	CAN0 message control register 5 (C0MCTL5)	
020616	CAN0 message control register 6 (C0MCTL6)	
020716	CAN0 message control register 7 (C0MCTL7)	
020816	CAN0 message control register 8 (C0MCTL8)	
020916	CAN0 message control register 9 (C0MCTL9)	
020A16	CAN0 message control register 10 (C0MCTL10)	
020B16	CAN0 message control register 11 (C0MCTL11)	
020C16	CAN0 message control register 12 (C0MCTL12)	
020D16	CAN0 message control register 13 (C0MCTL13)	
020E16	CAN0 message control register 14 (C0MCTL14)	
020F16	CAN0 message control register 15 (C0MCTL15)	
021016	CAN0 control register (C0CTLR)	
021116		
021216	CAN0 status register (C0STR)	
021316		
021416	CAN0 slot status register (C0SSTR)	
021516		
021616	CAN0 interrupt control register (C0ICR)	
021716		
021816	CAN0 extended register (C0IDR)	
021916		
021A16	CAN0 configuration register (C0CONR)	
021B16		
021C16	CAN0 receive error count register (C0RECR)	
021D16	CAN0 transmit error count register (C0TECR)	
021E16	CAN0 time stamp register (C0STR)	
021F16		
022016	CAN1 message control register 0 (C1MCTL0)	(Note1)
022116	CAN1 message control register 1 (C1MCTL1)	(Note1)
022216	CAN1 message control register 2 (C1MCTL2)	(Note1)
022316	CAN1 message control register 3 (C1MCTL3)	(Note1)
022416	CAN1 message control register 4 (C1MCTL4)	(Note1)
022516	CAN1 message control register 5 (C1MCTL5)	(Note1)
022616	CAN1 message control register 6 (C1MCTL6)	(Note1)
022716	CAN1 message control register 7 (C1MCTL7)	(Note1)
022816	CAN1 message control register 8 (C1MCTL8)	(Note1)
022916	CAN1 message control register 9 (C1MCTL9)	(Note1)
022A16	CAN1 message control register 10 (C1MCTL10)	(Note1)
022B16	CAN1 message control register 11 (C1MCTL11)	(Note1)
022C16	CAN1 message control register 12 (C1MCTL12)	(Note1)
022D16	CAN1 message control register 13 (C1MCTL13)	(Note1)
022E16	CAN1 message control register 14 (C1MCTL14)	(Note1)
022F16	CAN1 message control register 15 (C1MCTL15)	(Note1)
023016	CAN1 control register (C1CTLR)	(Note1)
023116		
023216	CAN1 status register (C1STR)	(Note1)
023316		
023416	CAN1 slot status register (C1SSTR)	(Note1)
023516		
023616	CAN1 interrupt control register (C1SICR)	(Note1)
023716		
023816	CAN1 extended register (C1IDR)	(Note1)
023916		
023A16	CAN1 configuration register (C1CONR)	(Note1)
023B16		
023C16	CAN1 receive error count register (C1RECR)	(Note1)
023D16	CAN1 transmit error count register (C1TECR)	(Note1)
023E16	CAN1 time stamp register (C1STR)	(Note1)
023F16		
024016		
024116		
024216	CAN0 acceptance filter support register (C0AFS)	
024316		
024416	CAN1 acceptance filter support register (C1AFS)	(Note1)
024516		
024616		
025D16		
025E16	Peripheral function clock select register (PCLKR)	
025F16	CAN0/1 clock select register (CCLKR)	(Note1)
026016	CAN1 message box 0: Identifier / DLC, Data Field, Time Stamp	(Note1)
026F16		
027016	CAN1 message box 1: Identifier / DLC, Data Field, Time Stamp	(Note1)
027F16		
028016	CAN1 message box 2: Identifier / DLC, Data Field, Time Stamp	(Note1)
028F16		
029016	CAN1 message box 3: Identifier / DLC, Data Field, Time Stamp	(Note1)
029F16		
02A016	CAN1 message box 4: Identifier / DLC, Data Field, Time Stamp	(Note1)
02AF16		
02B016	CAN1 message box 5: Identifier / DLC, Data Field, Time Stamp	(Note1)
02BF16		
02C016	CAN1 message box 6: Identifier / DLC, Data Field, Time Stamp	(Note1)
02CF16		
02D016	CAN1 message box 7: Identifier / DLC, Data Field, Time Stamp	(Note1)
02DF16		
02E016	CAN1 message box 8: Identifier / DLC, Data Field, Time Stamp	(Note1)
02EF16		
02F016	CAN1 message box 9: Identifier / DLC, Data Field, Time Stamp	(Note1)
02FF16		
030016	CAN1 message box 10: Identifier / DLC, Data Field, Time Stamp	(Note1)
030F16		
031016	CAN1 message box 11: Identifier / DLC, Data Field, Time Stamp	(Note1)
031F16		
032016	CAN1 message box 12: Identifier / DLC, Data Field, Time Stamp	(Note1)
032F16		
033016	CAN1 message box 13: Identifier / DLC, Data Field, Time Stamp	(Note1)
033F16		
034016	CAN1 message box 14: Identifier / DLC, Data Field, Time Stamp	(Note1)
034F16		
035016	CAN1 message box 15: Identifier / DLC, Data Field, Time Stamp	(Note1)
035F16		

Note 1: Channel CAN1 is not available for M16C/6N1 group.
Note 2: Locations in the SFR area where nothing is allocated are reserved areas. Do not access these areas for read or write.

Figure 1.6.2. Location of peripheral unit control registers (2)

0360₁₆	CAN1 global mask (C1GMR)	(Note1)
0365₁₆		
0366₁₆	CAN1 local mask A (C1LMAR)	(Note1)
036B₁₆		
036C₁₆	CAN1 local mask B (C1LMBR)	(Note1)
0371₁₆		
0372₁₆		
037F₁₆		
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 register (TA0)	
0387₁₆		
0388₁₆	Timer A1 register (TA1)	
0389₁₆		
038A₁₆	Timer A2 register (TA2)	
038B₁₆		
038C₁₆	Timer A3 register (TA3)	
038D₁₆		
038E₁₆	Timer A4 register (TA4)	
038F₁₆		
0390₁₆	Timer B0 register (TB0)	
0391₁₆		
0392₁₆	Timer B1 register (TB1)	
0393₁₆		
0394₁₆	Timer B2 register (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		
03B5₁₆		
03B6₁₆	Flash memory control register 1 (FMR1)	(Note2)
03B7₁₆	Flash memory control register 0 (FMR0)	(Note2)
03B8₁₆	DMA0 cause select register (DM0SL)	
03B9₁₆		
03BA₁₆	DMA1 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	
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	
03DF₁₆	
03E0₁₆	Port P0 register (P0)
03E1₁₆	Port P1 register (P1)
03E2₁₆	Port P0 direction register (PD0)
03E3₁₆	Port P1 direction register (PD1)
03E4₁₆	Port P2 register (P2)
03E5₁₆	Port P3 register (P3)
03E6₁₆	Port P2 direction register (PD2)
03E7₁₆	Port P3 direction register (PD3)
03E8₁₆	Port P4 register (P4)
03E9₁₆	Port P5 register (P5)
03EA₁₆	Port P4 direction register (PD4)
03EB₁₆	Port P5 direction register (PD5)
03EC₁₆	Port P6 register (P6)
03ED₁₆	Port P7 register (P7)
03EE₁₆	Port P6 direction register (PD6)
03EF₁₆	Port P7 direction register (PD7)
03F0₁₆	Port P8 register (P8)
03F1₁₆	Port P9 register (P9)
03F2₁₆	Port P8 direction register (PD8)
03F3₁₆	Port P9 direction register (PD9)
03F4₁₆	Port P10 register (P10)
03F5₁₆	
03F6₁₆	Port P10 direction register (PD10)
03F7₁₆	
03FB₁₆	
03FC₁₆	Pull-up control register 0 (PUR0)
03FD₁₆	Pull-up control register 1 (PUR1)
03FE₁₆	Pull-up control register 2 (PUR2)
03FF₁₆	Port control register (PCR)

Note 1: Channel CAN1 is not available for M16C/6N1 group.
Note 2: These registers are available on the flash version only.
Note 3: Locations in the SFR area where nothing is allocated are reserved areas. Do not access these areas for read or write.

Figure 1.6.3. Location of peripheral unit control registers (3)

Software Reset

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

Processor Mode

(1) Processor mode types

One of three processor modes can be selected: single-chip mode, memory expansion mode and microprocessor mode. The functions of some pins, the memory map and the access space differ according to the selected processor mode.

- **Single-chip mode**

  In single-chip mode, only internal memory space (SFR, internal RAM, and internal ROM) can be accessed. Ports P0 to P10 can be used as programmable I/O ports or as I/O ports for the internal peripheral functions.

- **Memory expansion mode**

  In memory expansion mode, external memory can be accessed in addition to the internal memory space (SFR, internal RAM, and internal ROM).

  In this mode, some of the pins function as the address bus, the data bus, and as control signals. The number of pins assigned to these functions depends on the bus width and register settings. (See "Bus Settings" for details.)

- **Microprocessor mode**

  In microprocessor mode, the SFR, internal RAM, and external memory space can be accessed. The internal ROM area cannot be accessed.

  In this mode, some of the pins function as the address bus, the data bus, and as control signals. The number of pins assigned to these functions depends on the bus width and register settings. (See "Bus Settings" for details.)

(2) Setting processor modes

The processor mode is set using the CNV$_{SS}$ pin and the processor mode bits (bits 1 and 0 at address 0004_{16}). Do not set the processor mode bits to "10_2".

Regardless of the level of the CNV$_{SS}$ pin, changing the processor mode bits selects the mode. Therefore, never change the processor mode bits when changing the contents of other bits. Do not change the processor mode bits simultaneously with other bits when changing the processor mode bits "01_2" or "11_2". Change the processor mode bits after changing the other bits. Also do not attempt to shift to or from the microprocessor mode within the program stored in the internal ROM area.

- **Applying Vss to CNVss pin**

  The microcomputer begins operation in single-chip mode after being reset. Memory expansion mode is selected by writing "01_2" to the processor mode bits.

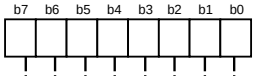
- **Applying Vcc to CNVss pin**

  The microcomputer starts to operate in microprocessor mode after being reset.

Figure 1.7.1 shows the processor mode register 0 and 1.

Figure 1.7.2 shows the memory maps applicable for each of the modes when memory area dose not be expanded (normal mode).

Processor mode register 0 (Note 1)

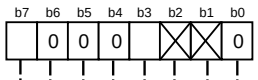


Symbol	Address	When reset
PM0	0004_{16}	00_{16} (Note 2)

Bit symbol	Bit name	Function	R	W
PM00	Processor mode bit	b1 b0 0 0: Single-chip mode 0 1: Memory expansion mode 1 0: Inhibited 1 1: Microprocessor mode	○	○
PM01			○	○
PM02	R/W mode select bit	0 : $\overline{RD}$,$\overline{BHE}$,$\overline{WR}$ 1 : $\overline{RD}$,$\overline{WRH}$,$\overline{WRL}$	○	○
PM03	Software reset bit	The device is reset when this bit is set to "1". The value of this bit is "0" when read.	○	○
PM04	Multiplexed bus space select bit	b5 b4 0 0 : Multiplexed bus is not used 0 1 : Allocated to CS2 space 1 0 : Allocated to CS1 space 1 1 : Allocated to entire space (Note4)	○	○
PM05			○	○
PM06	Port $P4_0$ to $P4_3$ function select bit (Note 3)	0 : Address output 1 : Port function (Address is not output)	○	○
PM07	BCLK output disable bit (Note 3)	0 : BCLK is output 1 : BCLK is not output (Pin is left floating)	○	○

Note 1: Set bit 1 of the protect register (address $000A_{16}$) to "1" when writing new values to this register.
Note 2: If the V_{CC} voltage is applied to the CNV_{SS}, the value of this register when reset is 03_{16}. (PM00 and PM01 both are set to "1".)
Note 3: Valid in microprocessor and memory expansion modes. In single-chip mode, ports $P4_0$ to $P4_3$ are not used for address output and BCLK is not output.
Note 4: If the entire space is of multiplexed bus in memory expansion mode, choose an 8-bit width.The processor operates using the separate bus after reset is revoked, space multiplexed bus cannot be chosen in microprocessor mode.
The higher-order address becomes a port if the entire space multiplexed bus is chosen, so only 256 bytes can be used in each chip select.

Processor mode register 1 (Note 1)



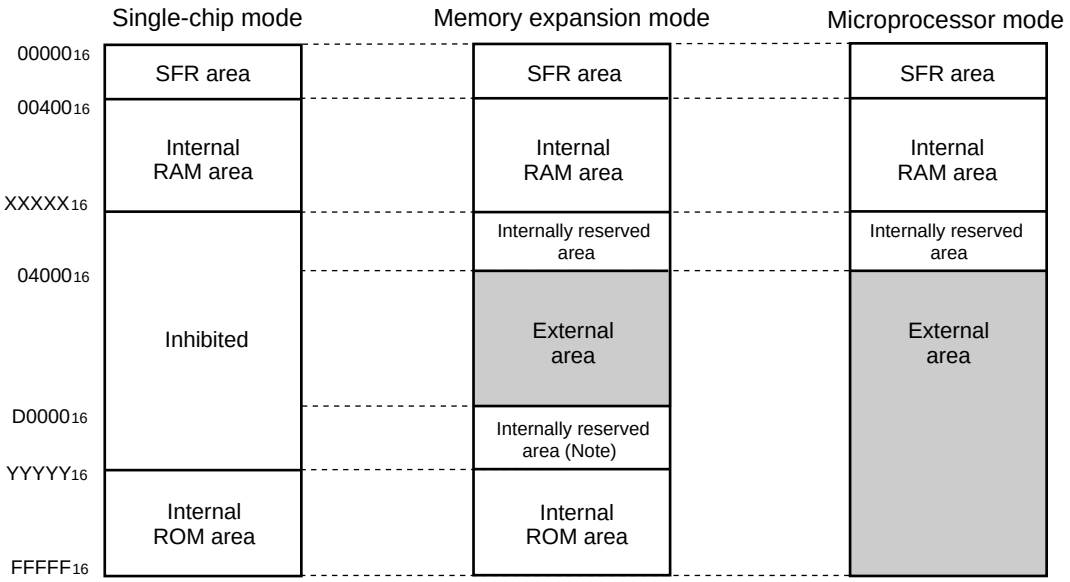
Symbol	Address	When reset
PM1	0005_{16}	$00000XX0_2$

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.			—	—
PM13	Internal reserved area expansion bit (Note 2)	0: The same internal reserved area as that of M16C/60, M16C/61 and M16C/62 group 1: Expands the internal ROM area up to 256K bytes respectively. (Note 2)	○	○
Reserved bits		Must always be set to "0"	○	○
PM17	Wait bit	0 : No wait state 1 : Wait state inserted	○	○

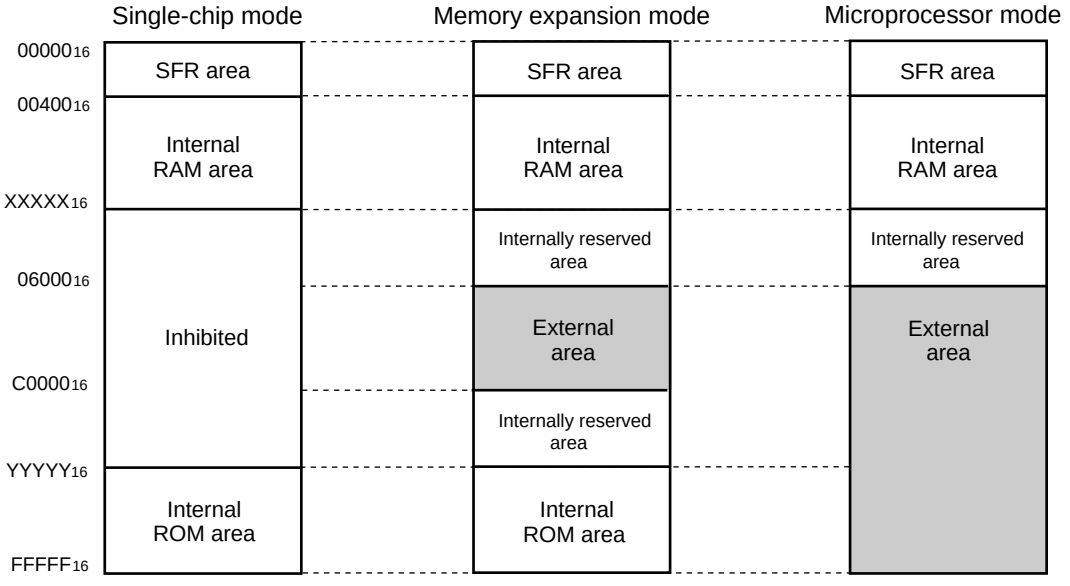
Note 1: Set bit 1 of the protect register (address $000A_{16}$) to "1" when writing new values to this register.
Note 2: Be sure to set this bit to 0 except for products whose ROM size exceeds 192K bytes. Set this bit to "1" for M306NAMG and M306NAFG.
Specify $D0000_{16}$ or a subsequent address as a reset address in the fixed vector table.

Figure 1.7.1. Processor mode register 0 and 1

Memory maps in each processor mode (Internal reserved area expansion bit PM13 is "0")



Memory maps in each processor mode (Internal reserved area expansion bit PM13 is "1")



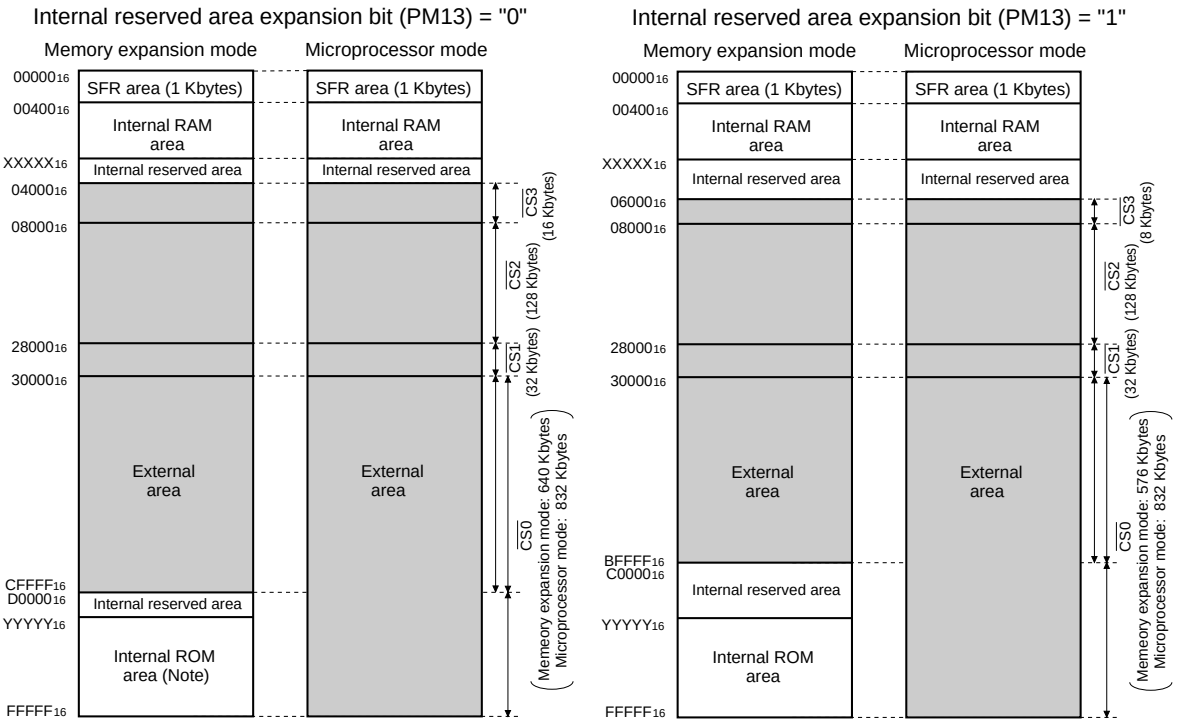
External area: Accessing this area allows the user to access a device connected externally to the microcomputer.

Type No.	Address XXXXX$_{16}$		Address YYYYY$_{16}$	
	PM13 = 0	PM13 = 1	PM13 = 0	PM13 = 1
M306NBMCT/FCT	017FF$_{16}$		E0000$_{16}$	
M306NAMCT	017FF$_{16}$		E0000$_{16}$	
M306NAMGT	02BFF$_{16}$		D0000$_{16}$	C0000$_{16}$
M306NAFGT	02BFF$_{16}$		D0000$_{16}$	C0000$_{16}$

Note: When YYYYY$_{16}$ is lower than D0000$_{16}$ (products with more than 192 Kbytes of ROM), internal ROM in the range of YYYYY$_{16}$ to CFFFF$_{16}$ cannot be accessed when PM13 is "0".

Figure 1.7.2. Memory maps in each processor mode

Processor Mode



External area : Accessing this area allows the user to access a device connected externally to the microcomputer.

Type No.	Address XXXXX₁₆		Address YYYYY₁₆	
	PM13 = 0	PM13 = 1	PM13 = 0	PM13 = 1
M306NBMCT/FCT	017FF₁₆		E0000₁₆	
M306NAMCT	017FF₁₆		E0000₁₆	
M306NAMGT	02BFF₁₆		D0000₁₆	C0000₁₆
M306NAFGT	02BFF₁₆		D0000₁₆	C0000₁₆

Note: When YYYYY₁₆ is lower than D0000₁₆ (products with more than 192 Kbytes of ROM), internal ROM in the range of YYYYY₁₆ to CFFFF₁₆ cannot be accessed.

	Chip select	PM13	Address range
Memory expansion mode	$\overline{CS0}$	0	30000₁₆ to CFFFF₁₆ (640Kbytes)
		1	30000₁₆ to BFFFF₁₆ (576Kbytes)
	$\overline{CS1}$	—	28000₁₆ to 2FFFF₁₆ (32Kbytes)
	$\overline{CS2}$	—	08000₁₆ to 27FFF₁₆ (128Kbytes)
	$\overline{CS3}$	0	04000₁₆ to 07FFF₁₆ (16Kbytes)
		1	06000₁₆ to 07FFF₁₆ (8Kbytes)
Microprocessor mode	$\overline{CS0}$	0	30000₁₆ to FFFFF₁₆ (832Kbytes)
		1	30000₁₆ to FFFFF₁₆ (832Kbytes)
	$\overline{CS1}$	—	28000₁₆ to 2FFFF₁₆ (32Kbytes)
	$\overline{CS2}$	—	08000₁₆ to 27FFF₁₆ (128Kbytes)
	$\overline{CS3}$	0	04000₁₆ to 07FFF₁₆ (16Kbytes)
		1	06000₁₆ to 07FFF₁₆ (8Kbytes)

Figure 1.7.3. Memory location and chip select area in each processor mode

Internal Reserved Area Expansion Bit (PM13)

This bit expands the internal RAM area and the internal ROM area, and changes the chip select area. In M306NAMGT/FGT, for example, to set this bit to "1" expandes the internal ROM area to 256 Kbytes respectively. Refer to Figure 1.7.2 and 1.7.3 for the chip select area. When the reset is revoked, this bit is set to "0". To expand the internal area, set this bit to "1" in user program. And the top of user program must be allocated to D0000₁₆ or subsequent address.

In the case of the product in which the internal ROM is 192 Kbytes or less set this bit to "0" when this product is used in the memory expansion mode or the microprocessor mode. When the product is used in the single chip mode, the internal area is not expanded and any action is not affected, even if this bit is set to "1".

Bus Settings

The BYTE pin and bits 4 to 6 of the processor mode register 0 (address 0004_{16}) are used to change the bus settings.

Table 1.8.1 shows the factors used to change the bus settings.

Table 1.8.1. Factors for switching bus settings

Bus setting	Switching factor
Switching external address bus width	Bit6 of processor mode register0
Switching external data bus width	BYTE pin
Switching between separate and multiplex bus	Bits4 and 5 of processor mode register0

(1) Selecting external address bus width

The address bus width for external output in the 1M bytes of address space can be set to 16 bits (64K bytes address space) or 20 bits (1M bytes address space). When bit 6 of the processor mode register 0 is set to "1", the external address bus width is set to 16 bits, and P2 and P3 become part of the address bus. $P4_0$ to $P4_3$ can be used as programmable I/O ports. When bit 6 of processor mode register 0 is set to "0", the external address bus width is set to 20 bits, and P2, P3, and $P4_0$ to $P4_3$ become part of the address bus.

(2) Selecting external data bus width

The external data bus width can be set to 8 or 16 bits. (Note, however, that only the separate bus can be set.) When the BYTE pin is "L", the bus width is set to 16 bits; when "H", it is set to 8 bits. (The internal bus width is permanently set to 16 bits.) While operating, fix the BYTE pin either to "H" or to "L".

(3) Selecting separate/multiplex bus

The bus format can be set to multiplex or separate bus using bits 4 and 5 of the processor mode register 0.

- **Separate bus**

  In this mode, the data and address are input and output separately. The data bus can be set using the BYTE pin to be 8 or 16 bits. When the BYTE pin is "H", the data bus is set to 8 bits and P0 functions as the data bus and P1 as a programmable I/O port. When the BYTE pin is "L", the data bus is set to 16 bits and P0 and P1 are both used for the data bus.

  When the separate bus is used for access, a software wait can be selected.

- **Multiplex bus**

  In this mode, data and address I/O are time multiplexed. With an 8-bit data bus selected (BYTE pin = "H"), the 8 bits from D_0 to D_7 are multiplexed with A_0 to A_7.

  With a 16-bit data bus selected (BYTE pin = "L"), the 8 bits from D_0 to D_7 are multiplexed with A_1 to A_8. D_8 to D_{15} are not multiplexed. In this case, the external devices connected to the multiplexed bus are mapped to the microcomputer's even addresses (every 2nd address). To access these external devices, access the even addresses as bytes.

  The ALE signal latches the address. It is output from $P5_6$.

  Before using the multiplex bus for access, always set the $\overline{CSi}$ wait bit of the chip select control register to "0".

  In microprocessor mode, multiplexed bus for the entire space cannot be selected.

  In memory expansion mode, when multiplexed bus for the entire space is selected, address bus range is 256 bytes in each chip select.

Table 1.8.2. Pin functions for each processor mode

Processor mode	Single-chip mode	Memory expansion mode/microprocessor modes				Memory expansion mode
External bus type		Multiplexed bus and separate bus		separate bus		Multiplexed bus (Note 1)
Multiplexed bus space select bit		"01", "10" Either CS1 or CS2 is for multiplexed bus and others are for separate bus.		"00" Separate bus		"11" (Note 2) Multiplexed bus for the entire space
Data bus width BYTE pin level		8 bits = "H"	16 bits = "L"	8 bits = "H"	16 bits = "L"	8 bits = "H"
$P0_0$ to $P0_7$	I/O port	Data bus	Data bus	Data bus	Data bus	I/O port (Note 3)
$P1_0$ to $P1_7$	I/O port	I/O port	Data bus	I/O port	Data bus	I/O port
$P2_0$	I/O port	Address bus /data bus(Note 4)	Address bus	Address bus	Address bus	Address bus /data bus
$P2_1$ to $P2_7$	I/O port	Address bus /data bus(Note 4)	Address bus /data bus (Note 4)	Address bus	Address bus	Address bus /data bus
$P3_0$	I/O port	Address bus	Address bus /data bus (Note 4)	Address bus	Address bus	A_8/D_7
$P3_1$ to $P3_7$	I/O port	Address bus	Address bus	Address bus	Address bus	I/O port
$P4_0$ to $P4_3$ Port $P4_0$ to $P4_3$ function select bit = "1"	I/O port	I/O port	I/O port	I/O port	I/O port	I/O port
$P4_0$ to $P4_3$ Port $P4_0$ to $P4_3$ function select bit = "0"	I/O port	Address bus	Address bus	Address bus	Address bus	I/O port
$P4_4$ to $P4_7$	I/O port	$\overline{CS}$ (chip select) or programmable I/O port (For details, refer to "Bus control")				
$P5_0$ to $P5_3$	I/O port	Outputs $\overline{RD}$, $\overline{WRL}$, $\overline{WRH}$, and BCLK or $\overline{RD}$, $\overline{BHE}$, $\overline{WR}$, and BCLK (For details, refer to "Bus control")				
$P5_4$	I/O port	$\overline{HLDA}$	$\overline{HLDA}$	$\overline{HLDA}$	$\overline{HLDA}$	$\overline{HLDA}$
$P5_5$	I/O port	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$
$P5_6$	I/O port	ALE	ALE	ALE	ALE	ALE
$P5_7$	I/O port	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$

Note 1: In memory expansion mode, do not select a 16-bit multiplex bus.
Note 2: In microprocessor mode, multiplexed bus for the entire space cannot be selected.
In memory expansion mode, when multiplexed bus for the entire space is selected, address bus range is 256 bytes in each chip select.
Note 3: These ports don't function as A-D converter input pins.
Note 4: Address bus when in separate bus mode.

Bus Control

The following explains the signals required for accessing external devices and software waits. The signals required for accessing the external devices are valid when the processor mode is set to memory expansion mode and microprocessor mode. The software waits are valid in all processor modes.

(1) Address bus/data bus

The address bus consists of the 20 pins A0 to A19 for accessing the 1M bytes of address space.

The data bus consists of the pins for data I/O. When the BYTE pin is "H", the 8 ports D0 to D7 function as the data bus. When BYTE is "L", the 16 ports D0 to D15 function as the data bus.

Both the address and data bus retain their previous states when internal ROM or RAM is accessed. Also, when a change is made from single-chip mode to memory expansion mode, the value of the address bus is undefined until external memory is accessed.

(2) Chip select signal

The chip select signal is output using the same pins as $P4_4$ to $P4_7$. Bits 0 to 3 of the chip select control register (address 0008_{16}) set each pin to function as a port or to output the chip select signal. The chip select control register is valid in memory expansion mode and microprocessor mode. In single-chip mode, $P4_4$ to $P4_7$ function as programmable I/O ports regardless of the value in the chip select control register.

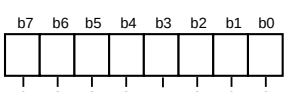
In microprocessor mode, only $\overline{CS0}$ outputs the chip select signal after the reset state has been cancelled. $\overline{CS1}$ to $\overline{CS3}$ function as input ports. Figure 1.9.1 shows the chip select control register.

The chip select signal can be used to split the external area into as many as four blocks. Table 1.9.1 shows the external memory areas specified using the chip select signal. Note that the address ranges for $\overline{CS0}$ and $\overline{CS3}$ vary according to processor mode and setting of the internal reserved area expansion bit (PM13).

Table 1.9.1. External areas specified by the chip select signals

Chip select	PM13	Address range	
		Memory expansion mode	Microprocessor mode
$\overline{CS0}$	0	30000_{16} to $CFFFF_{16}$ (640Kbytes)	30000_{16} to $FFFFF_{16}$ (832Kbytes)
	1	30000_{16} to $BFFFF_{16}$ (576Kbytes)	30000_{16} to $FFFFF_{16}$ (832Kbytes)
$\overline{CS1}$	—	28000_{16} to $2FFFF_{16}$ (32Kbytes)	28000_{16} to $2FFFF_{16}$ (32Kbytes)
$\overline{CS2}$	—	08000_{16} to $27FFF_{16}$ (128Kbytes)	08000_{16} to $27FFF_{16}$ (128Kbytes)
$\overline{CS3}$	0	04000_{16} to $07FFF_{16}$ (16Kbytes)	04000_{16} to $07FFF_{16}$ (16Kbytes)
	1	06000_{16} to $07FFF_{16}$ (8Kbytes)	06000_{16} to $07FFF_{16}$ (8Kbytes)

Chip select control register



Symbol: CSR | Address: 0008₁₆ | When reset: 01₁₆

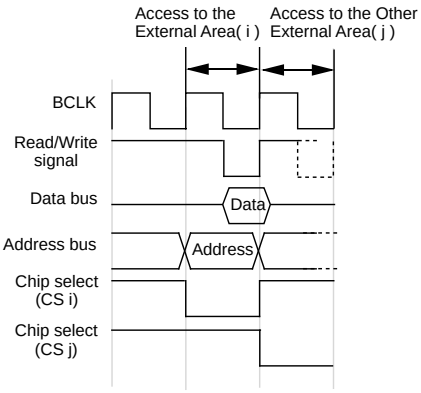
Bit symbol	Bit name	Function	R	W
CS0	$\overline{CS0}$ output enable bit	0 : Chip select output disabled (Normal port pin) 1 : Chip select output enabled	O	O
CS1	$\overline{CS1}$ output enable bit		O	O
CS2	$\overline{CS2}$ output enable bit		O	O
CS3	$\overline{CS3}$ output enable bit		O	O
CS0W	$\overline{CS0}$ wait bit	0 : Wait state inserted 1 : No wait state	O	O
CS1W	$\overline{CS1}$ wait bit		O	O
CS2W	$\overline{CS2}$ wait bit		O	O
CS3W	$\overline{CS3}$ wait bit		O	O

Figure 1.9.1. Chip select control register

The timing of the chip select signal changing to "L" (active) is synchronized with the address bus. But the timing of the chip select signal changing to "H" depends on the area which will be accessed in the next cycle. Figure 1.9.2 shows the output example of the address bus and chip select signal.

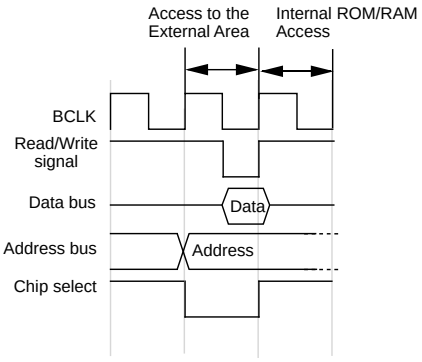
Example 1) After access the external area, both the address signal and the chip select signal change concurrently in the next cycle.

In this example, after access to the external area(i), an access to the area indicated by the other chip select signal(j) will occur in the next cycle. In this case, both the address bus and the chip select signal change between the two cycles.



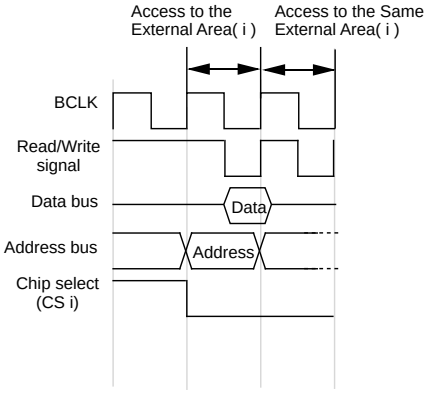
Example 2) After access the external area, only the chip select signal changes in the next cycle (the address bus does not change).

In this example, an access to the internal ROM or the internal RAM in the next cycle will occur, after access to the external area. In this case, the chip select signal changes between the two cycles, but the address does not change.



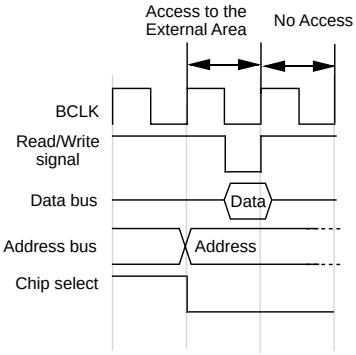
Example 3) After access the external area, only the address bus changes in the next cycle (the chip select signal does not change).

In this example, after access to the external area(i), an access to the area indicated by the same chip select signal(i) will occur in the next cycle. In this case, the address bus changes between the two cycles, but the chip select signal does not change.



Example 4) After access the external area, either the address signal and the chip select signal do not change in the next cycle.

In this example, any access to any area does not occur in the next cycle (either instruction prefetch does not occur). In this case,either the address bus and chip select signal do not change between the two cycles.



Note : These examples show the address bus and chip select signal within the successive two cycles.
According to the combination of these examples, the chip select can be elongated to over 2cycles.

Figure 1.9.2. Output examples about address bus and chip select signal (separated bus without wait)

(3) Read/write signals

With a 16-bit data bus (BYTE pin = "L"), bit 2 of the processor mode register 0 (address 0004_{16}) selects the combinations of $\overline{RD}$, $\overline{BHE}$, and $\overline{WR}$ signals or $\overline{RD}$, $\overline{WRL}$, and $\overline{WRH}$ signals. With an 8-bit data bus (BYTE pin = "H"), use the combination of $\overline{RD}$, $\overline{WR}$, and $\overline{BHE}$ signals. (Set bit 2 of the processor mode register 0 (address 0004_{16}) to "0".) Tables 1.9.2 and 1.9.3 show the operation of these signals.

After a reset has been cancelled, the combination of $\overline{RD}$, $\overline{WR}$, and $\overline{BHE}$ signals is automatically selected. When switching to the $\overline{RD}$, $\overline{WRL}$, and $\overline{WRH}$ combination, do not write to external memory until bit 2 of the processor mode register 0 (address 0004_{16}) has been set (Note).

Note: Before attempting to change the contents of the processor mode register 0, set bit 1 of the protect register (address $000A_{16}$) to "1".

Table 1.9.2. Operation of $\overline{RD}$, $\overline{WRL}$, and $\overline{WRH}$ signals

Data bus width	$\overline{RD}$	$\overline{WRL}$	$\overline{WRH}$	Status of external data bus
16-bit (BYTE = "L")	L	H	H	Read data
	H	L	H	Write 1 byte of data to even address
	H	H	L	Write 1 byte of data to odd address
	H	L	L	Write data to both even and odd addresses

Table 1.9.3. Operation of $\overline{RD}$, $\overline{WRL}$, and $\overline{BHE}$ signals

Data bus width	$\overline{RD}$	$\overline{WR}$	$\overline{BHE}$	A0	Status of external data bus
16-bit (BYTE = "L")	H	L	L	H	Write 1 byte of data to odd address
	L	H	L	H	Read 1 byte of data from odd address
	H	L	H	L	Write 1 byte of data to even address
	L	H	H	L	Read 1 byte of data from even address
	H	L	L	L	Write data to both even and odd addresses
	L	H	L	L	Read data from both even and odd addresses
8-bit (BYTE = "H")	H	L	Not used	H / L	Write 1 byte of data
	L	H	Not used	H / L	Read 1 byte of data

(4) ALE signal

The ALE signal latches the address when accessing the multiplex bus space. Latch the address when the ALE signal falls.

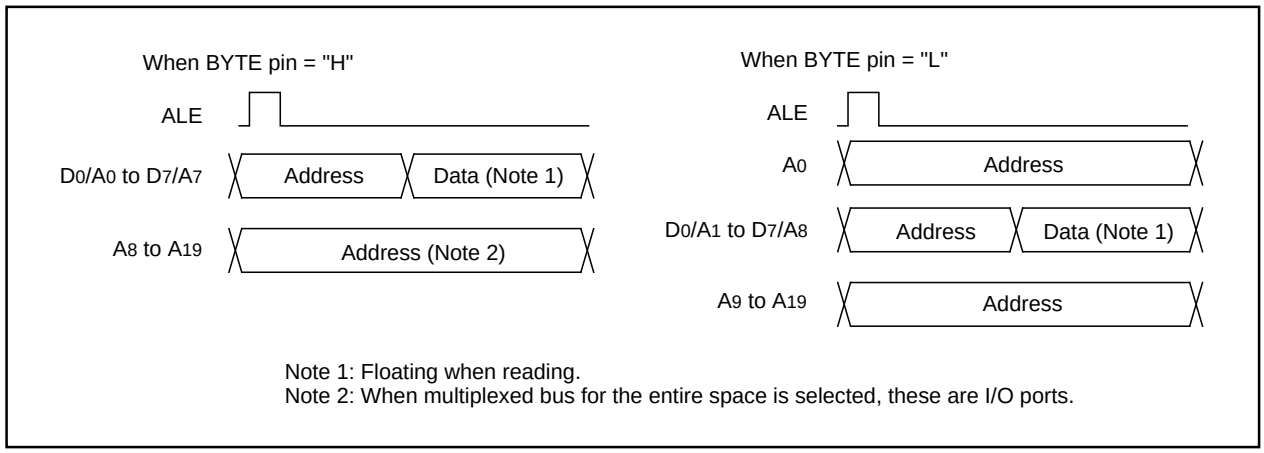


Figure 1.9.3. ALE sigal and address/data bus

(5) $\overline{RDY}$ signal

The ready signal facilitates access of external devices that require a long time for access. As shown in Figure 1.9.4, inputting "L" to the $\overline{RDY}$ pin at the falling edge of BCLK causes the microcomputer to enter the ready state. Inputting "H" to the $\overline{RDY}$ pin at the falling edge of BCLK cancels the ready state. Table 1.9.4 shows the microcomputer status in the ready state. Figure 1.9.4 shows the example of the $\overline{RD}$ signal being extended using the $\overline{RDY}$ pin.

The $\overline{RDY}$ signal is valid when accessing the external area during the bus cycle in which bits 4 to 7 of the chip select control register (address 0008_{16}) are set to "0". The $\overline{RDY}$ signal is invalid when setting "1" to all bits 4 to 7 of the chip select control register (address 0008_{16}), but the $\overline{RDY}$ pin should be treated as properly as in non-using.

Table 1.9.4. Microcomputer status in ready state (Note)

Item	Status
Oscillation	On
R/$\overline{W}$ signal, address bus, data bus, $\overline{CS}$ ALE signal, $\overline{HLDA}$, programmable I/O ports	Maintain status when $\overline{RDY}$ signal accepted
Internal peripheral circuits	On

Note: The $\overline{RDY}$ signal cannot be accepted immediately prior to a software wait.

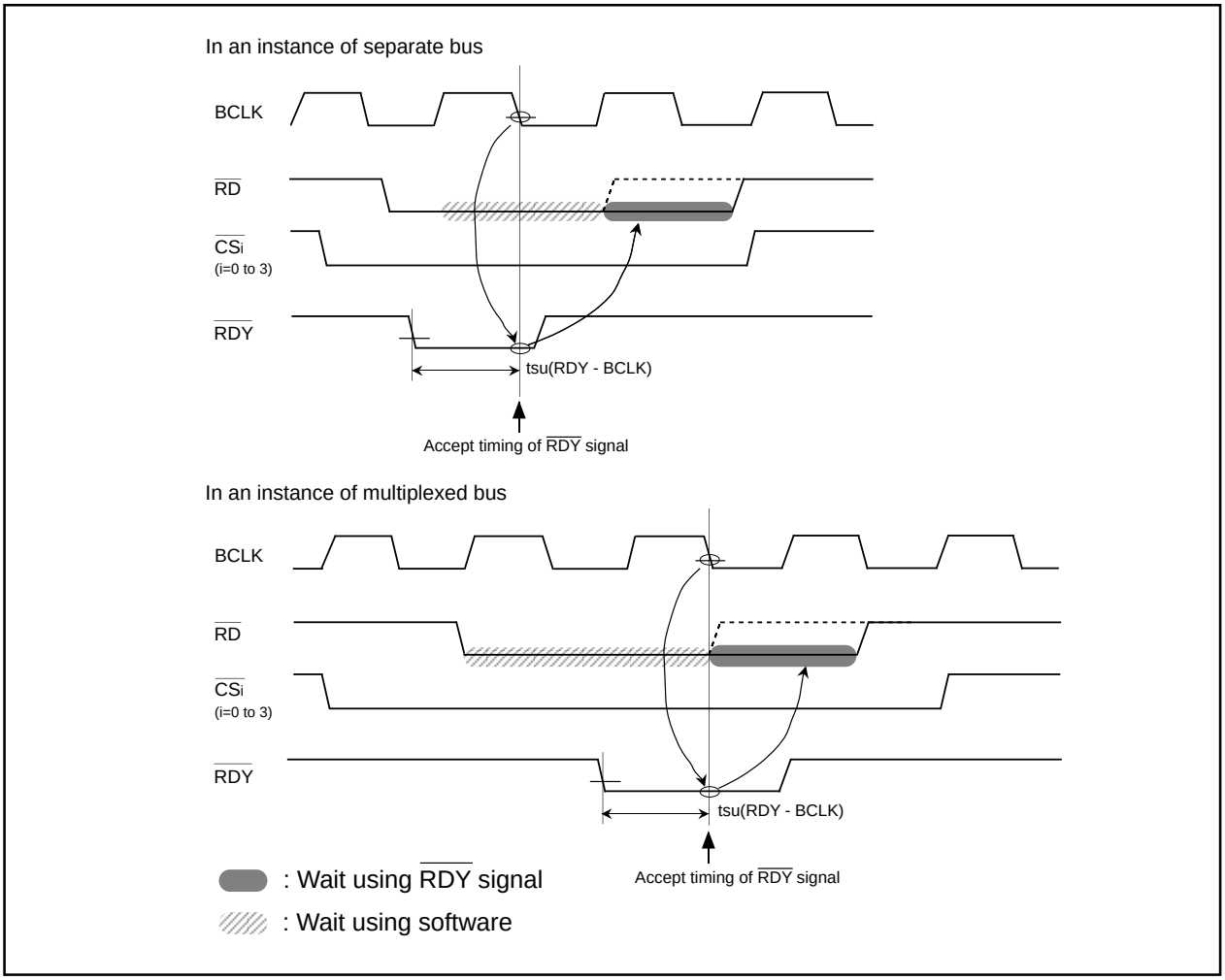


Figure 1.9.4. Example of $\overline{RD}$ signal extended by $\overline{RDY}$ signal

(6) Hold signal

The hold signal is used to transfer the bus privileges from the CPU to the external circuits. Inputting "L" to the $\overline{HOLD}$ pin places the microcomputer in the hold state at the end of the current bus access. This status is maintained and "L" is output from the $\overline{HLDA}$ pin as long as "L" is input to the $\overline{HOLD}$ pin. Table 1.9.5 shows the microcomputer status in the hold state.

Bus-using priorities are given to $\overline{HOLD}$, DMAC, and CPU in order of decreasing precedence.

$\overline{HOLD}$ > DMAC > CPU

Figure 1.9.5. Bus-using priorities

Table 1.9.5. Microcomputer status in hold state

Item		Status
Oscillation		On
R/$\overline{W}$ signal, address bus, data bus, $\overline{CS}$, $\overline{BHE}$		Floating
Programmable I/O ports	P0, P1, P2, P3, P4, P5	Floating
	P6, P7, P8, P9, P10	Maintain status when hold signal is received
$\overline{HLDA}$		Output "L"
Internal peripheral circuits		On (but watchdog timer stops)
ALE signal		Undefined

(7) External bus status when the internal area is accessed

Table 1.9.6 shows the external bus status when the internal area is accessed.

Table 1.9.6. External bus status when the internal area is accessed

Item		SFR accessed	Internal ROM/RAM accessed
Address bus		Address output	Maintain status before accessed address of external area
Data bus	When read	Floating	Floating
	When write	Output data	Undefined
$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$		$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$ output	Output "H"
$\overline{BHE}$		$\overline{BHE}$ output	Maintain status before accessed status of external area
$\overline{CS}$		Output "H"	Output "H"
ALE		Output "L"	Output "L"

(8) BCLK output

The output of the internal clock ø can be selected using bit 7 of the processor mode register 0 (address 0004_{16}) (Note). The output is floating when bit 7 is set to "1".

Note: Before attempting to change the contents of the processor mode register 0, set bit 1 of the protect register (address $000A_{16}$) to "1".

(9) 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) and bits 4 to 7 of the chip select control register (address 0008_{16}).

A software wait is inserted in the internal ROM/RAM area and in the external memory 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 or three BCLK cycles. After the microcomputer has been reset, this bit defaults to "0". When set to "1", bits 4 to 7 of the chip select control register are invalid and a wait is applied to all external memory areas (two or three BCLK cycles).

When the wait bit of the processor mode register 1 is "0", software waits can be set independently for each of the 4 areas selected using the chip select signal. Bits 4 to 7 of the chip select control register correspond to chip selects $\overline{CS0}$ to $\overline{CS3}$. When one of these bits is set to "1", the bus cycle is executed in one BCLK cycle. When set to "0", the bus cycle is executed in two or three BCLK cycles. These bits default to "0" after the microcomputer has been reset.

The SFR area is always accessed in two BCLK cycles regardless of the setting of these control bits. Also, the corresponding bits of the chip select control register must be set to "0" if using the multiplex bus to access the external memory area.

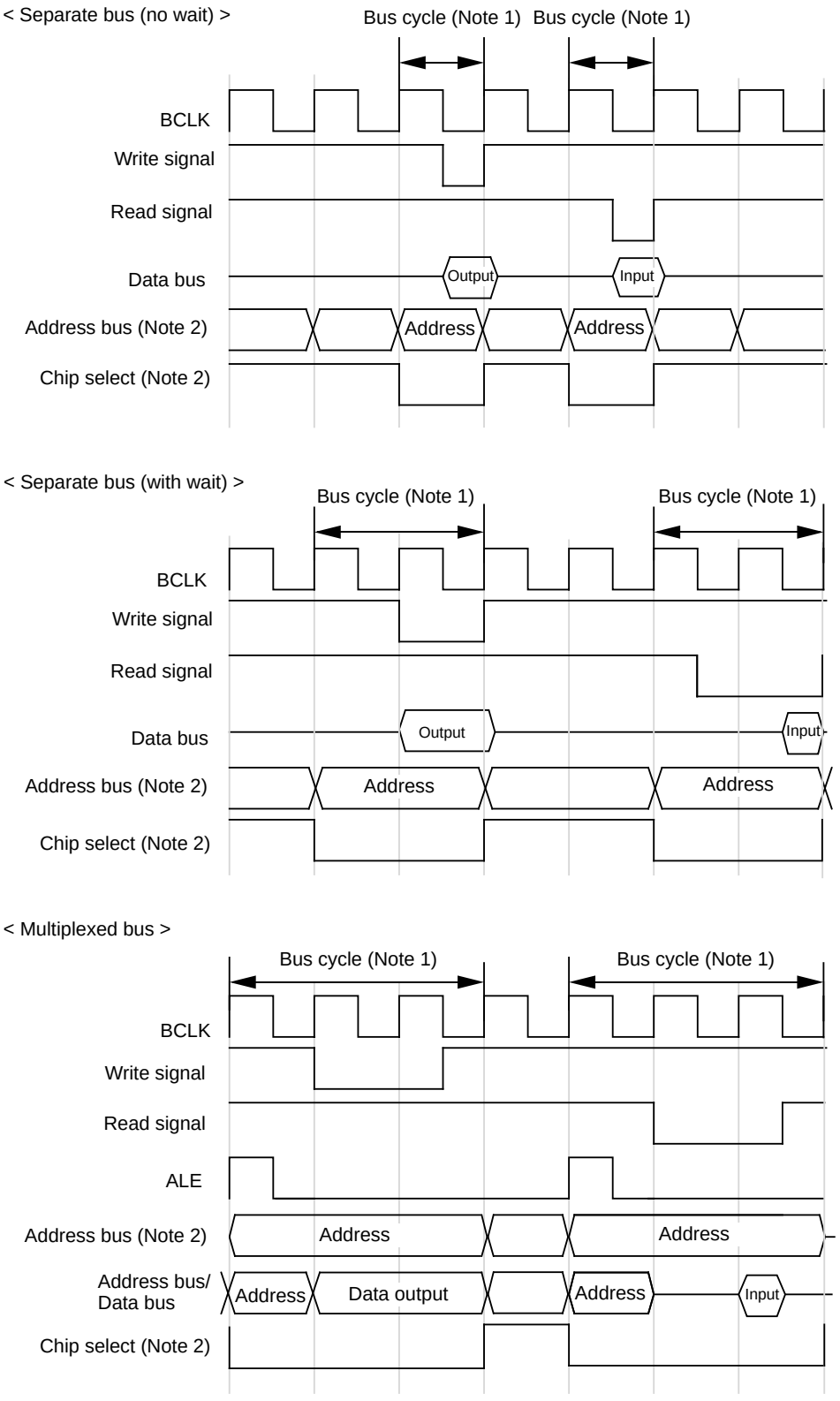
Table 1.9.7 shows the software wait and bus cycles. Figure 1.9.6 shows example bus timing when using software waits.

Note: 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 1.9.7. Software waits and bus cycles

Area	Bus status	Wait bit	Bits 4 to 7 of chip select control register	Bus cycle
SFR	———	Invalid	Invalid	2 BCLK cycles
Internal ROM/RAM	———	0	Invalid	1 BCLK cycle
	———	1	Invalid	2 BCLK cycles
External memory area	Separate bus	0	1	1 BCLK cycle
	Separate bus	0	0	2 BCLK cycles
	Separate bus	1	0 (Note)	2 BCLK cycles
	Multiplex bus	0	0 (Note)	3 BCLK cycles
	Multiplex bus	1	0 (Note)	3 BCLK cycles

Note: When using the $\overline{RDY}$ signal, always set to "0".



Note 1: These example timing charts indicate bus cycle length.
After this bus cycle sometimes come read and write cycles in succession.
Note 2: The address bus and chip select may be extended depending on the CPU status such as that of the instruction queue buffer.

Figure 1.9.6. Typical bus timings using software wait

Clock Generating Circuit

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

Table 1.10.1. 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 unit's operating clock source	●CPU's operating clock source ●Timer A/B's count clock source
Usable oscillator	Ceramic or quartz crystal oscillator	Quartz 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 1.10.1 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 1.10.2 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 1.10.1 and 1.10.2 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 1.10.1. 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 XIN and XOUT following the instruction.

Figure 1.10.2. Examples of sub clock

Internal Ring-Oscillator

A ring oscillator is built in the microcomputer. It can be used instead of XIN as a main clock by setup of the bit 1 of the oscillation stop detect register. Lower power dissipation can be realized because the oscillating frequency of the ring oscillator is much lower compared to that of XIN.

Clock Control

Figure 1.10.3 shows the block diagram of the clock generating circuit.

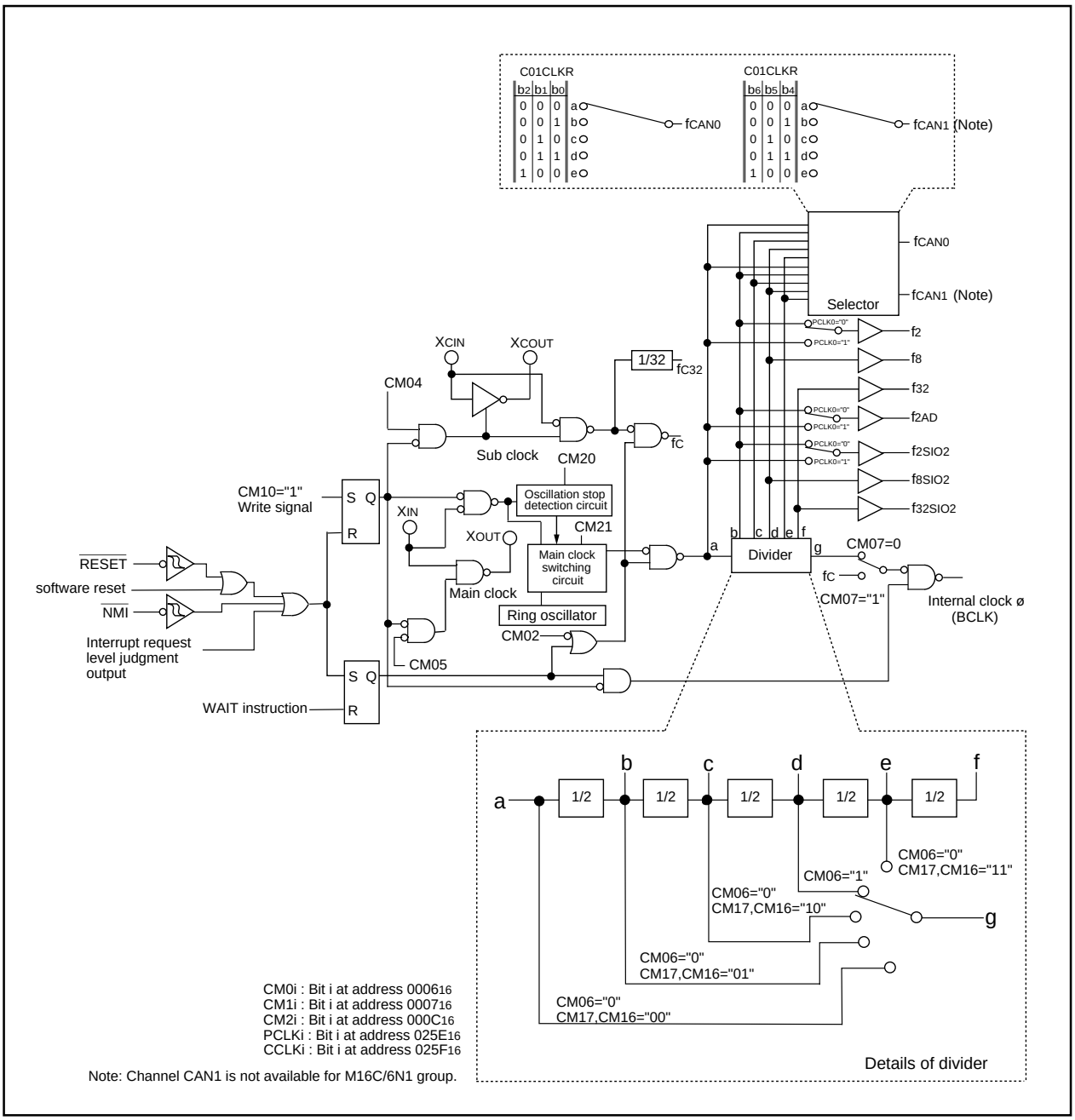


Figure 1.10.3. Clock generating circuit

The following paragraphs describe 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 XOUT pin can be reduced using the XIN-XOUT drive capacity select bit (bit 5 at address 0007_{16}). Reducing the drive capacity of the XOUT pin reduces the power dissipation. This bit defaults to "1" when shifting to stop mode and after a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

You can switch over from the main clock to the ring oscillator by changing the value of the main clock switch bit (bit 1 at address $000C_{16}$).

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

When the XCIN/XCOUT is used, set ports $P8_6$ and $P8_7$ as the input ports without pull-up.

(3) BCLK

The BCLK is the clock that drives the CPU and the watchdog timer, i.e. the internal clock ø, and is either the main clock or fc or is derived by dividing the main clock by 2, 4, 8, or 16. After a reset the BCLK is derived by dividing the main clock by 8 .

When shifting to stop mode, the main clock division select bit (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 retainded.

(4) Peripheral function clocks

- **f_2, f_8, f_{32}, f_{2SIO2}, f_{8SIO2}, f_{32SIO2}**

The clock for the peripheral devices is derived by dividing the main clock by 2 (or no division), 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 address 0006_{16}) to "1" and then executing a WAIT instruction.

As to f_2 and f_{2SIO2}, you can select division by 2 or no division by changing the value of the peripheral function clock select register.

- **f_{2AD}**

This clock is derived by dividing the main clock by 2 (or no division) and is used for A-D conversion. The peripheral function clock is stopped by stopping the main clock or by setting the WAIT peripheral function clock stop bit (bit 2 at address 0006_{16}) to "1" and then executing a WAIT instruction. You can select division by 2 or no division by changing the value of the peripheral function clock select register.

• **f_{CAN0}, f_{CAN1} (Note)**

These clocks are derived by dividing the main clock by 1, 2, 4, 8 or 16 and they are used for the corresponding CAN module. The peripheral function clock is stopped by stopping the main clock or by setting the WAIT peripheral function clock stop bit (bit 2 at address 0006_{16}) to "1" and then executing a WAIT instruction. When CAN modules are shifted to sleep mode, these clocks stop at "H".

(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 may be selected as the BCLK and for the watchdog timer.

(7) f_{RING}

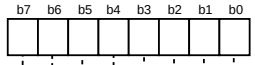
This clock is supplied by the ring oscillator circuit. Immediately after a reset, this clock is not supplied. The ring oscillator oscillation can be set to BCLK when oscillation stop is detected or with the main clock switch bit (bit 1 at address $000C_{16}$).

After the oscillation of the ring oscillator circuit has stabilized, the X_{IN} clock driver can be stopped by setting the main clock stop bit (bit 5 at address 0006_{16}) to "1". This can reduce the power dissipation even more.

Note: Channel CAN1 is not available for M16C/6N1 group.

Figure 1.10.4 shows the system clock control registers 0 and 1.

System clock control register 0 (Note 1)



Symbol: CM0 | Address: 000616 | When reset: 4816

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 P57 0 1 : fC output 1 0 : f8 output 1 1 : f32 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 (Note 9)	O	O
CM05	Main clock (XIN-XOUT) stop bit (Note 3, 4, 5, 10)	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 000A16) 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 the low power dissipation mode and the ring oscillator mode, main clock stops by using this bit. To stop the main clock, when the sub clock oscillation is stable, 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 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: fC32 is not included. Do not set to "1" when using low-speed or low power dissipation mode.
Note 9: When the XCIN/XCOUT is used, set ports P86 and P87 as the input ports without pull-up.
Note10: Setting this bit to "1" disables the main clock buffer. If the oscillation is generated by a quartz crystal or ceramic oscillator, oscillation stops. To avoid stopping the MCU, set CM07 to "1" or CM21 to "1" prior to setting this bit to "1".

System clock control register 1 (Note 1)

Symbol: CM1 | Address: 000716 | When reset: 2016

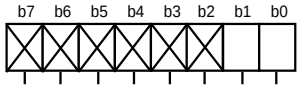
Bit symbol	Bit name	Function	R	W
CM10	All clock stop control bit (Note4)	0 : Clock on 1 : All clocks off (stop mode)	O	O
Reserved bits		Must always be set to "0"	O	O
CM15	XIN-XOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	O	O
CM16	Main clock division select bit 1 (Note 3)	b7 b6 0 0 : No division 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 000A16) 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 000616) 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 1.10.4. Clock control registers 0 and 1

Figure 1.10.5 shows the peripheral function clock select register and Figure 1.10.6 shows the CAN0/1 clock select register.

Peripheral function clock select register (Note 1)



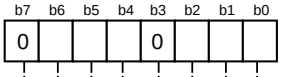
Symbol	Address	When reset
PCLKR	$025E_{16}$	$XXXXXX00_2$

Bit symbol	Bit name	Function	R	W
PCLK0	TimerA, TimerB, A-D converter function clock	(f_2 and f_{2AD} are) 0: Division by 2 mode 1: Division by 1 mode	○	○
PCLK1	UART0-2, S I/O3 function clock	(f_{2SIO2} is) 0: Division by 2 mode 1: Division by 1 mode	○	○
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are "0".			○	–

Note: Set bit 0 of the protect register (address $000A_{16}$) to "1" before writing in this register.

Figure 1.10.5. Peripheral function clock select register

CAN0/1 clock select register (Note 1, 2)



Symbol	Address	When reset
CCLKR	$025F_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
CCLK0	CAN0 Clock select bit	b2 b1 b0 0 0 0: No division mode 0 0 1: Division by 2 mode 0 1 0: Division by 4 mode 0 1 1: Division by 8 mode 1 0 0: Division by 16 mode 1 0 1: Inhibited 1 1 0: Inhibited 1 1 1: Inhibited	○	○
CCLK1			○	○
CCLK2			○	○
Reserved bit		Always set to "0"	○	○
CCLK4	CAN1 Clock select bit (Note3)	b6 b5 b4 0 0 0: No division mode 0 0 1: Division by 2 mode 0 1 0: Division by 4 mode 0 1 1: Division by 8 mode 1 0 0: Division by 16 mode 1 0 1: Inhibited 1 1 0: Inhibited 1 1 1: Inhibited	○	○
CCLK5			○	○
CCLK6			○	○
Reserved bit		Always set to "0"	○	○

Note 1: Set bit 0 of the protect register (address $000A_{16}$) to "1" before writing in this register.

Note 2: Change the register value only when the CAN module is in Reset/Initialization mode (the bit 0 of the CAN Control Register (address 0210_{16} and 0230_{16}) is "1").

Note 3: Channel CAN1 is not available for M16C/6N1 group.

Figure 1.10.6. CAN0/1 clock select register

Clock Output

In single-chip mode, the clock output function select bits (bits 0 and 1 at address 0006_{16}) enable f_8, f_{32}, or fc to be output from the $P5_7$/CLK_{OUT} 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} stops when a WAIT instruction is executed.

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_2 to f_{32}, f_{1SIO2} to f_{32SIO2}, f_{2AD}, f_{CAN0}, f_{CAN1}, f_{C32}, and fC stop in stop mode, peripheral functions such as the 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 UARTi (i = 0 to 2), S I/O3 functions provided an external clock is selected. Table 1.10.2 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, and the priority level of the interrupt which is not used to cancel must have been changed to "0". If returning by an interrupt, that interrupt routine is executed. If only a hardware reset or an $\overline{NMI}$ interrupt is used to cancel stop mode, change the priority level of all interrupt to "0", then shift to stop mode.

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 1.10.2. Port status during stop mode

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
Address bus, data bus, $\overline{CS0}$ to $\overline{CS3}$, $\overline{BHE}$		Retains status before stop mode	——
$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$		"H"	——
$\overline{HLDA}$, BCLK		"H"	——
ALE		"H"	——
Port		Retains status before stop mode	Retains status before stop mode
CLK_{OUT}	When fc selected	Valid only in single-chip mode	"H"
	When f_8, f_{32} selected	Valid only in single-chip mode	Retains status before stop mode

Wait Mode

When a WAIT instruction is executed, the BCLK stops and the microcomputer enters the wait mode. In this mode, oscillation continues but the BCLK and the watchdog timer stop. Writing "1" to the WAIT peripheral function clock stop bit and executing a WAIT instruction also stops the clock being supplied to the internal peripheral functions, allowing power dissipation to be reduced. However, peripheral function clock f_{C32} does not stop so that the peripherals using f_{C32} do not contribute to the power saving. When the MCU running in low-speed or low power dissipation mode, do not enter WAIT mode with this bit set to "1". Table 1.10.3 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, that interrupt must first have been enabled, and the priority level of the interrupt which is not used to cancel must have been changed to "0". If returning by an interrupt, the clock in which the WAIT instruction executed is set to BCLK by the microcomputer, and the action is resumed from the interrupt routine. If only a hardware reset or an $\overline{NMI}$ interrupt is used to cancel wait mode, change the priority level of all interrupt to "0", then shift to wait mode.

Table 1.10.3. Port status during wait mode

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
Address bus, data bus, $\overline{CS0}$ to $\overline{CS3}$, $\overline{BHE}$		Retains status before wait mode	——
$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$		"H"	——
$\overline{HLDA}$,BCLK		"H"	——
ALE		"H"	——
Port		Retains status before wait mode	Retains status before wait mode
CLK_{OUT}	When f_C selected	Valid only in single-chip mode	Does not stop
	When f_8, f_{32} selected	Valid only in single-chip mode	Does not stop when the WAIT peripheral function clock stop bit is "0". When the WAIT peripheral function clock stop bit is "1", the status immediately prior to entering wait mode is maintained.

Status Transition of BCLK

Power dissipation can be reduced and low-voltage operation achieved by changing the count source for BCLK. Table 1.10.4 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 low power dissipation mode, make sure the sub clock is oscillating stable.

(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 BCLK. Note that oscillation of both the main and sub clocks must have stabilized before transferring from this mode to another or vice versa. 2 to 3 seconds or more are required before the sub clock is fully stabilized. 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.

(8) Ring oscillator mode

BCLK is generated by the ring oscillator. You can use it by dividing it by 2, 4, 8 or 16, and also no division is possible.

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 stable. Allow a wait time in software for the oscillation to stabilize before switching over the clock.

Status Transition of BCLK

Table 1.10.4. Operating modes dictated by settings of system clock control registers 0 and 1

CM21	CM17	CM16	CM07	CM06	CM05	CM04	Operating mode of BCLK
0	0	0	0	0	0	Invalid	No division mode
0	0	1	0	0	0	Invalid	Division by 2 mode
0	1	0	0	0	0	Invalid	Division by 4 mode
0	Invalid	Invalid	0	1	0	Invalid	Division by 8 mode
0	1	1	0	0	0	Invalid	Division by 16 mode
0	Invalid	Invalid	1	Invalid	0	1	Low-speed mode
0	Invalid	Invalid	1	Invalid	1	1	Low power dissipation mode
1	0	0	0	0	1 (Note)	Invalid	Ring oscillator mode/1
1	0	1	0	0	1 (Note)	Invalid	Ring oscillator mode/2
1	1	0	0	0	1 (Note)	Invalid	Ring oscillator mode/4
1	Invalid	Invalid	0	1	1 (Note)	Invalid	Ring oscillator mode/8
1	1	1	0	0	1 (Note)	Invalid	Ring oscillator mode/16

Note: Set CM21 to "1" before setting this bit to "1".

Power Control

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

Modes

Power control is available in four modes.

(1) 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 selected. The fc clock is supplied by the sub clock. Each peripheral function operates according to its assigned clock.

- **Low power dissipation 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 sub clock. The only peripheral functions that operate are those with the sub clock selected as the count source.

(2) Wait mode

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

(3) Stop mode

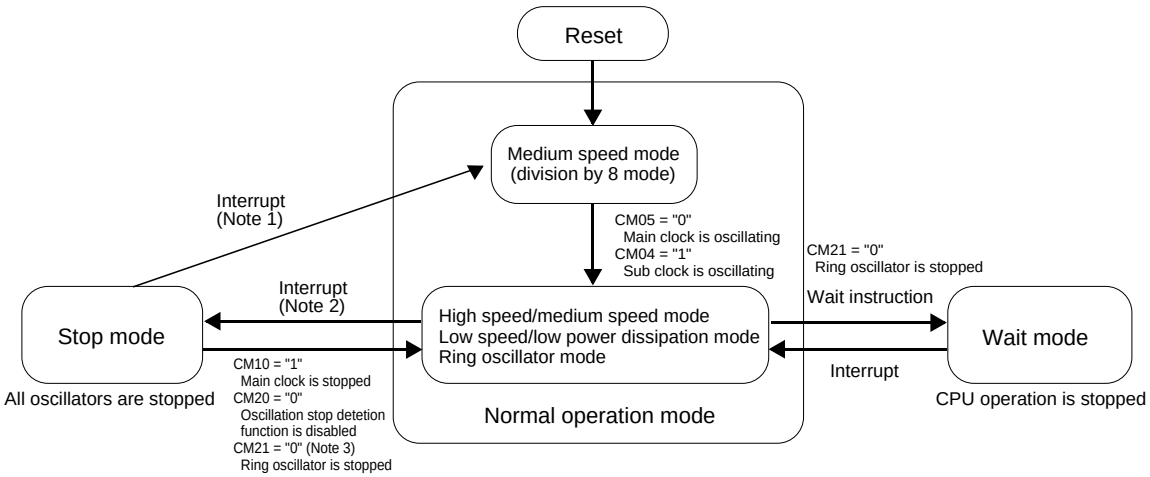
All oscillators stop. The CPU and all built-in peripheral functions stop. This mode, among the three modes listed here, is the most effective in reducing power dissipation.

(4) Ring oscillator mode

The ring oscillator replaces XIN. No-division-, divide-by-2-, 4-, 8- or 16 mode can be selected by changing the values in CM06, CM16 and CM17. The higher the division ratio is, the lower power dissipation. The clock driver of XIN can be stopped by changing the value of the main clock stop bit to "0" when the CPU operates using the ring oscillator. Through this the power dissipation will be still lower.

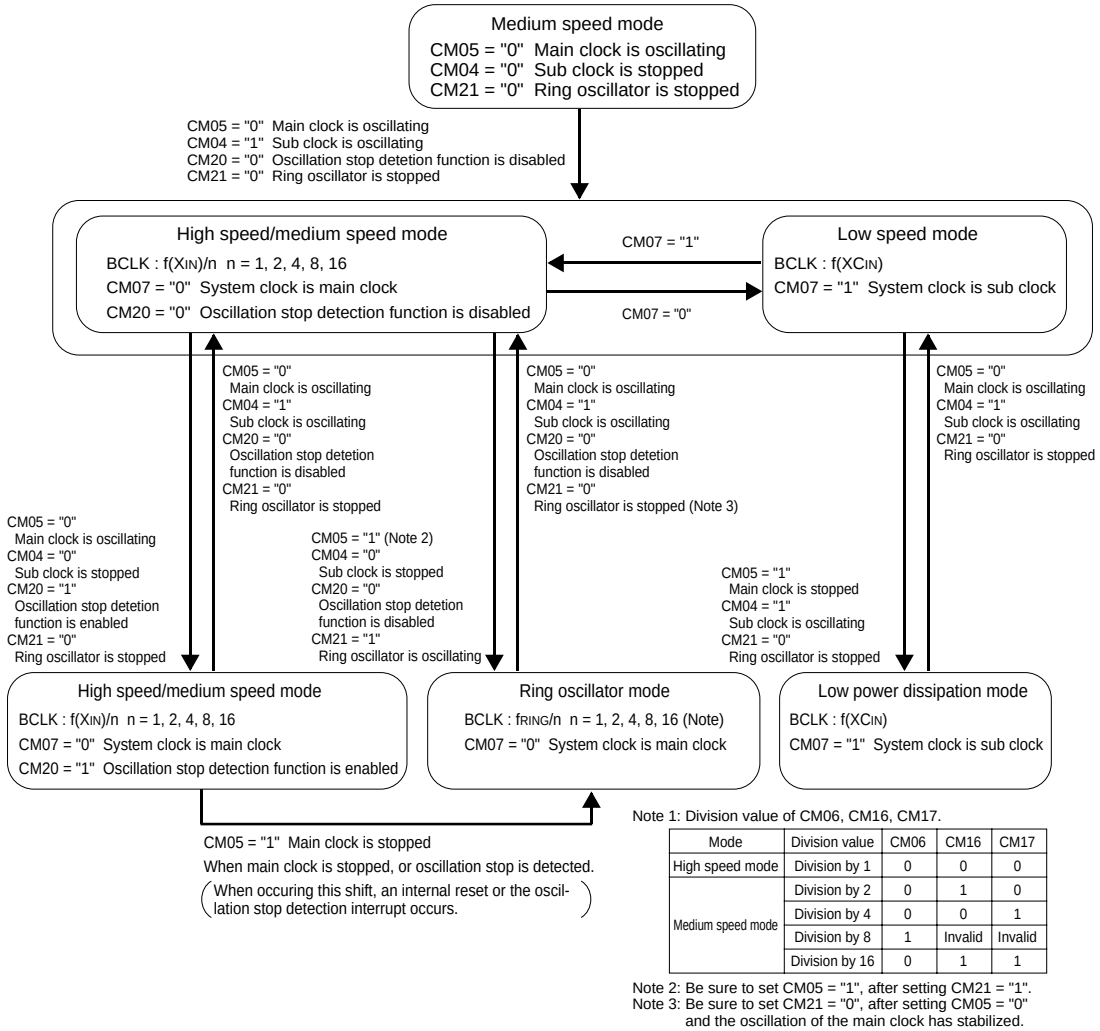
Figure 1.10.7 shows the state transition of power control modes.

Transition of stop mode, wait mode



Note 1: When the mode returns to normal operation mode again after shifting from high speed/medium speed mode to stop mode, the mode shifts to division by 8 mode of medium speed mode.
Note 2: When the mode returns to normal operation mode again after shifting from low speed/low power dissipation mode to stop mode, the mode shifts to low speed/low power dissipation mode.
Note 3: The mode does not shift directly from ring oscillator mode to stop mode.

Transition of normal operation mode



Note 1: Division value of CM06, CM16, CM17.

Mode	Division value	CM06	CM16	CM17
High speed mode	Division by 1	0	0	0
Medium speed mode	Division by 2	0	1	0
	Division by 4	0	0	1
	Division by 8	1	Invalid	Invalid
	Division by 16	0	1	1

Note 2: Be sure to set CM05 = "1", after setting CM21 = "1".
Note 3: Be sure to set CM21 = "0", after setting CM05 = "0" and the oscillation of the main clock has stabilized.

Figure 1.10.7. State transition diagram of power control mode

Oscillation Stop Detection Function

The oscillation stop detection function detects abnormal stopping of the clock by causes such as opening and shorting of the XIN oscillation circuit. When oscillation stop is detected, either an internal reset or an oscillation stop detection interrupt is generated. The selection depends on the value in the bit 7 of the oscillation stop detection register (address 000C16). When an oscillation stop detection interrupt is generated, the ring oscillator in the microcomputer operates automatically and is used as the system clock in place of the XIN clock. This allows interrupt processing.

The oscillation stop detection function can be enabled/disabled with bit 0 of the oscillation stop detection register. When this bit is set to "1", the function is enabled. After the reset is released, the oscillation stop detection function becomes invalid because the bit value is "0".

Note that an oscillator input signal that violates the timing or voltage specification for XIN may cause the CPU to hang. Therefore the oscillation stop detection interrupt may not be processed. Countermeasures on system level should be taken for the CPU to recover from such kind of status.

Table 1.10.5 gives an specification overview of the oscillation stop detection function.

Table 1.10.5. Specification outlint of the oscillation stop detection function

Item	Specification
Oscillation stop detectable clock and frequency range	XIN : 2MHz or higher
Enabling condition for oscillation stop detection function	When the oscillation stop detection bit (bit 0 of address 000C16) is set to "1"
Operation at oscillation stop detection	When internal reset is generated (bit 7 at address 000C16 is cleared to "0".)
	When oscillation stop detection interrupt is generated (bit 7 at address 000C16 is set to "1".)
Notes on STOP mode	Before setting up the stop mode, write "0" in the oscillation stop detection valid bit to invalidate the oscillation stop detection function. Write "1" into the bit again after the stop mode is released.

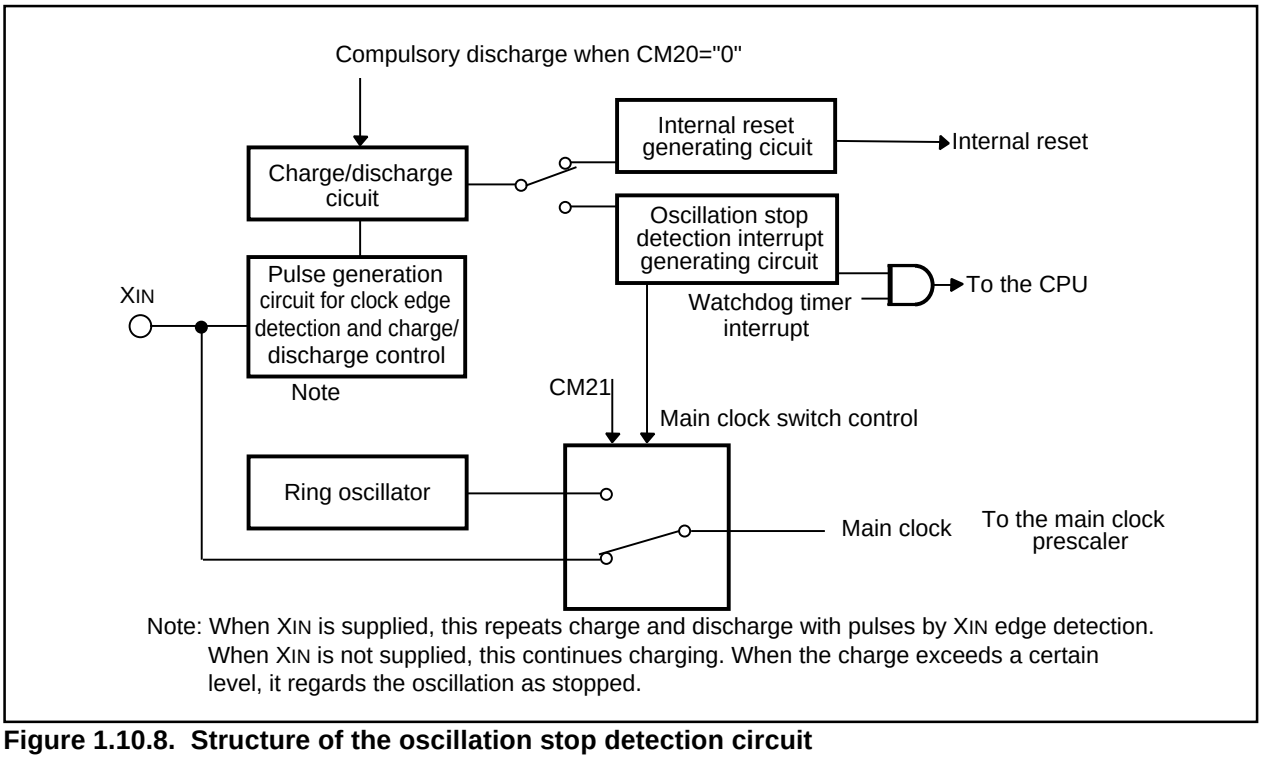


Figure 1.10.8. Structure of the oscillation stop detection circuit

Oscillation stop detection register (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
	0	0	0				

Symbol: CM2 — Address: 000C16 — When reset: 0016

Bit symbol	Bit name	Function	R	W
CM20	Oscillation stop detection bit (Note 2)	0: The function is disabled 1: The function is enabled	○	○
CM21	Main clock source select bit (Note 3)	0: Select XIN 1: Select ring oscillator	○	○
CM22	Oscillation stop detection interrupt request bit (Note 4)	0: Invalid 1: Oscillation stop has been detected	○	○
CM23	Clock monitor bit (Note 5)	0: XIN is in operation 1: Invalid	○	×
Reserved bits		Must be set to "0"	○	○
CM27	Action selection on stop detection (Note 5)	0: Issue internal reset 1: Switch to ring oscillators and request an interrupt	○	○

Note 1: Set bit 0 of the protect register to "1" before writing to this register.
Note 2: Not valid during wait mode with disabled peripheral clocks (CM02 = "1"), main clock stop (CM10 = "1"), or if the ring oscillator is selected manually (CM21 = "1").
Note 3: If CM07 = "1", the selection of XCIN takes precedence over fRING.
Note 4: CM22 may be reset but not set
Note 5: Valid when CM20 = "1".

Figure 1.10.9. Structure of the oscillation stop detection register

Oscillation Stop Detection Bit (CM20)

The oscillation stop detection is activated by setting CM20 to "1".

Set this bit to "0" before entering the stop mode. When returning from stop mode, the external oscillator may be unstable for a period of time. The detection may be enabled again after returning from stop mode.

Set this bit to "0" also before setting main clock stop bit (bit 5 at address 000616).

Do not enable the detection function if XIN is lower than 2MHz.

Main Clock Source Selection Bit (CM21)

The internal ring oscillator may be selected as the main clock source independently of the oscillation stop detection by setting this bit to "1". The division ratio of the ring oscillator, like that of XIN, is governed by CM06, CM16, and CM17. Note that the selection of the sub clock (XCIN) takes precedence over the ring oscillator selection.

Switching to fRING manually disables the oscillation stop detection regardless of CM20.

Oscillation Stop Detection Interrupt Request Bit (CM22)

This bit signals an oscillation stop detection interrupt. The oscillation stop/restart detection and the watchdog timer share an interrupt request line. The interrupt service routine can determine which unit requested the interrupt by sampling this bit. Whe an oscillation stop is detected, this bit goes high. It has to be reset manually during interrupt servicing.

See also Figure 1.10.10.

Clock Monitor Bit (CM23)

The operational status of XIN can be monitored with this bit. When XIN is operating, this bit is "0". After a clock stop detection, this bit can be polled to check if XIN has restarted. This bit is valid when CM20 is "1".

Reserved Bits (CM24-CM26)

These bits are reserved. They must always remain at "0".

Action Selection (when an oscillation stop is detected) Bit (CM27)

(i) Operation when internal reset is selected (CM27="0")

In case an abnormal stop of XIN is detected when the oscillation stop detection valid bit (CM20) is "1", an internal reset is generated. The microcomputer stops in reset state, and it does not operate further.

Note: Release from this state is possible by external reset only. However, if XIN clock includes some errors, further operation cannot be guaranteed.

Table 1.10.6 shows the status of each port after an internal reset is generated.

(ii) Operation when oscillation stop detection interrupt is selected (CM27="1")

In case an abnormal stop of XIN is detected when the oscillation stop detection valid bit (CM20) is "1", an oscillation stop detection interrupt is generated. In this case, the ring oscillator operates instead of the XIN stopped abnormally. Further operation can be done to the ring oscillation. Oscillation stop detection interrupt shares the vector table with watchdog timer interrupt. Accordingly, the interrupt factor should be judged. For this purpose, use the CM22, oscillation stop detection status.

Figure 1.10.10 shows how to judge the factor by the oscillation stop detection interrupt process program.

Stop Mode (CM10="1")

It is recommended the stop detection be disabled before entering stop mode. When returning from stop mode, the external quartz crystal oscillator is instable for a period of time and may falsely cause a clock stop to be signalled. It may be enabled again after returning from the stop mode.

Wait Mode (WAIT instruction issued)

When peripheral clocks during wait are enabled (CM02="0"), the oscillation stop/restart detection circuit can continue to monitor XIN. Should XIN fail, the wait mode is cancelled and an interrupt is generated (CM27="1"). The microcomputer issues an internal reset if CM27 is "0".

When the wait mode is entered with the peripheral clocks disabled (CM02="1"), the clock stop detection is disabled. Clock stops will not be detected. The oscillation stop detection circuit will, however, detect and react to a stopped clock if the wait mode is cancelled (for instance by an $\overline{\text{NMI}}$).

Generally, it is recommended to disable the detection circuit before entering wait mode if the peripheral clocks are to be disabled during wait.

Table1.10.6. Port status after an internal reset is generated

Pin name	Pin status		
	Single chip mode	Operating mode of BCLK	
		BYTE = V_{SS}	BYTE = V_{CC}
P0	Input port (floating)	Data input (floating)	Data input (floating)
P1	Input port (floating)	Data input (floating)	Data input (floating)
P2, P3, $P4_0$ to $P4_3$	Input port (floating)	Address output (indeterminate)	Address output (indeterminate)
$P4_4$	Input port (floating)	$\overline{CS0}$ output ("H" level output)	$\overline{CS0}$ output ("H" level output)
$P4_5$ to $P4_7$	Input port (floating)	Input port (floating) (Pull-up resistance is ON.)	Input port (floating) (Pull-up resistance is ON.)
$P5_0$	Input port (floating)	$\overline{WR}$ output ("H" level output)	$\overline{WR}$ output ("H" level output)
$P5_1$	Input port (floating)	$\overline{BHE}$ output (indeterminate)	$\overline{BHE}$ output (indeterminate)
$P5_2$	Input port (floating)	$\overline{RD}$ output ("H" level output)	$\overline{RD}$ output ("H" level output)
$P5_3$	Input port (floating)	BCLK output	BCLK output
$P5_4$	Input port (floating)	$\overline{HLDA}$ output (output value depends on $\overline{HOLD}$ pin input)	$\overline{HLDA}$ output (output value depends on $\overline{HOLD}$ pin input)
$P5_5$	Input port (floating)	$\overline{HOLD}$ input (floating)	HOLD input (floating)
$P5_6$	Input port (floating)	ALE output ("L" level output)	ALE output ("L" level output)
$P5_7$	Input port (floating)	$\overline{RDY}$ input (floating)	$\overline{RDY}$ input (floating)
P6, P7, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, P9, P10	Input port (floating)	Input port (floating)	Input port (floating)

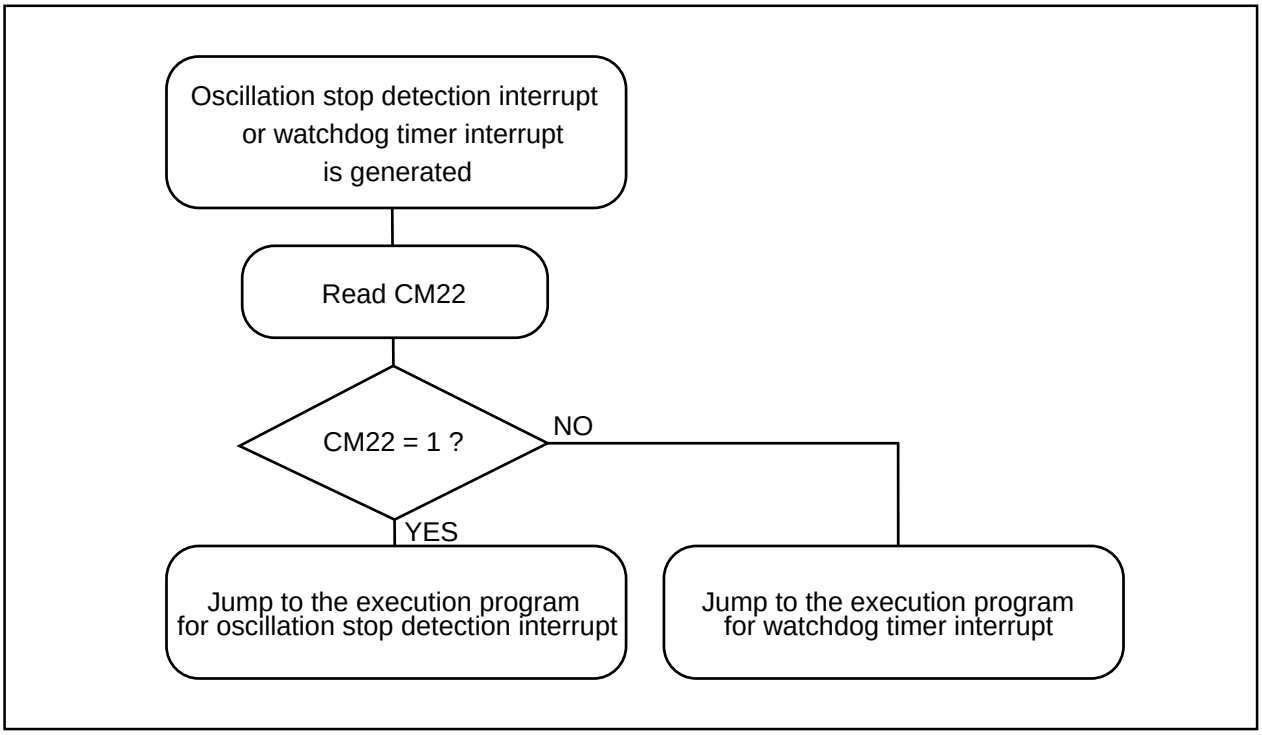


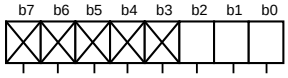
Figure 1.10.10. Flow of the judgment

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 1.10.11 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}), system clock control register 1 (address 0007_{16}), peripheral function clock select register (address $025E_{16}$), CAN0/1 clock select register (address $025F_{16}$), S I/O3 control register (address $01E2_{16}$), oscillation stop detection register (address $000C_{16}$) port P7 direction register (address $03EF_{16}$) and port P9 direction register (address $03F3_{16}$) can only be changed when the respective bit in the protect register is set to "1". Therefore, important outputs can be allocated to port P7 or port P9.

If, after "1" (write-enabled) has been written to the port P7 or port P9 direction registers write-enable bit (bit 2 at address $000A_{16}$), a value is written to any address, the bit automatically reverts to "0" (write-inhibited). However, 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	Clock register protection bit	Enables writing to the system clock control registers 0 and 1 (addresses 0006_{16} and 0007_{16}), oscillation stop detection register (address $000C_{16}$), peripheral function clock select register (address $025E_{16}$), and CAN0/1 clock select register (address $025F_{16}$) (Note 1) 0 : Write-inhibited 1 : Write-enabled	○	○
PRC1	Processor mode register protection bit	Enables writing to processor mode registers 0 and 1 (addresses 0004_{16} and 0005_{16}) 0 : Write-inhibited 1 : Write-enabled	○	○
PRC2	P7/P9 direction register and S I/O3 control register protection bit	Enables writing to port P7/9 direction register (addresses $03F3_{16}$ and $03EF_{16}$) and S I/O3 control register (address $01E2_{16}$) (Note 2) 0 : Write-inhibited 1 : Write-enabled	○	○
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.			–	–

Note 1: There is only one CAN channel available for M16C/6N1 group.

Note 2: Writing a value to an address after "1" is written to this bit returns the bit to "0" . Other bits do not automatically return to "0" and they must therefore be reset by the program.

Figure 1.10.11. Protect register

Overview of Interrupt

Type of Interrupts

Figure 1.11.1 lists the types of interrupts.

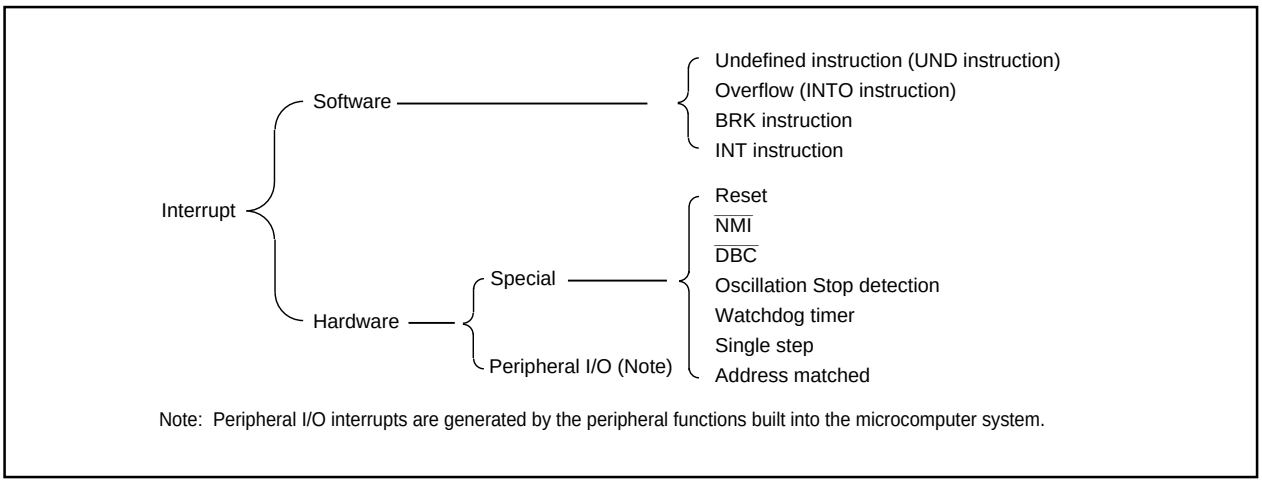


Figure 1.11.1. Classification of interrupts

- 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 assigning one of software interrupt numbers 0 through 63 and executing the INT instruction. Software interrupt numbers 0 through 31 are assigned to peripheral interrupt 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.

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

  An $\overline{\text{NMI}}$ interrupt occurs if an "L" is input to the $\overline{\text{NMI}}$ pin.

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

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

- **Oscillation stop detection interrupt**

  Generated by the oscillation stop detection function.

- **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 no address match interrupt occurs. For address match interrupt, see "Address match interrupt".

(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.

- **Bus collision detection interrupt**

  This is an interrupt that the serial I/O bus collision detection generates.

- **DMA0 interrupt, DMA1 interrupt**

  These are interrupts that DMA generates.

- **Key-input interrupt**

  A key-input interrupt occurs if an "L" is input to the $\overline{\text{KI}}$ pin.

- **A-D conversion interrupt**

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

- **UART0, UART1, UART2/NACK, and S I/O3 transmission interrupt**

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

- **UART0, UART1, UART2/ACK, and S I/O3 reception interrupt**

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

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

  These are interrupts that timer A generates.

- **Timer B0 interrupt through timer B5 interrupt**

  These are interrupts that timer B generates.

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

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

- **CAN0/1 wake up interrupts** (Note)

  These interrupts are generated when a falling edge is input to CRX_0 ro CRX_1 pin.

- **CAN0, 1 transmission interrupts** (Note)

  These are interrupts that a CAN transmission generates.

- **CAN0, 1 reception interrupts** (Note)

  These are interrupts that a CAN reception generates.

- **CAN0/1 error interrupts** (Note)

  These are interrupts that a CAN error generates.

  Note: Channel CAN1 is not available for M16C/6N1 group.
  Interrupts relating to channel CAN1 are invalid for M16C/6N1 group.

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 1.11.2 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.

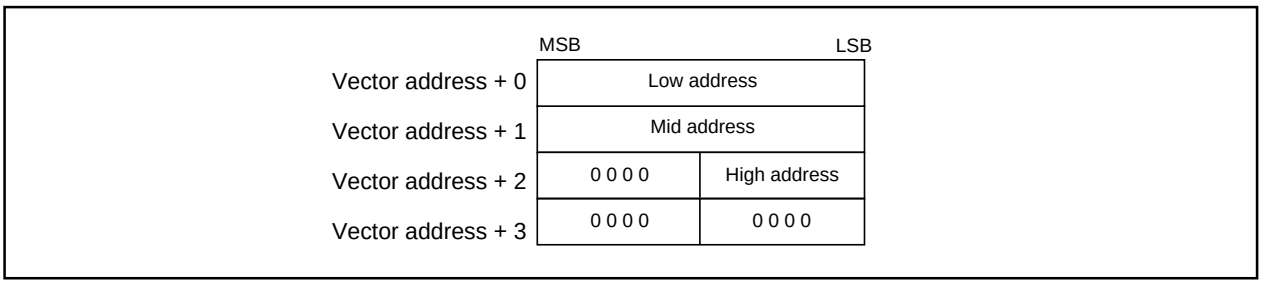


Figure 1.11.2. Format for specifying interrupt vector addresses

• 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 1.11.1 shows the interrupts assigned to the fixed vector tables and addresses of vector tables.

Table 1.11.1. Interrupts assigned to the fixed vector tables and addresses of vector tables

Interrupt source	Vector table addresse 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
Oscillation stop detection/ Watchdog timer	FFFF0 16 to FFFF316	
$\overline{\text{DBC}}$ (Note)	FFFF416 to FFFF716	Do not use
$\overline{\text{NMI}}$	FFFF816 to FFFFB16	External interrupt by input to $\overline{\text{NMI}}$ pin
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 setting. 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 1.11.2 shows the interrupts assigned to the variable vector tables and addresses of vector tables.

Table 1.11.2. Interrupt 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 1)	BRK instr.	
Software interrupt number 1	+4 to +7 (Note 1)	CAN0/1 Wake Up	(Note 4)
Software interrupt number 2	+8 to +11 (Note 1)	CAN0 reception	
Software interrupt number 3	+12 to +15 (Note 1)	CAN0 transmission	
Software interrupt number 4	+16 to +19 (Note 1)	$\overline{\text{INT3}}$	
Software interrupt number 5	+20 to +23 (Note 1)	Timer B5	
Software interrupt number 6	+24 to +27 (Note 1)	Timer B4	
Software interrupt number 7	+28 to +31 (Note 1)	Timer B3	
Software interrupt number 8	+32 to +35 (Note 1,2)	CAN1 reception, $\overline{\text{INT5}}$	(Note 4)
Software interrupt number 9	+36 to +39 (Note 1,2)	CAN1 transmission, $\overline{\text{INT4}}$, S I/O3	(Note 4)
Software interrupt number 10	+40 to +43 (Note 1)	Bus collision detection	
Software interrupt number 11	+44 to +47 (Note 1)	DMA0	
Software interrupt number 12	+48 to +51 (Note 1)	DMA1	
Software interrupt number 13	+52 to +55 (Note 1)	CAN0/1 Error int.	(Note 4)
Software interrupt number 14	+56 to +59 (Note 1,2)	A-D conversion, Key input	
Software interrupt number 15	+60 to +63 (Note 1,3)	UART2 transmission/NACK	
Software interrupt number 16	+64 to +67 (Note 1,3)	UART2 reception/ACK	
Software interrupt number 17	+68 to +71 (Note 1)	UART0 transmission	
Software interrupt number 18	+72 to +75 (Note 1)	UART0 reception	
Software interrupt number 19	+76 to +79 (Note 1)	UART1 transmission	
Software interrupt number 20	+80 to +83 (Note 1)	UART1 reception	
Software interrupt number 21	+84 to +87 (Note 1)	Timer A0	
Software interrupt number 22	+88 to +91 (Note 1)	Timer A1	
Software interrupt number 23	+92 to +95 (Note 1)	Timer A2	
Software interrupt number 24	+96 to +99 (Note 1)	Timer A3	
Software interrupt number 25	+100 to +103 (Note 1)	Timer A4	
Software interrupt number 26	+104 to +107 (Note 1)	Timer B0	
Software interrupt number 27	+108 to +111 (Note 1)	Timer B1	
Software interrupt number 28	+112 to +115 (Note 1)	Timer B2	
Software interrupt number 29	+116 to +119 (Note 1)	$\overline{\text{INT0}}$	
Software interrupt number 30	+120 to +123 (Note 1)	$\overline{\text{INT1}}$	
Software interrupt number 31	+124 to +127 (Note 1)	$\overline{\text{INT2}}$	
Software interrupt number 32 to Software interrupt number 63	+128 to +131 (Note 1) to +252 to +255 (Note 1)	Software interrupt	Cannot be masked I flag

Note 1: Address relative to address in interrupt table register (INTB).
Note 2: It is selected by interrupt request cause select bits (bit 0 and bit 1 at address 01DE16 and bit 6 and bit 7 at address 01DF16).
Note 3: When IIC mode is selected, NACK and ACK interrupts are selected.
Note 4: Channel CAN1 is not available for M16C/6N1 group.

Interrupt Control

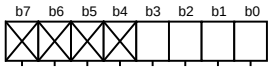
Descriptions are given here regarding how to enable or disable interrupts and how to set the priority to be accepted. What is described here does not apply to non-maskable interrupts.

Enable or disable a maskable interrupt using the interrupt enable flag (I flag), interrupt priority 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 1.11.3 shows the memory map of the interrupt control registers.

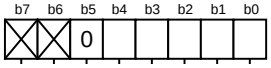
Interrupt control register (Note 2)

Symbol	Address	When reset
C01WKIC, (Note 3)	0041_{16}	$XXXXX000_2$
C0RECIC, C0TRMIC	0042_{16}, 0043_{16}	$XXXXX000_2$
TBiIC(i=3 to 5)	0045_{16} to 0047_{16}	$XXXXX000_2$
BCNIC	$004A_{16}$	$XXXXX000_2$
DMiIC(i=0, 1)	$004B_{16}$, $004C_{16}$	$XXXXX000_2$
C01ERRIC (Note 3)	$004D_{16}$	$XXXXX000_2$
ADIC/KUPIC	$004E_{16}$	$XXXXX000_2$
SiTIC(i=0 to 2)	0051_{16}, 0053_{16}, $004F_{16}$	$XXXXX000_2$
SiRIC(i=0 to 2)	0052_{16}, 0054_{16}, 0050_{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 "0".			–	–

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.
Note 3: Channel CAN1 is not available for M16C/6N1 group.



Symbol		Address	When reset
INT3IC		0044_{16}	$XX00X000_2$
C1RECIC/INT5IC	(Notes 3, 4)	0048_{16}	$XX00X000_2$
C1TRMIC/S3IC/INT4IC	(Notes 3, 4)	0049_{16}	$XX00X000_2$
INTiIC(i=0 to 2)		$005D_{16}$ to $005F_{16}$	$XX00X000_2$

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 "0".			–	–

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.
Note 3: Channel CAN1 is not available for M16C/6N1 group.
Note 4: Use IFSR0/ISFR1 (addresses $01DE_{16}$ and $01DF_{16}$) for interrupt request cause selection.

Figure 1.11.3. 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 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. Table 1.11.3 shows the settings of interrupt priority levels and Table 1.11.4 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 1.11.3. 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 1.11.4. 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 when no interrupt request for that register can be generated. If there is possibility of the interrupt request to 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                 ;Four NOP instructions are required when using HOLD function.
    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         ;Push Flag register onto stack
    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 (four when using the HOLD function) or dummy read is 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 an 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 00000_{16}. After this, the corresponding interrupt request bit becomes "0".

(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.

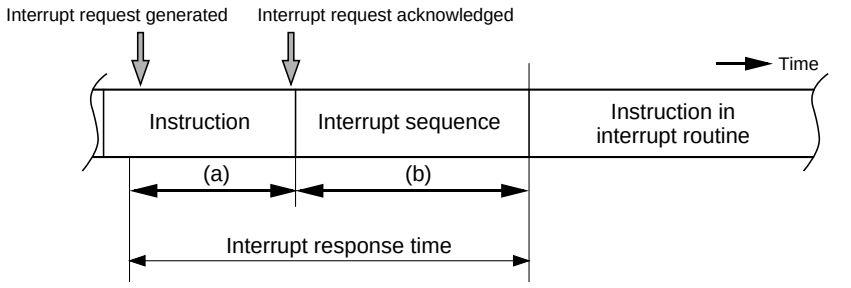
(6) Sets the interrupt priority level of the accepted instruction in the IPL.

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 occurence 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 1.11.4 shows the interrupt responce 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 1.11.4. Interrupt response time

Time (a) is dependent on the instruction under execution. Thirty cycles is the maximum required for the DIVX instruction (without wait).

Time (b) is as shown in Table 1.11.5.

Table 1.11.5. Time required for executing the interrupt sequence

Interrupt vector address	Stack pointer (SP) value	16-Bit bus, without wait	8-Bit bus, without wait
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.

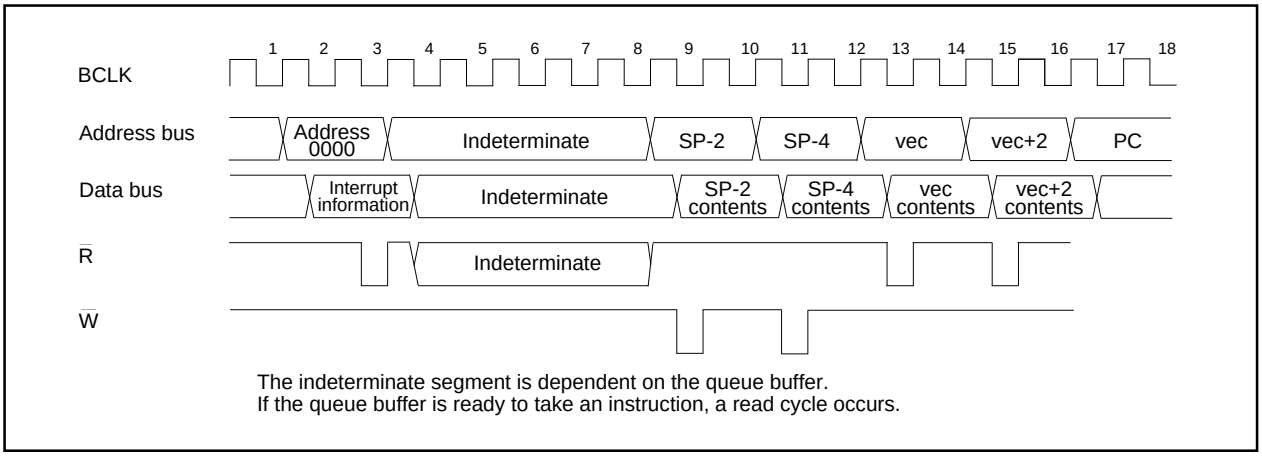


Figure 1.11.5. Time required for executing the interrupt sequence

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 1.11.6 is set in the IPL.

Table 1.11.6. Relation between interrupts without interrupt priority levels and IPL

Interrupt sources without priority levels	Value set in the IPL
Oscillation stop detection, Watchdog timer, $\overline{NMI}$	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 1.11.6 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 ecept the stack pointer (SP).

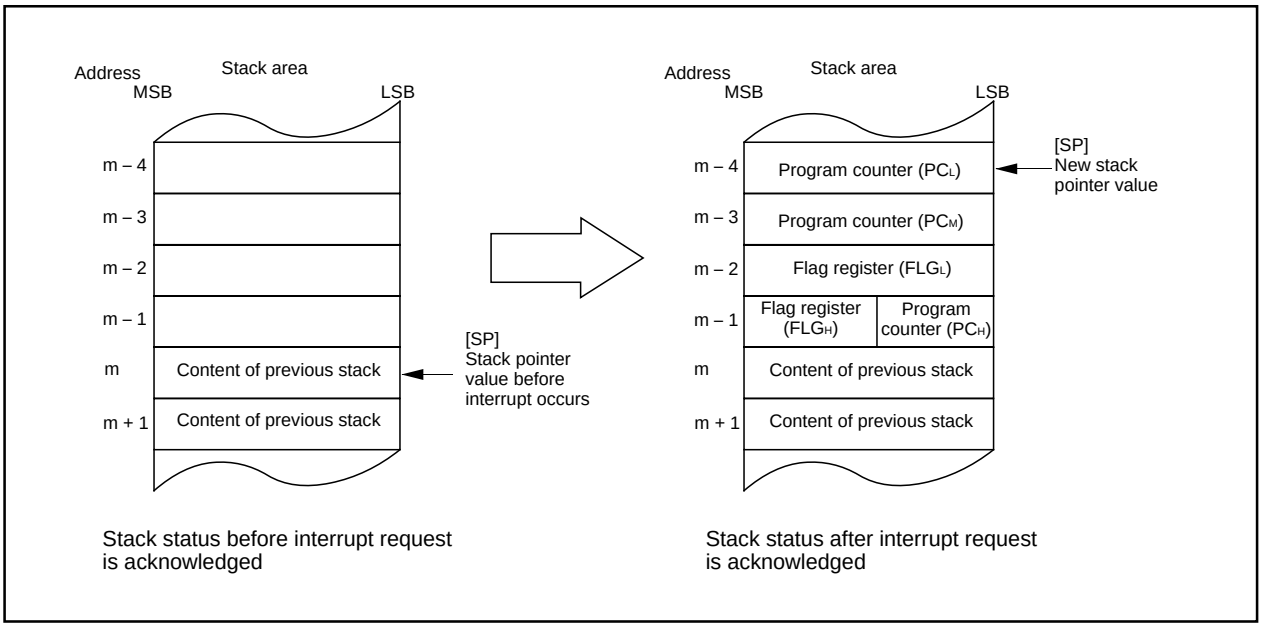
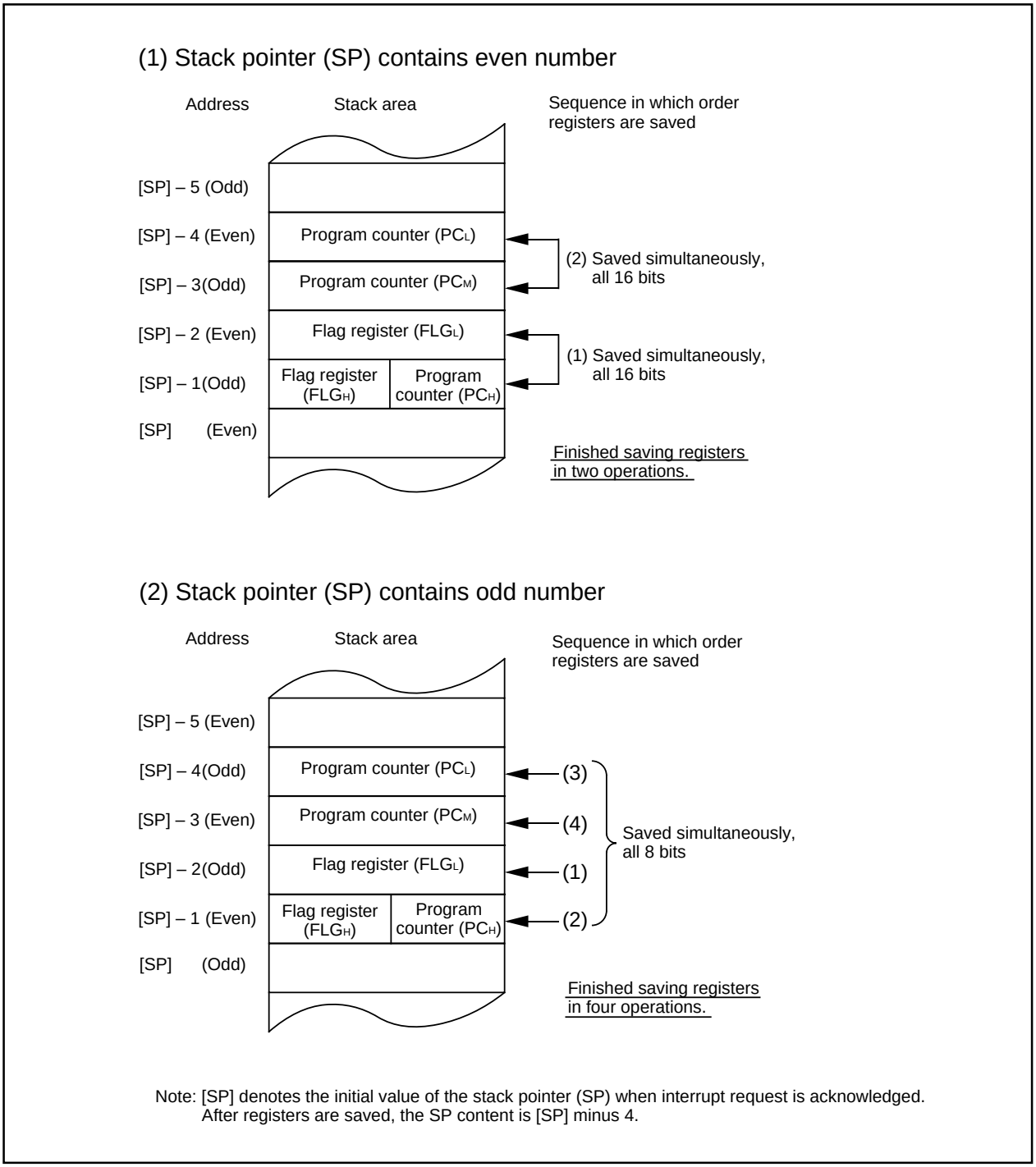


Figure 1.11.6. State of stack before and after acceptance of interrupt request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the content of the stack pointer (Note), at the time of acceptance of an interrupt equest, is even or odd. If the counter of the stack pointer (Note) is even, the counter 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 1.11.7 shows the operation of the saving registers.

Note: When any INT instruction in software numbers 32 to 63 has been executed, this is the stack pointer indicated by the U flag. Otherwise, it is the interrupt stack pointer (ISP).



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 1.11.7. 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 suspendedd process resumes.

Return the other registers saved by software within the interrupt routine using the POPM or similar instruction befoere 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 aqssigned 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 1.11.8 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 > $\overline{\text{NMI}}$ > $\overline{\text{DBC}}$ > Oscillation stop detection / Watchdog timer > Peripheral I/O > Single step > Address match

Figure 1.11.8. Hardware interrupts priorities

Interrupt Resolution Circuit

When two or more interupts are generated simultaneously, this circuit selects the interrupt with the highest priority level. Figure 1.11.9 shows the circuit that judges the interrupt priority level.

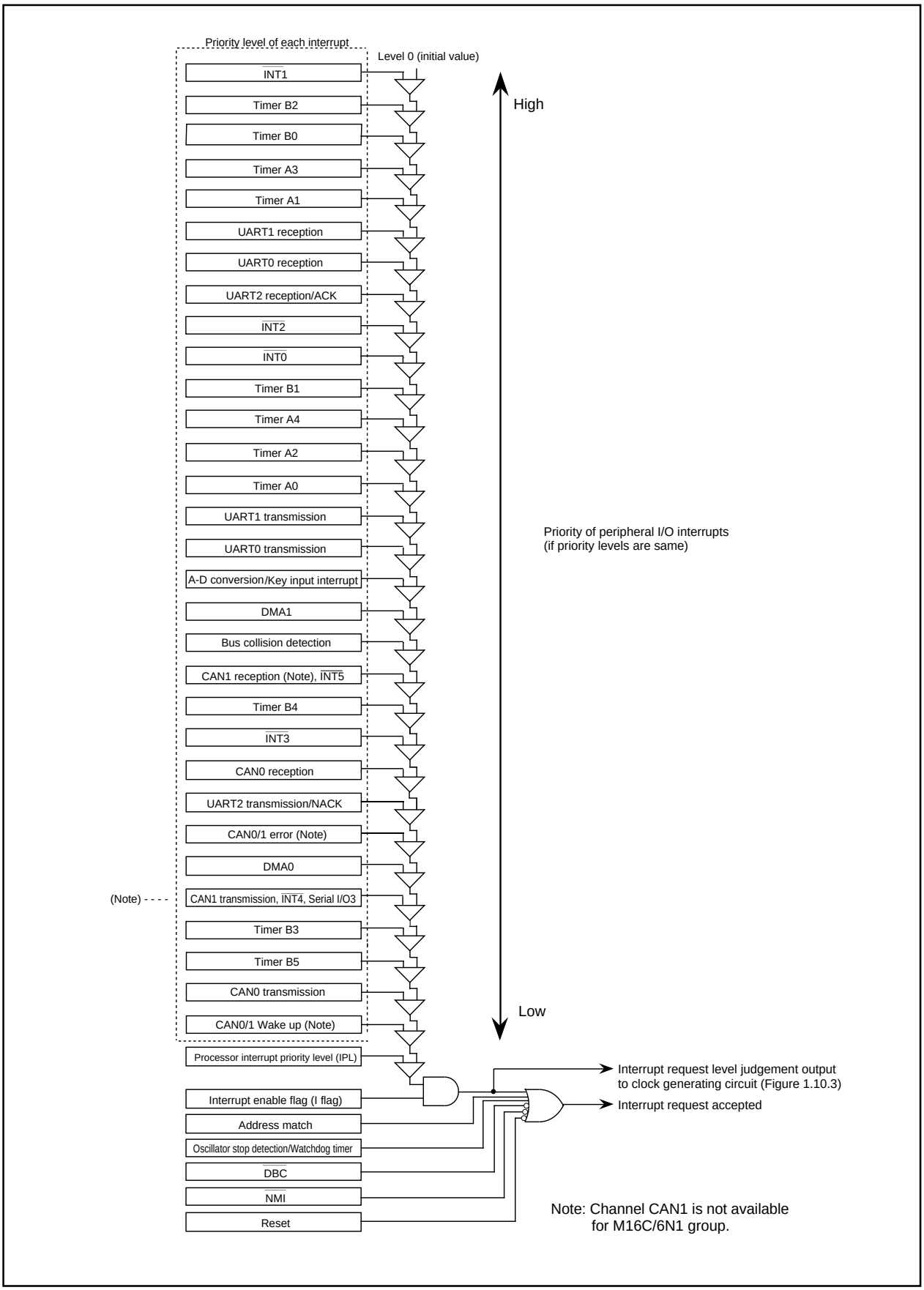


Figure 1.11.9. Maskable interrupts priorities (peripheral I/O interrupts)

INT Interrupt

INT0 to INT5 are triggered by the edges of external inputs. The edge polarity is selected using the polarity select bit.

Of interrupt control registers, address 0048$_{16}$ is used both as CAN1 reception and external interrupt INT5 input control register, and 0049$_{16}$ is used as S I/O3, CAN1 transmission and as external interrupt INT4 input control register. Use the interrupt request cause select bits (bit 0 and 1 at address 01DE$_{16}$ and bit 6 and 7 at address 01DF$_{16}$) to specify which interrupt request cause to select. After having set an interrupt request cause, be sure to clear the corresponding interrupt request bit before enabling an interrupt. (Note)

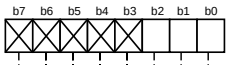
The interrupt control registers (address 0049$_{16}$ and address 0048$_{16}$) have the polarity-switching bit. Be sure to set this bit to "0" when selecting the S I/O3, CAN1 reseption or CAN1 transmission as the interrupt request cause. (Note)

As to external interrupt input, an interrupt can be generated both at the rising edge and at the falling edge by setting "1" in the INTi interrupt polarity switching bit of the interrupt request cause select register (address 01DF$_{16}$). To select both edges, set the polarity switching bit of the correponding interrupt control register to 'falling edge' ("0").

Note: Channel CAN1 is not available for M16C/6N1 group.

Figures 1.11.10 and 1.11.11 show the interrupt request cause select registers 0 and 1.

Interrupt request cause select register 0



Symbol: IFSR0 — Address: 01DE$_{16}$ — When reset: XXXXX000$_{16}$

Bit symbol	Bit name	Function	R	W
IFSR00	Interrupt request cause select bit	0 : C1TRMIC (Note) 1 : SIO3	○	○
IFSR01	Interrupt request cause select bit	0 : AD Converter 1 : Key On Wake Up	○	○
IFSR02	Reserved bit	Always set to "0"	○	○
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.			–	–

Note: Channel CAN1 is not available for M16C/6N1 group.

Figure 1.11.10. Interrupt request cause select register 0

Interrupt request cause select register 1

Symbol: IFSR1 — Address: 01DF$_{16}$ — When reset: 00$_{16}$

Bit symbol	Bit name	Fumction	R	W
IFSR10	INT0 interrupt polarity swiching bit	0 : One edge 1 : Two edges	○	○
IFSR11	INT1 interrupt polarity swiching bit	0 : One edge 1 : Two edges	○	○
IFSR12	INT2 interrupt polarity swiching bit	0 : One edge 1 : Two edges	○	○
IFSR13	INT3 interrupt polarity swiching bit	0 : One edge 1 : Two edges	○	○
IFSR14	INT4 interrupt polarity swiching bit	0 : One edge 1 : Two edges	○	○
IFSR15	INT5 interrupt polarity swiching bit	0 : One edge 1 : Two edges	○	○
IFSR16	Interrupt request cause select bit	0 : SIO3 / C1TRMIC (Note) 1 : INT4	○	○
IFSR17	Interrupt request cause select bit	0 : C1RECIC (Note) 1 : INT5	○	○

Note: Channel CAN1 is not available for M16C/6N1 group.

Figure 1.11.11. Interrupt request cause select register 1

NMI Interrupt

An NMI interrupt is generated when the input to the P85/NMI pin changes from "H" to "L". The NMI interrupt is a non-maskable external interrupt. The pin level can be checked in the port P85 register (bit 5 at address 03F016).

This pin cannot be used as a normal port input.

Key Input Interrupt

If the direction register of any of P104 to P107 is set for input and a falling edge is input to that port, a key input interrupt is generated. A key input interrupt can also be used as a key-on wakeup function for cancelling the wait mode or stop mode. However, if you intend to use the key input interrupt, do not use P104 to P107 as A-D input ports. Figure 1.11.12 shows the block diagram of the key input interrupt. Note that if an "L" level is input to any pin that has not been disabled for input, inputs to the other pins are not detected as an interrupt.

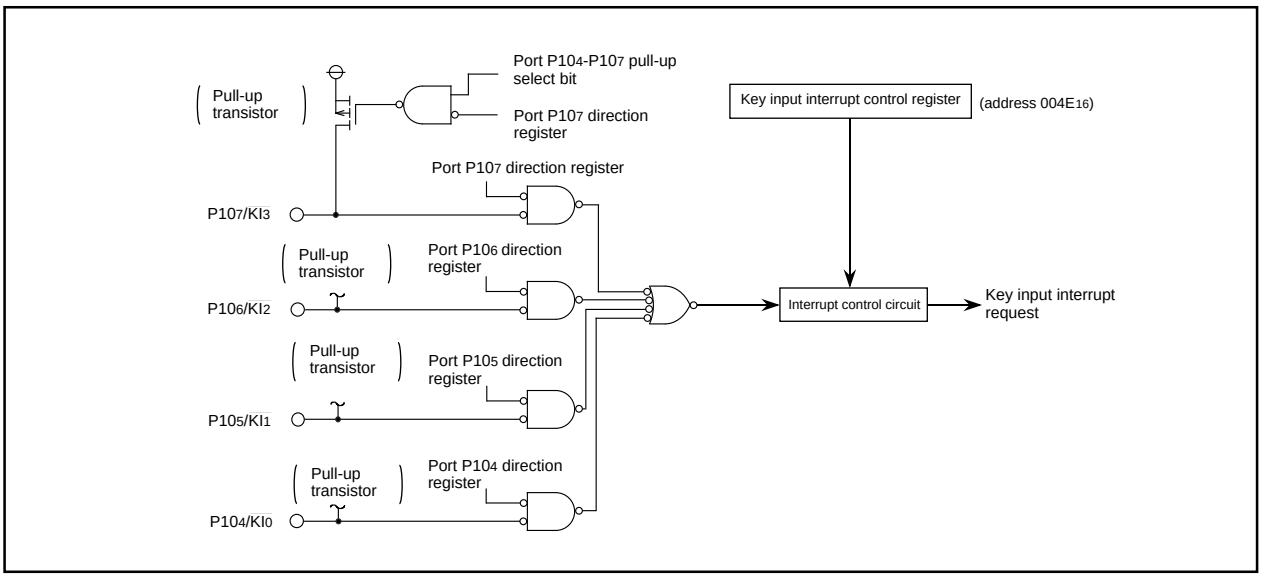


Figure 1.11.12. Block diagram of key input interrupt

CAN0/1 Wake up Interrupt

CAN0/1 wake up interrupt occurs when a falling edge is input to CRX0 or CRX1. Use the interrupt in stop/wait mode or CAN sleep mode. The CAN0/1 wake up interrupt is enabled only when the port is defined as the CAN port. One interrupt is allocated to CAN0/1. Figure 1.11.13 shows the block diagram of the CAN0/1 wake up interrupt. (Note)

Please note that the wake up message will be lost.

Note: There is only one CAN channel available for M16C/6N1 group.

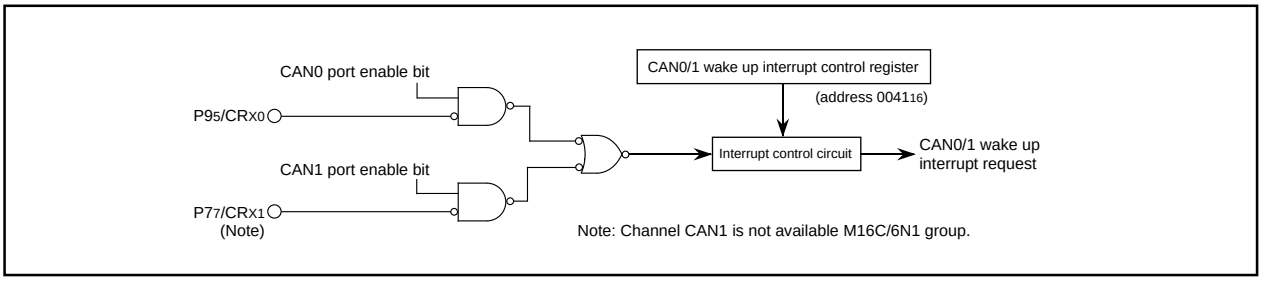


Figure 1.11.13. Block diagram of CAN0/1 wake up interrupt

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 1.11.14 shows the address match interrupt-related registers.

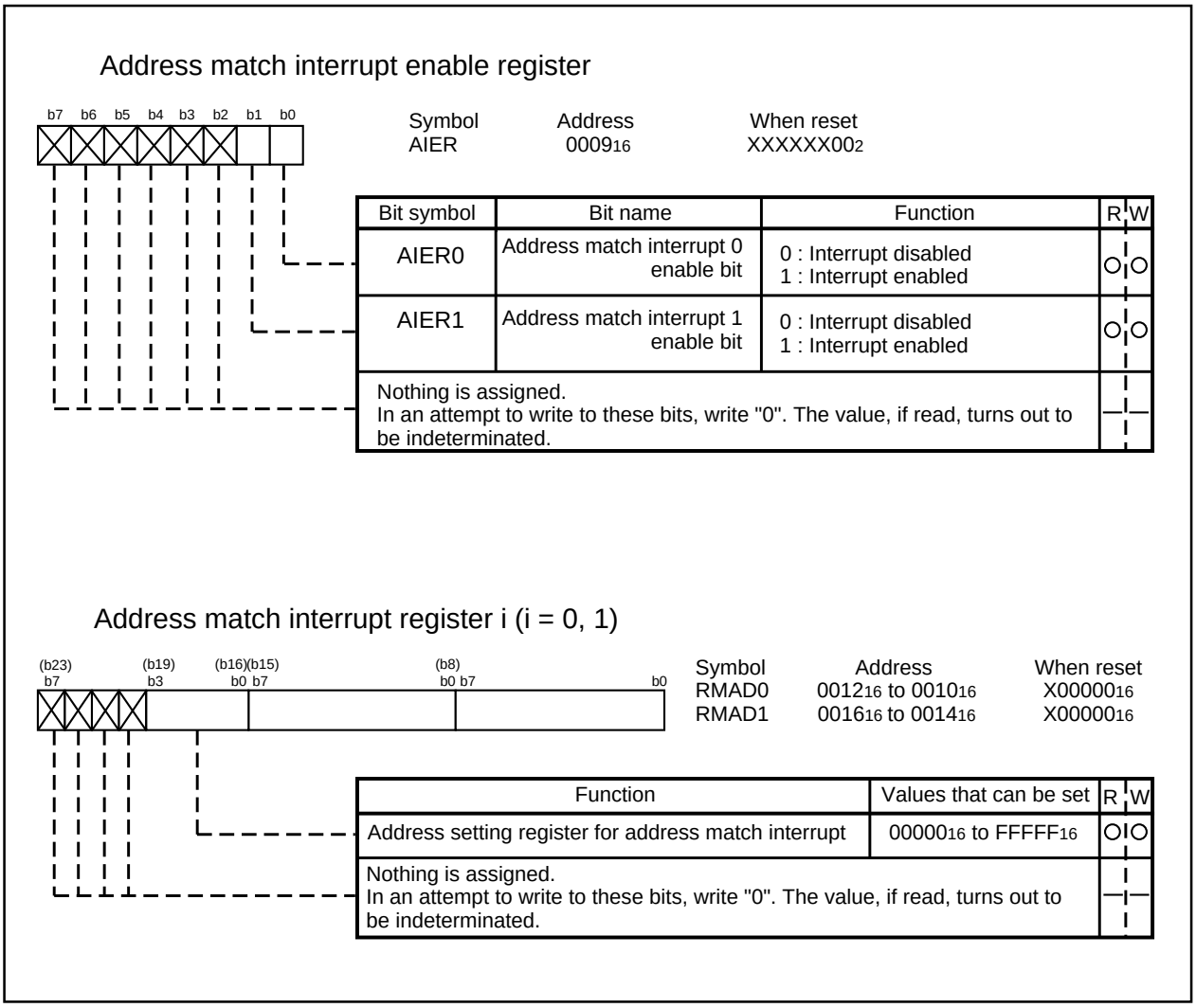


Figure 1.11.14. Address match interrupt-related registers

Precautions for Interrupts

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

- When maskable interrupt occurs, CPU reads 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".

  Even if the address 00000$_{16}$ is read out by software, "0" is set to the enabled highest priority interrupt source request bit. Therefore interrupt can be canceled and unexpected interrupt can occur.
  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 for the stack pointer before accepting an interrupt. When using the $\overline{\text{NMI}}$ interrupt, initialize the stack point at the beginning of a program. Concerning the first instruction immediately after reset, generating any interrupt including the $\overline{\text{NMI}}$ interrupt is prohibited.

(3) The $\overline{\text{NMI}}$ interrupt

- As for the $\overline{\text{NMI}}$ pin, an interrupt cannot be disabled. Connect it to the Vcc pin via a resistor (pull-up) if unused. Be sure to work on it.
- The $\overline{\text{NMI}}$ pin also serves as P8$_5$, which is exclusively for input. Reading the contents of the P8 register allows reading the pin value. Use the reading of this pin only for establishing the pin level at the time when the $\overline{\text{NMI}}$ interrupt is input.
- Do not reset the CPU with the input to the $\overline{\text{NMI}}$ pin being in the "L" state.
- Do not attempt to go into stop mode with the input to the $\overline{\text{NMI}}$ pin being in the "L" state. With the input to the $\overline{\text{NMI}}$ pin being in the "L" state, the CM10 is fixed to "0", so attempting to go into stop mode is turned down.
- Do not attempt to go into wait mode with the input to the $\overline{\text{NMI}}$ pin being in the "L" state. With the input to the $\overline{\text{NMI}}$ pin being in the "L" state, the CPU stops but the oscillation does not stop, so no power is saved. In this instance, the CPU is returned to the normal state by a later interrupt.
- Signals input to the $\overline{\text{NMI}}$ pin require an "L" level of 1 clock or more, from the operation clock of the CPU.

(4) External interrupt

- Either an "L" level or an "H" level of at least 250 ns width is necessary for the signal input to pins $\overline{\text{INT0}}$ through $\overline{\text{INT5}}$ regardless of the CPU operation clock.
- When the polarity of the $\overline{\text{INT0}}$ to $\overline{\text{INT5}}$ pins is changed, the interrupt request bit is sometimes set to "1". After changing the polarity, set the interrupt request bit to "0". Figure 1.11.15 shows the procedure for changing the $\overline{\text{INT}}$ interrupt generate factor.

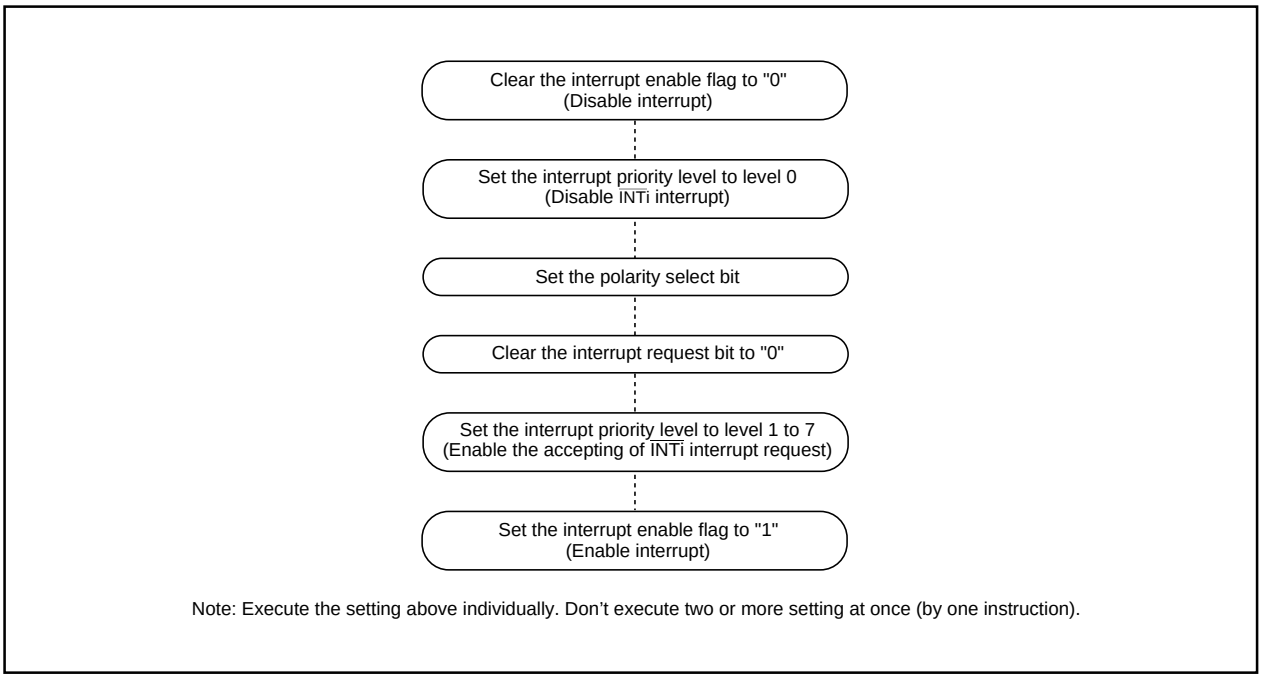


Figure 1.11.15. Switching condition of $\overline{\text{INT}}$ interrupt request

(5) Rewrite the interrupt control register

- To rewrite the interrupt control register, do so at a point when no interrupt request for that register can be generated. If there is possibility of the interrupt request to 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                 ;Four NOP instructions are required when using HOLD function.
    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         ;Push Flag register onto stack
    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 (four when using the HOLD function) or dummy read is 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 an 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. Therefore, we recommend using the watchdog timer to improve reliability of a system. 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 XIN is selected as BCLK, bit 7 of the watchdog timer control register (address 000F16) selects the prescaler division ratio (by 16 or by 128). When XCIN is selected as the BCLK, the prescaler is set for division by 2 regardless of bit 7 of the watchdog timer control register (address 000F16). 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 XIN chosen for BCLK

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

With XCIN 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 16 MHz and that 16 has been chosen for the dividing ratio of the prescaler, then the watchdog timer's period becomes approximately 32.8 ms.

The watchdog timer is initialized by writing to the watchdog timer start register (address 000E16) 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 000E16). In stop mode, wait mode and hold state, the watchdog timer and prescaler are stopped. Counting is resumed from the held value when the modes or state are released.
Figure 1.12.1 shows the block diagram of the watchdog timer. Figure 1.12.2 shows the watchdog timer-related registers.

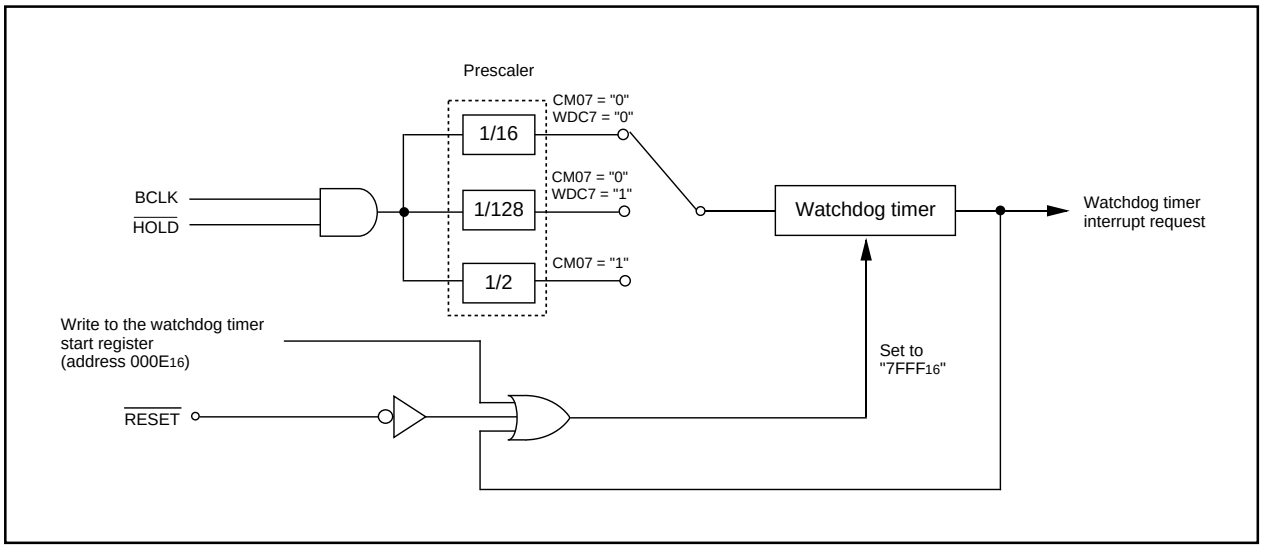


Figure 1.12.1. Block diagram of watchdog timer

Watchdog Timer during Wait and Clock Stop Modes

The watchdog timer is supplied by the BCLK. If the BCLK stops, the watchdog timer stops also. When executing a WAIT instruction, the BCLK stops and watchdog timer operation is suspended. The same applies when entering Stop Mode (CM10 = "1"). Watchdog timer operation recommences when either mode is cancelled.

Watchdog timer control register

b7	b6	b5	b4	b3	b2	b1	b0
	0	0					

Symbol	Address	When reset
WDC	$000F_{16}$	$000XXXXX_2$

Bit symbol	Bit name	Function	R	W
High-order bits of watchdog timer			○	×
Reserved bit		This bit can neither be set nor reset.	○	○
Reserved bit		Must always be set to "0"	○	○
WDC7	Prescaler select bit	0 : Divided by 16 1 : Divided by 128	○	○

Watchdog timer start register

b7	b0

Symbol	Address	When reset
WDTS	$000E_{16}$	Indeterminate

Function	R	W
The watchdog timer is initialized and starts counting after a write instruction to this register. The watchdog timer value is always initialized to "$7FFF_{16}$" regardless of whatever value is written.	×	○

Figure 1.12.2. 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, performing the cycle steal 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 1.13.1 shows the block diagram of the DMAC. Table 1.13.1 shows the DMAC specifications. Figures 1.13.2 to 1.13.4 show the registers used by the DMAC.

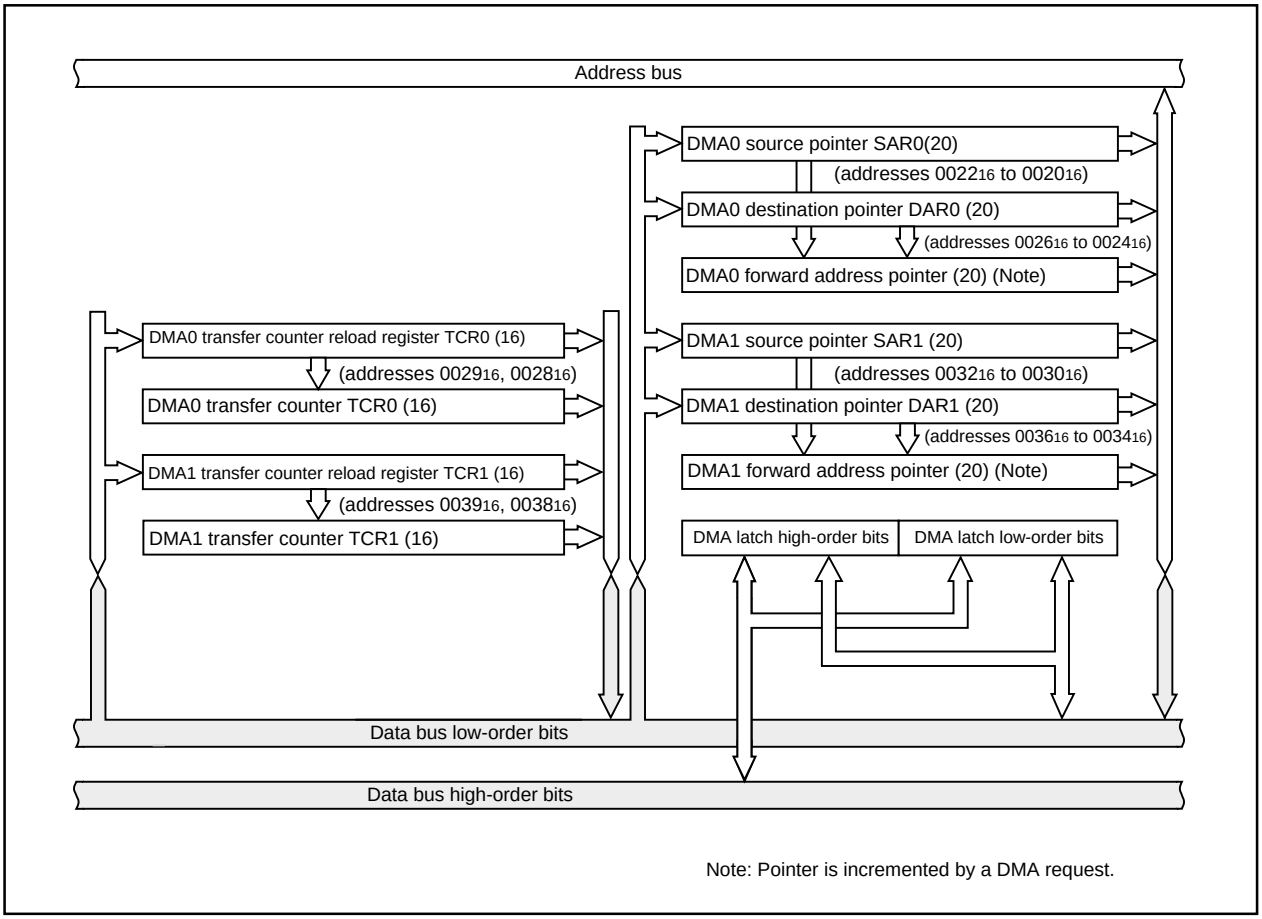


Figure 1.13.1. Block diagram of DMAC

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.

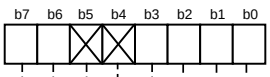
If the DMAC is active (the DMA enable bit is set to "1"), data transfer starts every time a DMA transfer request signal occurs. If the cycle of the occurrences of DMA transfer request signals is higher than the DMA transfer cycle, there can be instances in which the number of transfer requests doesn't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

Table 1.13.1. 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) or both edge Timer A0 to timer A4 interrupt requests Timer B0 to timer B5 interrupt requests UART0 transmission and reception interrupt requests UART1 transmission and reception interrupt requests UART2 transmission and reception interrupt requests Serial I/O3 interrupt request 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/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.
Reload timing for forward address pointer and transfer counter	At the time of starting data transfer immediately after turning the DMAC active, 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.

DMA0 request cause select register



Symbol	Address	When reset
DM0SL	$03B8_{16}$	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 $\overline{INT_0}$ pin 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 (DMS=0) /two edges of $\overline{INT_0}$ pin (DMS=1) 0 1 1 1 : Timer B0 (DMS=0) Timer B3 (DMS=1) 1 0 0 0 : Timer B1 (DMS=0) Timer B4 (DMS=1) 1 0 0 1 : Timer B2 (DMS=0) Timer B5 (DMS=1) 1 0 1 0 : UART0 transmit 1 0 1 1 : UART0 receive 1 1 0 0 : UART2 transmit 1 1 0 1 : UART2 receive 1 1 1 0 : A-D conversion 1 1 1 1 : UART1 transmit	○	○
DSEL1			○	○
DSEL2			○	○
DSEL3			○	○
Nothing is assigned. These bits can neither be set nor reset. When read, the value of these bits is "0".			–	–
DMS	DMA request cause expansion bit	0 : Normal 1 : Expanded cause	○	○
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")	○	○

Figure 1.13.2. DMAC register (1)

DMA1 request cause select register

b7	b6	b5	b4	b3	b2	b1	b0
		X	X				

Symbol	Address	When reset
DM1SL	03BA16	0016

Bit symbol	Bit name	Function	R	W
DSEL0	DMA request cause select bit	b3 b2 b1 b0 0 0 0 0 : Falling edge of $\overline{INT_1}$ pin 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(DMS=0) /serial I/O3 (DMS=1) 0 1 1 0 : Timer A4 (DMS=0) 0 1 1 1 : Timer B0 (DMS=0) /two edges of $\overline{INT_1}$ (DMS=1) 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 : UART2 transmit 1 1 0 1 : UART2 receive 1 1 1 0 : A-D conversion 1 1 1 1 : UART1 receive	○	○
DSEL1			○	○
DSEL2			○	○
DSEL3			○	○
Nothing is assigned. These bits can neither be set nor reset. When read, the value of these bits is "0".			–	–
DMS	DMA request cause expansion select bit	0 : Normal 1 : Expanded cause	○	○
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")	○	○

DMAi control register

b7	b6	b5	b4	b3	b2	b1	b0
X	X						

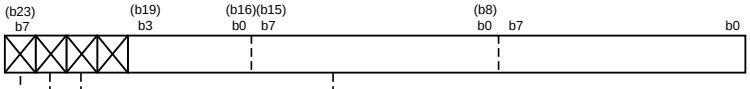
Symbol	Address	When reset
DMiCON(i=0,1)	002C16, 003C16	00000X002

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. These bits can neither be set nor reset. When read, the value of these bits is "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 1.13.3. DMAC register (2)

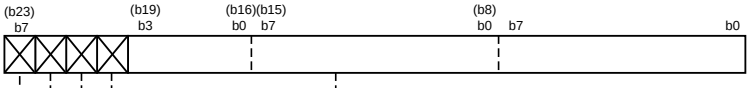
DMAi source pointer (i = 0, 1)



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. These bits can neither be set nor reset. When read, the value of these bits is "0".		—	—

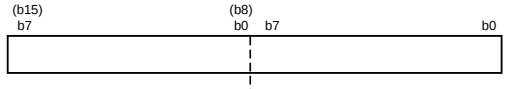
DMAi destination pointer (i = 0, 1)



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. These bits can neither be set nor reset. When read, the value of these bits is "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 1.13.4. DMAC register (3)

(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. In memory expansion mode and microprocessor mode, the number of read and write bus cycles also depends on the level of the BYTE pin. Also, the bus cycle itself is longer when software waits are inserted.

(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.

(b) Effect of BYTE pin level

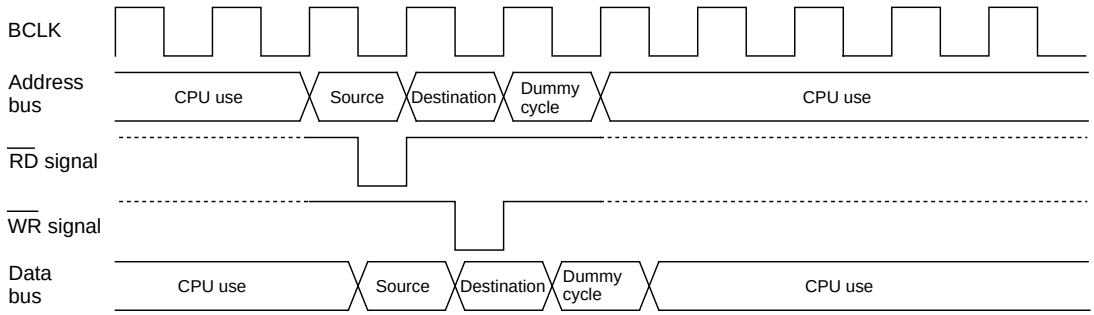
When transferring 16-bit data over an 8-bit data bus (BYTE pin = "H") in memory expansion mode and microprocessor mode, the 16 bits of data are sent in two 8-bit blocks. Therefore, two bus cycles are required for reading the data and two are required for writing the data. Also, in contrast to when the CPU accesses internal memory, when the DMAC accesses internal memory (internal ROM, internal RAM, and SFR), these areas are accessed using the data size selected by the BYTE pin.

(c) Effect of software wait

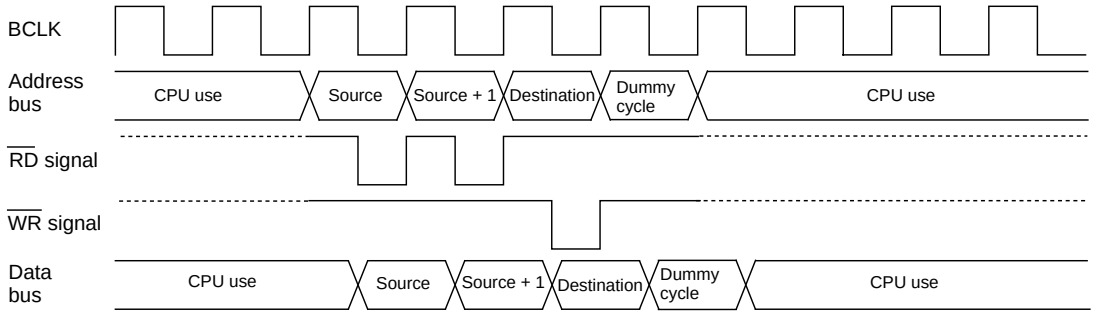
When the SFR area or a memory area with a software wait is accessed, the number of cycles is increased for the wait by 1 bus cycle. The length of the cycle is determined by BCLK.

Figure 1.13.5 shows the example of the transfer cycles 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. For example (2) in Figure 1.13.5, if data is being transferred in 16-bit units on an 8-bit bus, two bus cycles are required for both the source read cycle and the destination write cycle.

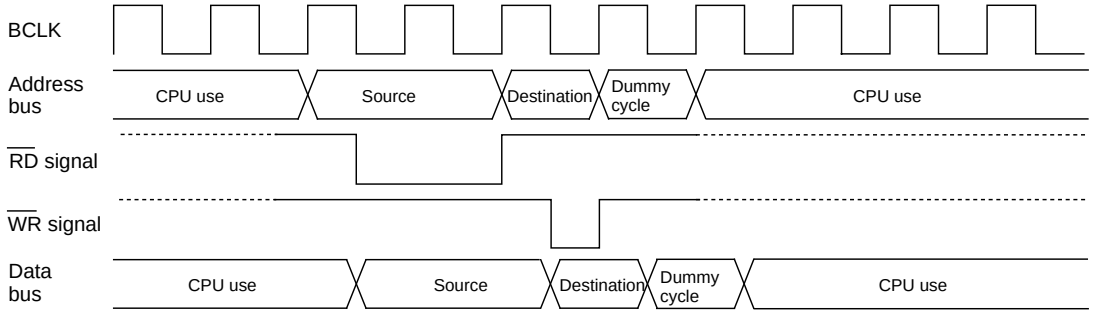
(1) 8-bit transfers
16-bit transfers and the source address is even.



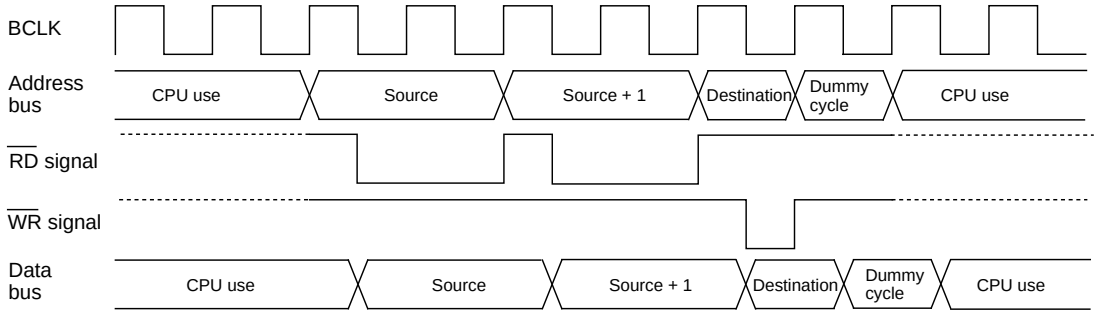
(2) 16-bit transfers and the source address is odd
Transferring 16-bit data on an 8-bit data bus (In this case, there are also two destination write cycles).



(3) One wait is inserted into the source read under the conditions in (1)



(4) One wait is inserted into the source read under the conditions in (2)
(When 16-bit data is transferred on an 8-bit data bus, there are two destination cycles).



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

Figure 1.13.5. Example of the transfer cycles for a source read

(2) DMAC transfer cycles

Any combination of even or odd transfer read and write addresses is possible. Table 1.13.2 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 1.13.2. No. of DMAC transfer cycles

Transfer unit	Bus width	Access address	Single-chip mode		Memory expansion mode Microprocessor mode	
			No. of read cycles	No. of write cycles	No. of read cycles	No. of write cycles
8-bit transfers (DMBIT= "1")	16-bit (BYTE= "L")	Even	1	1	1	1
		Odd	1	1	1	1
	8-bit (BYTE = "H")	Even	—	—	1	1
		Odd	—	—	1	1
16-bit transfers (DMBIT= "0")	16-bit (BYTE = "L")	Even	1	1	1	1
		Odd	2	2	2	2
	8-bit (BYTE = "H")	Even	—	—	2	2
		Odd	—	—	2	2

Coefficient j, k

Internal memory			External memory		
InternalROM/RAM No wait	InternalROM/RAM With wait	SFR area	Separate bus No wait	Separate bus With wait	Multiplex bus
1	2	2	1	2	3

DMA Enable Bit

Setting the DMA enable bit to "1" makes the DMAC active. The DMAC carries out the following operations at the time data transfer starts immediately after DMAC is turned active.

(1) Reloads the value of the source pointer or the destination pointer - the one specified for the forward direction - to the forward direction address pointer.

(2) Reloads the value of the transfer counter reload register to the transfer counter.

Thus overwriting "1" to the DMA enable bit with the DMAC being active carries out the operations described above, so the DMA operates again, beginning from the initial state instantly when "1" is overwritten to the DMA enable bit.

DMA Request Bit

The DMAC can generate a DMA transfer request signal triggered by a factor chosen in advance out of DMA request factors for each channel.

DMA request factors include the following.

- Factors effected by using the interrupt request signals from the built-in peripheral functions and software DMA factors (internal factors) effected by a program.
- External factors effected by utilizing the input from external interrupt signals.

The DMA request factor is selected by the DMA request cause select bits (bit 0 to 3 at addresses $03B8_{16}$ and $03BA_{16}$).

The DMA request bit turns to "1" if the DMA transfer request signal occurs regardless of the DMAC's state (Whether the DMA enable bit is set to "1" or "0"). It turns to "0" immediately before data transfer starts.

In addition, the DMA request bit can be set to "0" by use of a program, but cannot be set to "1".

There can be instances in which a change in DMA request cause select bits cause the DMA request bit to turn to "1". So be sure to set the DMA request bit to "0" after the DMA request cause select bits are changed.

The DMA request bit turns to "1" if a DMA transfer request signal occurs, and turns to "0" immediately before data transfer starts. If the DMAC is active, data transfer starts immediately, so the value of the DMA request bit, if read by use of a program, turns out to be "0" in most cases. To examine whether the DMAC is active, read the DMA enable bit.

Here follows the timing of changes in the DMA request bit.

(1) Internal factors

Except the DMA request factors triggered by software, the timing for the DMA request bit to turn to "1" due to an internal factor is the same as the timing for the interrupt request bit of the interrupt control register to turn to "1" due to several factors.

Turning the DMA request bit to "0" due to an internal factor is timed to be effected immediately before the transfer starts.

(2) External factors

An external factor is caused to occur by the leading edge of input from the $\overline{\text{INTi}}$ pin (i depends on which DMAC channel is used).

Selecting the $\overline{\text{INTi}}$ pins as external factors using the DMA request cause select bits cause input from these pins to become the DMA transfer request signals.

The timing for the DMA request bit to turn to "1" when an external factor is selected, synchronizes with the signal's edge applicable to the function specified by the DMA request cause select bits (synchronizes with the trailing edge of the input signal to each $\overline{\text{INTi}}$ pin, for example).

With an external factor selected, the DMA request bit is timed to turn to "0" immediately before data transfer starts similarly to the state in which an internal factor is selected.

(3) The priorities of channels and DMA transfer timing

If a DMA transfer request signal falls on a single sampling cycle (a sampling cycle means one period from the leading edge to the trailing edge of BCLK), the DMA request bits of applicable channels concurrently turn to "1". If the channels are active at that moment, DMA0 is given a high priority to start data transfer. When DMA0 finishes data transfer, it gives the bus right to the CPU. When the CPU finishes single bus access, then DMA1 starts data transfer and gives the bus right to the CPU after transfer is finished.
An example which shows DMA transfer is carried out in minimum cycles at the time when DMA transfer request signals due to external factors concurrently occur, is given below.
Figure 1.13.6 An example of DMA transfer effected by external factors.

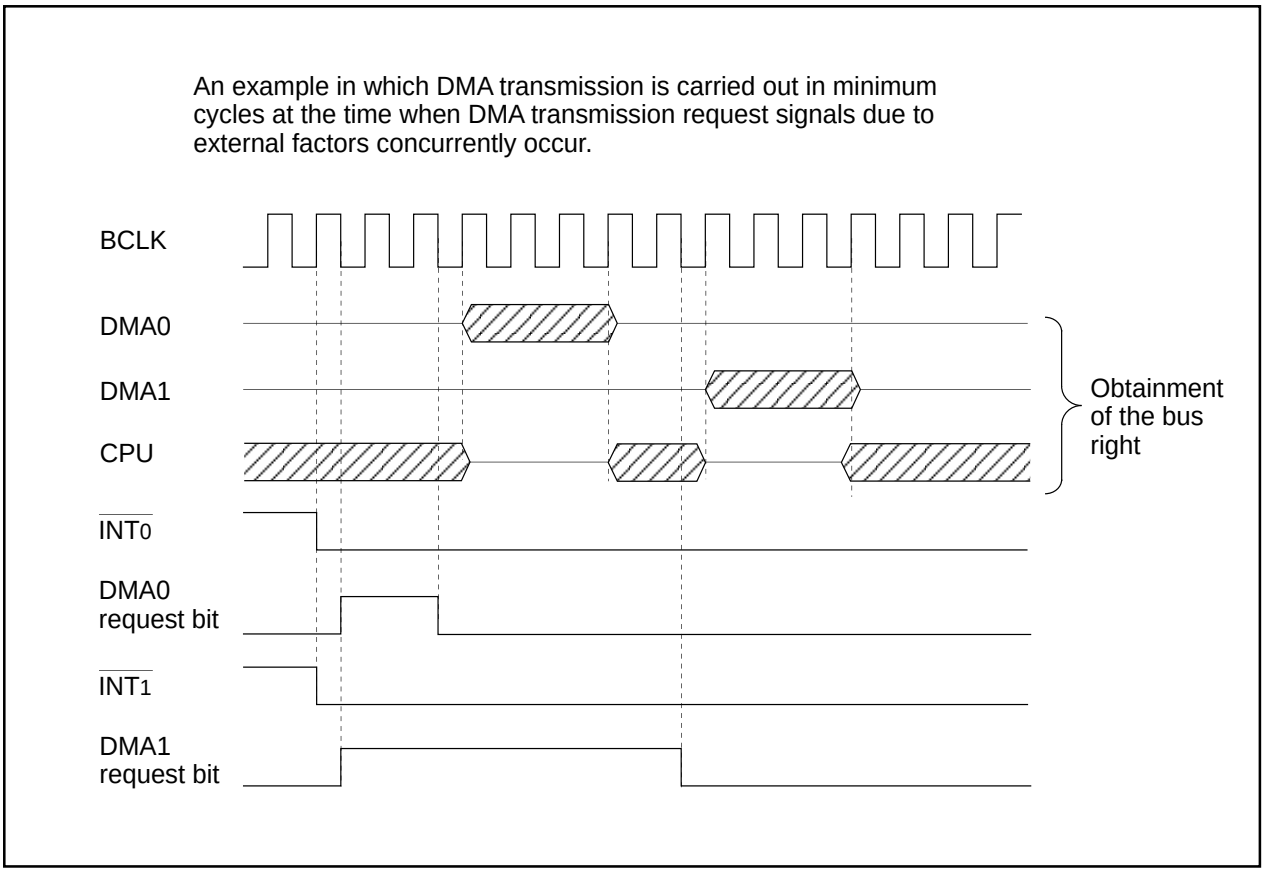
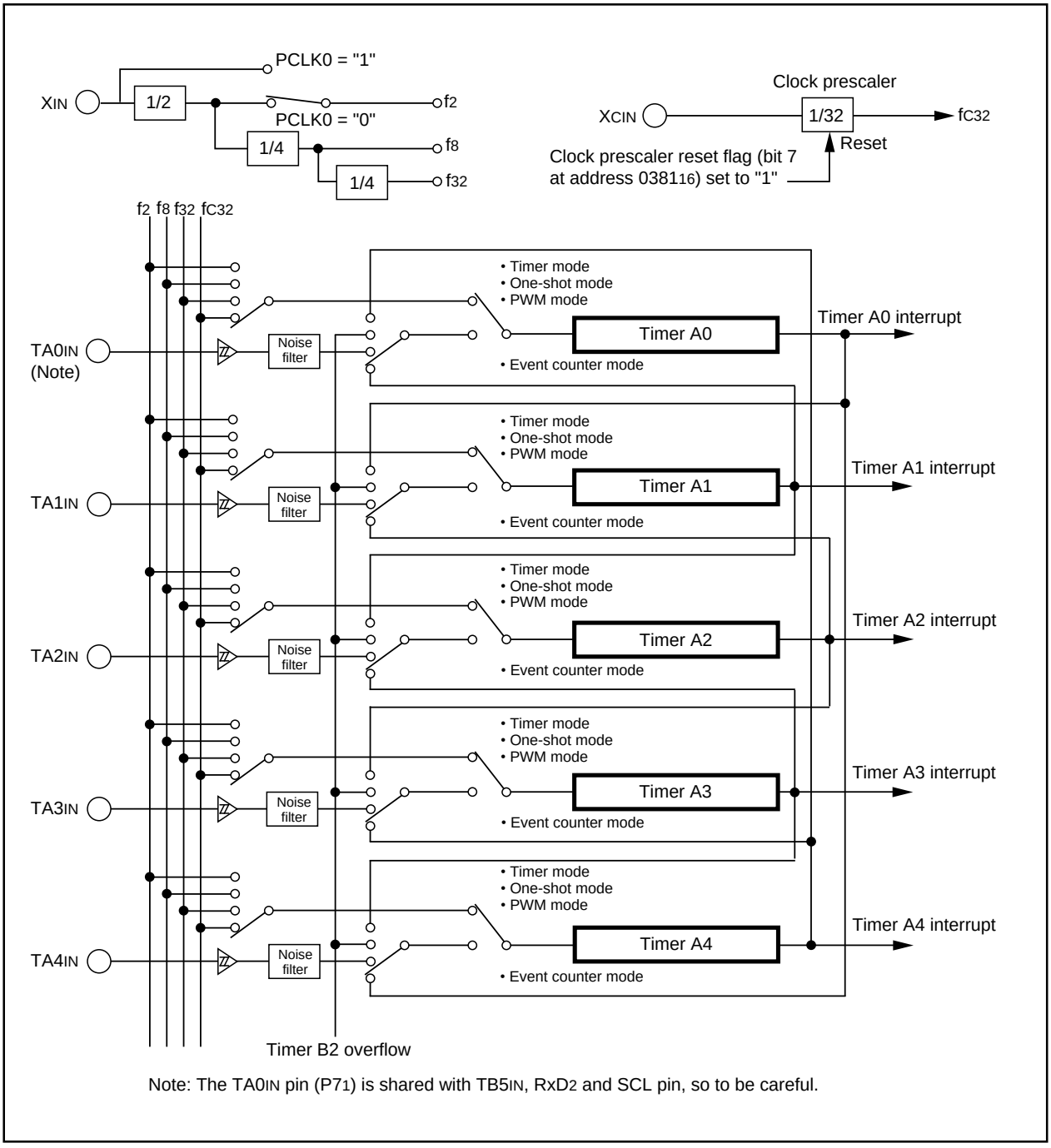


Figure 1.13.6. An example of DMA transfer effected by external factors

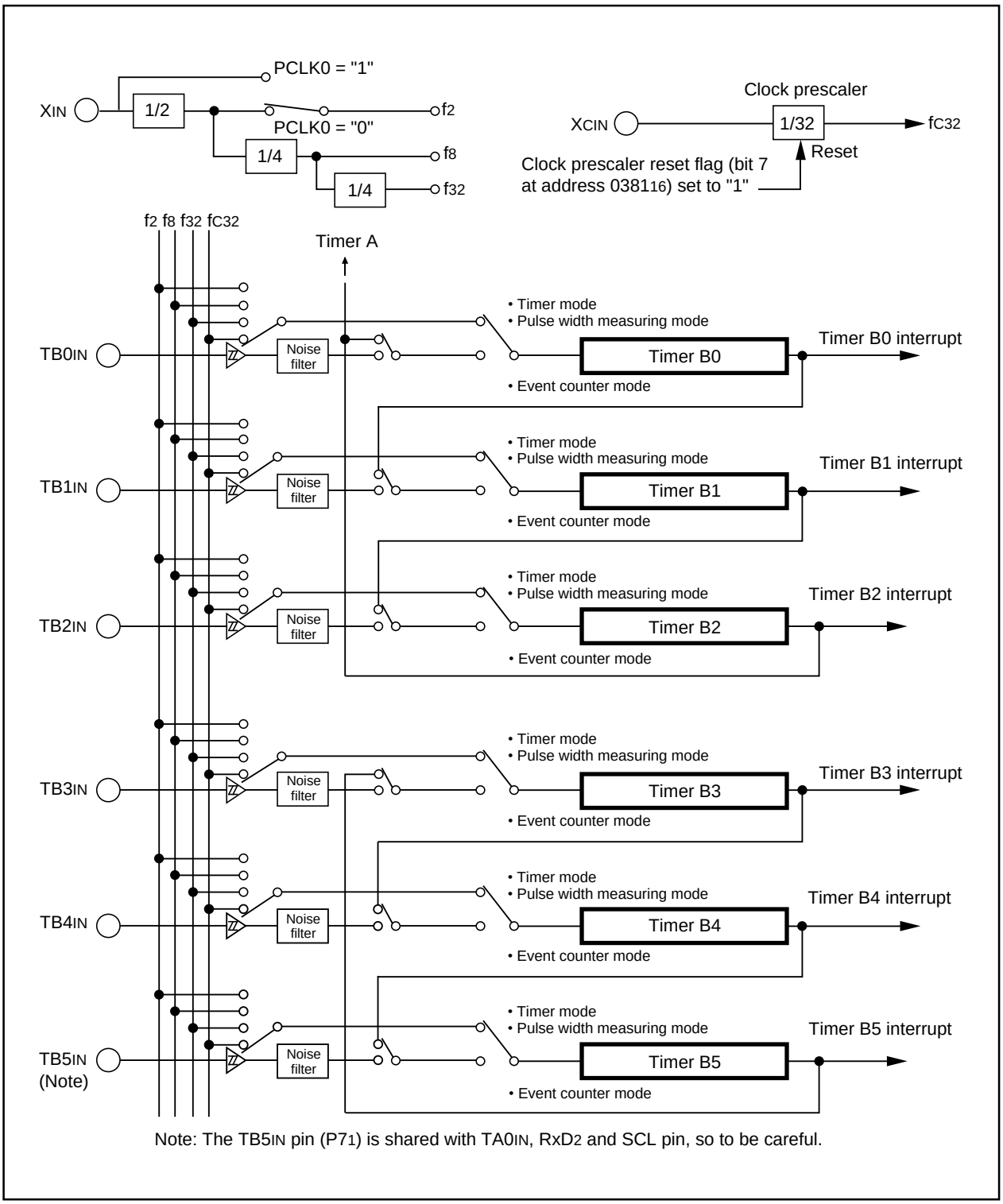
Timer

There are eleven 16-bit timers. These timers can be classified by function into timers A (five) and timers B (six). All these timers function independently. Figures 1.14.1 and 1.14.2 show the block diagram of timers.



Note: The TA0IN pin (P71) is shared with TB5IN, RxD2 and SCL pin, so to be careful.

Figure 1.14.1. Timer A block diagram



Note: The TB5IN pin (P71) is shared with TA0IN, RxD2 and SCL pin, so to be careful.

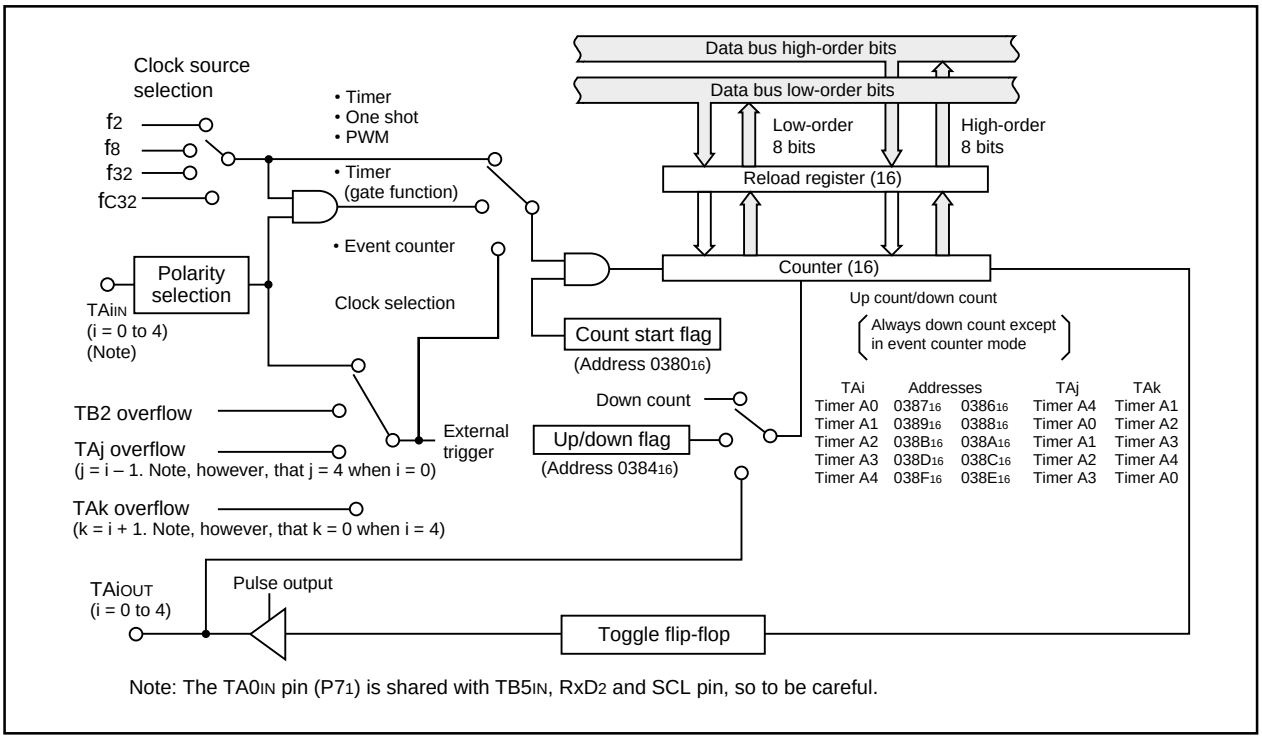
Figure 1.14.2. Timer B block diagram

Timer A

Figure 1.14.3 shows the block diagram of timer A. Figures 1.14.4 to 1.14.6 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 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.



Note: The TA0IN pin (P71) is shared with TB5IN, RxD2 and SCL pin, so to be careful.

Figure 1.14.3. Block diagram of timer A

Timer Ai mode register

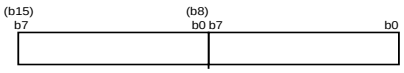
Symbol	Address	When reset
TAiMR(i=0 to 4)	039616 to 039A16	0016

b7 b6 b5 b4 b3 b2 b1 b0

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 : One-shot timer mode 1 1 : Pulse width modulation (PWM) mode	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
MR2			○	○
MR3			○	○
TCK0	Count source select bit (Function varies with each operation mode)		○	○
TCK1			○	○

Figure 1.14.4. Timer A-related registers (1)

Timer Ai register (Note 1)



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 $FFFF_{16}$	O	O
• Event counter mode Counts pulses from an external source or timer overflow	0000_{16} to $FFFF_{16}$	O	O
• One-shot timer mode Counts a one shot width	0000_{16} to $FFFF_{16}$ (Note 2,4)	X	O
• Pulse width modulation mode (16-bit PWM) Functions as a 16-bit pulse width modulator	0000_{16} to $FFFE_{16}$ (Note 3,4)	X	O
• 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} (High-order address) 00_{16} to FF_{16} (Low-order address) (Note 3,4)	X	O

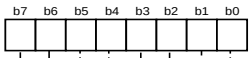
Note 1: Read and write data in 16-bit units.
Note 2: When the timer Ai register is set to "0000_{16}", the counter does not operate and the timer Ai interrupt request is not generated. When the pulse is set to output, the pulse does not output from the TAiOUT pin.
Note 3: When the timer Ai register is set to "0000_{16}", the pulse width modulator does not operate and the output level of the TAiOUT pin remains "L" level, therefore the timer Ai interrupt request is not generated. This also occurs in the 8-bit pulse width modulator mode when the significant 8 high-order bits in the timer Ai register are set to "00_{16}".
Note 4: Use MOV instruction to write to this register.

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	O	O
TA1S	Timer A1 count start flag		O	O
TA2S	Timer A2 count start flag		O	O
TA3S	Timer A3 count start flag		O	O
TA4S	Timer A4 count start flag		O	O
TB0S	Timer B0 count start flag		O	O
TB1S	Timer B1 count start flag		O	O
TB2S	Timer B2 count start flag		O	O

Up/down flag (Note 1)



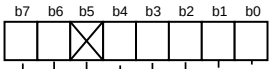
Symbol	Address	When reset
UDF	0384_{16}	00_{16}

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	O	O
TA1UD	Timer A1 up/down flag		O	O
TA2UD	Timer A2 up/down flag		O	O
TA3UD	Timer A3 up/down flag		O	O
TA4UD	Timer A4 up/down flag		O	O
TA2P	Timer A2 two-phase pulse signal processing select bit	0 : two-phase pulse signal processing disabled 1 : two-phase pulse signal processing enabled (Note 2) When not using the two-phase pulse signal processing function, set the select bit to "0"	X	O
TA3P	Timer A3 two-phase pulse signal processing select bit		X	O
TA4P	Timer A4 two-phase pulse signal processing select bit		X	O

Note 1: Use MOV instruction to write to this register.
Note 2: Set the TAiIN and TAiOUT pins correspondent port direction registers to "0".

Figure 1.14.5. Timer A-related registers (2)

One-shot start flag

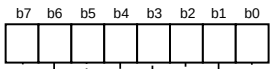


Symbol	Address	When reset
ONSF	0382_{16}	$00X00000_2$

Bit symbol	Bit name	Function	R	W
TA0OS	Timer A0 one-shot start flag	1 : Timer start When read, the value is "0"	○	○
TA1OS	Timer A1 one-shot start flag		○	○
TA2OS	Timer A2 one-shot start flag		○	○
TA3OS	Timer A3 one-shot start flag		○	○
TA4OS	Timer A4 one-shot start flag		○	○
Nothing is assigned. This bit can neither be set nor reset. When read, its content is indeterminate.			—	—
TA0TGL	Timer A0 event/trigger select bit	b7 b6 0 0 : Input on $TA0_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA4 overflow is selected 1 1 : TA1 overflow is selected	○	○
TA0TGH			○	○

Note: Set the corresponding port direction register to "0".

Trigger select register

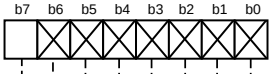


Symbol	Address	When reset
TRGSR	0383_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TA1TGL	Timer A1 event/trigger select bit	b1 b0 0 0 : Input on $TA1_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA0 overflow is selected 1 1 : TA2 overflow is selected	○	○
TA1TGH			○	○
TA2TGL	Timer A2 event/trigger select bit	b3 b2 0 0 : Input on $TA2_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA1 overflow is selected 1 1 : TA3 overflow is selected	○	○
TA2TGH			○	○
TA3TGL	Timer A3 event/trigger select bit	b5 b4 0 0 : Input on $TA3_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA2 overflow is selected 1 1 : TA4 overflow is selected	○	○
TA3TGH			○	○
TA4TGL	Timer A4 event/trigger select bit	b7 b6 0 0 : Input on $TA4_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA3 overflow is selected 1 1 : TA0 overflow is selected	○	○
TA4TGH			○	○

Note: Set the corresponding port direction register to "0".

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")	○	○

Figure 1.14.6. Timer A-related registers (3)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 1.14.1.) Figure 1.14.7 shows the timer Ai mode register in timer mode.

Table 1.14.1. Specifications of timer mode

Item	Specification
Count source	f_2, 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: 039616 to $039A16$ — When reset: 0016

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) 1 1 : Timer counts only when TAiIN pin is held "H" (Note 3)	○	○
MR2			○	○
MR3	0 (Must always be fixed to "0" in timer mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f_2 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 1.14.7. 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 1.14.2 lists timer specifications when counting a single-phase external signal. Figure 1.14.8 shows the timer Ai mode register in event counter mode.

Table 1.14.2 lists timer specifications when counting a two-phase external signal. Figure 1.14.9 shows the timer Ai mode register in event counter mode.

Table 1.14.2. 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, it reloads the reload register con tents before continuing counting (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	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
(When not using two-phase pulse signal processing)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

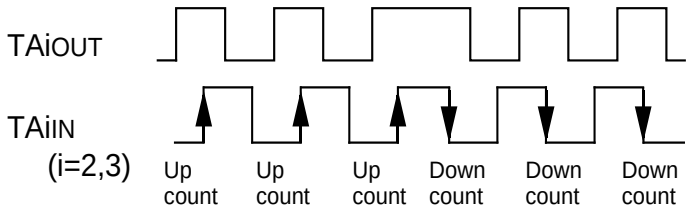
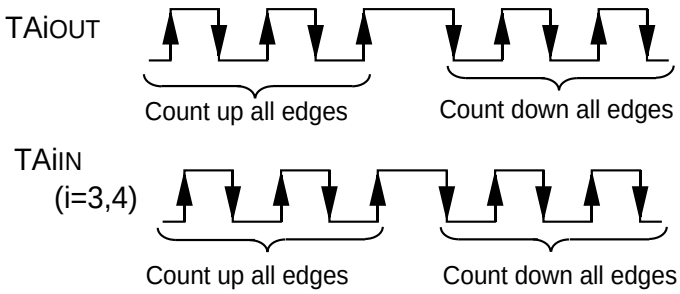
Symbol: TAiMR(i = 0, 1) Address: 039616, 039716 When reset: 0016

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 038216 and 038316).
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 1.14.8. Timer Ai mode register in event counter mode

Table 1.14.3. Timer specifications in event counter mode (when processing two-phase pulse signal with timers 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 1)
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 (Set the TAiIN pin correspondent port direction register to "0")
TAiOUT pin function	Two-phase pulse input (Set the TAiOUT pin correspondent port direction register to "0")
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 (Note 2)	• Normal processing operation (timer A2 and timer A3) 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 (timer A3 and timer A4) 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 1: This does not apply when the free-run function is selected.
Note 2: Timer A3 can be selected. Timer A2 is fixed to normal processing operation, and timer A4 is fixed to multiply-by-4 processing operation.

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	○	○
TMOD1			○	○
MR0	0 (Must always be "0" when using two-phase pulse signal processing)		○	○
MR1	0 (Must always be "0" when using two-phase pulse signal processing)		○	○
MR2	1 (Must always be "1" when using two-phase pulse signal processing)		○	○
MR3	0 (Must always be "0" when using two-phase pulse signal processing)		○	○
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Two-phase pulse processing operation select bit (Note 1)(Note 2)	0 : Normal processing operation 1 : Multiply-by-4 processing operation	○	○

Note 1: This bit is valid for timer A3 mode register.
Timer A2 is fixed to normal processing operation, and timer A4 is fixed to multiply-by-4 processing operation.

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 (address 0383$_{16}$) to "00".

Figure 1.14.9. Timer Ai mode register in event counter mode

(3) One-shot timer mode

In this mode, the timer operates only once. (See Table 1.14.4.) When a trigger occurs, the timer starts up and continues operating for a given period. Figure 1.14.10 shows the timer Ai mode register in one-shot timer mode.

Table 1.14.4. Timer specifications in one-shot timer mode

Item	Specification
Count source	f_2, 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 × (m+1) / fi n : values set to timer Ai register's high-order address • Cycle time $(2^8-1)\times(m+1)$ / 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 progressWhen 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: 039616 to $039A16$ When reset: 0016

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 bits	○	○
MR3	0 (Must always be "0" in one-shot timer mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f_2 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: Valid only when the TAiIN pin is selected by the event/trigger select bit (addresses 038216 and 038316). If timer overflow is selected, this bit can be "1" or "0".
Note 3: Set the corresponding port direction register to "0".

Figure 1.14.10. 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 1.14.5.) In this mode, the counter functions as either a 16-bit pulse width modulator or an 8-bit pulse width modulator. Figure 1.14.11 shows the timer Ai mode register in pulse width modulation mode. Figure 1.14.12 shows the example of how a 16-bit pulse width modulator operates. Figure 1.14.13 shows the example of how an 8-bit pulse width modulator operates.

Table 1.14.5. Timer specifications in pulse width modulation mode

Item	Specification
Count source	f_2, 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 × (m+1) / fi n : values set to timer Ai register's high-order address • Cycle time $(2^8-1) \times (m+1)$ / 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 progressWhen 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 "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_2 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 0382_{16} and 0383_{16}). If timer overflow is selected, this bit can be "1" or "0".
Note 2: Set the corresponding port direction register to "0".

Figure 1.14.11. Timer Ai mode register in pulse width modulation mode

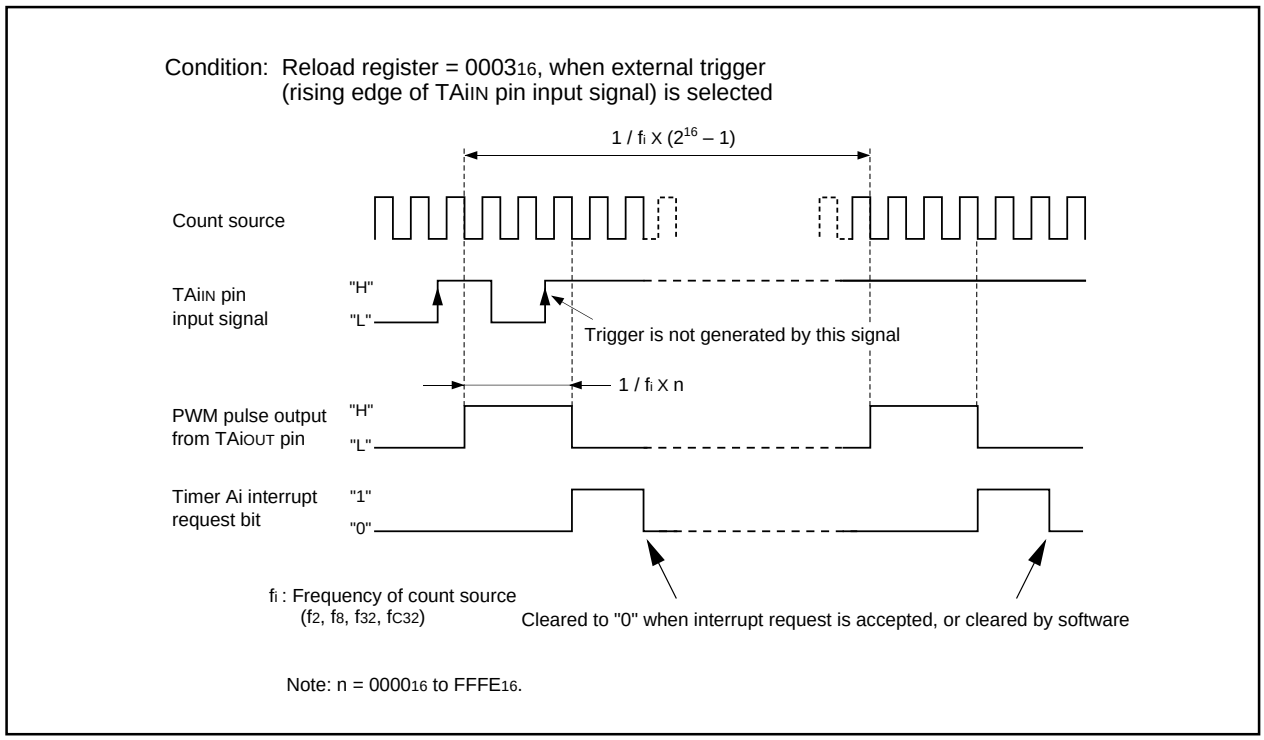


Figure 1.14.12. Example of how a 16-bit pulse width modulator operates

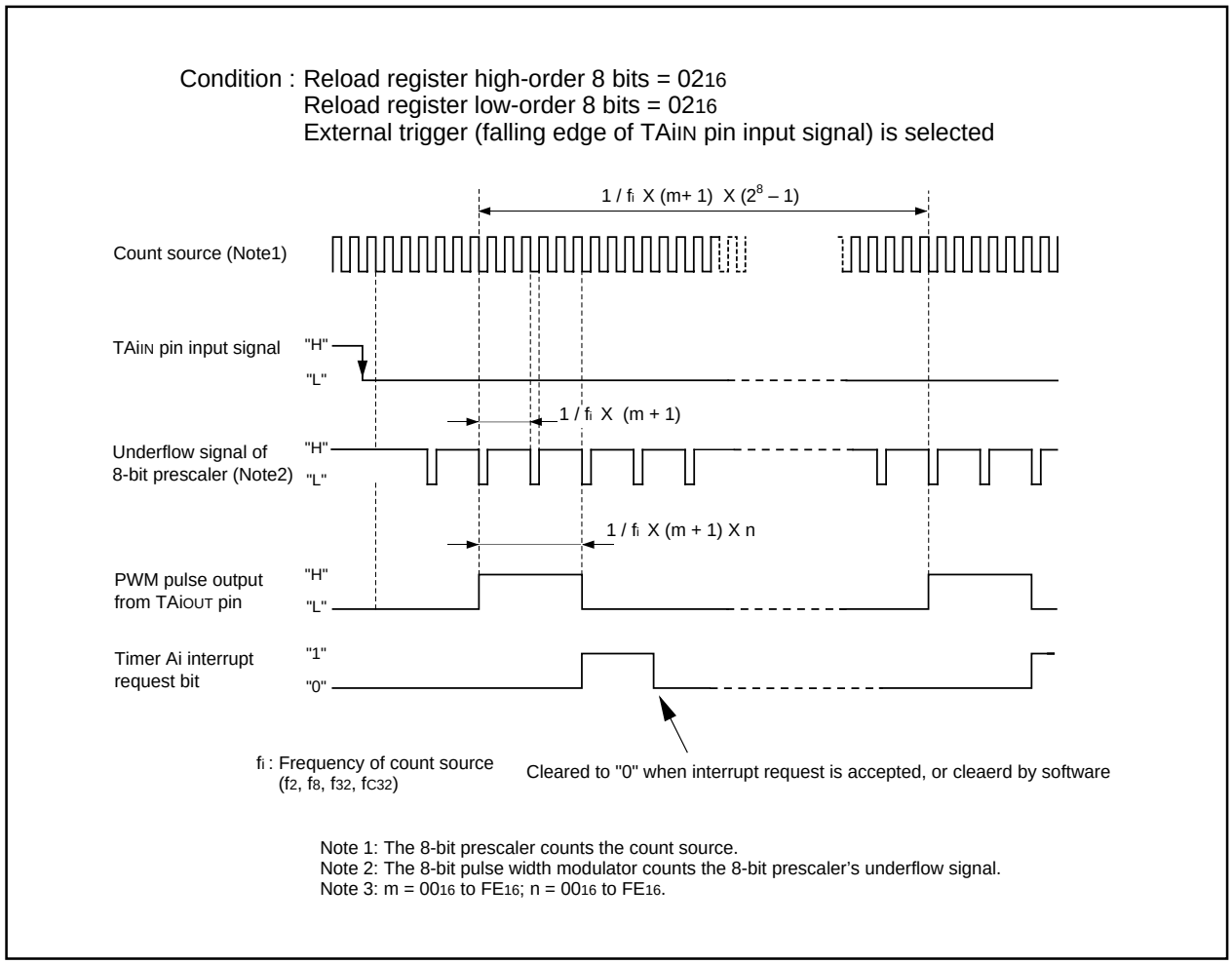


Figure 1.14.13. Example of how an 8-bit pulse width modulator operates

Timer B

Figure 1.14.14 shows the block diagram of timer B. Figures 1.14.15 and 1.14.16 show the timer B-related registers.

Use the timer Bi mode register (i = 0 to 5) 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.

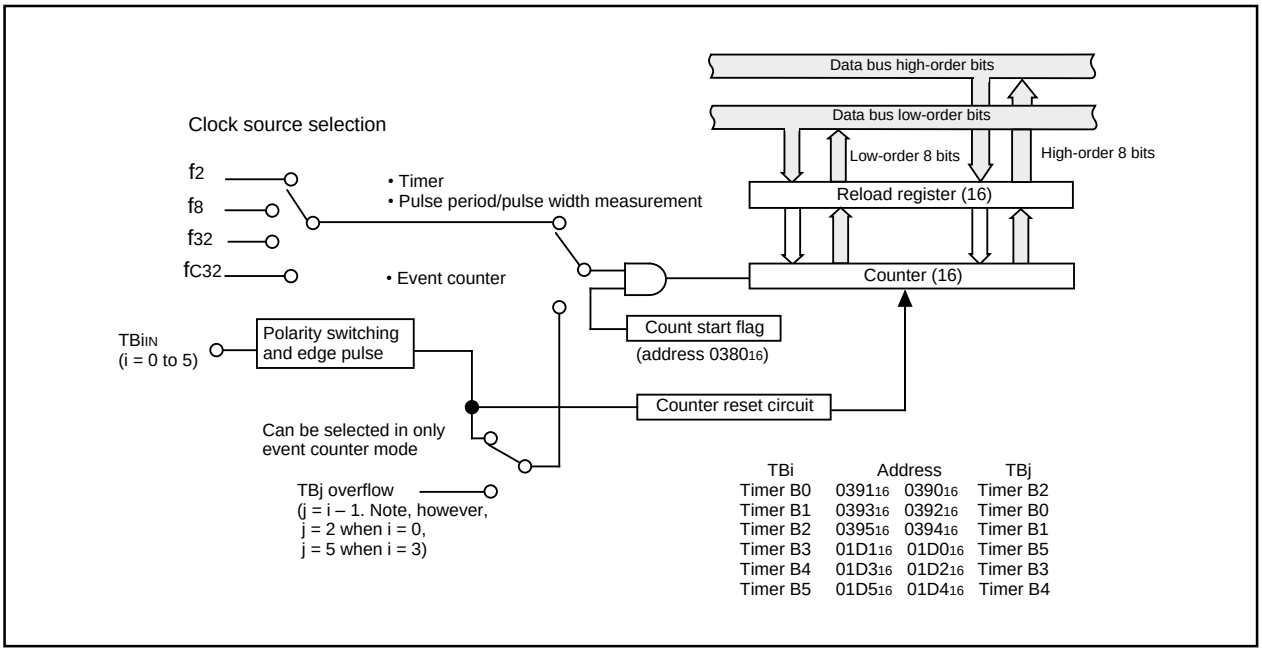


Figure 1.14.14. Block diagram of timer B

Timer Bi mode register

b7 b6 b5 b4 b3 b2 b1 b0

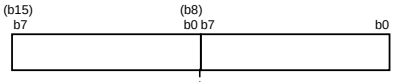
Symbol	Address	When reset
TBiMR(i = 0 to 5)	039B16 to 039D16	00XX00002
	01DB16 to 01DD16	00XX00002

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 : Must not be set	○	○
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, timer B3.
Note 2: Timer B1, timer B2, timer B4, timer B5.

Figure 1.14.15. Timer B-related registers (1)

Timer Bi register (Note)

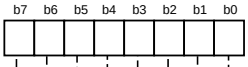


Symbol	Address	When reset
TB0	0391_{16}, 0390_{16}	Indeterminate
TB1	0393_{16}, 0392_{16}	Indeterminate
TB2	0395_{16}, 0394_{16}	Indeterminate
TB3	$01D1_{16}$, $01D0_{16}$	Indeterminate
TB4	$01D3_{16}$, $01D2_{16}$	Indeterminate
TB5	$01D5_{16}$, $01D4_{16}$	Indeterminate

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.

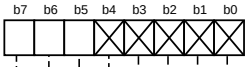
Count start flag



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		○	○

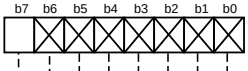
Timer B3, 4, 5 count start flag



Symbol	Address	When reset
TBSR	$01C0_{16}$	$000XXXXX_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. These bits can neither be set nor reset. When read, the value of these bits is "0".			—	—
TB3S	Timer B3 count start flag	0 : Stops counting 1 : Starts counting	○	○
TB4S	Timer B4 count start flag		○	○
TB5S	Timer B5 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. 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")	○	○

Figure 1.14.16. Timer B-related registers (2)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 1.14.6.) Figure 1.14.17 shows the timer Bi mode register in timer mode.

Table 1.14.6. Timer specifications in timer mode

Item	Specification
Count source	f_2, f_8, f_{32}, f_{C32}
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
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	Address	When reset
TBiMR(i = 0 to 5)	$039B_{16}$ to $039D_{16}$	$00XX0000_2$
	$01DB_{16}$ to $01DD_{16}$	$00XX0000_2$

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, 3)		○ (Note 1)	○
	Nothing is assiigned (i = 1, 2, 4, 5). This bit can neither be set nor reset. When read, its content is indeterminate.		× (Note 2)	×
MR3	Invalid in timer mode. This bit can neither be set nor reset. When read in timer mode, its content is indeterminate.		○	×
TCK0	Count source select bit	b7 b6 0 0 : f_2 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: Timer B0, timer B3.
Note 2: Timer B1, timer B2, timer B4, timer B5.

Figure 1.14.17. 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 1.14.7.)

Figure 1.14.18 shows the timer Bi mode register in event counter mode.

Table 1.14.7. 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	Address	When reset
TBiMR(i = 0 to 5)	039B16 to 039D16	00XX00002
	01DB16 to 01DD16	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 : Must not be set	○	○
MR1			○	○
MR2	0 (Fixed to "0" in event counter mode; i = 0, 3)		○ (Note 2)	○
	Nothing is assigned (i = 1, 2, 4, 5). This bit can neither be set nor reset. When read, its content is indeterminate.		× (Note 3)	
MR3	Invalid in event counter mode. This bit can neither be set nor reset. When read in event counter mode, its content is 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, j = 5 when i = 3)	○	○

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, timer B3.
Note 3: Timer B1, timer B2, timer B4, timer B5.
Note 4: Set the corresponding port direction register to "0".

Figure 1.14.18. 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 1.14.8.) Figure 1.14.19 shows the timer Bi mode register in pulse period/pulse width measurement mode. Figure 1.14.20 shows the operation timing when measuring a pulse period. Figure 1.14.21 shows the operation timing when measuring a pulse width.

Table 1.14.8. Timer specifications in pulse period/pulse width measurement mode

Item	Specification
Count source	f_2, 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	Address	When reset
TBiMR(i = 0 to 5)	$039B_{16}$ to $039D_{16}$ $01DB_{16}$ to $01DD_{16}$	$00XX0000_2$ $00XX0000_2$

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	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)	○	○
MR1		1 0 : Pulse width measurement (Interval between measurement pulse's falling edge to rising edge, and between rising edge to falling edge) 1 1 : Must not be set	○	○
MR2	0 (Fixed to "0" in pulse period/pulse width measurement mode; i = 0, 3)		○ (Note 2)	○
	Nothing is assigned (i = 1, 2, 4, 5). This bit can neither be set nor reset. When read, its content is 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_2 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: It is indeterminate when reset.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, timer B3.
Note 3: Timer B1, timer B2, timer B4, timer B5.

Figure 1.14.19. Timer Bi mode register in pulse period/pulse width measurement mode

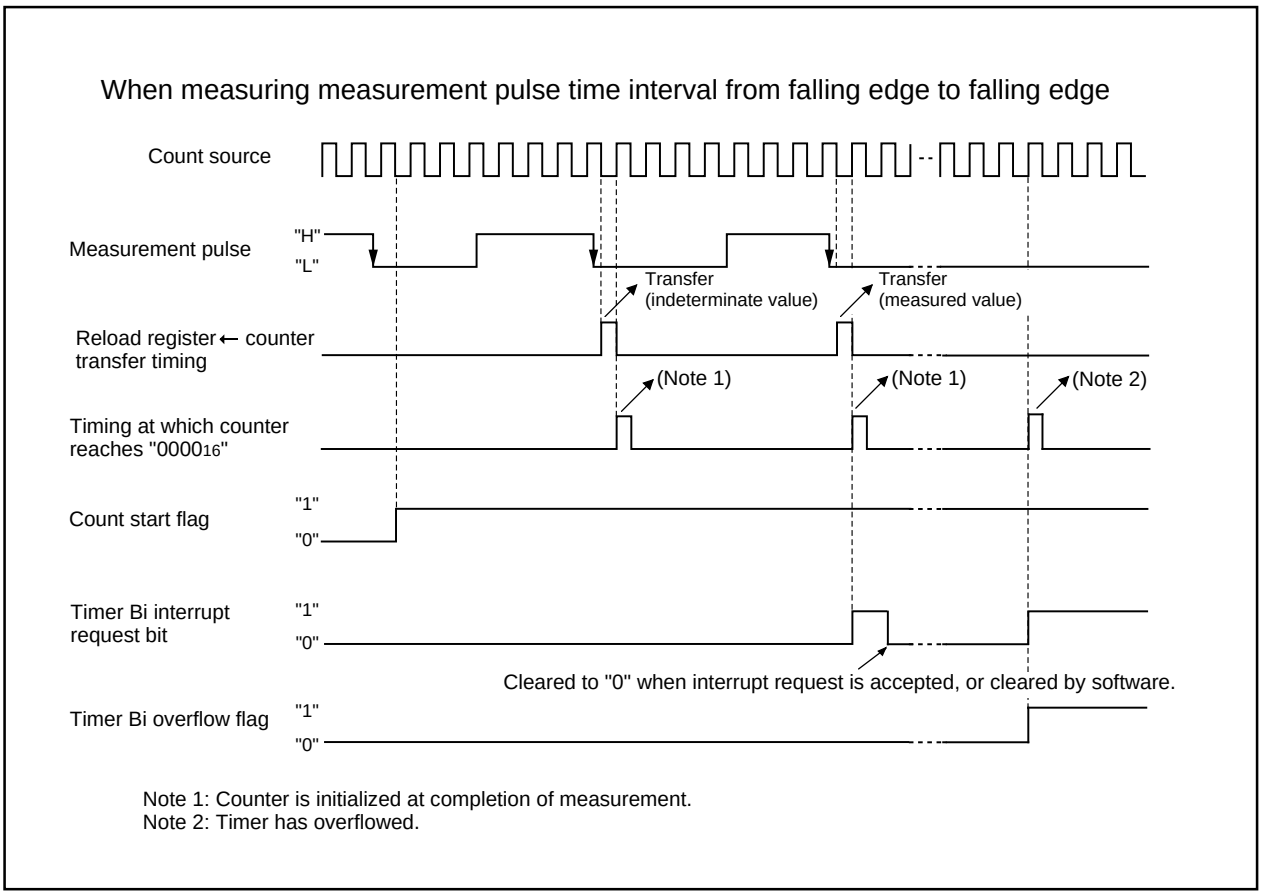


Figure 1.14.20. Operation timing when measuring a pulse period

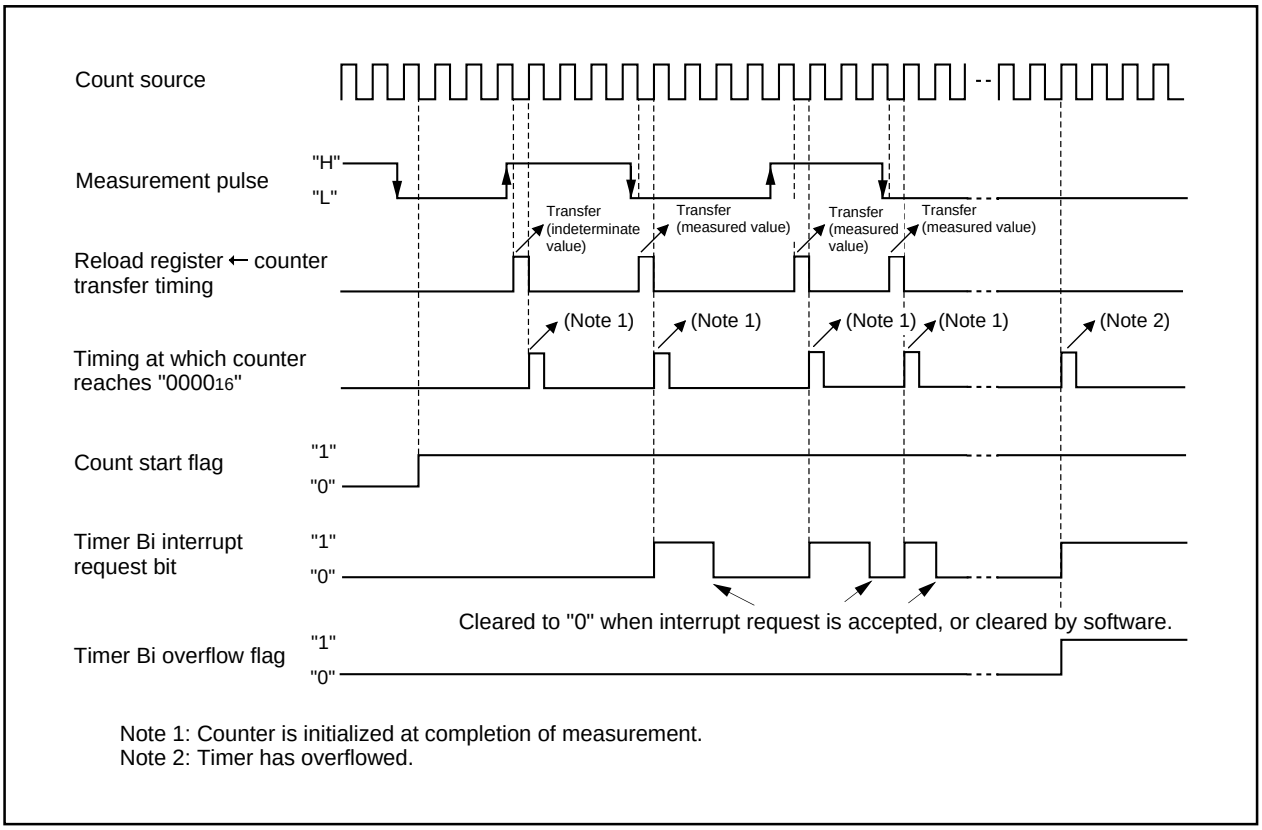


Figure 1.14.21. Operation timing when measuring a pulse width

Timer's Functions for Three-Phase Motor Control

Use of more than one built-in timer A and timer B provides the means of outputting three-phase motor driving waveforms.

Figures 1.15.1 through 1.15.3 show registers related to timers for three-phase motor control.

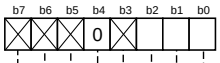
Three-phase PWM control register 0

Symbol: INVC0 | Address: 01C8₁₆ | When reset: 00₁₆

Bit symbol	Bit name	Description	R	W
INV00	Effective interrupt output polarity select bit	0: A timer B2 interrupt occurs when the timer A1 reload control signal is "1". 1: A timer B2 interrupt occurs when the timer A1 reload control signal is "0". **Effective only in three-phase mode 1**	○	○
INV01	Effective interrupt output specification bit (Note 4)	0: Not specified. 1: Selected by the effective interrupt output polarity selection bit. **Effective only in three-phase mode 1**	○	○
INV02	Mode select bit (Note 2)	0: Normal mode 1: Three-phase PWM output mode	○	○
INV03	Output control bit	0: Output disabled 1: Output enabled	○	○
INV04	Positive and negative phases concurrent L output disable function enable bit	0: Feature disabled 1: Feature enabled	○	○
INV05	Positive and negative phases concurrent L output detect flag	0: Not detected yet 1: Already detected	○	○ (Note 1)
INV06	Modulation mode select bit (Note 3)	0: Triangular wave modulation mode 1: Sawtooth wave modulation mode	○	○
INV07	Software trigger bit	1: Trigger generated The value, when read, is "0".	○	○

Note 1: No value other than "0" can be written.
Note 2: Selecting three-phase PWM output mode causes P8₀, P8₁, and P7₂ through P7₅ to output U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$, and works the timer for setting short circuit prevention time, the U, V, W phase output control circuits, and the circuit for setting timer B2 interrupt frequency.
Note 3: **In triangular wave modulation mode:**
The short circuit prevention timer starts in synchronization with the falling edge of timer Ai output.
The data transfer from the three-phase buffer register to the three-phase output shift register is made only once in synchronization with the transfer trigger signal after writing to the three-phase output buffer register.
In sawtooth wave modulation mode:
The short circuit prevention timer starts in synchronization with the falling edge of timer A output and with the transfer trigger signal. The data transfer from the three-phase output buffer register to the three-phase output shift register is made with respect to every transfer trigger.
Note 4: To write "1" to bit 1 (INV01) of the three-phase PWM control register 0, set in advance the contentof the timer B2 interrupt occurrences frequency set counter.

Three-phase PWM control register 1



Symbol: INVC1 | Address: 01C9₁₆ | When reset: 00₁₆

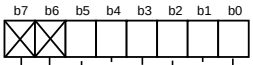
Bit symbol	Bit name	Description	R	W
INV10	Timer Ai start trigger signal select bit	0: Timer B2 overflow signal 1: Timer B2 overflow signal, signal for writing to timer B2	○	○
INV11	Timer A1-1, A2-1, A4-1 control bit	0: Three-phase mode 0 1: Three-phase mode 1	○	○
INV12	Short circuit timer count source select bit	0 : inhibited 1 : f2/2 (Note)	○	○
Noting is assigned. These bits can be set nor reset. When read, their contents are indeterminate.			—	—
Reserved bit		Always set to "0"	○	○
Noting is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".			—	—

Note: To use three-phase PWM output mode, write "1" to INV12.

Figure 1.15.1. Registers related to timers for three-phase motor control

Timer's Functions for Three-Phase Motor Control

Three-phase output buffer register 0

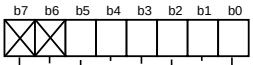


Symbol: IDB0 | Address: $01CA_{16}$ | When reset: 00_{16}

Bit Symbol	Bit name	Function	R	W
DU0	U phase output buffer 0	Setting in U phase output buffer 0	(Note)	○
DUB0	$\overline{U}$ phase output buffer 0	Setting in $\overline{U}$ phase output buffer 0	(Note)	○
DV0	V phase output buffer 0	Setting in V phase output buffer 0	(Note)	○
DVB0	$\overline{V}$ phase output buffer 0	Setting in $\overline{V}$ phase output buffer 0	(Note)	○
DW0	W phase output buffer 0	Setting in W phase output buffer 0	(Note)	○
DWB0	$\overline{W}$ phase output buffer 0	Setting in $\overline{W}$ phase output buffer 0	(Note)	○
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".			—	—

Note: When executing read instruction of this register, the contents of three-phase shift register is read out.

Three-phase output buffer register 1

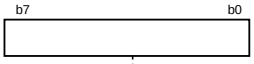


Symbol: IDB1 | Address: $01CB_{16}$ | When reset: 00_{16}

Bit Symbol	Bit name	Function	R	W
DU1	U phase output buffer 1	Setting in U phase output buffer 1	(Note)	○
DUB1	$\overline{U}$ phase output buffer 1	Setting in $\overline{U}$ phase output buffer 1	(Note)	○
DV1	V phase output buffer 1	Setting in V phase output buffer 1	(Note)	○
DVB1	$\overline{V}$ phase output buffer 1	Setting in $\overline{V}$ phase output buffer 1	(Note)	○
DW1	W phase output buffer 1	Setting in W phase output buffer 1	(Note)	○
DWB1	$\overline{W}$ phase output buffer 1	Setting in $\overline{W}$ phase output buffer 1	(Note)	○
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".			—	—

Note: When executing read instruction of this register, the contents of three-phase shift register is read out.

Dead time timer (Note)

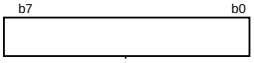


Symbol: DTT | Address: $01CC_{16}$ | When reset: Indeterminate

Function	Values that can be set	R	W
Set dead time timer	1 to 255	—	○

Note: Use MOV instruction to write to this register.

Timer B2 interrupt occurrences frequency set counter (Note 1, 2, 3)



Symbol: ICTB2 | Address: $01CD_{16}$ | When reset: Indeterminate

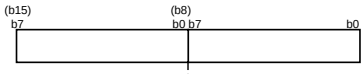
Function	Values that can be set	R	W
Set occurrence frequency of timer B2 interrupt request	1 to 15	—	○

Note 1: In setting 1 to bit 1 (INV01) - the effective interrupt output specification bit - of three-phase PWM control register 0, do not change the B2 interrupt occurrences frequency set counter to deal with the timer function for three-phase motor control.
Note 2: Do not write at the timing of an overflow occurrence in timer B2.
Note 3: Use MOV instruction to write to this register.

Figure 1.15.2. Registers related to timers for three-phase motor control

Timer's Functions for Three-Phase Motor Control

Timer Ai register (Note 1)

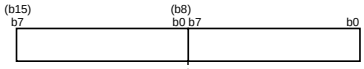


Symbol	Address	When reset
TA1	0389_{16},$038B_{16}$	Indeterminate
TA2	$038B_{16}$,$038A_{16}$	Indeterminate
TA4	$038F_{16}$,$038E_{16}$	Indeterminate
TB2	0395_{16},0394_{16}	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts an internal count source	0000_{16} to $FFFF_{16}$	○	○
• One-shot timer mode Counts a one shot width	0000_{16} to $FFFF_{16}$ (Note 2, 3)	×	○

Note 1: Read and write data in 16-bit units.
Note 2: When the timer Ai register is set to "0000_{16}", the counter does not operate and a timer Ai register interrupt does not occur.
Note 3: Use MOV instruction to write to this segister.

Timer Ai-1 register (Note)

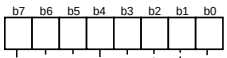


Symbol	Address	When reset
TA11	$01C3_{16}$,$01C2_{16}$	Indeterminate
TA21	$01C5_{16}$,$01C4_{16}$	Indeterminate
TA41	$01C7_{16}$,$01C6_{16}$	Indeterminate

Function	Values that can be set	R	W
Counts an internal count source	0000_{16} to $FFFF_{16}$	○	○

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

Trigger select register



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

Bit symbol	Bit name	Function	R	W
TA1TGL	Timer A1 event/trigger select bit	b1 b0 0 0 : Input on $TA1_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA0 overflow is selected 1 1 : TA2 overflow is selected	○	○
TA1TGH			○	○
TA2TGL	Timer A2 event/trigger select bit	b3 b2 0 0 : Input on $TA2_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA1 overflow is selected 1 1 : TA3 overflow is selected	○	○
TA2TGH			○	○
TA3TGL	Timer A3 event/trigger select bit	b5 b4 0 0 : Input on $TA3_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA2 overflow is selected 1 1 : TA4 overflow is selected	○	○
TA3TGH			○	○
TA4TGL	Timer A4 event/trigger select bit	b7 b6 0 0 : Input on $TA4_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA3 overflow is selected 1 1 : TA0 overflow is selected	○	○
TA4TGH			○	○

Note: Set the corresponding port direction register to "0".

Count start flag

Symbol: TABSR — Address: 0380_{16} — When reset: 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		○	○

Figure 1.15.3. Registers related to timers for three-phase motor control

Three-Phase Motor Driving Waveform Output Mode (three-phase PWM output mode)

Setting "1" in the mode select bit (bit 2 at address 01C8$_{16}$) shown in Figure 1.15.1 - causes three-phase PWM output mode that uses four timers A1, A2, A4, and B2 to be selected. As shown in Figure 1.15.4, set timers A1, A2, and A4 in one-shot timer mode, set the trigger in timer B2, and set timer B2 in timer mode using the respective timer mode registers.

Timer Ai mode register

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

Symbol	Address	When reset
TA1MR	0397$_{16}$	00$_{16}$
TA2MR	0398$_{16}$	00$_{16}$
TA4MR	039A$_{16}$	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 : (Must always be "0" in three-phase PWM output mode)	○	○
MR1	External trigger select bit	Invalid in three-phase PWM output mode	○	○
MR2	Trigger select bit	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 : f_2 0 1 : f_8 1 0 : f_{32} 1 1 : fc_{32}	○	○
TCK1			○	○

Timer B2 mode register

b7	b6	b5	b4	b3	b2	b1	b0
			0			0	0

Symbol	Address	When reset
TB2MR	039D$_{16}$	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)		○	○
MR3	Invalid in timer mode. This bit can neither be set nor reset. When read in timer mode, its content is indeterminate.		○	×
TCK0	Count source select bit	b7 b6 0 0 : f_2 0 1 : f_8 1 0 : f_{32} 1 1 : fc_{32}	○	○
TCK1			○	○

Figure 1.15.4. Timer mode registers in three-phase waveform mode

Figure 1.15.5 shows the block diagram for three-phase PWM output mode. In three-phase PWM output mode, the positive-phase PWM output (U phase, V phase, and W phase) and negative waveforms ($\overline{U}$ phase, $\overline{V}$ phase, and $\overline{W}$ phase), six waveforms in total, are output from $P8_0$, $P8_1$, $P7_2$, $P7_3$, $P7_4$, and $P7_5$ as active on the "L" level. Of the timers used in this mode, timer A4 controls the U phase and $\overline{U}$ phase, timer A1 controls the V phase and $\overline{V}$ phase, and timer A2 controls the W phase and $\overline{W}$ phase respectively; timer B2 controls the periods of one-shot pulse output from timers A4, A1, and A2.

In outputting a waveform, dead time can be set so as to cause the "L" level of the positive waveform output (U phase, V phase, and W phase) not to lap over the "L" level of the negative waveform output ($\overline{U}$ phase, $\overline{V}$ phase, and $\overline{W}$ phase).

To set short circuit time, use three 8-bit timers sharing the reload register for setting dead time. A value from 1 through 255 can be set as the count of the timer for setting dead time. The timer for setting dead time works as a one-shot timer. If a value is written to the dead timer (address $01CC_{16}$), the value is written to the reload register shared by the three timers for setting dead time.

Any of the timers for setting dead time takes the value of the reload register into its counter, if a start trigger comes from its corresponding timer, and performs a down count in line with the clock source selected by the dead time timer count source select bit (bit 2 at address $01C9_{16}$). The timer can receive another trigger again before the workings due to the previous trigger are completed. In this instance, the timer performs a down count from the reload register's content after its transfer, provoked by the trigger, to the timer for setting dead time.

Since the timer for setting dead time works as a one-shot timer, it starts outputting pulses if a trigger comes; it stops outputting pulses as soon as its content becomes 00_{16}, and waits for the next trigger to come.

The positive waveforms (U phase, V phase, and W phase) and the negative waveforms ($\overline{U}$ phase, $\overline{V}$ phase, and $\overline{W}$ phase) in three-phase PWM output mode are output from respective ports by means of setting "1" in the output control bit (bit 3 at address $01C8_{16}$). Setting "0" in this bit causes the ports to be the state of set by port direction register. This bit can be set to "0" not only by use of the applicable instruction, but by entering a falling edge in the $\overline{NMI}$ terminal or by resetting. Also, if "1" is set in the positive and negative phases concurrent "L" output disable function enable bit (bit4 at address $01C8_{16}$) causes one of the pairs of U phase and $\overline{U}$ phase, V phase and $\overline{V}$ phase, and W phase and $\overline{W}$ phase concurrently go to "L", as a result, the output control bit become the state of set by port direction register.

Timer's Functions for Three-Phase Motor Control

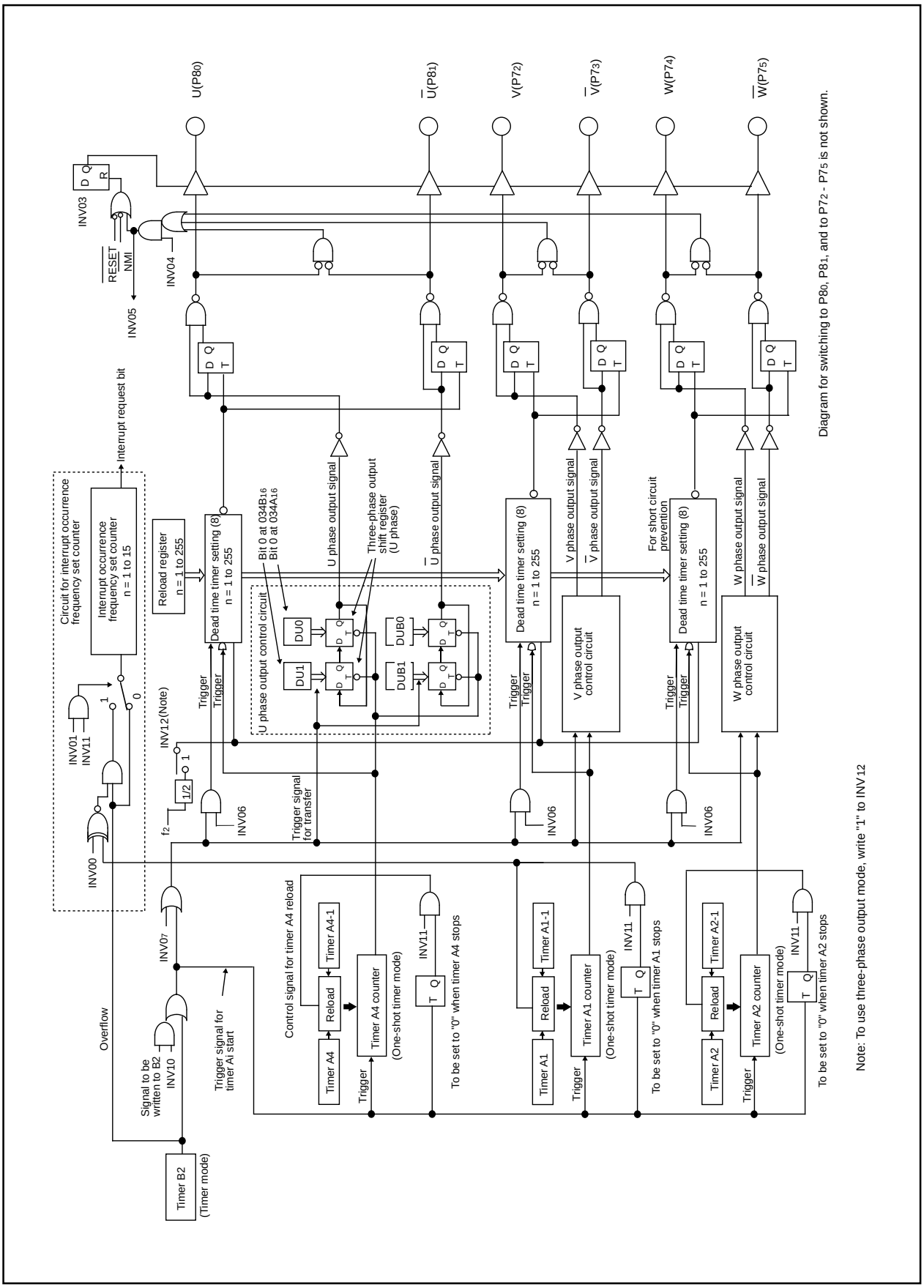


Figure 1.15.5. Block diagram for three-phase waveform mode

Triangular Wave Modulation

To generate a PWM waveform of triangular wave modulation, set "0" in the modulation mode select bit (bit 6 at address $01C8_{16}$). Also, set "1" in the timers A4-1, A1-1, A2-1 control bit (bit 1 at address $01C9_{16}$). In this mode, each of timers A4, A1, and A2 has two timer registers, and alternately reloads the timer register's content to the counter every time timer B2 counter's content becomes 0000_{16}. If "0" is set to the effective interrupt output specification bit (bit 1 at address $01C8_{16}$), the frequency of interrupt requests that occur every time the timer B2 counter's value becomes 0000_{16} can be set by use of the timer B2 counter (address $01CD_{16}$) for setting the frequency of interrupt occurrences. The frequency of occurrences is given by a set value ($\neq 0$).

Setting "1" in the effective interrupt output specification bit (bit 1 at address $01C8_{16}$) provides the means to choose which value of the timer A1 reload control signal to use, "0" or "1", to cause timer B2's interrupt request to occur. To make this selection, use the effective interrupt output polarity selection bit (bit 0 at address $01C8_{16}$).

An example of U phase waveform is shown in Figure 1.15.6, and the description of waveform output workings is given below. Set "1" in DU0 bit (bit 0 at address $01CA_{16}$). And set "0" in DUB0 bit (bit 1 at address $01CA_{16}$). In addition, set "0" in DU1 bit (bit 0 at address $01CB_{16}$) and set "1" in DUB1 bit (bit 1 at address $01CB_{16}$). Also, set "0" in the effective interrupt output specification bit (bit 1 at address $01C8_{16}$) to set a value in the timer B2 interrupt occurence frequency set counter. By this setting, a timer B2 interrupt occurs when the timer B2 counter's content becomes 0000_{16} as many as (setting) times. Furthermore, set "1" in the effective interrupt output specification bit (bit 1 at address $01C8_{16}$), set "0" in the effective interrupt output polarity select bit (bit 0 at address $01C8_{16}$) and set "1" in the interrupt occurence frequency set counter (address $01CD_{16}$). These settings cause a timer B2 interrupt to occur every other interval when the U phase output goes to "H".

When the timer B2 counter's content becomes 0000_{16}, timer A4 starts outputting one-shot pulses. In this instance, the content of DU1 bit (bit 0 at address $01CB_{16}$) and that of DU0 bit (bit 0 at address $01CA_{16}$) are set in the three-phase output shift register (U phase), the content of DUB1 bit (bit 1 at address $01CB_{16}$) and that of DUB0 bit (bit 1 at address $01CA_{16}$) are set in the three-phase output shift register ($\overline{U}$ phase). After triangular wave modulation mode is selected, however, no setting is made in the shift register even though the timer B2 counter's content becomes 0000_{16}.

The value of DU0 bit and that of DUB0 bit are output to the U terminal ($P8_0$) and to the $\overline{U}$ terminal ($P8_1$) respectively. When the timer A4 counter counts the value written to timer A4 (addresses $038F_{16}$ and $038E_{16}$) and when timer A4 finishes outputting one-shot pulses, the three-phase shift register's content is shifted one position, and the value of DU1 bit and that of DUB1 bit are output to the U phase output signal and to $\overline{U}$ phase output signal respectively. At the this time, one-shot pulses are output from the timer for setting dead time used for setting the time over which the "L" level of the U phase waveform doesn't lap over the "L" level of the $\overline{U}$ phase waveform, which has the opposite phase of the former. The U phase waveform output that started from the "H" level keeps its level until the timer for setting dead time finishes outputting one-shot pulses even though the three-phase output shift register's content changes from "1" to "0" by the effect of the one-shot pulses. When the timer for setting dead time finishes outputting one-shot pulses, "0" already shifted in the three-phase shift register goes effective, and the U phase waveform changes to the "L" level. When the timer B2 counter's content becomes 0000_{16}, the timer A4 counter starts counting the value written to timer A4-1 (addresses $01C1_{16}$ and $01C0_{16}$), and starts outputting one-shot pulses. When timer A4 finishes outputting one-shot pulses, the three-phase shift register's content is shifted one position, but if the three-phase output shift register's content changes from "0" to "1" as a result of the shift, the output level changes from "L" to "H" without waiting for the timer for setting dead time to finish outputting one-shot

pulses. A U phase waveform is generated by these workings repeatedly. With the exception that the three-phase output shift register on the $\overline{U}$ phase side is used, the workings in generating a $\overline{U}$ phase waveform, which has the opposite phase of the U phase waveform, are the same as in generating a U phase waveform. In this way, a waveform can be picked up from the applicable terminal in a manner in which the "L" level of the U phase waveform doesn't lap over that of the $\overline{U}$ phase waveform, which has the opposite phase of the U phase waveform. The width of the "L" level too can be adjusted by varying the values of timer B2, timer A4, and timer A4-1. In dealing with the V and W phases, and $\overline{V}$ and $\overline{W}$ phases, the latter are of opposite phase of the former, have the corresponding timers work similarly to dealing with the U and $\overline{U}$ phases to generate an intended waveform.

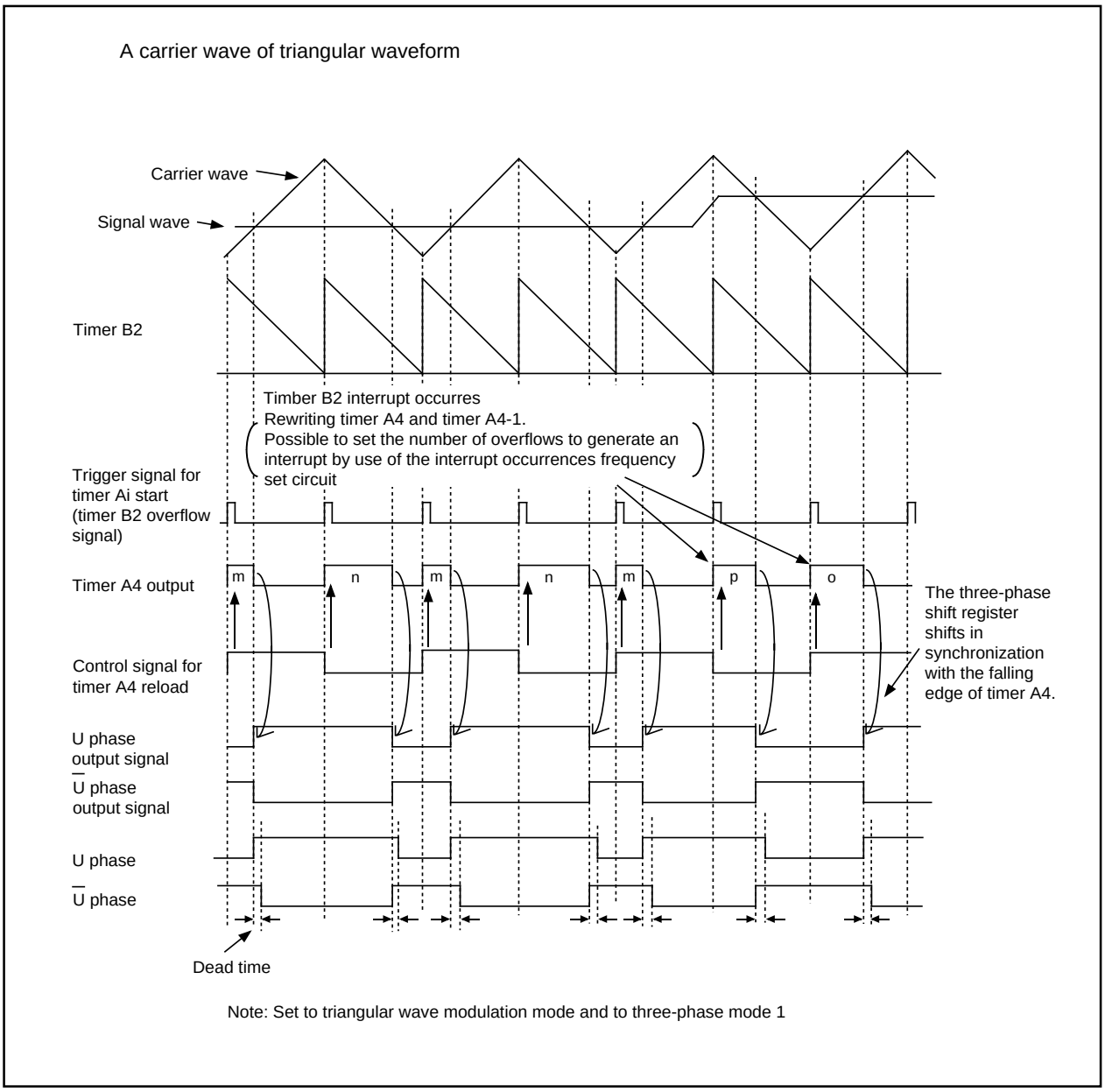
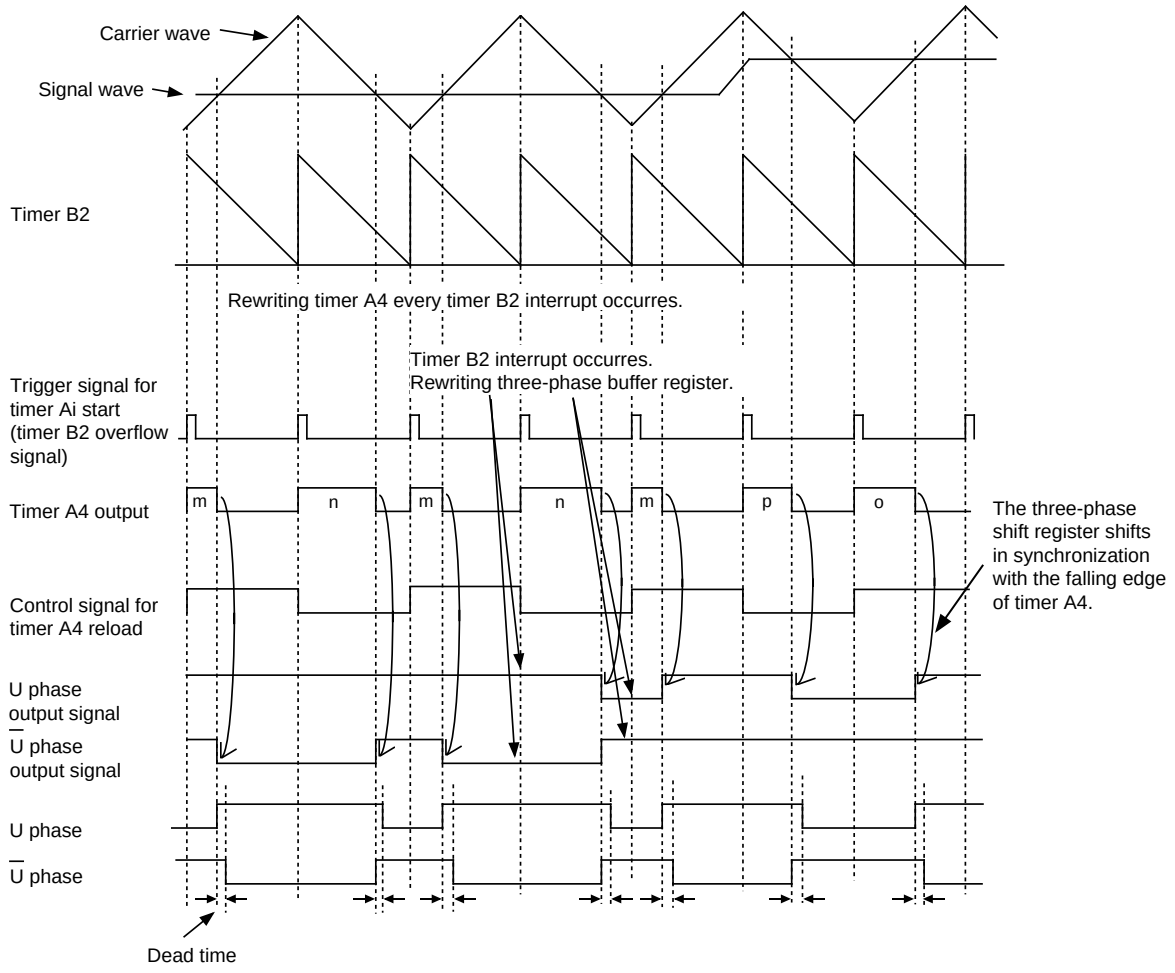


Figure 1.15.6. Timing chart of operation (1)

Assigning certain values to DU0 bit (bit 0 at address 01CA16) and DUB0 bit (bit 1 at address 01CA16), and to DU1 bit (bit 0 at address 01CB16) and DUB1 bit (bit 1 at address 01CB16) allows the user to output the waveforms as shown in the Figure 1.15.7, that is, to output the U phase alone, to fix $\overline{U}$ phase to "H", to fix the U phase to "H", or to output the $\overline{U}$ phase alone.

A carrier wave of triangular waveform



Note: Set to triangular wave modulation mode and to three-phase mode 0.

Figure 1.15.7. Timing chart of operation (2)

Sawtooth Modulation

To generate a PWM waveform of sawtooth wave modulation, set "1" in the modulation mode select bit (bit 6 at address $01C8_{16}$). Also, set "0" in the timers A1-1, A2-1, A4-1 control bit (bit 1 at address $01C9_{16}$). In this mode, the timer registers of timers A4, A1, and of A2 comprise conventional timers A4, A1, and A2 alone, and reload the corresponding timer register's content to the counter every time the timer B2 counter's content becomes 0000_{16}. The effective interrupt output specification bit (bit 1 at address $01C8_{16}$) and the effective interrupt output polarity selection bit (bit 0 at address $01C8_{16}$) turn nullified.

An example of U phase waveform is shown in Figure 1.15.8, and the description of waveform output workings is given below. Set "1" in DU0 bit (bit 0 at address $01CA_{16}$) and set "0" in DUB0 bit (bit 1 at address $01CA_{16}$). In addition, set "0" in DU1 bit (bit 0 at address $01CA_{16}$) and set "1" in DUB1 bit (bit 1 at address $01CA_{16}$).

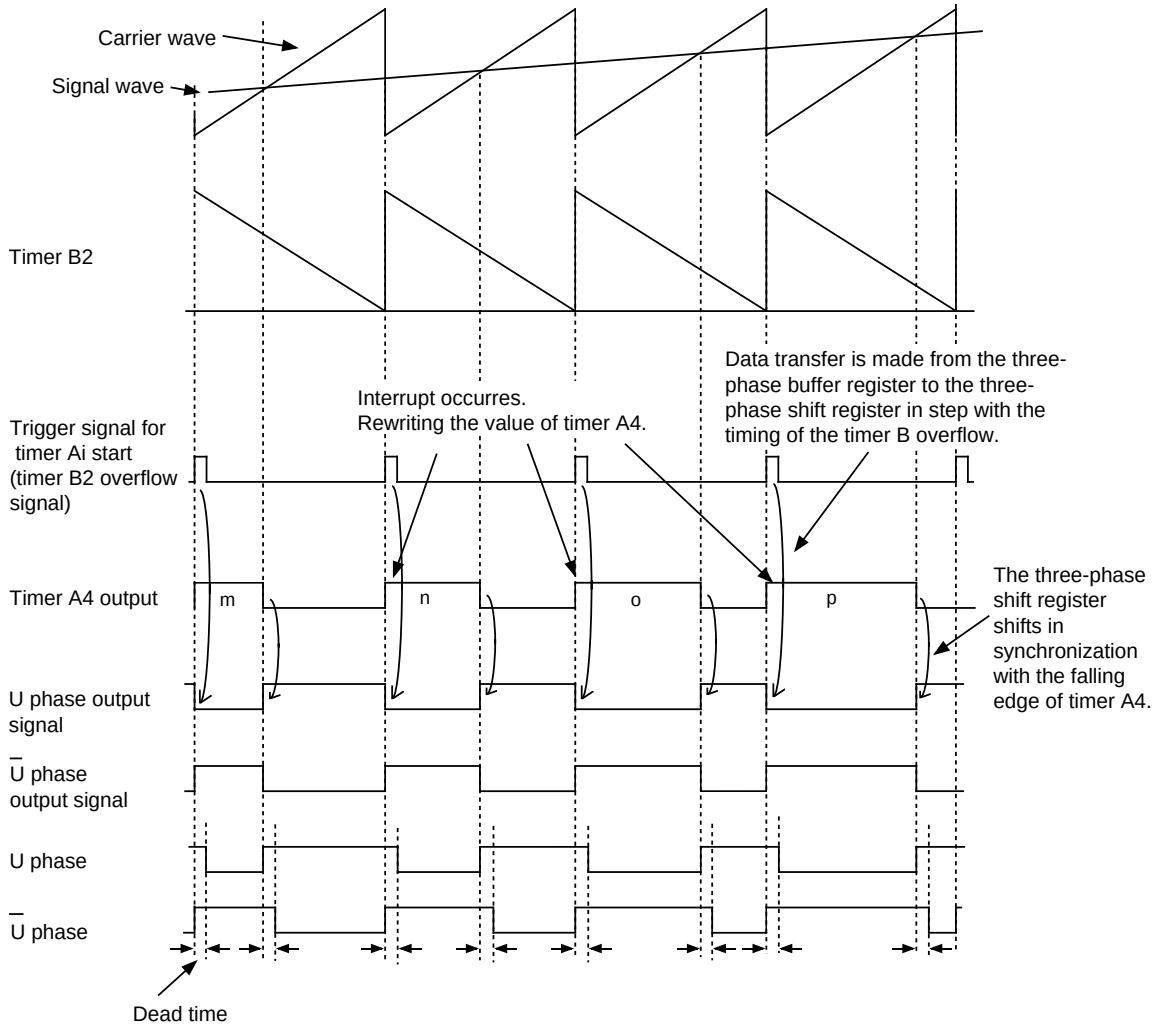
When the timer B2 counter's content becomes 0000_{16}, timer B2 generates an interrupt, and timer A4 starts outputting one-shot pulses at the same time. In this instance, the contents of the three-phase buffer registers DU1 bit and DU0 bit are set in the three-phase output shift register (U phase), and the contents of DUB1 bit and DUB0 bit are set in the three-phase output register (U phase). After this, the three-phase buffer register's content is set in the three-phase shift register every time the timer B2 counter's content becomes 0000_{16}.

The value of DU0 bit and that of DUB0 bit are output to the U terminal ($P8_0$) and to the $\overline{U}$ terminal ($P8_1$) respectively. When the timer A4 counter counts the value written to timer A4 (addresses $038F_{16}$ and $038E_{16}$) and when timer A4 finishes outputting one-shot pulses, the three-phase output shift register's content is shifted one position, and the value of DU1 bit and that of DUB1 bit are output to the U phase output signal and to the $\overline{U}$ output signal respectively. At the this time, one-shot pulses are output from the timer for setting dead time used for setting the time over which the "L" level of the U phase waveform doesn't lap over the "L" level of the $\overline{U}$ phase waveform, which has the opposite phase of the former. The U phase waveform output that started from the "H" level keeps its level until the timer for setting dead time finishes outputting one-shot pulses even though the three-phase output shift register's content changes from "1" to "0" by the effect of the one-shot pulses. When the timer for setting dead time finishes outputting one-shot pulses, 0 already shifted in the three-phase shift register goes effective, and the U phase waveform changes to the "L" level. When the timer B2 counter's content becomes 0000_{16}, the contents of the three-phase buffer registers DU1 bit and DU0 bit are set in the three-phase shift register (U phase), and the contents of DUB1 bit and DUB0 bit are set in the three-phase shift register ($\overline{U}$ phase) again.

A U phase waveform is generated by these workings repeatedly. With the exception that the three-phase output shift register on the $\overline{U}$ phase side is used, the workings in generating a $\overline{U}$ phase waveform, which has the opposite phase of the U phase waveform, are the same as in generating a U phase waveform. In this way, a waveform can be picked up from the applicable terminal in a manner in which the "L" level of the U phase waveform doesn't lap over that of the $\overline{U}$ phase waveform, which has the opposite phase of the U phase waveform. The width of the "L" level too can be adjusted by varying the values of timer B2 and timer A4. In dealing with the V and W phases, and $\overline{V}$ and $\overline{W}$ phases, the latter are of opposite phase of the former, have the corresponding timers work similarly to dealing with the U and $\overline{U}$ phases to generate an intended waveform.

Setting "1" both in DUB0 bit and DUB1 bit provides a means to output the U phase alone and to fix the $\overline{U}$ phase output to "H" as shown in Figure 1.15.9.

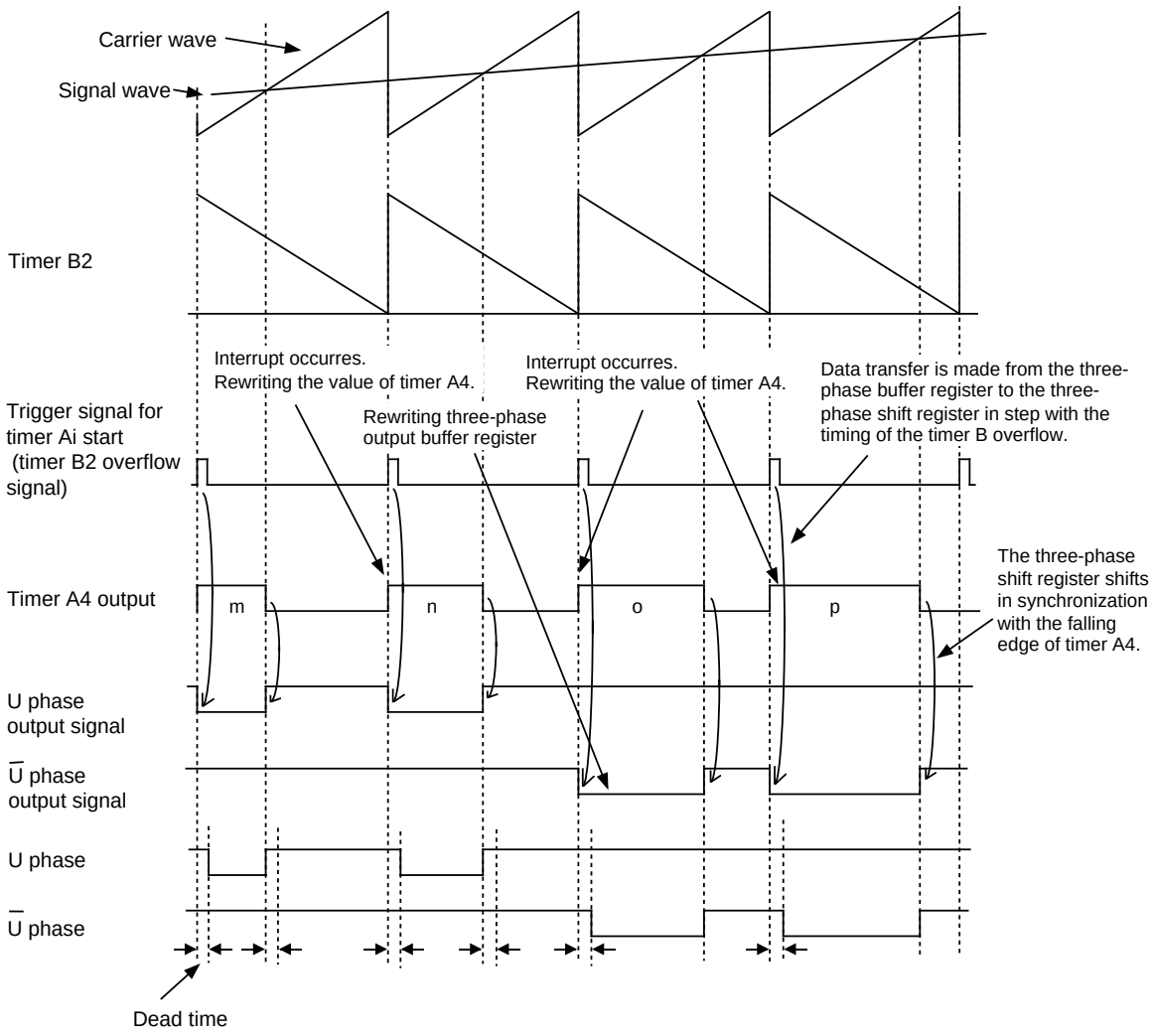
A carrier wave of sawtooth waveform



Note: Set to sawtooth modulation mode and to three-phase mode 0.

Figure 1.15.8. Timing chart of operation (3)

A carrier wave of sawtooth waveform



Note: Set to sawtooth modulation mode and to three-phase mode 0.

Figure 1.15.9. Timing chart of operation (4)

Serial I/O

Serial I/O is configured as four channels: UART0, UART1, UART2 and S I/O3.

UART0 to 2

UART0, UART1 and UART2 each have an exclusive timer to generate a transfer clock, so they operate independently of each other.

Figure 1.16.1 shows the block diagram of UART0, UART1 and UART2. Figures 1.16.2 and 1.16.3 show the block diagram of the transmit/receive unit.

UARTi (i = 0 to 2) 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 $03A0_{16}$, $03A8_{16}$ and $01F8_{16}$) determine whether UARTi is used as a clock synchronous serial I/O or as a UART. Although a few functions are different, UART0, UART1 and UART2 have almost the same functions.

UART2, in particular, is used for the SIM (Subscriber Identity Module) interface with some extra settings added in clock-asynchronous serial I/O mode (Note). It also has the bus collision detection function that generates an interrupt request if the TxD pin and the RxD pin are different in level.

Table 1.16.1 shows the comparison of functions of UART0 through UART2, and Figures 1.16.4 through 1.16.8 show the registers related to UARTi.

Table 1.16.1. Comparison of functions of UART0 through UART2

Function	UART0	UART1	UART2
CLK polarity selection	Possible (Note 1)	Possible (Note 1)	Possible (Note 1)
LSB first / MSB first selection	Possible (Note 1)	Possible (Note 1)	Possible (Note 2)
Continuous receive mode selection	Possible (Note 1)	Possible (Note 1)	Possible (Note 1)
Transfer clock output from multiple pins selection	Impossible	Possible (Note 1)	Impossible
Separate $\overline{CTS}/\overline{RTS}$ pins	Possible	Impossible	Impossible
Serial data logic switch	Impossible	Impossible	Possible (Note 4)
Sleep mode selection	Possible (Note 3)	Possible (Note 3)	Impossible
TxD, RxD I/O polarity switch	Impossible	Impossible	Possible
TxD port output format	N-channel open-drain /CMOS output (Note 5)	N-channel open-drain /CMOS output (Note 5)	N-channel open-drain /CMOS output (Note 5)
Parity error signal output	Impossible	Impossible	Possible (Note 4)
Bus collision detection	Impossible	Impossible	Possible (Note 6)

Note 1: Only when clock synchronous serial I/O mode.
Note 2: Only when clock synchronous serial I/O mode and 8-bit UART mode.
Note 3: Only when UART mode.
Note 4: Using for SIM interface.
Note 5: When selecting N-channel open-drain output, connect via pull-up resister to V_{CC} outside.
Note 6: Generally, it use in case of IE bus-emulation.

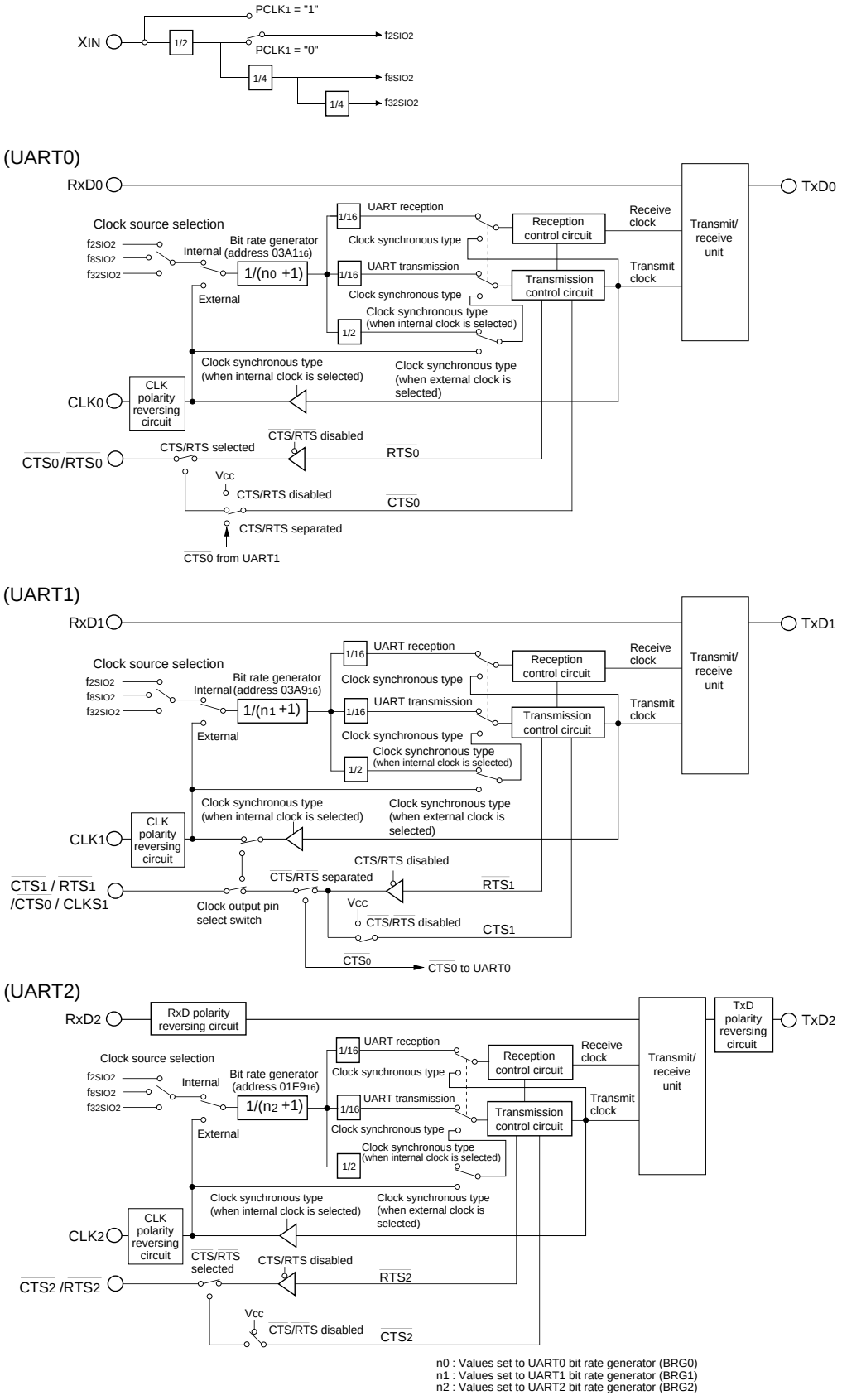


Figure 1.16.1. Block diagram of UARTi (i = 0 to 2)

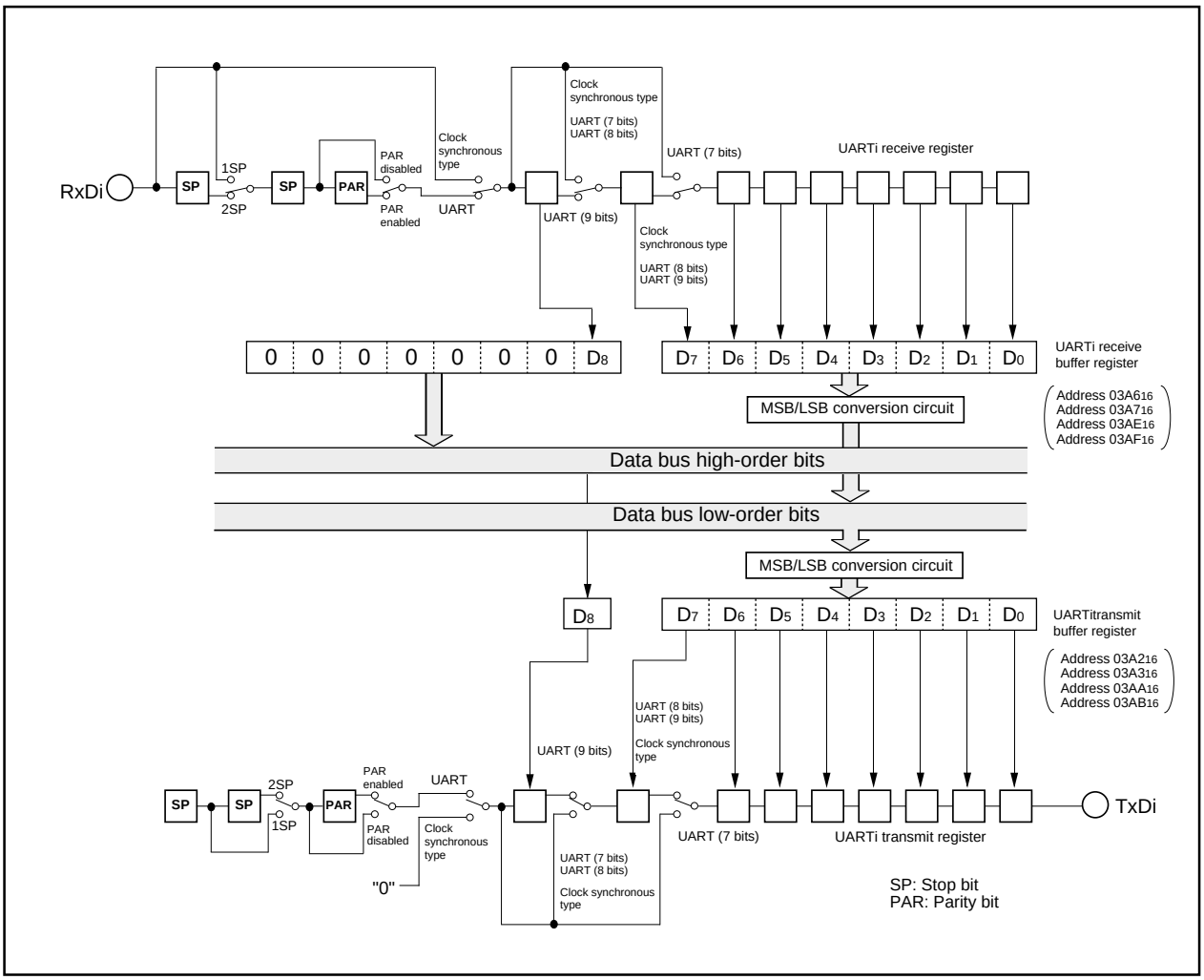


Figure 1.16.2. Block diagram of UARTi (i = 0, 1) transmit/receive unit

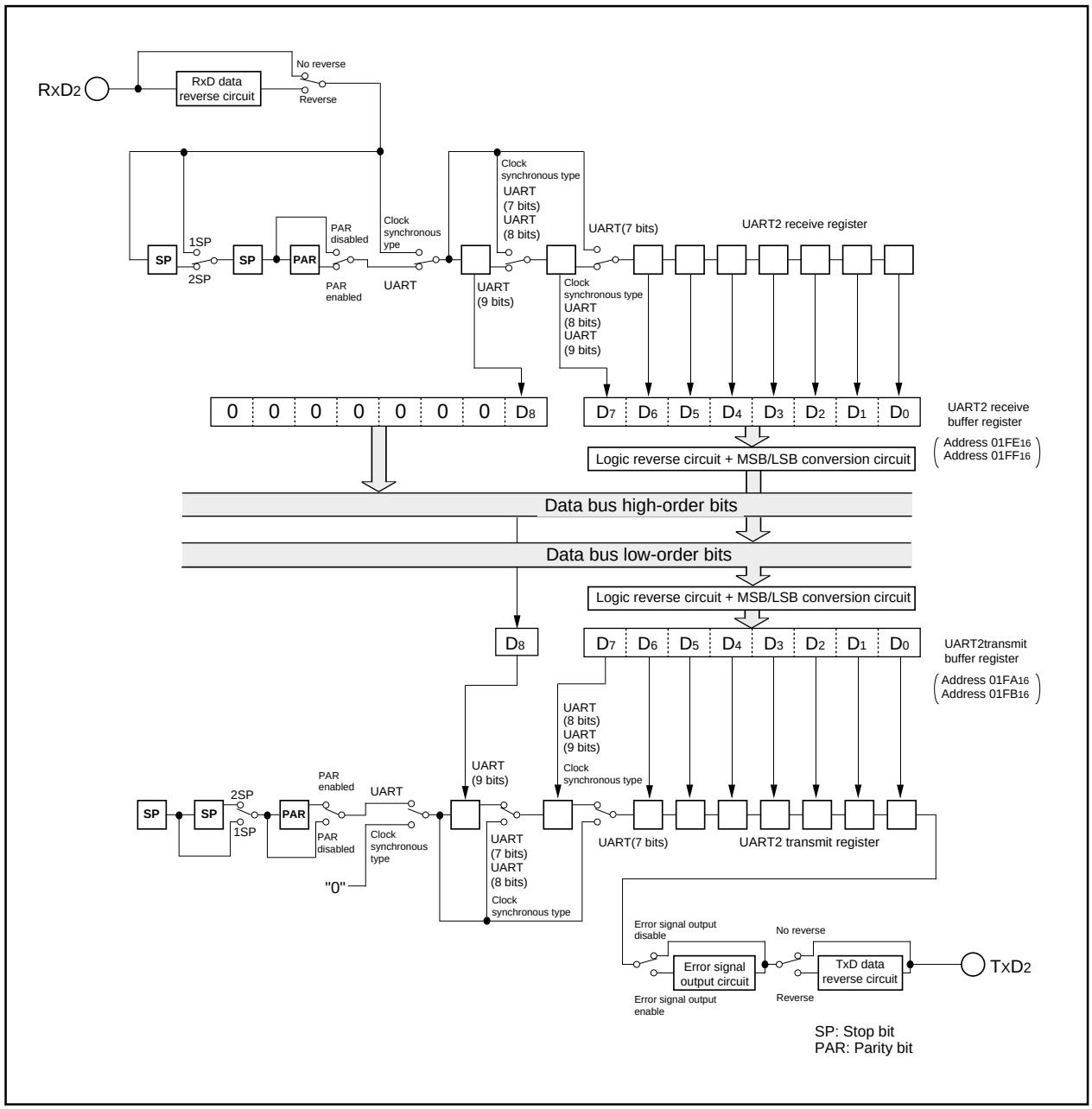
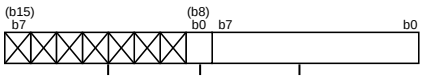


Figure 1.16.3. Block diagram of UART2 transmit/receive unit

UARTi transmit buffer register (Note)

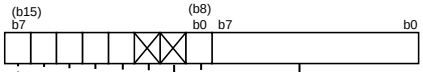


Symbol	Address	When reset
U0TB	$03A3_{16}$, $03A2_{16}$	Indeterminate
U1TB	$03AB_{16}$, $03AA_{16}$	Indeterminate
U2TB	$01FB_{16}$, $01FA_{16}$	Indeterminate

Function	R	W
Transmit data	×	○
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.	—	—

Note: Use MOV instruction to write to this register.

UARTi receive buffer register

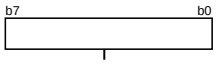


Symbol	Address	When reset
U0RB	$03A7_{16}$, $03A6_{16}$	Indeterminate
U1RB	$03AF_{16}$, $03AE_{16}$	Indeterminate
U2RB	$01FF_{16}$, $01FE_{16}$	Indeterminate

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
—	———	Receive data	Receive data	○	×
Nothing is assigned. These bits can neither be set nor reset. When read, the value of these bits is "0".				—	—
ABT	Arbitration lost detecting flag (Note 2)	0 : Not detected 1 : Detected	Invalid	○	○
OER	Overrun error flag (Note 1)	0 : No overrun error 1 : Overrun error found	0 : No overrun error 1 : Overrun error found	○	×
FER	Framing error flag (Note 1)	Invalid	0 : No framing error 1 : Framing error found	○	×
PER	Parity error flag (Note 1)	Invalid	0 : No parity error 1 : Parity error found	○	×
SUM	Error sum flag (Note 1)	Invalid	0 : No error 1 : Error found	○	×

Note 1: Bits 15 through 12 are set to "0" when the serial I/O mode select bit (bits 2 to 0 at addresses $03A0_{16}$, $03A8_{16}$ and $01F8_{16}$) 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 $03A6_{16}$, $03AE_{16}$ and $01FE_{16}$) is read out.
Note 2: Arbitration lost detecting flag is allocated to U2RB and noting but "0" may be written. Nothing is assigned in bit 11 of U0RB and U1RB. These bits can neither be set or reset. When read, the value of this bit is "0".

UARTi bit rate generator (Note 1, 2)



Symbol	Address	When reset
U0BRG	$03A1_{16}$	Indeterminate
U1BRG	$03A9_{16}$	Indeterminate
U2BRG	$01F9_{16}$	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}	×	○

Note 1: Write a value to this register while transmit/receive halts.
Note 2: Use MOV instruction to write to this register.

Figure 1.16.4. 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 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	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				○	○
SMD2				○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (Note)	0 : Internal clock 1 : External clock (Note)	○	○
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	○	○

Note: Set the corresponding port direction register to "0".

UART2 transmit/receive mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: U2MR | Address: 01F8$_{16}$ | 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 0 0 0 : Serial I/O invalid 0 1 0 : (Note 1) 0 1 1 : Inhibited 1 1 1 : Inhibited	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				○	○
SMD2				○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (Note 2)	Must always be fixed to "0"	○	○
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	○	○
IOPOL	TxD, RxD I/O polarity reverse bit	0 : No reverse 1 : Reverse Usually set to "0"	0 : No reverse 1 : Reverse Usually set to "0"	○	○

Note 1: Bit 2 to bit 0 are set to "010$_2$" when I^2C mode is used.
Note 2: Set the corresponding port direction register to "0".

Figure 1.16.5. Serial I/O-related registers (2)

UARTi transmit/receive control register 0

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_{2SIO2} is selected 0 1 : f_{8SIO2} is selected 1 0 : f_{32SIO2} is selected 1 1 : Inhibited	b1 b0 0 0 : f_{2SIO2} is selected 0 1 : f_{8SIO2} is selected 1 0 : f_{32SIO2} 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 ($P6_0$ and $P6_4$ function as programmable I/O port)	0 : $\overline{CTS}$/$\overline{RTS}$ function enabled 1 : $\overline{CTS}$/$\overline{RTS}$ function disabled ($P6_0$ and $P6_4$ 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.

UART2 transmit/receive control register 0

Symbol: U2C0 — Address: 01FC$_{16}$ — 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_{2SIO2} is selected 0 1 : f_{8SIO2} is selected 1 0 : f_{32SIO2} is selected 1 1 : Inhibited	b1 b0 0 0 : f_{2SIO2} is selected 0 1 : f_{8SIO2} is selected 1 0 : f_{32SIO2} 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 : CTS/RTS function disabled ($P7_3$ functions programmable I/O port)	0 : $\overline{CTS}$/$\overline{RTS}$ function enabled 1 : CTS/RTS function disabled ($P7_3$ functions programmable I/O port)	○	○
NCH	Data output select bit	0 : TXD_2 pin is CMOS output 1 : TXD_2 pin is N-channel open-drain output (Note 4)	0 : TXD_2 pin is CMOS output 1 : TXD_2 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 (Note 3)	0 : LSB first 1 : MSB first	0 : LSB first 1 : MSB first	○	○

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.
Note 3: Only clock synchronous serial I/O mode and 8-bit UART mode are valid.

Figure 1.16.6. Serial I/O-related registers (3)

UARTi transmit/receive control register 1

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

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. These bits can neither be set nor reset. When read, the value of these bits is "0".				–	–

UART2 transmit/receive control register 1

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: U2C1 — Address: 01FD$_{16}$ — 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	○	×
U2IRS	UART2 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmit is completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmit is completed (TXEPT = 1)	○	○
U2RRM	UART2 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	Must always be "0"	○	○
U2LCH	Data logic select bit	0 : No reverse 1 : Reverse	0 : No reverse 1 : Reverse	○	○
U2ERE	Error signal output enable bit	Must be fixed to "0"	0 : Output disabled 1 : Output enabled	○	○

Figure 1.16.7. Serial I/O-related registers (4)

UART transmit/receive control register 2

b7	b6	b5	b4	b3	b2	b1	b0
⊠							

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	Must always be "0"	○	○
U1RRM	UART1 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	Must always be "0"	○	○
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"	○	○
RCSP	Separate $\overline{CTS}/\overline{RTS}$ bit	0 : $\overline{CTS}/\overline{RTS}$ shared pin 1 : $\overline{CTS}/\overline{RTS}$ separated	0 : $\overline{CTS}/\overline{RTS}$ shared pin 1 : $\overline{CTS}/\overline{RTS}$ separated	○	○
Nothing is assigned. This bit can neither be set nor reset. When read, its content is indeterminate.				–	–

Note: When using multiple pins to output the transfer clock, the following requirements must be met:
• UART1 internal/external clock select bit (bit 3 at address $03A8_{16}$) = "0".

UART2 special mode register

b7	b6	b5	b4	b3	b2	b1	b0
⊠							

Symbol: U2SMR — Address: $01F7_{16}$ — When reset: 00_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
IICM	IIC mode selection bit	0 : Normal mode 1 : IIC mode	Must always be "0"	○	○
ABC	Arbitration lost detecting flag control bit	0 : Update per bit 1 : Update per byte	Must always be "0"	○	○
BBS	Bus busy flag	0 : STOP condition detected 1 : START condition detected	Must always be "0"	○	○ (Note)
LSYN	SCLL sync output enable bit	0 : Disabled 1 : Enabled	Must always be "0"	○	○
ABSCS	Bus collision detect sampling clock select bit	Must always be "0"	0 : Rising edge of transfer clock 1 : Underflow signal of timer A0	○	○
ACSE	Auto clear function select bit of transmit enable bit	Must always be "0"	0 : No auto clear function 1 : Auto clear at occurrence of bus collision	○	○
SSS	Transmit start condition select bit	Must always be "0"	0 : Ordinary 1 : Falling edge of RxD_2	○	○
Nothing is assigned. This bit can neither be set nor reset. When read, its content is indeterminate.				–	–

Note: Nothing but "0" may be written.

Figure 1.16.8. Serial I/O-related registers (5)

(1) Clock synchronous serial I/O mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Tables 1.16.2 and 1.16.3 list the specifications of the clock synchronous serial I/O mode. Figure 1.16.9 shows the UARTi transmit/receive mode register.

Table 1.16.2. Specifications of clock synchronous serial I/O mode (1)

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• When internal clock is selected (bit 3 at addresses 03A0$_{16}$, 03A8$_{16}$, 01F8$_{16}$ = "0") : f_{SIO2}/ 2(n+1)(Note 1) j = 2, 8, 32 • When external clock is selected (bit 3 at addresses 03A0$_{16}$, 03A8$_{16}$, 01F8$_{16}$ = "1") : Input from CLKi pin
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$_{16}$, 01FD$_{16}$) = "1" – Transmit buffer empty flag (bit 1 at addresses 03A5$_{16}$, 03AD$_{16}$, 01FD$_{16}$) = "0" – When $\overline{CTS}$ function 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 addresses 03A4$_{16}$, 03AC$_{16}$, 01FC$_{16}$) = "0": CLKi input level = "H" – CLKi polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 01FC$_{16}$) = "1": CLKi 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$_{16}$, 01FD$_{16}$) = "1" – Transmit enable bit (bit 0 at addresses 03A5$_{16}$, 03AD$_{16}$, 01FD$_{16}$) = "1" – Transmit buffer empty flag (bit 1 at addresses 03A5$_{16}$, 03AD$_{16}$, 01FD$_{16}$) = "0" • Furthermore, if external clock is selected, the following requirements must also be met: – CLKi polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 01FC$_{16}$) = "0": CLKi input level = "H" – CLKi polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 01FC$_{16}$) = "1": CLKi input level = "L"
Interrupt request generation timing	• When transmitting – Transmit interrupt cause select bit (bits 0, 1 at address 03B0$_{16}$, bit 4 at address 01FD$_{16}$) = "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$_{16}$, bit 4 at address 01FD$_{16}$) = "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 2) This error occurs when the next data is ready before contents of UARTi receive buffer register are read out

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

Note 2: 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".

Table 1.16.3. Specifications of clock synchronous serial I/O mode (2)

Item	Specification
Select function	• CLK polarity selection Whether transmit data is output/input timing 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) (Note) UART1 transfer clock can be chosen by software to be output from one of the two pins set • Separate $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ pins (UART0) (Note) UART0 $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ pins each can be assigned to separate pins • Switching serial data logic (UART2) Whether to reverse data in writing to the transmission buffer register or reading the reception buffer register can be selected • TxD, RxD I/O polarity reverse (UART2) This function is reversing TxD port output and RxD port input. All I/O data level is reversed

Note: The transfer clock output from multiple pins and the separate $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ pins functions cannot be selected simultaneously.

UARTi transmit/receive mode registers

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 (Note)	○	○
STPS	Invalid in clock synchronous serial I/O mode		○	○
PRY			○	○
PRYE			○	○
SLEP	0 (Must always be "0" in clock synchronous serial I/O mode)		○	○

Note: Set the corresponding port direction register to "0".

UART2 transmit/receive mode registers

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

Symbol: U2MR — Address: 01F8$_{16}$ — 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 (Note 2)	○	○
STPS	Invalid in clock synchronous serial I/O mode		○	○
PRY			○	○
PRYE			○	○
IOPOL	TxD, RxD I/O polarity reverse bit (Note 1)	0 : No reverse 1 : Reverse	○	○

Note 1: Usually set to "0".
Note 2: Set the corresponding port direction register to "0".

Figure 1.16.9. UARTi transmit/receive mode register in clock synchronous serial I/O mode

Table 1.16.4 lists the functions of the input/output pins during clock synchronous serial I/O mode. This table shows the pin functions when the transfer clock output from multiple pins and the separate $\overline{CTS}$/$\overline{RTS}$ pins functions are not selected. 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.)

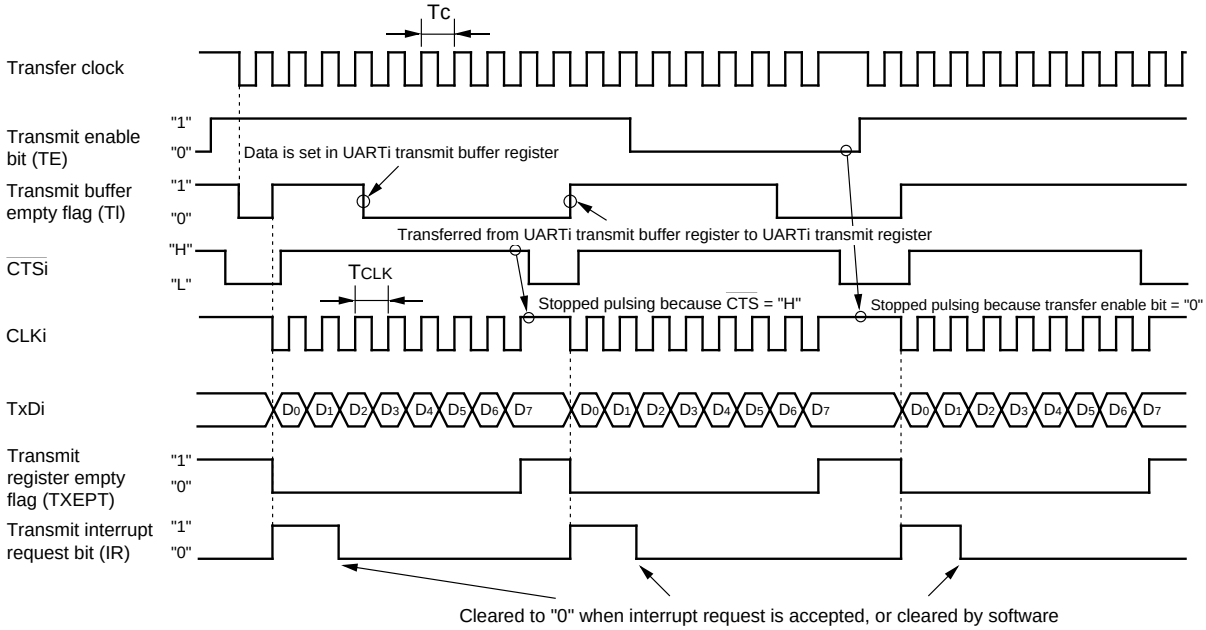
Table 1.16.4. Input/output pin functions in clock synchronous serial I/O mode

(when transfer clock output from multiple pins and separate $\overline{CTS}$/$\overline{RTS}$ pins functions are not selected)

Pin name	Function	Method of selection
TxDi (P6$_3$, P6$_7$, P7$_0$)	Serial data output	(Outputs dummy data when performing reception only)
RxDi (P6$_2$, P6$_6$, P7$_1$)	Serial data input	Port P6$_2$, P6$_6$ and P7$_1$ direction register (bits 2 and 6 at address 03EE$_{16}$, bit 1 at address 03EF$_{16}$)= "0" (Can be used as an input port when performing transmission only)
CLKi (P6$_1$, P6$_5$, P7$_2$)	Transfer clock output	Internal/external clock select bit (bit 3 at address 03A0$_{16}$, 03A8$_{16}$, 01F8$_{16}$) = "0"
	Transfer clock input	Internal/external clock select bit (bit 3 at address 03A0$_{16}$, 03A8$_{16}$, 01F8$_{16}$) = "1" Port P6$_1$, P6$_5$ and P7$_2$ direction register (bits 1 and 5 at address 03EE$_{16}$, bit 2 at address 03EF$_{16}$) = "0"
$\overline{CTS}$i/$\overline{RTS}$i (P6$_0$, P6$_4$, P7$_3$)	$\overline{CTS}$ input	$\overline{CTS}$/$\overline{RTS}$ disable bit (bit 4 at address 03A4$_{16}$, 03AC$_{16}$, 01FC$_{16}$) = "0" $\overline{CTS}$/$\overline{RTS}$ function select bit (bit 2 at address 03A4$_{16}$, 03AC$_{16}$, 01FC$_{16}$) = "0" Port P6$_0$, P6$_4$ and P7$_3$ direction register (bits 0 and 4 at address 03EE$_{16}$, bit 3 at address 03EF$_{16}$) = "0"
	$\overline{RTS}$ output	$\overline{CTS}$/$\overline{RTS}$ disable bit (bit 4 at address 03A4$_{16}$, 03AC$_{16}$, 01FC$_{16}$) = "0" $\overline{CTS}$/$\overline{RTS}$ function select bit (bit 2 at address 03A4$_{16}$, 03AC$_{16}$, 01FC$_{16}$) = "1"
	Programmable I/O port	$\overline{CTS}$/$\overline{RTS}$ disable bit (bit 4 at address 03A4$_{16}$, 03AC$_{16}$, 01FC$_{16}$) = "1"

Clock Synchronous Serial I/O Mode

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

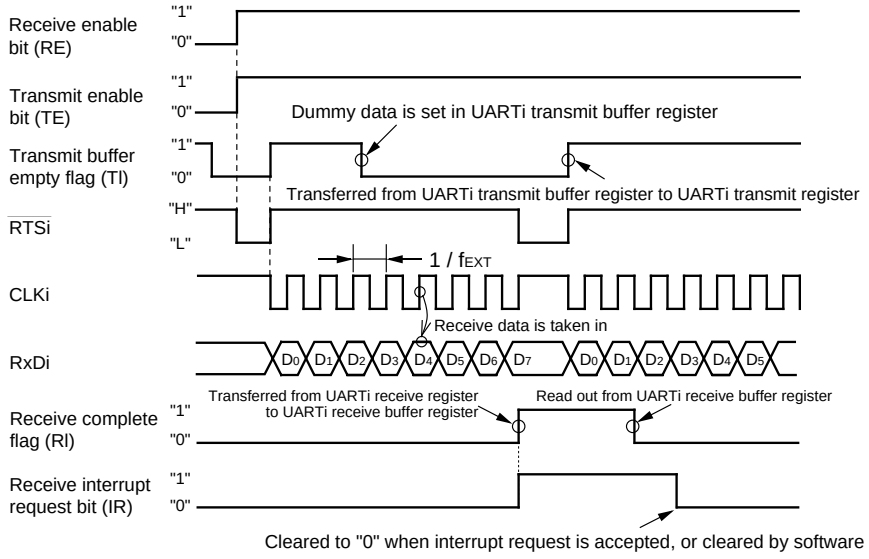


Shown in ( ) are bit symbols.

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 = TCLK = 2(n + 1) / fjSIO2
fjSIO2: frequency of BRGi count source (j = 2, 8, 32)
n: value set to BRGi

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



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".

fEXT: frequency of external clock

Meet the following conditions are met 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 1.16.10. Typical transmit/receive timings in clock synchronous serial I/O mode

(a) Polarity select function

As shown in Figure 1.16.11, the CLK polarity select bit (bit 6 at addresses 03A4$_{16}$, 03AC$_{16}$, 01FC$_{16}$) allows selection of the polarity of the transfer clock.

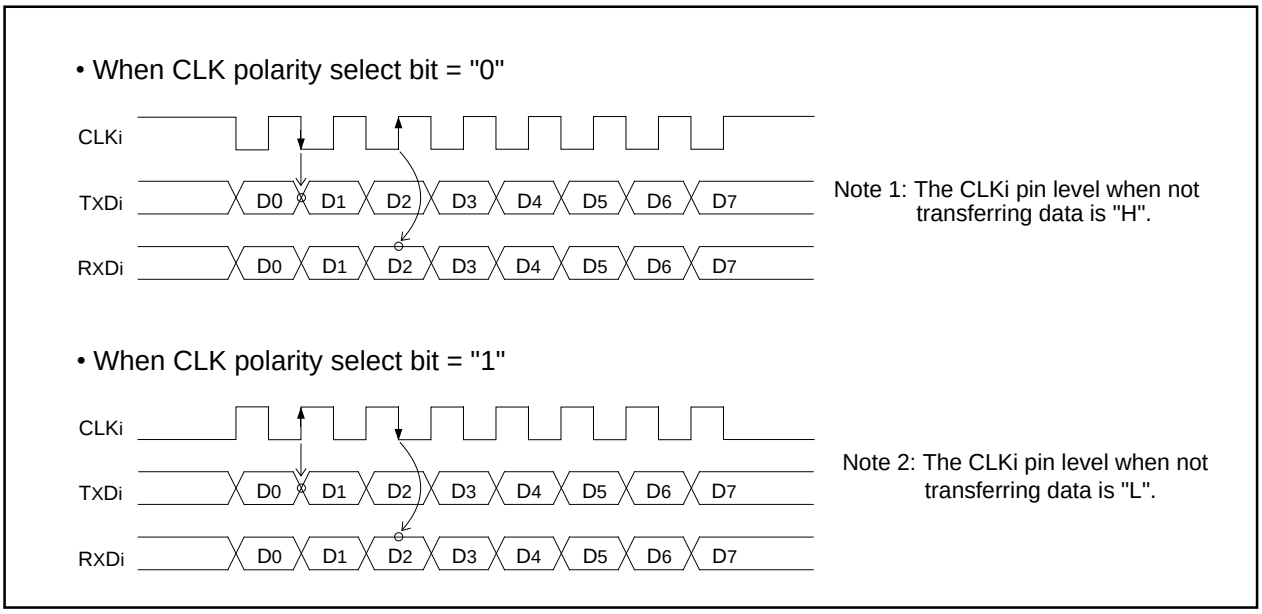


Figure 1.16.11. Polarity of transfer clock

(b) LSB first/MSB first select function

As shown in Figure 1.16.12, when the transfer format select bit (bit 7 at addresses 03A4$_{16}$, 03AC$_{16}$, 01FC$_{16}$) = "0", the transfer format is "LSB first"; when the bit = "1", the transfer format is "MSB first".

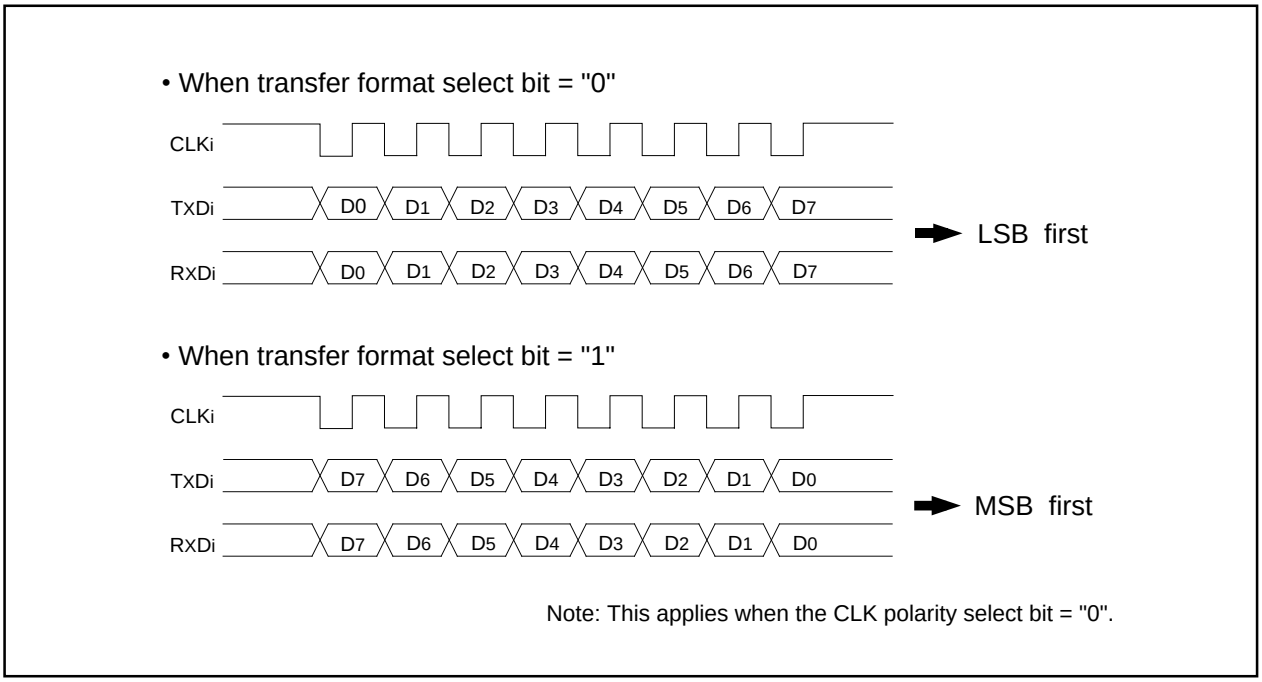


Figure 1.16.12. Transfer format

(c) Transfer clock output from multiple pins function (UART1)

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 1.16.13.) The multiple pins function is valid only when the internal clock is selected for UART1. Note that when this function is selected, UART1 $\overline{CTS}$/$\overline{RTS}$ function cannot be used.

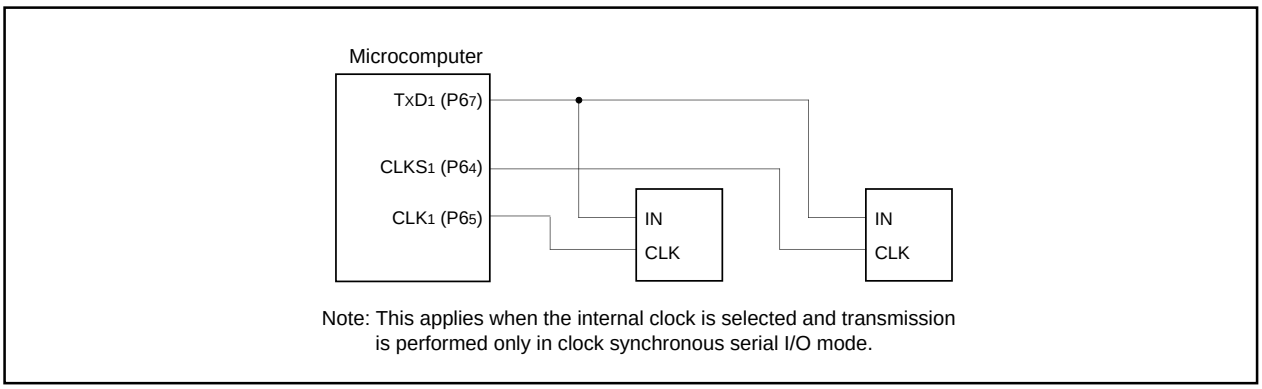


Figure 1.16.13. The transfer clock output from the multiple pins function usage

(d) Continuous receive mode

If the continuous receive mode enable bit (bits 2 and 3 at address 03B0$_{16}$, bit 5 at address 01FD$_{16}$) 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.

(e) Separate $\overline{CTS}$/$\overline{RTS}$ pins function (UART0)

This function works the same way as in the clock asynchronous serial I/O (UART) mode. The method of setting and the input/output pin functions are both the same, so refer to select function in the next section, "(2) Clock asynchronous serial I/O (UART) mode". Note that this function is invalid if the transfer clock output from the multiple pins function is selected.

(f) Serial data logic switch function (UART2)

When the data logic select bit (bit6 at address 01FD$_{16}$) = "1", and writing to transmit buffer register or reading from receive buffer register, data is reversed. Figure 1.16.14 shows the example of serial data logic switch timing.

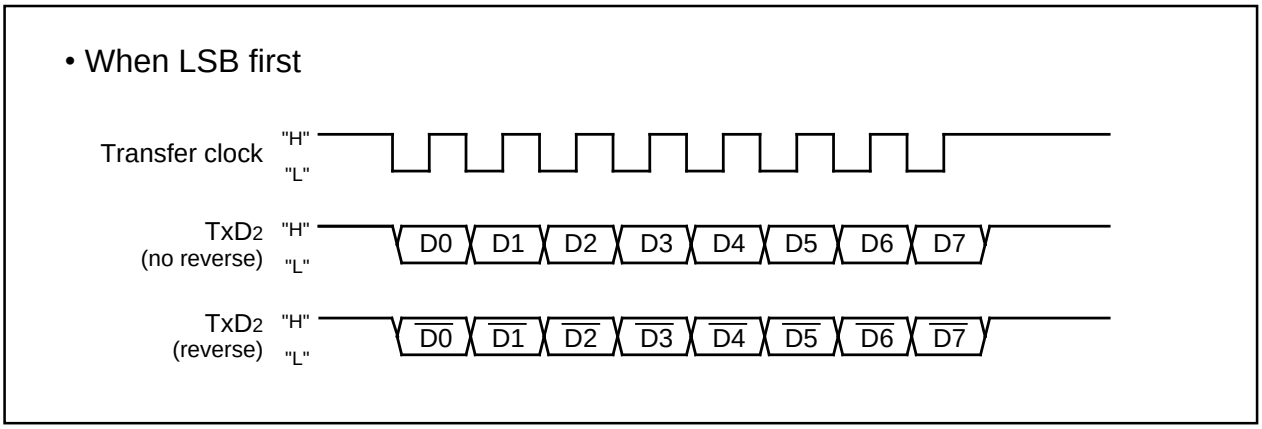


Figure 1.16.14. Serial data logic switch timing

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

The UART mode allows transmitting and receiving data after setting the desired transfer rate and transfer data format. Tables 1.16.5 and 1.16.6 list the specifications of the UART mode. Figure 1.16.15 shows the UARTi transmit/receive mode register.

Table 1.16.5. Specifications of UART mode (1)

Item	Specification
Transfer data format	• Start bit: 1 bit • Parity bit: Odd, even, or nothing as selected • Stop bit: 1 bit or 2 bits as selected • Character bit (transfer data): 7 bits, 8 bits, or 9 bits as selected
Transfer clock	• When internal clock is selected (bit 3 at addresses $03A0_{16}$, $03A8_{16}$, $01F8_{16}$="0") : $fj_{SIO2}/16(n+1)$ (Note 1) j = 2, 8, 32 • When external clock is selected (bit 3 at addresses $03A0_{16}$, $03A8_{16}$, $01F8_{16}$ ="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 $03A5_{16}$, $03AD_{16}$, $01FD_{16}$) = "1" - Transmit buffer empty flag (bit 1 at addresses $03A5_{16}$, $03AD_{16}$, $01FD_{16}$) = "0" - When $\overline{CTS}$ function 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 $03A5_{16}$, $03AD_{16}$, $01FD_{16}$) = "1" - Start bit detection
Interrupt request generation timing	• When transmitting - Transmit interrupt cause select bits (bits 0, 1 at address $03B0_{16}$, bit4 at address $01FD_{16}$) = "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 $03B0_{16}$, bit4 at address $01FD_{16}$) = "1": Interrupts requested when data transmission fromUARTi 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

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

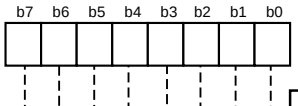
Note 2: f_{EXT} is input from the CLKi pin.

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".

Table 1.16.6. Specifications of UART mode (2)

Item	Specification
Select function	• Separate $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ pins (UART0) UART0 $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ pins each can be assigned to separate pins • Sleep mode selection (UART0, UART1) This mode is used to transfer data to and from one of multiple slave microcomputers • Serial data logic switch (UART2) This function is reversing logic value of transferring data. Start bit, parity bit and stop bit are not reversed. • TxD, RxD I/O polarity switch (UART2) This function is reversing TxD port output and RxD port input. All I/O data level is reversed.

UARTi transmit / receive mode registers

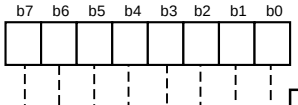


Symbol: UiMR(i=0,1) | Address: $03A0_{16}$, $03A8_{16}$ | When reset: 00_{16}

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 (Note)	○	○
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	○	○

Note: Set the corresponding port direction register to "0".

UART2 transmit / receive mode register



Symbol: U2MR | Address: $01F8_{16}$ | 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	Must always be fixed to "0"	○	○
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	○	○
IOPOL	TxD, RxD I/O polarity reverse bit (Note)	0 : No reverse 1 : Reverse	○	○

Note: Usually set to "0".

Figure 1.16.15. UARTi transmit/receive mode register in UART mode

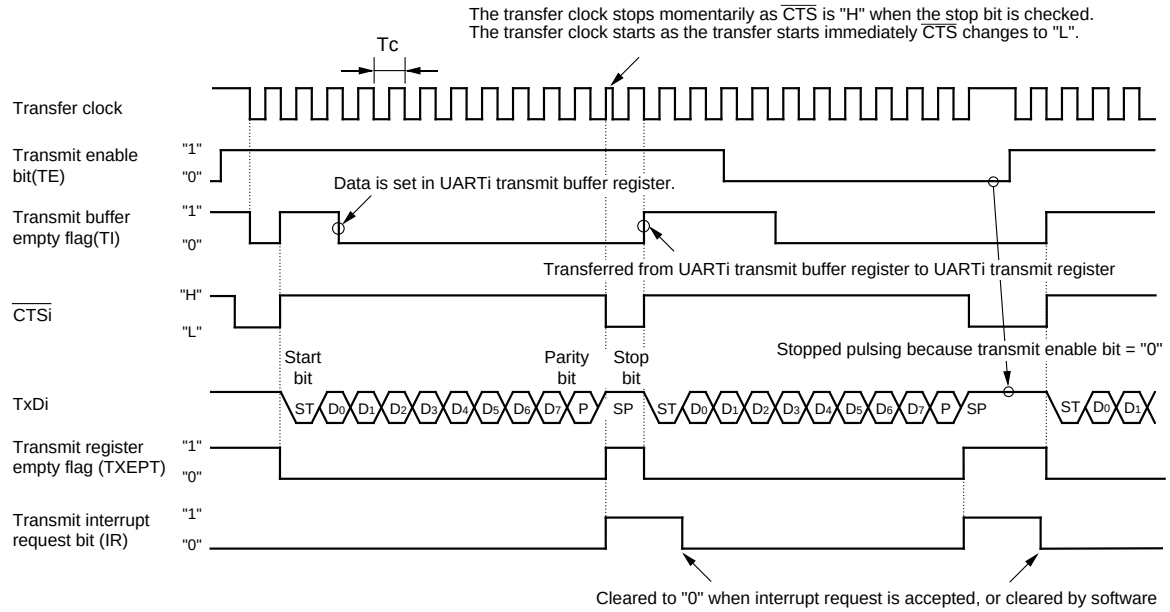
Table 1.16.7 lists the functions of the input/output pins during UART mode. This table shows the pin functions when the separate $\overline{CTS}/\overline{RTS}$ pins function is not selected. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs a "H". (If the N-channel open-drain is selected, this pin is in floating state.)

Table 1.16.7. Input/output pin functions in UART mode
(when separate $\overline{CTS}/\overline{RTS}$ pins function is not selected)

Pin name	Function	Method of selection
TxDi ($P6_3$, $P6_7$, $P7_0$)	Serial data output	
RxDi ($P6_2$, $P6_6$, $P7_1$)	Serial data input	Port $P6_2$, $P6_6$ and $P7_1$ direction register (bits 2 and 6 at address $03EE_{16}$, bit 1 at address $03EF_{16}$)= "0" (Can be used as an input port when performing transmission only)
CLKi ($P6_1$, $P6_5$, $P7_2$)	Programmable I/O port	Internal/external clock select bit (bit 3 at address $03A0_{16}$, $03A8_{16}$, $01F8_{16}$) = "0"
	Transfer clock input	Internal/external clock select bit (bit 3 at address $03A0_{16}$, $03A8_{16}$, $01F8_{16}$) = "1" Port $P6_1$, $P6_5$ and $P7_2$ direction register (bits 1 and 5 at address $03EE_{16}$, bit 2 at address $03EF_{16}$) = "0"
$\overline{CTS}i/\overline{RTS}i$ ($P6_0$, $P6_4$, $P7_3$)	$\overline{CTS}$ input	$\overline{CTS}/\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $01FC_{16}$) = "0" $\overline{CTS}$/RTS function select bit (bit 2 at address $03A4_{16}$, $03AC_{16}$, $01FC_{16}$) = "0" Port $P6_0$, $P6_4$ and $P7_3$ direction register (bits 0 and 4 at address $03EE_{16}$, bit 3 at address $03EF_{16}$) = "0"
	$\overline{RTS}$ output	$\overline{CTS}/\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $01FC_{16}$) = "0" $\overline{CTS}$/RTS function select bit (bit 2 at address $03A4_{16}$, $03AC_{16}$, $01FC_{16}$) = "1"
	Programmable I/O port	$\overline{CTS}/\overline{RTS}$ disable bit (bit 4 at address $03A4_{16}$, $03AC_{16}$, $01FC_{16}$) = "1"

Clock Asynchronous Serial I/O (UART) Mode

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

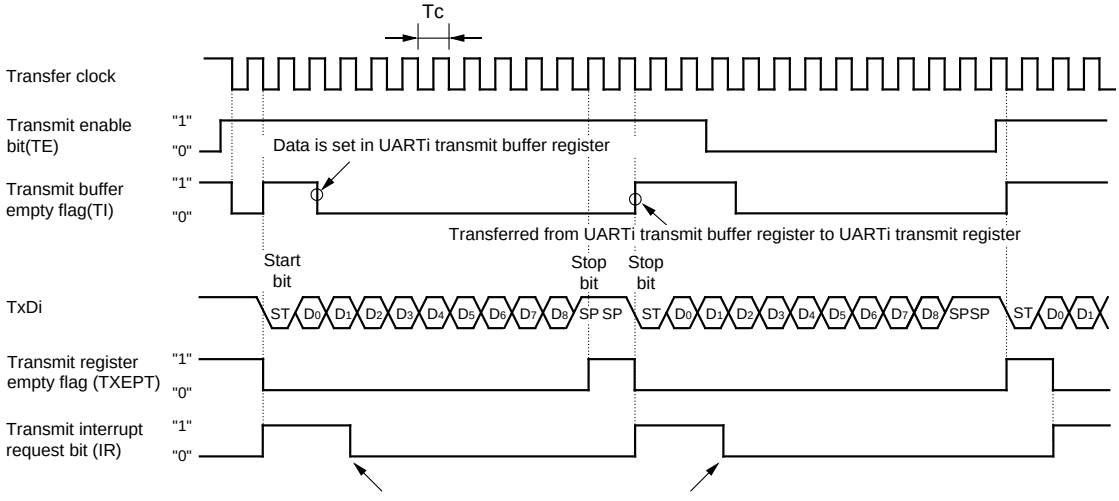


Shown in ( ) are bit symbols.

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) / fjSIO2 or 16 (n + 1) / fEXT
fjSIO2 : frequency of BRGi count source (j = 2, 8, 32)
fEXT : frequency of BRGi 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)



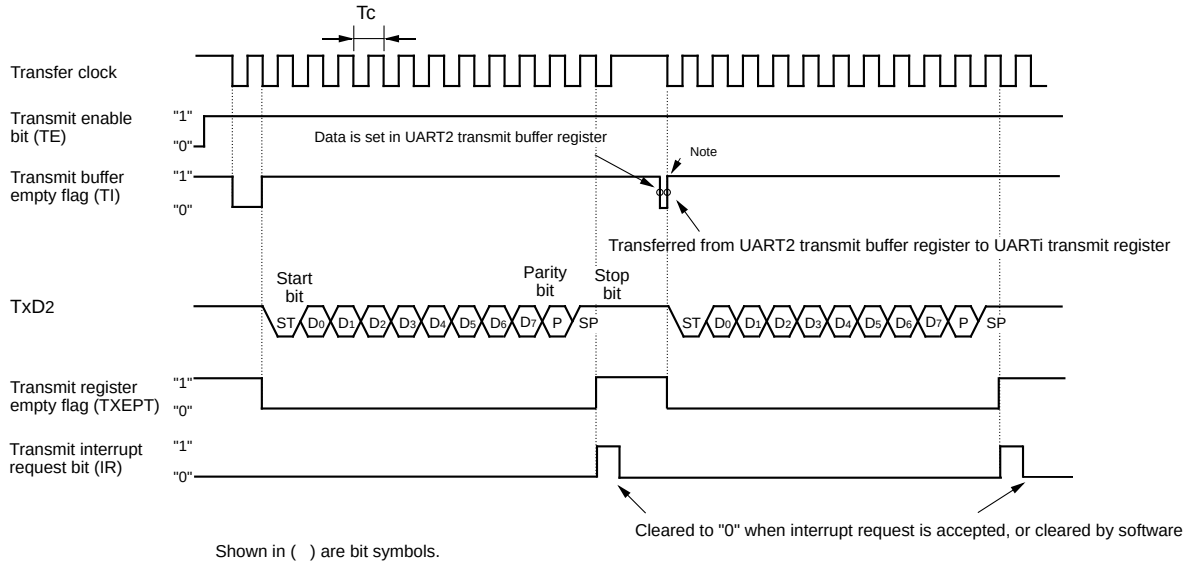
Shown in ( ) are bit symbols.

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

Tc = 16 (n + 1) / fjSIO2 or 16 (n + 1) / fEXT
fjSIO2 : frequency of BRGi count source (j = 2, 8, 32)
fEXT : frequency of BRGi count source (external clock)
n : value set to BRGi

Figure 1.16.16. Typical transmit timings in UART mode (UART0, UART1)

• 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.
- Transmit interrupt cause select bit = "1".

Tc = 16 (n + 1) / fjSIO2
fjSIO2 : frequency of BRG2 count source (j = 2, 8, 32)
n : value set to BRG2

Note: The transmit is started with overflow timing of BRG after having written in a value at the transmit buffer in the above timing.

Figure 1.16.17. Typical transmit timings in UART mode (UART2)

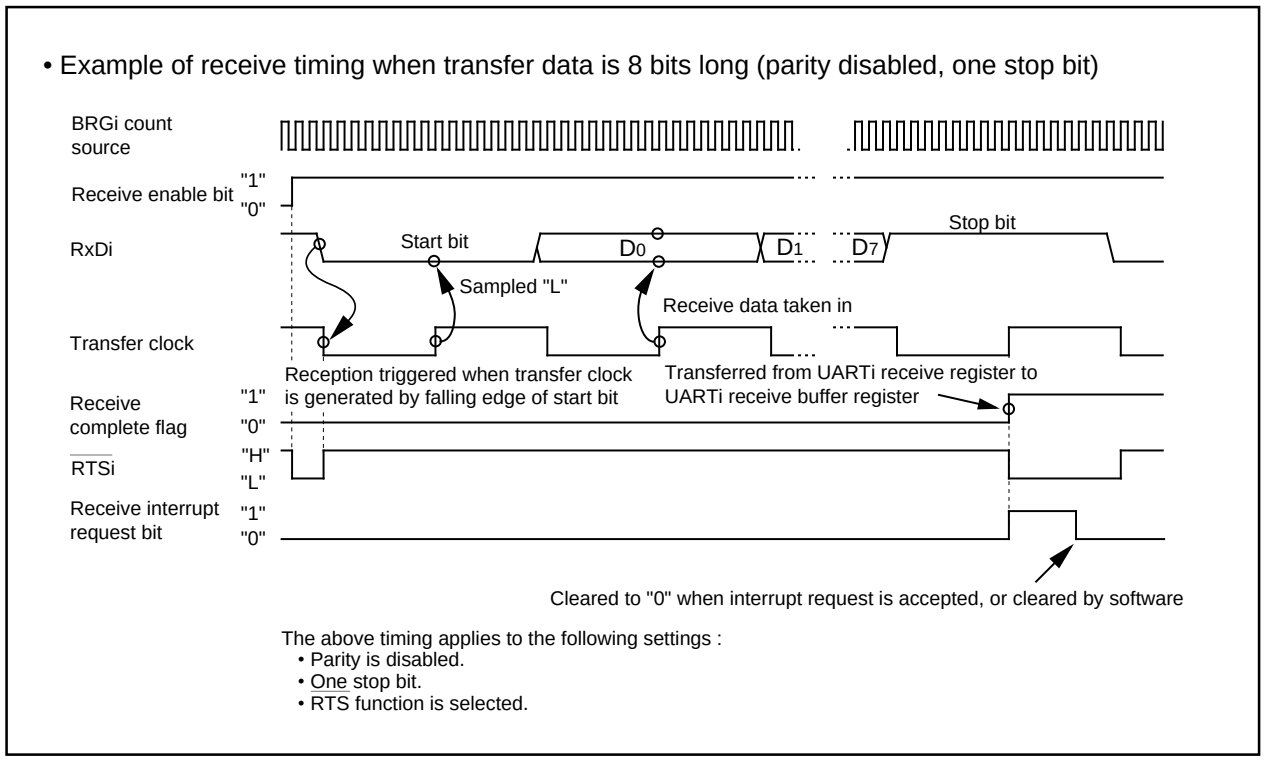


Figure 1.16.18. Typical receive timing in UART mode

(a) Separate $\overline{CTS}/\overline{RTS}$ pins function (UART0)

With the separate $\overline{CTS}/\overline{RTS}$ bit (bit 6 at address $03B0_{16}$) is set to "1", the unit outputs/inputs the $\overline{CTS}$ and $\overline{RTS}$ signals on different pins. (See Figure 1.16.19.) This function is valid only for UART0. Note that if this function is selected, the $\overline{CTS}/\overline{RTS}$ function for UART1 cannot be used, but set to "0", both the $\overline{CTS}/\overline{RTS}$ function select bit (bit 2 at address $03AC_{16}$) and the $\overline{CTS}/\overline{RTS}$ disable bit (bit 4 at address $03AC_{16}$).

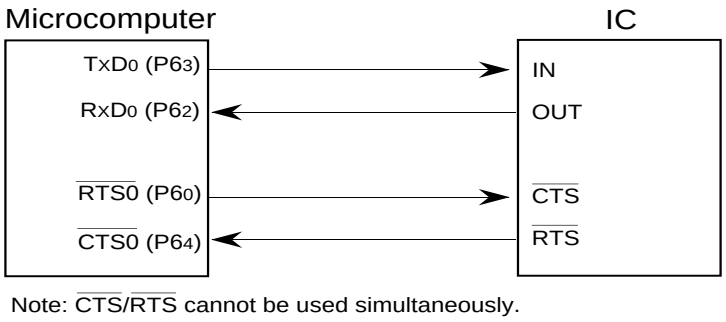


Figure 1.16.19. The separate $\overline{CTS}/\overline{RTS}$ pins function usage

(b) 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 $03A0_{16}$ and $03A8_{16}$) 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".

(c) Function for switching serial data logic (UART2)

When the data logic select bit (bit 6 at address 01FD16) is assigned 1, data is inverted in writing to the transmission buffer register or reading the reception buffer register. Figure 1.16.20 shows the example of timing for switching serial data logic.

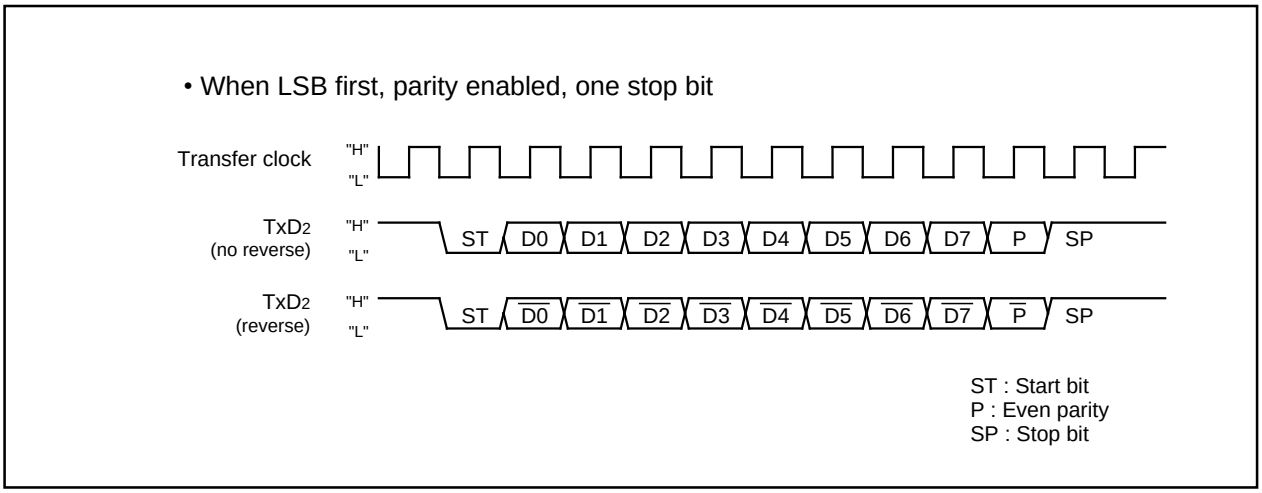


Figure 1.16.20. Timing for switching serial data logic

(d) TxD, RxD I/O polarity reverse function (UART2)

This function is to reverse TxD pin output and RxD pin input. The level of any data to be input or output (including the start bit, stop bit (s), and parity bit) is reversed. Set this function to "0" (not to reverse) for usual use.

(e) Bus collision detection function (UART2)

This function is to sample the output level of the TxD pin and the input level of the RxD pin at the rising edge of the transfer clock; if their values are different, then an interrupt request occurs. Figure 1.16.21 shows the example of detection timing of a buss collision (in UART mode).

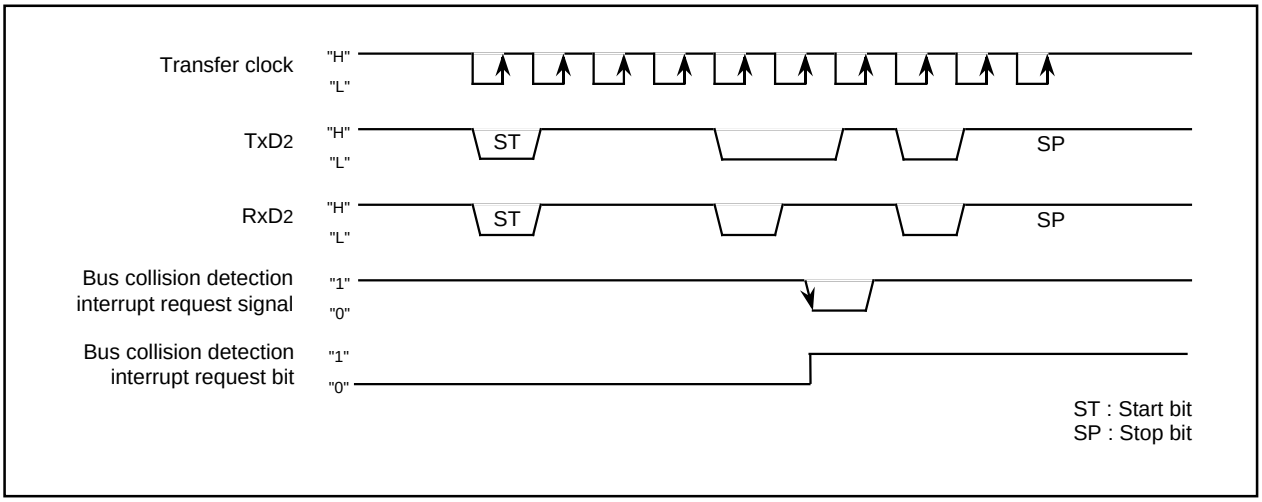


Figure 1.16.21. Detection timing of a bus collision (in UART mode)

(3) Clock-asynchronous serial I/O mode (compliant with the SIM interface)

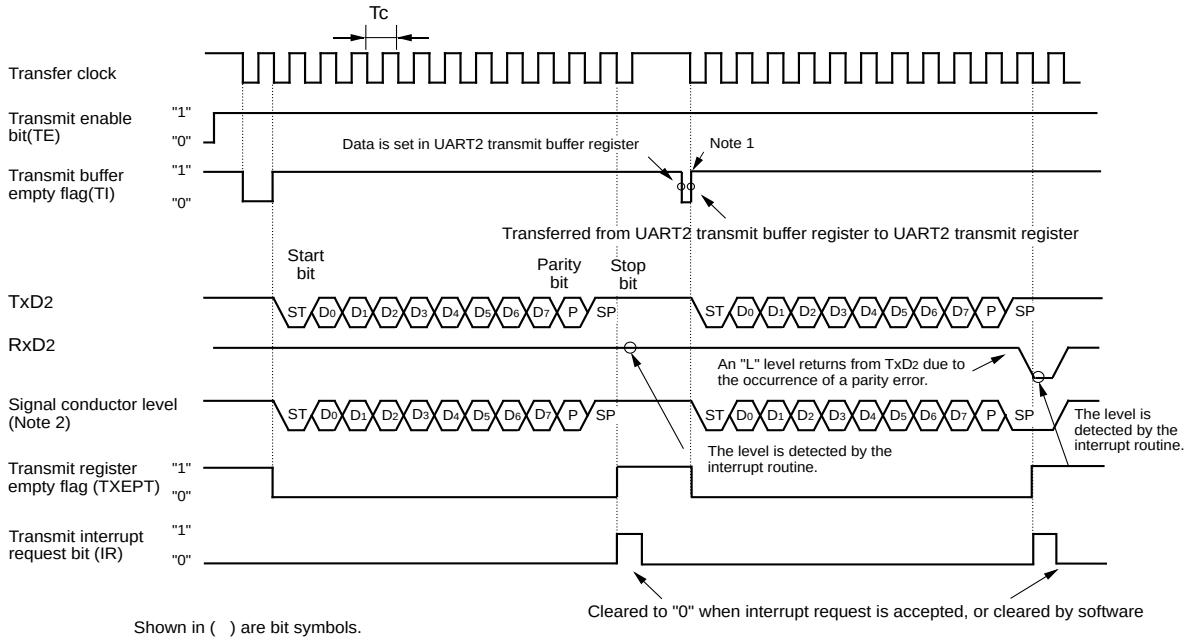
The SIM interface is used for connecting the microcomputer with a memory card I/C or the like; adding some extra settings in UART2 clock-asynchronous serial I/O mode allows the user to effect this function. Table 1.16.8 shows the specifications of clock-asynchronous serial I/O mode (used for the SIM interface).

Table 1.16.8. Specifications of clock-asynchronous serial I/O mode (compliant with the SIM interface)

Item	Specification
Transfer data format	• Transfer data 8-bit UART mode (bit 2 through bit 0 at address 01F8$_{16}$ = "101$_2$") • One stop bit (bit 4 at address 01F8$_{16}$ = "0") • With the direct format chosen Set parity to "even" (bit 5 and bit 6 at address 01F8$_{16}$ = "1" and "1" respectively) Set data logic to "direct" (bit 6 at address 01FD$_{16}$ = "0"). Set transfer format to LSB (bit 7 at address 01FC$_{16}$ = "0"). • With the inverse format chosen Set parity to "odd" (bit 5 and bit 6 at address 01F8$_{16}$ = "0" and "1" respectively) Set data logic to "inverse" (bit 6 at address 01FD$_{16}$ = "1") Set transfer format to MSB (bit 7 at address 01FC$_{16}$ = "1")
Transfer clock	• With the internal clock chosen (bit 3 at address 01F8$_{16}$ = "0") : f_{jSIO2} / 16 (n + 1) (Note 1) : j=2, 8, 32 (Do not set external clock)
Transmission / reception control	• Disable the $\overline{CTS}$ and $\overline{RTS}$ function (bit 4 at address 01FC$_{16}$ = "1")
Other settings	• The sleep mode select function is not available for UART2 • Set transmission interrupt factor to "transmission completed" (bit 4 at address 01FD$_{16}$ = "1")
Transmission start condition	• To start transmission, the following requirements must be met: - Transmit enable bit (bit 0 at address 01FD$_{16}$ = "1") - Transmit buffer empty flag (bit 1 at address 01FD$_{16}$ = "1")
Reception start condition	• To start reception, the following requirements must be met: - Reception enable bit (bit 2 at address 01FD$_{16}$ = "1") - Detection of a start bit
Interrupt request generation timing	• When transmitting When data transmission from the UART2 transfer register is completed (bit 4 at address 01FD$_{16}$ = "1") • When receiving When data transfer from the UART2 receive register to the UART2 receive buffer register is completed
Error detection	• Overrun error (see the specifications of clock-asynchronous serial I/O) (Note 2) • Framing error (see the specifications of clock-asynchronous serial I/O) • Parity error (see the specifications of clock-asynchronous serial I/O) - On the reception side, an "L" level is output from the TxD$_2$ pin by use of the parity error signal output function (bit 7 at address 01FD$_{16}$ = "1") when a parity error is detected - On the transmission side, a parity error is detected by the level of input to the RxD$_2$ pin when a transmission interrupt occurs • The error sum flag (see the specifications of clock-asynchronous serial I/O)

Note 1: "n" denotes the value 00_{16} to FF$_{16}$ that is set to the UART2 bit rate generator.

Note 2: If an overrun error occurs, the UART2 receive buffer will have the next data written in. Note also that the UART2 receive interrupt request bit is not set to "1".

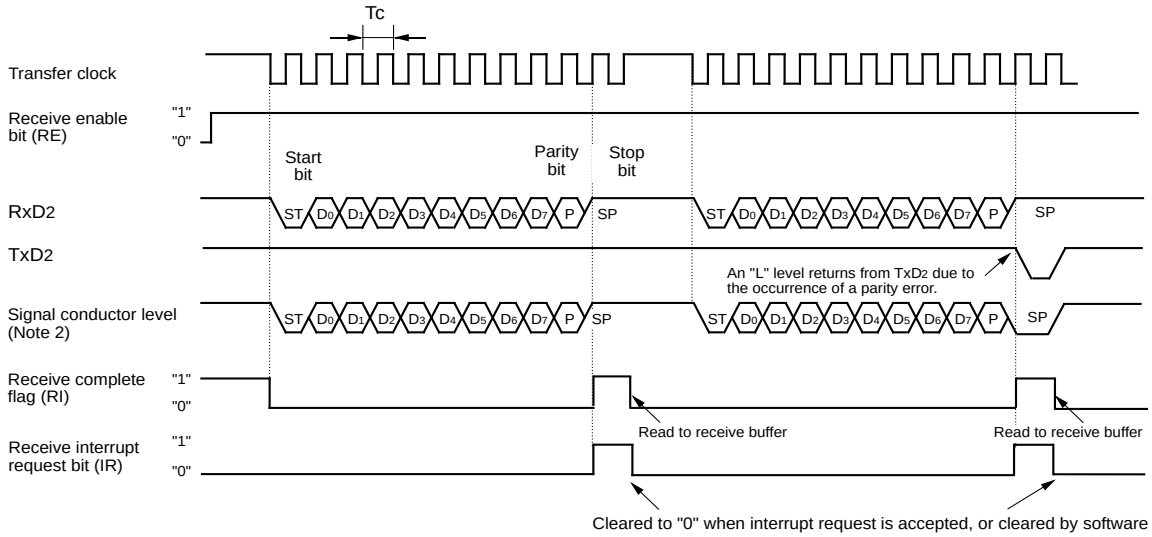


Shown in ( ) are bit symbols.

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

Tc = 16 (n + 1) / fjSIO2
fjSIO2 : frequency of BRG2 count source (j = 2, 8, 32)
n : value set to BRG2

Note 1: The transmit is started with overflow timing of BRG after having written in a value at the transmit buffer in the above timing.



Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- Transmit interrupt cause select bit = "0".

Tc = 16 (n + 1) / fjSIO2
fjSIO2 : frequency of BRG2 count source (j = 2, 8, 32)
n : value set to BRG2

Note 2: Equal in waveform because TxD2 and RxD2 are connected.

Figure 1.16.22. Typical transmit/receive timing in UART mode (compliant with the SIM interface)

(a) Function for outputting a parity error signal

During reception, with the error signal output enable bit (bit 7 at address 01FD16) assigned "1", you can output an "L" level from the TxD2 pin when a parity error is detected. And during transmission, comparing with the case in which the error signal output enable bit (bit 7 at address 01FD16) is assigned "0", the transmission completion interrupt occurs in the half cycle later of the transfer clock. Therefore parity error signals can be detected by a transmission completion interrupt program. Figure 1.16.23 shows the output timing of the parity error signal.

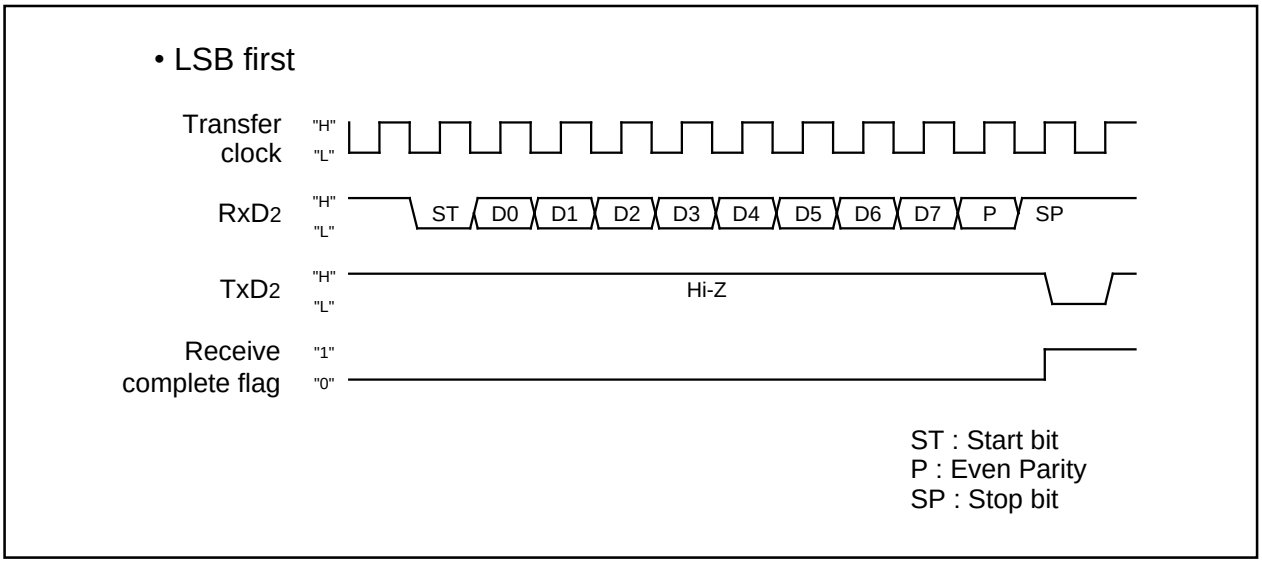


Figure 1.16.23. Output timing of the parity error signal

(b) Direct format/inverse format

Connecting the SIM card allows you to switch between direct format and inverse format. If you choose the direct format, D0 data is output from TxD2. If you choose the inverse format, D7 data is inverted and output from TxD2.

Figure 1.16.24 shows the SIM interface format.

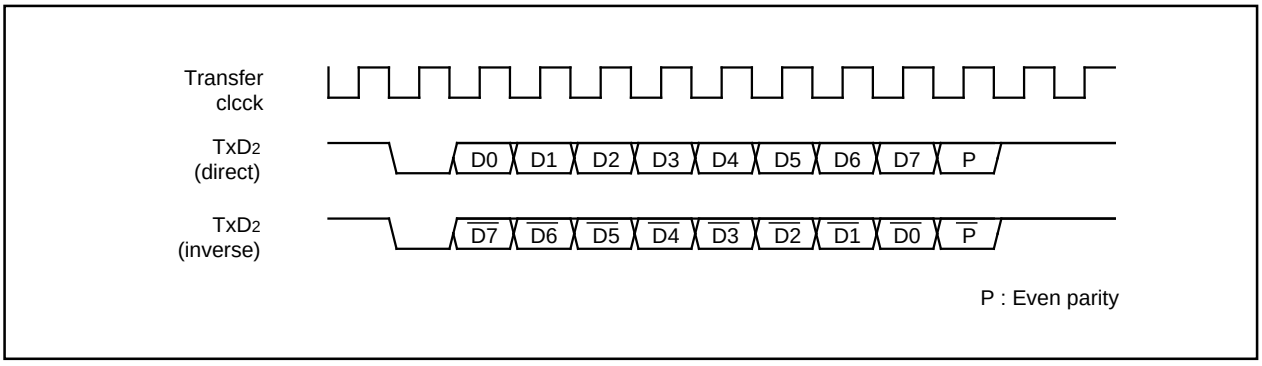


Figure 1.16.24. SIM interface format

Figure 1.16.25 shows the example of connecting the SIM interface. When setting the data output select bit (bit 5 at address 01FC16) to "1", connect TxD2 and RxD2 and apply pull-up.

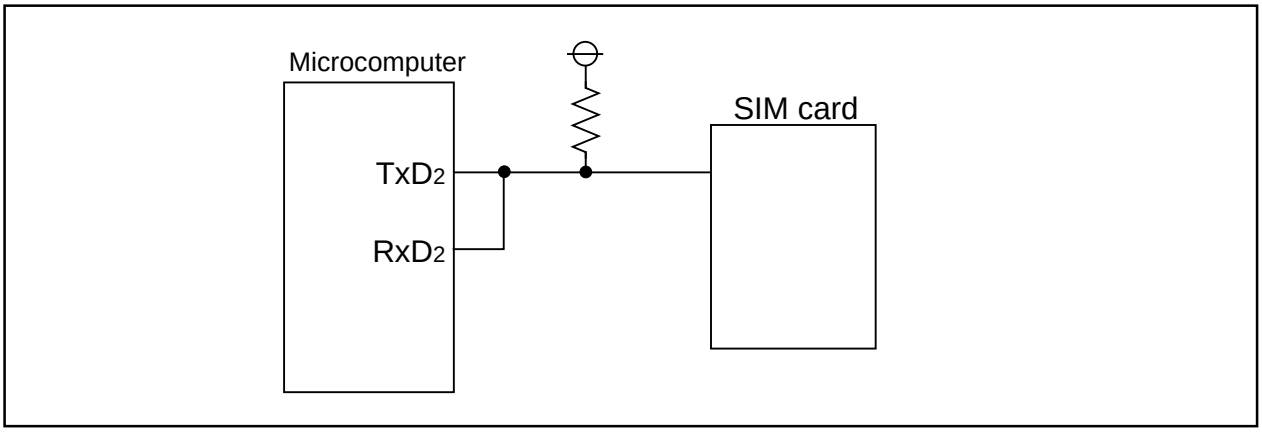


Figure 1.16.25. Connecting the SIM interface (When setting NCH bit to "1")

UART2 Special Mode Register

The UART2 special mode register (address 01F7$_{16}$) is used to control UART2 in various ways.
Figure 1.16.26 shows the special UART2 mode register.
In the first place, the control bits related to the I^{2}C bus(simplified I^{2}C bus) interface are explained.
Bit 0 of the UART2 special mode register (address 01F7$_{16}$) is used as the I^{2}C mode selection bit.
Setting "1" in the I^{2}C mode selection bit (bit 0 at address 01F7$_{16}$) goes the circuit to achieve the I^{2}C bus (simplified I^{2}C bus) interface effective.
Since this function uses clock-synchronous serial I/O mode, be sure to set this bit to "0" in UART mode.
Table 1.16.9 shows the relation between the I^{2}C mode selection bit and respective control workings.
In order to configure P7$_0$ as N-channel set the data output select bit (bit 5 at address 01FC$_{16}$) to "1".

UART2 special mode register

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	When reset
0								U2SMR	01F7$_{16}$	00$_{16}$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
IICM	I^{2}C mode selection bit	0 : Normal mode 1 : I^{2}C mode	Must always be "0"	○	○
ABC	Arbitration loss detecting flag control bit	0 : Update per bit 1 : Update per byte	Must always be "0"	○	○
BBS	Bus busy flag	0 : STOP condition detected 1 : START condition detected	Must always be "0"	○	○ (Note)
LSYN	SCLL sync output enable bit	0 : Disabled 1 : Enabled	Must always be "0"	○	○
ABSCS	Bus collision detect sampling clock select bit	Must always be "0"	0 : Rising edge of transfer clock 1 : Underflow signal of timer A0	○	○
ACSE	Auto clear function select bit of transmit enable bit	Must always be "0"	0 : No auto clear function 1 : Auto clear at occurrence of bus collision	○	○
SSS	Transmit start condition select bit	Must always be "0"	0 : Ordinary 1 : Falling edge of RxD$_2$	○	○
Reserved bit		Must always be "0"		○	○

Note: Nothing but "0" may be written.

Figure 1.16.26. UART2 special mode register

UART2 Special Mode Register

Table 1.16.9. Features in I^2C mode

	Function	Normal mode	I^2C mode (Note 1)
1	Factor of interrupt number 10 (Note 2)	Bus collision detection	Start condition detection or stop condition detection
2	Factor of interrupt number 15 (Note 2)	UART2 transmission	No acknowledgment detection (NACK)
3	Factor of interrupt number 16 (Note 2)	UART2 reception	Acknowledgment detection (ACK)
4	UART2 transmission output delay	Not delayed	Delayed
5	$P7_0$ at the time when UART2 is in use	TxD_2 (output)	SDA (input/output) (Note 3)
6	$P7_1$ at the time when UART2 is in use	RxD_2 (input)	SCL (input/output)
7	$P7_2$ at the time when UART2 is in use	CLK2	$P7_2$
8	DMA1 factor at the time when 1 1 0 1 is assigned to the DMA request cause select bits	UART2 reception	Acknowledgment detection (ACK)
9	Noise filter width	15ns	50ns
10	Reading $P7_1$	Reading the terminal when 0 is assigned to the direction register	Reading the terminal regardless of the value of the direction register
11	Initial value of UART2 output	H level (when "0" is assigned to the CLK polarity select bit)	The value set in latch $P7_0$ when the port is selected

Note 1: Make the settings given below when I^2C mode is in use.
Set "0 1 0" in bits 2, 1, 0 of the UART2 transmission/reception mode register.
Disable the CTS/RTS function. Select TxD_2 as Nch. Choose the MSB First function.
Note 2: Follow the steps given below to switch from a factor to another.
1. Disable the interrupt of the corresponding number.
2. Switch from a factor to another.
3. Reset the interrupt request flag of the corresponding number.
4. Set an interrupt level of the corresponding number.

Note 3: Set an initial value of SDA transmission output when serial I/O is invalid.

$P7_0$ through $P7_2$ conforming to the simplified I^2C bus

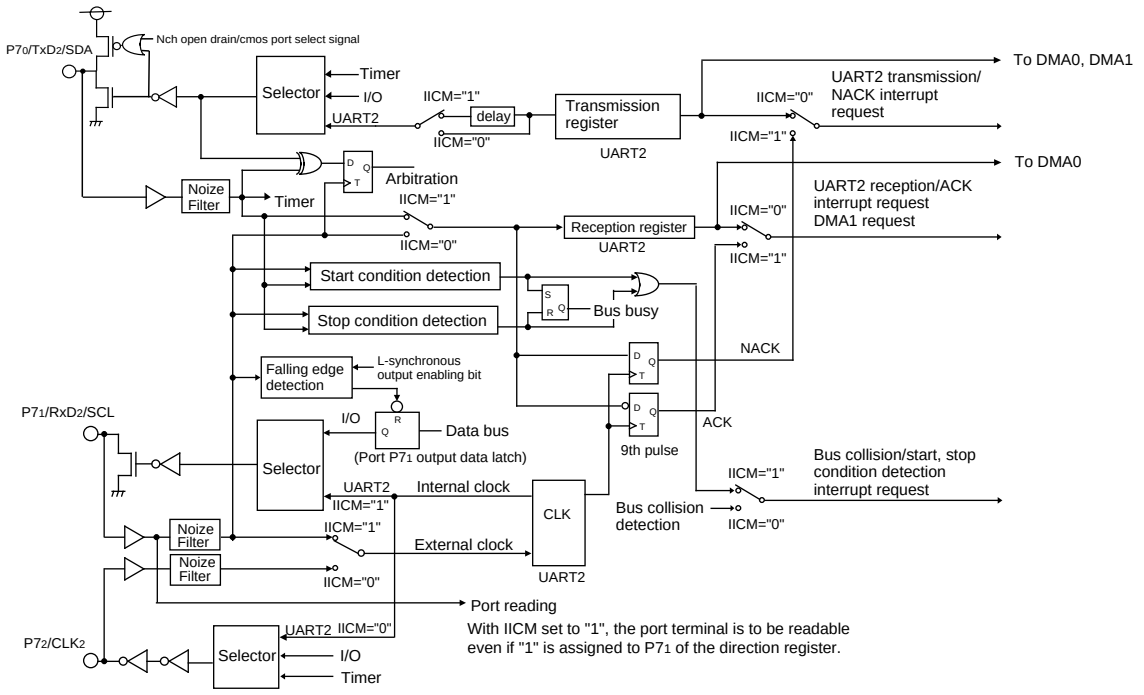


Figure 1.16.27. Functional block diagram for I^2C mode

UART2 Special Mode Register

Figure 1.16.27 shows the functional block diagram for I^2C mode. Setting "1" in the I^2C mode selection bit (IICM) causes ports $P7_0$, $P7_1$, and $P7_2$ to work as data transmission-reception terminal SDA, clock input-output terminal SCL, and port $P7_2$ respectively. A delay circuit is added to the SDA transmission output, so the SDA output changes after SCL fully goes to "L". An attempt to read Port $P7_1$ (SCL) results in getting the terminal's level regardless of the content of the port direction register. The initial value of SDA transmission output in this mode goes to the value set in port $P7_0$. The interrupt factors of the bus collision detection interrupt, UART2 transmission interrupt, and of UART2 reception interrupt turn to the start/stop condition detection interrupt, acknowledgment non-detection interrupt, and acknowledgment detection interrupt respectively.

The start condition detection interrupt refers to the interrupt that occurs when the falling edge of the SDA terminal ($P7_0$) is detected with the SCL terminal ($P7_1$) staying "H". The stop condition detection interrupt refers to the interrupt that occurs when the rising edge of the SDA terminal ($P7_0$) is detected with the SCL terminal ($P7_1$) staying "H". The bus busy flag (bit 2 of the special UART2 mode register) is set to "1" by the start condition detection, and set to "0" by the stop condition detection. The acknowledgment non-detection interrupt refers to the interrupt that occurs when the SDA terminal level is detected still staying "H" at the rising edge of the 9th transmission clock. The acknowledgment detection interrupt refers to the interrupt that occurs when SDA terminal's level is detected already went to "L" at the 9th transmission clock. Also, assigning "1 1 0 1" (UART2 reception) to the DMA1 request cause select bits provides the means to start up the DMA transfer by the effect of acknowledgment detection.

Bit 1 of the special UART2 mode register (address $01F7_{16}$) is used as the arbitration lost detecting flag control bit. Arbitration means the act of detecting the nonconformity between transmission data and SDA terminal data at the timing of the SCL rising edge. This detecting flag is located at bit 3 of the UART2 reception buffer register (address $01FF_{16}$), and "1" is set in this flag when nonconformity is detected. Use the arbitration lost detecting flag control bit to choose which way to use to update the flag, bit by bit or byte by byte. When setting this bit to "1" and updated the flag byte by byte if nonconformity is detected, the arbitration lost detecting flag is set to "1" at the falling edge of the 9th transmission clock. If updated the flag byte by byte, must judge and clear ("0") the arbitration lost detecting flag after completing the first byte acknowledge detect and before starting the next one byte transmission.

Bit 3 of the UART2 special mode register is used as SCL- and L-synchronous output enabling bit. Setting this bit to "1" resets the $P7_1$ data register to "0" in synchronization with the SCL terminal level going to "L".

Some other functions added are explained here. Figure 1.16.28 shows their workings.
Bit 4 of the UART2 special mode register is used as the bus collision detect sampling clock select bit. The bus collision detect interrupt occurs when the RxD2 level and TxD2 level do not match, but the nonconformity is detected in synchronization with the rising edge of the transfer clock signal if the bit is set to "0". If this bit is set to "1", the nonconformity is detected at the timing of the overflow of timer A0 rather than at the rising edge of the transfer clock.
Bit 5 of the UART2 special mode register is used as the auto clear function select bit of transmit enable bit. Setting this bit to "1" automatically resets the transmit enable bit to "0" when "1" is set in the bus collision detect interrupt request bit (nonconformity).
Bit 6 of the UART2 special mode register is used as the transmission start condition select bit. Setting this bit to "1" starts the TxD transmission in synchronization with the falling edge of the RxD terminal.

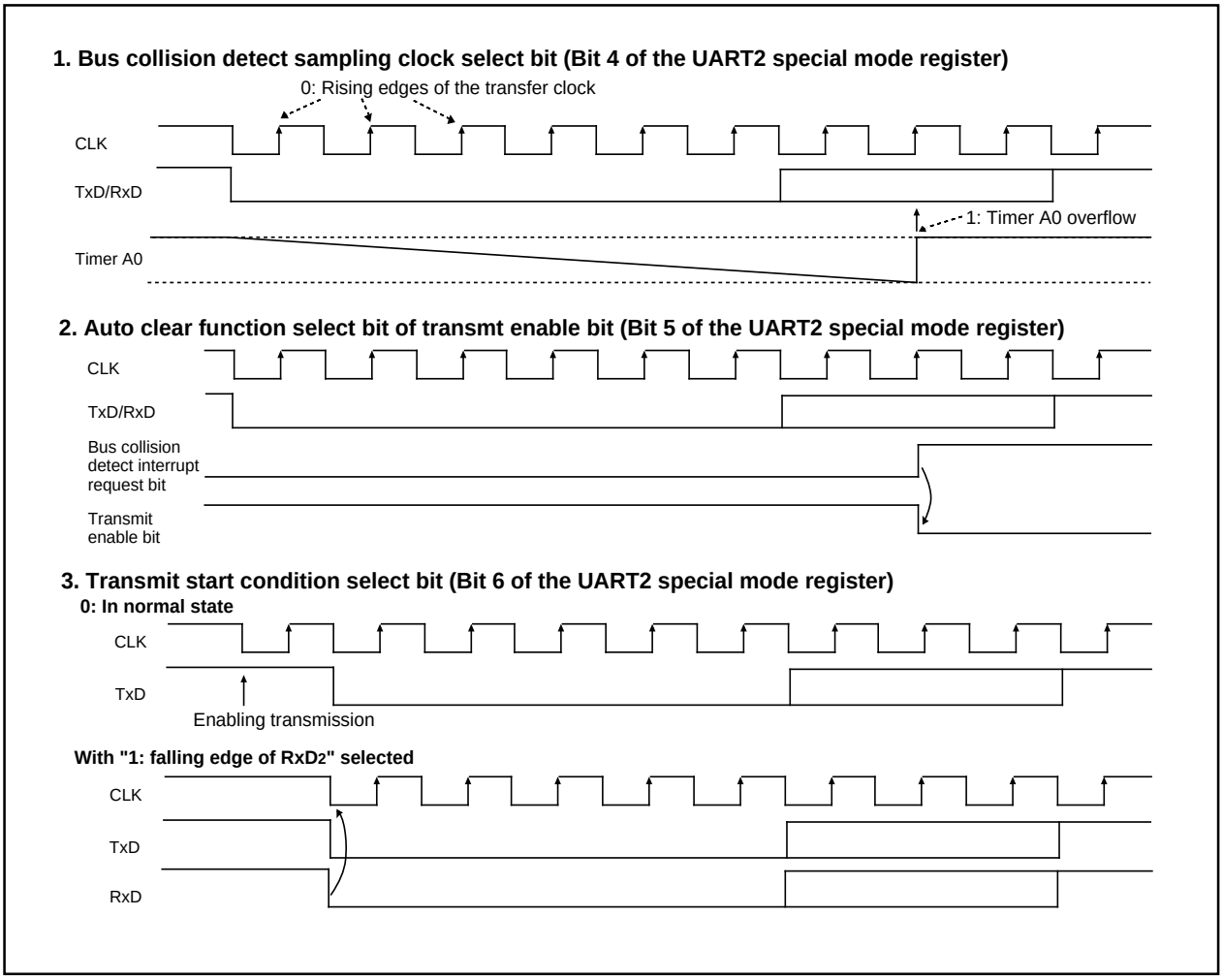


Figure 1.16.28. Some other functions added

UART2 Special Mode Register 2

The UART2 special mode register 2 (address 01F6$_{16}$) is used to further control UART2 in I^{2}C mode. Figure 1.16.29 shows the UART2 special mode register 2.

UART2 special mode register 2

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: U2SMR2 — Address: 01F6$_{16}$ — When reset: 00$_{16}$

Bit symbol	Bit name	Function	R	W
IICM2	I^{2}C mode selection bit 2	Refer to Table 1.16.10.	○	○
CSC	Clock-synchronous bit	0 : Disabled 1 : Enabled	○	○
SWC	SCL wait output bit	0 : Disabled 1 : Enabled	○	○ (Note)
ASL	SDA output stop bit	0 : Disabled 1 : Enabled	○	○
STAC	UART2 initialization bit	0 : Disabled 1 : Enabled	○	○
SWC2	SCL wait output bit 2	0: UART2 clock 1: 0 output	○	○
SDHI	SDA output disable bit	0: Enabled 1: Disabled (high impedance)	○	○
SHTC	Start/stop condition control bit	Set this bit to "1" in I^{2}C mode (refer to Table 1.16.11)	○	○

Figure 1.16.29. UART2 special mode register 2

UART2 Special Mode Register 2

Bit 0 of the UART2 special mode register 2 (address 01F6$_{16}$) is used as the I^{2}C mode selection bit 2. Table 1.16.10 shows the types of control to be changed by I^{2}C mode selection bit 2 when the I^{2}C mode selection bit is set to "1". Table 1.16.11 shows the timing characteristics of detecting the start condition and the stop condition. Set the start/stop condition control bit (bit 7 of UART2 special mode register 2) to "1" in I^{2}C mode.

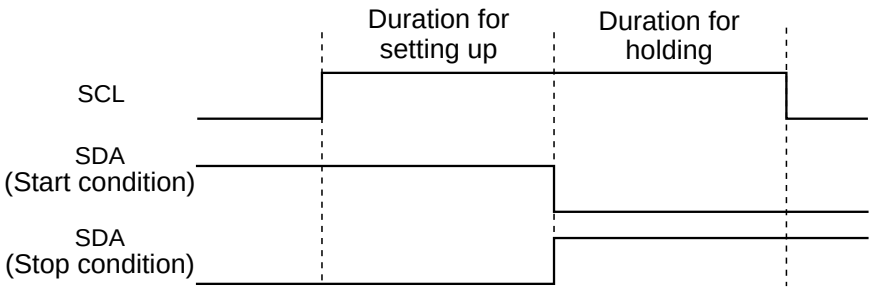
Table 1.16.10. Functions changed by I^{2}C mode selection bit 2

	Function	IICM2 = 0	IICM2 = 1
1	Factor of interrupt number 15	No acknowledgment detection (NACK)	UART2 transmission (the rising edge of the final bit of the clock)
2	Factor of interrupt number 16	Acknowledgment detection (ACK)	UART2 reception (the falling edge of the final bit of the clock)
3	DMA1 factor at the time when "1 1 0 1" is assigned to the DMA request cause select bits	Acknowledgment detection (ACK)	UART2 reception (the falling edge of the final bit of the clock)
4	Timing for transferring data from the UART2 reception shift register to the reception buffer.	The rising edge of the final bit of the reception clock	The falling edge of the final bit of the reception clock
5	Timing for generating a UART2 reception/ACK interrupt request	The rising edge of the final bit of the reception clock	The falling edge of the final bit of the reception clock

Table 1.16.11. Timing characteristics of detecting the start condition and the stop condition (Note 1)

3 to 6 cycles < duration for setting-up (Note 2)
3 to 6 cycles < duration for holding (Note 2)

Note 1: When the start/stop condition count bit is "1".
Note 2: "cycles" is in terms of the input oscillation frequency f(XIN) of the main clock.



UART2 Special Mode Register 2

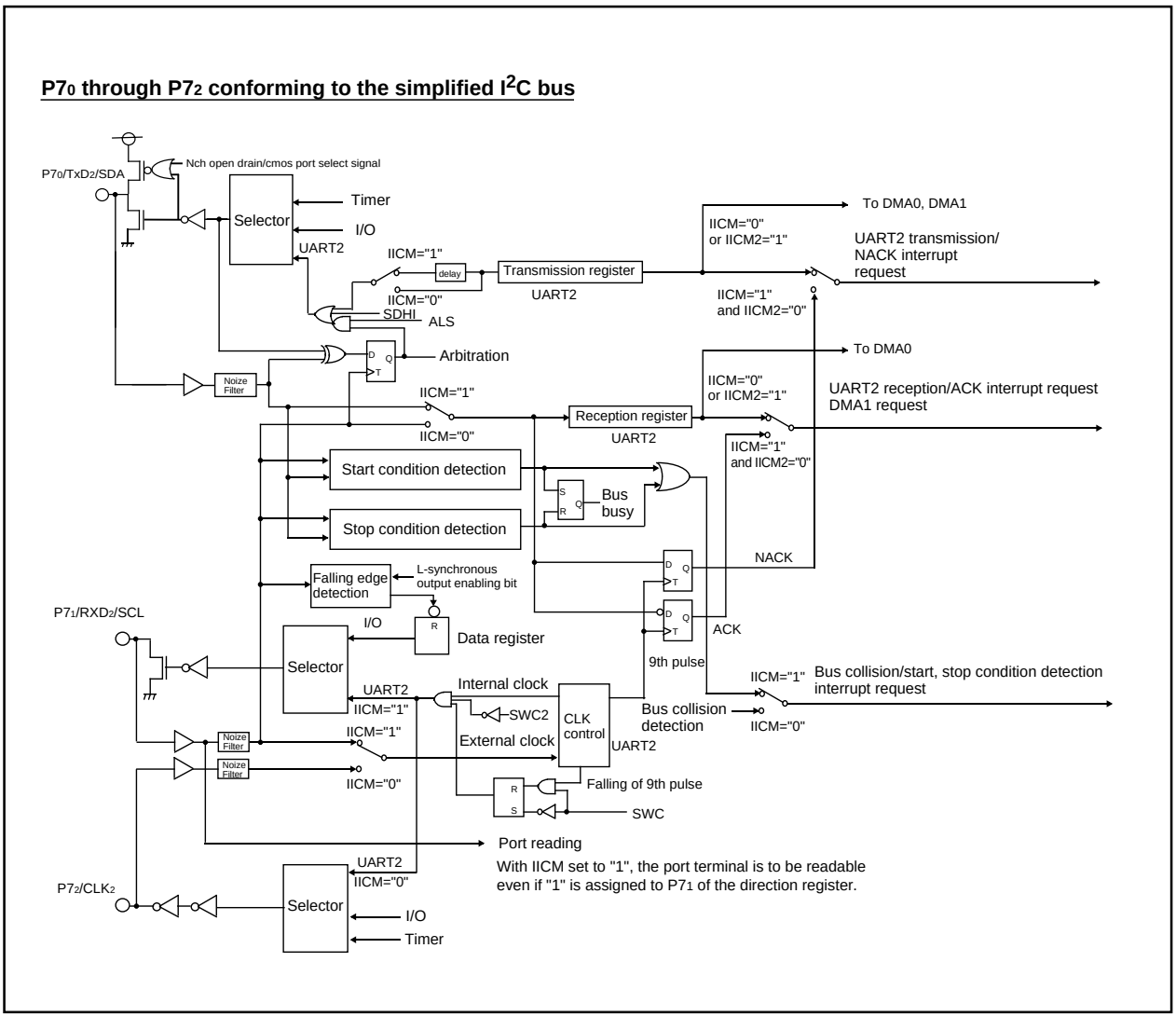


Figure 1.16.30. Functional block diagram for I²C mode

Functions available in I²C mode are shown in Figure 1.16.30 — a functional block diagram.

Bit 3 of the UART2 special mode register 2 (address 01F6₁₆) is used as the SDA output stop bit. Setting this bit to "1" causes an arbitration loss to occur, and the SDA pin turns to high-impedance state the instant when the arbitration lost detecting flag is set to "1".

Bit 1 of the UART2 special mode register 2 (address 01F6₁₆) is used as the clock synchronization bit. With this bit set to "1" at the time when the internal SCL is set to "H", the internal SCL turns to "L" if the falling edge is found in the SCL pin; and the baud rate generator reloads the set value, and start counting within the "L" interval. When the internal SCL changes from "L" to "H" with the SCL pin set to "L", stops counting the baud rate generator, and starts counting it again when the SCL pin turns to "H". Due to this function, the UART2 transmission-reception clock becomes the logical product of the signal flowing through the internal SCL and that flowing through the SCL pin. This function operates over the period from the moment earlier by a half cycle than falling edge of the UART2 first clock to the rising edge of the ninth bit. To use this function, choose the internal clock for the transfer clock.

Bit 2 of the UART2 special mode register 2 (address 01F6₁₆) is used as the SCL wait output bit. Setting this bit to "1" causes the SCL pin to be fixed to "L" at the falling edge of the ninth bit of the clock. Setting this bit to "0" frees the output fixed to "L".

UART2 Special Mode Register 2

Bit 4 of the UART2 special mode register 2 (address 01F6$_{16}$) is used as the UART2 initialization bit. Setting this bit to "1", and when the start condition is detected, the microcomputer operates as follows:

(1) The transmission shift register is initialized, and the content of the transmission register is transferred to the transmission shift register. This starts transmission by dealing with the clock entered next as the first bit. The UART2 output value, however, does not change until the first bit data is output after the entrance of the clock, and remains unchanged from the value at the moment when the microcomputer detected the start condition.

(2) The reception shift register is initialized, and the microcomputer starts reception by dealing with the clock entered next as the first bit.

(3) The SCL wait output bit turns to "1". This turns the SCL pin to "L" at the falling edge of the ninth bit of the clock.

Starting to transmit/receive signals to/from UART2 using this function does not change the value of the transmission buffer empty flag. To use this function, choose the external clock for the tansfer clock.

Bit 5 of the UART2 special mode register 2 (address 01F6$_{16}$) is used as the SCL pin wait output bit 2. Setting this bit to "1" with the serial I/O specified allows the user to forcibly output an "L" from the SCL pin even if UART2 is in operation. Setting this bit to "0" frees the "L" output from the SCL pin, and the UART2 clock is input/output.

Bit 6 of the UART special mode register 2 (address 01F6$_{16}$) is used as the SDA output enable bit. Setting this bit to "1" forces the SDA pin to turn to the high-impedance state. Refrain from changing the value of this bit at the rising edge of the UART2 transfer clock. There can be instances in which arbitration lost detecting flag is turned on.

S I/O3

S I/O3 is exclusive clock-synchronous serial I/O.

Figure 1.16.31 shows the S I/O3 block diagram, and Figure 1.16.32 shows the S I/O3 control register.

Table 1.16.12 shows the specifications of S I/O3.

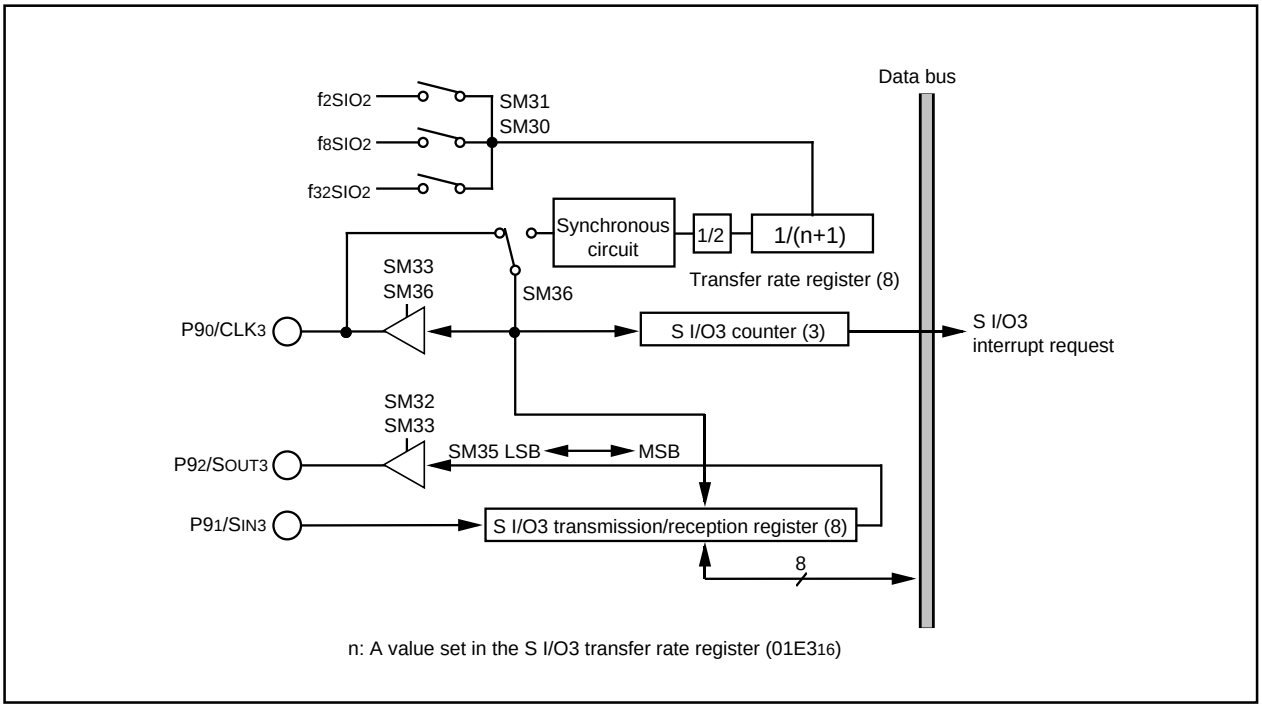


Figure 1.16.31. S I/O3 block diagram

S I/O3 control register (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
			X				

Symbol	Address	When reset
S3C	$01E2_{16}$	40_{16}

Bit symbol	Bit name	Description	R	W
SM30	Internal synchronous clock select bit	b1 b0 0 0 : Selecting f_{2SIO2} 0 1 : Selecting f_{8SIO2} 1 0 : Selecting f_{32SIO2} 1 1 : Not to be used	○	○
SM31			○	○
SM32	S_{OUT3} output disable bit	0 : S_{OUT3} output 1 : S_{OUT3} output disable (high impedance)	○	○
SM33	S I/O3 port select bit (Note 2)	0 : Input-output port 1 : S_{OUT3} output, CLK function	○	○
Nothing is assigned. This bit can neither be set nor read. When read, the value of this bit is "0".			—	—
SM35	Transfer direction select bit	0 : LSB first 1 : MSB first	○	○
SM36	Synchronous clock select bit (Note 2)	0 : External clock 1 : Internal clock	○	○
SM37	S_{OUT3} initial value set bit	Effective when SM33 = 0 0 : L output 1 : H output	○	○

Note 1: Set "1" in bit 2 of the protection register ($000A_{16}$) in advance to write to the S I/O3 control register.
Note 2: When set "0" to SM33 bit and select input - output port, set "1" to SM36 bit and select internal clock.

Figure 1.16.32. S I/O3 control register

S I/O3 bit rate generator (Note 1, 2)

b7	b0

Symbol	Address	When reset
S3BRG	$01E3_{16}$	Indeterminate

Indeterminate	Values that can be set	R	W
Assuming that set value = n, BRG3 divides the count source by n + 1	00_{16} to FF_{16}		○

Note 1: Write a value to this register while transmit/receive halts.
Note 2: Use MOV instruction to write to this register.

S I/O3 transmit/receive register (Note)

b7	b0

Symbol	Address	When reset
S3TRR	$01E0_{16}$	Indeterminate

Indeterminate	R	W
Transmission/reception starts by writing data to this register. After transmission/reception finishes, reception data is input.	○	○

Note: Write a value to this register while transmit/receive halts.

Figure 1.16.33. S I/O3 related register

Table 1.16.12. Specifications of S I/O3

Item	Specifications
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• With the internal clock selected (bit 6 at address $01E2_{16}$ = "1"): $fj_{SIO2}/2(n+1)$ (j = 2, 8, 32) (Note 1) • With the external clock selected (bit 6 at address $01E2_{16}$ = "0"): Input from the CLK3 terminal (Note 2)
Conditions for transmission/ reception start	• To start transmit/reception, the following requirements must be met: - Select the synchronous clock (use bit 6 at address $01E2_{16}$). Select a frequency dividing ratio if the internal clock has been selected (use bits 0 and 1 at address $01E2_{16}$). - $S_{OUT}3$ initial value set bit (use bit 7 at address $01E2_{16}$) = "1". - S I/O3 port select bit (bit 3 at address $01E2_{16}$) = "1". - Select the transfer direction (use bit 5 at address $01E2_{16}$) - Write transfer data to the S I/O3 transmit/receive register (address $01E0_{16}$) • To use S I/O3 interrupts, the following requirements must be met: - Clear the S I/O3 interrupt request bit (bit 3 at address 0049_{16}) before writing transfer data to the S I/O3 transmit/receive register.
Interrupt request generation timing	• Rising edge of the last transfer clock. (Note 3)
Select function	• LSB first or MSB first selection Whether transmission/reception begins with bit 0 (LSB) or bit 7 (MSB) can be selected.
Precaution	• Unlike UART0 to 2, S I/O3 is not divided for transfer register and buffer. Therefore, do not write the next transfer data to the S I/O3 transmit/receive register (address $01E0_{16}$) during a transfer. • When the internal clock is selected for the transfer clock, S_{OUT3} holds the last data for a 1/2 transfer clock period after it finished transferring and then goes to a high-impedance state. However, if the transfer data is written to the S I/O3 transmit/receive register (address $01E0_{16}$) during this time, S_{OUT3} is placed in the high-impedance state immediately upon writing and the data hold time is thereby reduced.

Note 1: "n" is a value from 00_{16} to FF_{16} set in the S I/O3 transfer rate generator.

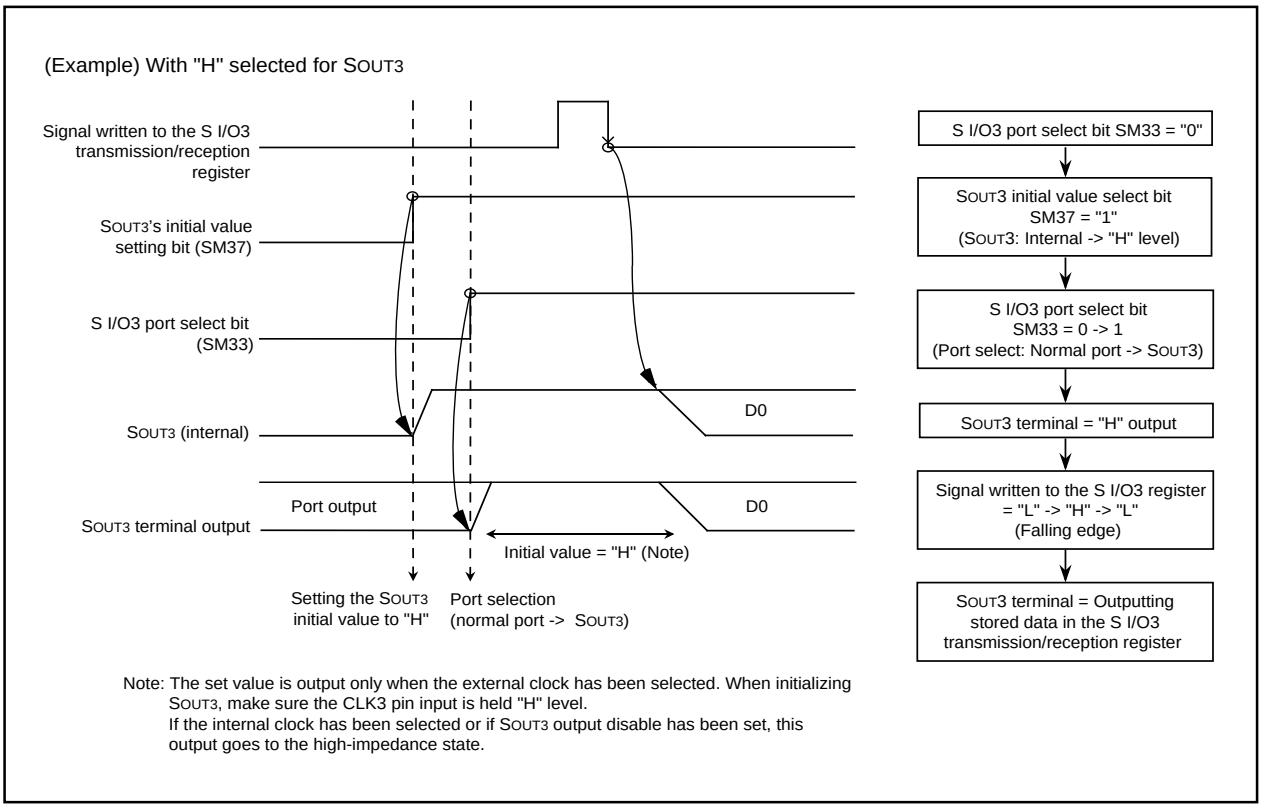
Note 2: With the external clock selected:

- Before data can be written to the S I/O3 transmit/receive register (address $01E0_{16}$), the CLK3 pin input must be in the high state. Also, before rewriting bit 7 of the S I/O3 control register (address $01E2_{16}$), make sure the CLK3 pin input is held high.
- The S I/O3 circuit keeps on with the shift operation as long as the synchronous clock is entered in it, so stop the synchronous clock at the instant when it counts to eight. The internal clock, if selected, automatically stops.

Note 3: If the internal clock is used for the synchronous clock, the transfer clock signal stops at the "H" state.

■Functions for setting an SOUT3 initial value

When using an external clock for the transfer clock, the SOUT3 pin output level during a non-transfer time can be set to high or low state. Figure 1.16.34 shows the timing chart for setting an SOUT3 initial value and how to set it.

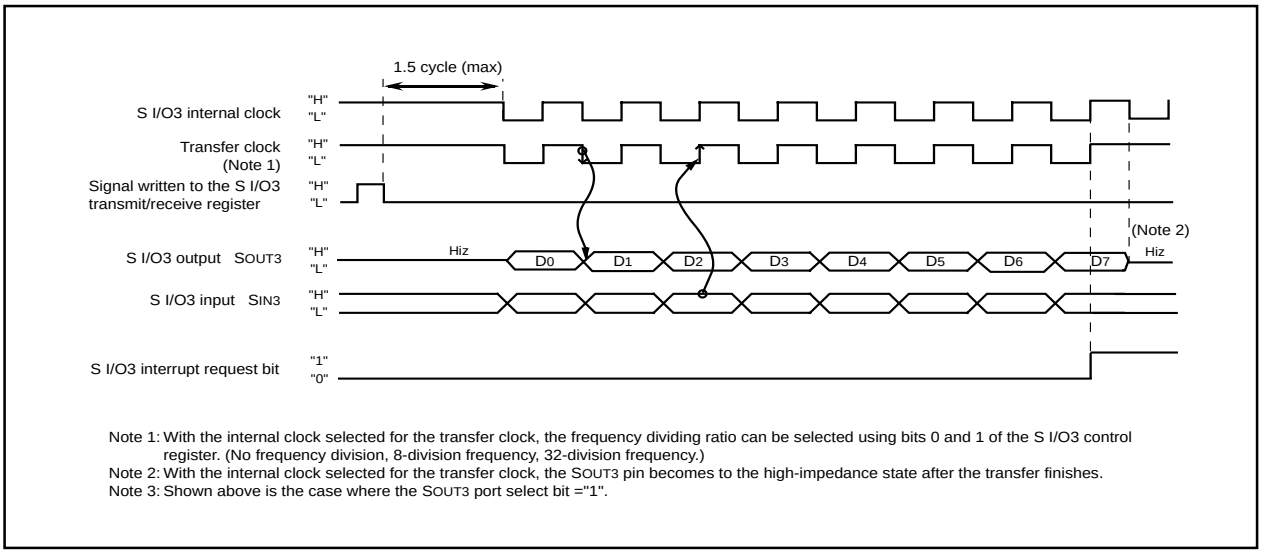


Note: The set value is output only when the external clock has been selected. When initializing SOUT3, make sure the CLK3 pin input is held "H" level.
If the internal clock has been selected or if SOUT3 output disable has been set, this output goes to the high-impedance state.

Figure 1.16.34. Timing chart for setting SOUT3's initial value and how to set it

■S I/O3 operation timing

Figure 1.16.35 shows the S I/O3 operation timing



Note 1: With the internal clock selected for the transfer clock, the frequency dividing ratio can be selected using bits 0 and 1 of the S I/O3 control register. (No frequency division, 8-division frequency, 32-division frequency.)
Note 2: With the internal clock selected for the transfer clock, the SOUT3 pin becomes to the high-impedance state after the transfer finishes.
Note 3: Shown above is the case where the SOUT3 port select bit ="1".

Figure 1.16.35. S I/O3 operation timing chart

A-D Converter

The A-D converter consists of one 10-bit successive approximation A-D converter circuit with a capacitive coupling amplifier. Pins $P0_0$ to $P0_7$, $P2_0$ to $P2_7$, $P10_0$ to $P10_7$, $P9_5$, and $P9_6$ function as the analog signal input pins. The direction registers of these pins for A-D conversion must therefore be set to input. The V_{REF} 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 the V_{REF} connect bit (bit 5 at address $03D7_{16}$) 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 1.17.1 shows the performance of the A-D converter. Figure 1.17.1 shows the block diagram of the A-D converter, and Figures 1.17.2 and 1.17.3 show the A-D converter-related registers.

Table 1.17.1. 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 condition $ø_{AD}$ (Note 2)	V_{CC} = 5V, f_{2AD} divided by 1, 2, or 4. $f_{2AD}=f(X_{IN})/2$ (PCLK0 ="0") $f_{2AD}=f(X_{IN})$ (PCLK0 ="1")
Resolution	8-bit or 10-bit (selectable)
Absolute precision	V_{CC} = 5 • Without sample and hold function ±3LSB • With sample and hold function (8-bit resolution) ±2LSB • With sample and hold function (10-bit resolution) AN_0 to AN_7, AN_{00} to AN_{07}, and AN_{20} to AN_{27} input : ±3LSB ANEX0 and ANEX1 input (including mode in which external operation amp is connected) : ±7LSB
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog input pins	24 pins (AN_0 to AN_7, AN_{00} to AN_{07} and AN_{20} to AN_{27}) + 2 pins (ANEX0 and ANEX1)
A-D conversion start condition	• Software trigger A-D conversion starts when the A-D conversion start flag changes to "1" • External trigger (can be retriggered) A-D conversion starts when the A-D conversion start flag is "1" and the $\overline{ADTRG}$/$P9_7$ input changes from "H" to "L"
Conversion speed per pin	• Without sample and hold function 8-bit resolution: 49 $ø_{AD}$ cycles, 10-bit resolution: 59 $ø_{AD}$ cycles • With sample and hold function 8-bit resolution: 28 $ø_{AD}$ cycles, 10-bit resolution: 33 $ø_{AD}$ cycles

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

Note 2: Without sample and hold function, set the $ø_{AD}$ frequency to 250kHz min.
With the sample and hold function, set the $ø_{AD}$ frequency to 1MHz min.
In either case, the $ø_{AD}$ frequency may not exceed 10 MHz.

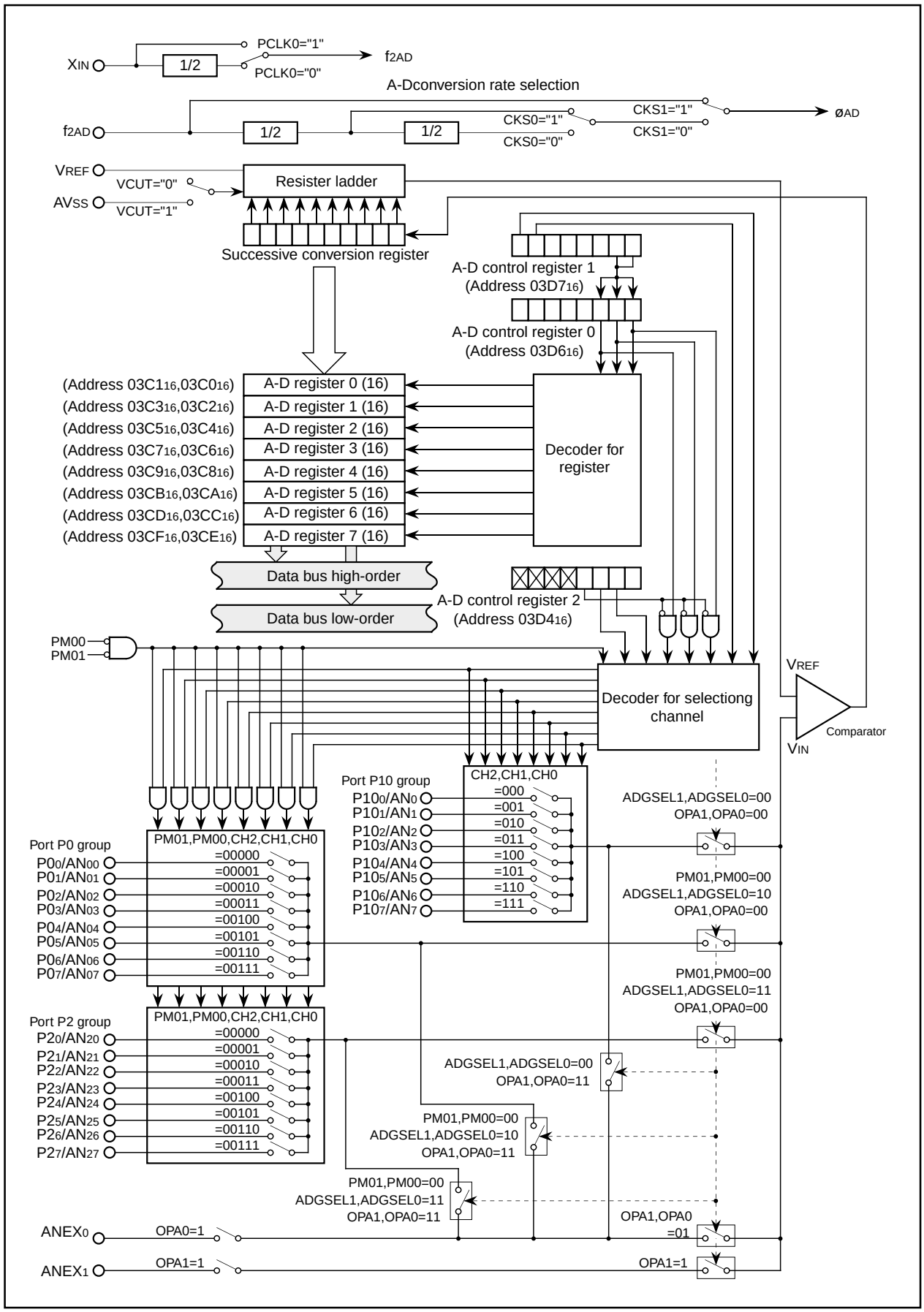


Figure 1.17.1. Block diagram of A-D converter

A-D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0

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

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit (Note 2, 3)	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			○	○
TRG	Trigger select bit	0 : Software trigger 1 : ADTRG trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : f2AD/4 is selected 1 : f2AD/2 is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN$_{00}$ to AN$_{07}$ and AN$_{20}$ to AN$_{27}$ can be used in the same way as AN$_0$ to AN$_7$.
Note 3: When changing A-D operation mode, set analog input pin again.

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0

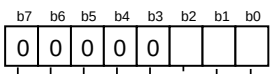
Symbol	Address	When reset
ADCON1	03D7$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit (Note 2)	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 (Note 3)	0 : f2AD/2 or f2AD/4 is selected 1 : f2AD is selected	○	○
VCUT	VREF connect bit	0 : VREF not connected 1 : VREF connected	○	○
OPA0	External op-amp connection mode bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A-D converted 1 0 : ANEX1 input is A-D converted 1 1 : External op-amp connection mode	○	○
OPA1			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN$_{00}$ to AN$_{07}$ and AN$_{20}$ to AN$_{27}$ can be used in the same way as AN$_0$ to AN$_7$.
Note 3: Divide the frequency if f(XIN) exceeds 10MHz, and make øAD frequency equal to or less than 10MHz.

Figure 1.17.2. A-D converter-related registers (1)

A-D control register 2 (Note)



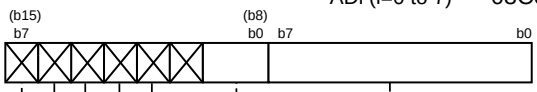
Symbol: ADCON2 | Address: $03D4_{16}$ | When reset: $XXXX0000_2$

Bit symbol	Bit name	Function	R	W
SMP	A-D conversion method select bit	0 : Without sample and hold 1 : With sample and hold	○	○
ADGSEL0	A-D group select bit	b2 b1 0 0 : Port P10 group select 0 1 : inhibited 1 0 : Port P0 group select 1 1 : Port P2 group select	○	○
ADGSEL1				
Reserved bits		Always set to "0"	○	○

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

A-D register i

Symbol: ADi (i=0 to 7) | Address: $03C0_{16}$ to $03CF_{16}$ | When reset: 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. These bits can neither be set nor reset. When read, their content is "0".	—	—

Figure 1.17.3. 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 1.17.2 shows the specifications of one-shot mode. Figure 1.17.4 shows the A-D control register in one-shot mode.

Table 1.17.2. 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", 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	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 1)

b7	b6	b5	b4	b3	b2	b1	b0
			0	0			

Symbol ADCON0 — Address 03D6₁₆ — When reset 00000XXX₂

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit (Note 2, 3)	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 (Note 3)	b4 b3 0 0 : One-shot mode	○	○
MD1			○	○
TRG	Trigger select bit	0 : Software trigger 1 : $\overline{AD_{TRG}}$ trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : $f_{2AD}/4$ is selected 1 : $f_{2AD}/2$ is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN_{00} to AN_{07} and AN_{20} to AN_{27} can be used in the same way as AN_0 to AN_7.
Note 3: When changing A-D operation mode, set analog input pin again.

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		1			0		

Symbol ADCON1 — Address 03D7₁₆ — When reset 00₁₆

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 : Set to "0" when this mode is selected	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit1 (Note 2)	0 : $f_{2AD}/2$ or $f_{2AD}/4$ is selected 1 : f_{2AD} is selected	○	○
VCUT	V_{REF} connect bit	1 : V_{REF} connected	○	○
OPA0	External op-amp connection mode bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A-D converted 1 0 : ANEX1 input is A-D converted 1 1 : External op-amp connection mode	○	○
OPA1			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: Divide the frequency if f(X_{IN}) exceeds 10MHz, and make $ø_{AD}$ frequency equal to or less than 10MHz.

Figure 1.17.4. 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 1.17.3 shows the specifications of repeat mode. Figure 1.17.5 shows the A-D control register in repeat mode.

Table 1.17.3. Repeat mode specifications

Item	Specification
Function	The pin selected by the analog input pin select bit is used for repeated 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	One of AN_0 to AN_7, as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)

A-D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
			0	1			

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

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit (Note 2, 3)	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 (Note 3)	b4 b3 0 1 : Repeat mode	○	○
MD1			○	○
TRG	Trigger select bit	0 : Software trigger 1 : AD_{TRG} trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : $f_{2AD}/4$ is selected 1 : $f_{2AD}/2$ is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN_{00} to AN_{07} and AN_{20} to AN_{27} can be used in the same way as AN_0 to AN_7.
Note 3: When changing A-D operation mode, set analog input pin again.

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		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 : Set to "0" when this mode is selected	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit1 (Note 2)	0 : $f_{2AD}/2$ or $f_{2AD}/4$ is selected 1 : f_{2AD} is selected	○	○
VCUT	V_{REF} connect bit	1 : V_{REF} connected	○	○
OPA0	External op-amp connection mode bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A-D converted 1 0 : ANEX1 input is A-D converted 1 1 : External op-amp connection mode	○	○
OPA1			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: Divide the frequency if $f(X_{IN})$ exceeds 10MHz, and make $ø_{AD}$ frequency equal to or less than 10MHz.

Figure 1.17.5. 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 1.17.4 shows the specifications of single sweep mode. Figure 1.17.6 shows the A-D control register in single sweep mode.

Table 1.17.4. 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	AN0 and AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), or AN0 to AN7 (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
			1	0			

Symbol ADCON0 — Address 03D6₁₆ — When reset 00000XXX₂

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in single sweep mode	O	O
CH1			O	O
CH2			O	O
MD0	A-D operation mode select bit 0	b4 b3 1 0 : Single sweep mode	O	O
MD1			O	O
TRG	Trigger select bit	0 : Software trigger 1 : ADTRG trigger	O	O
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	O	O
CKS0	Frequency select bit 0	0 : f2AD/4 is selected 1 : f2AD/2 is selected	O	O

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

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		1			0		

Symbol ADCON1 — Address 03D7₁₆ — When reset 00₁₆

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit (Note 2)	When single sweep and repeat sweep mode 0 are selected: b1 b0 0 0 : AN0, AN1 (2 pins) 0 1 : AN0 to AN3 (4 pins) 1 0 : AN0 to AN5 (6 pins) 1 1 : AN0 to AN7 (8 pins)	O	O
SCAN1			O	O
MD2	A-D operation mode select bit 1	0 : Set to "0" when this mode is selected	O	O
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	O	O
CKS1	Frequency select bit1 (Note 3)	0 : fAD2/2 or fAD2/4 is selected 1 : fAD2 is selected	O	O
VCUT	VREF connect bit	1 : VREF connected	O	O
OPA0	External op-amp connection mode bit (Note 4)	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A-D converted 1 0 : ANEX1 input is A-D converted 1 1 : External op-amp connection mode	O	O
OPA1			O	O

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN00 to AN07 and AN20 to AN27 can be used in the same way as AN0 to AN7.
Note 3: Divide the frequency if f(XIN) exceeds 10MHz, and make øAD frequency equal to or less than 10MHz.
Note 4: Neither "01" nor "10" can be selected with the external op-amp connection mode bit.

Figure 1.17.6. 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 1.17.5 shows the specifications of repeat sweep mode 0. Figure 1.17.7 shows the A-D control register in repeat sweep mode 0.

Table 1.17.5. 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), 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 (at any time)

A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
			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	O	O
CH1			O	O
CH2			O	O
MD0	A-D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 0	O	O
MD1			O	O
TRG	Trigger select bit	0 : Software trigger 1 : AD_{TRG} trigger	O	O
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	O	O
CKS0	Frequency select bit 0	0 : $f2_{AD}$/4 is selected 1 : $f2_{AD}$/2 is selected	O	O

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

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		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 (Note 2)	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)	O	O
SCAN1			O	O
MD2	A-D operation mode select bit 1	0 : Set to "0" when this mode is selected	O	O
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	O	O
CKS1	Frequency select bit1 (Note 3)	0 : $f2_{AD}$/2 or $f2_{AD}$/4 is selected 1 : $f2_{AD}$ is selected	O	O
VCUT	V_{REF} connect bit	1 : V_{REF} connected	O	O
OPA0	External op-amp connection mode bit (Note 4)	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A-D converted 1 0 : ANEX1 input is A-D converted 1 1 : External op-amp connection mode	O	O
OPA1			O	O

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN_{00} to AN_{07} and AN_{20} to AN_{27} can be used in the same way as AN_0 to AN_7.
Note 3: Divide the frequency if $f(X_{IN})$ exceeds 10MHz, and make $ø_{AD}$ frequency equal to or less than 10MHz.
Note 4: Neither "01" nor "10" can be selected with the external op-amp connection mode bit.

Figure 1.17.7. 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 1.17.6 shows the specifications of repeat sweep mode 1. Figure 1.17.8 shows the A-D control register in repeat sweep mode 1.

Table 1.17.6. 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; AN0 selected AN0 → AN1 → AN0 → AN2 → AN0 → AN3, 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	With emphasis on these pins; AN0 (1 pin), AN0 and AN1 (2 pins), AN0 to AN2 (3 pins), AN0 to AN3 (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
			1	1			

Symbol ADCON0 — Address 03D6₁₆ — When reset 00000XXX₂

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bits	Invalid in repeat sweep mode 1	O	O
CH1			O	O
CH2			O	O
MD0	A-D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 1	O	O
MD1			O	O
TRG	Trigger select bit	0 : Software trigger 1 : ADTRG trigger	O	O
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	O	O
CKS0	Frequency select bit 0	0 : f2AD/4 is selected 1 : f2AD/2 is selected	O	O

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

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		1			1		

Symbol ADCON1 — Address 03D7₁₆ — When reset 00₁₆

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bits (Note 2)	When repeat sweep mode 1 is selected: b1 b0 0 0 : AN0 (1 pin) 0 1 : AN0, AN1 (2 pins) 1 0 : AN0 to AN2 (3 pins) 1 1 : AN0 to AN3 (4 pins)	O	O
SCAN1			O	O
MD2	A-D operation mode select bit 1	1 : Set to "1" when this mode is selected	O	O
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	O	O
CKS1	Frequency select bit1 (Note 3)	0 : f2AD/2 or f2AD/4 is selected 1 : f2AD is selected	O	O
VCUT	VREF connect bit	1 : VREF connected	O	O
OPA0	External op-amp connection mode bits (Note 4)	b7 b6 0 0 : ANEX0 and ANEX1 are not used 0 1 : ANEX0 input is A-D converted 1 0 : ANEX1 input is A-D converted 1 1 : External op-amp connection mode	O	O
OPA1			O	O

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN00 to AN07 and AN20 to AN27 can be used in the same way as AN0 to AN7.
Note 3: Divide the frequency if f(XIN) exceeds 10MHz, and make øAD frequency equal to or less than 10MHz.
Note 4: Neither "01" nor "10" can be selected with the external op-amp connection mode bit.

Figure 1.17.8. 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, a 28 øAD cycle is achieved with 8-bit resolution and 33 øAD 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.

(b) Extended analog input pins

In one-shot mode and repeat mode, the input via the extended analog input pins ANEX0 and ANEX1 can also be converted from analog to digital.

When bit 6 of the A-D control register 1 (address 03D7$_{16}$) is "1" and bit 7 is "0", input via ANEX0 is converted from analog to digital. The result of conversion is stored in A-D register 0.

When bit 6 of the A-D control register 1 (address 03D7$_{16}$) is "0" and bit 7 is "1", input via ANEX1 is converted from analog to digital. The result of conversion is stored in A-D register 1.

(c) External operation amp connection mode

In this mode, multiple external analog inputs via the extended analog input pins, ANEX0 and ANEX1, can be amplified together by just one operation amp and used as the input for A-D conversion.

When bit 6 of the A-D control register 1 (address 03D7$_{16}$) is "1" and bit 7 is "1", input via AN$_0$ to AN$_7$ is output from ANEX0. The input from ANEX1 is converted from analog to digital and the result stored in the corresponding A-D register. The speed of A-D conversion depends on the response of the external operation amp. Do not connect the ANEX0 and ANEX1 pins directly. Figure 1.17.9 is an example of how to connect the pins in external operation amp mode.

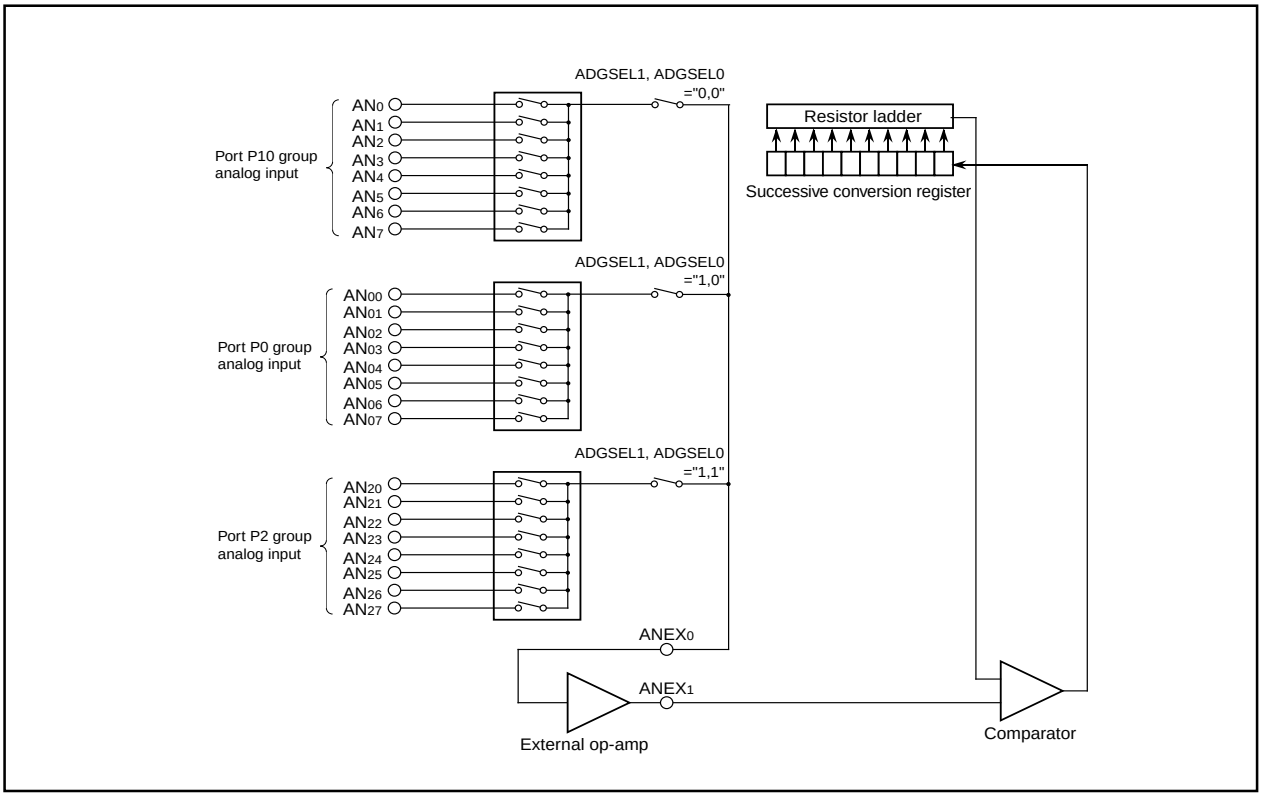


Figure 1.17.9. Example of external op-amp connection mode

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. The D-A output enable bits (bits 0 and 1 at address $03DC_{16}$) 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.

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

V_{REF} : reference voltage

Table 1.18.1 lists the performance of the D-A converter. Figure 1.18.1 shows the block diagram of the D-A converter. Figure 1.18.2 shows the D-A control register.

Table 1.18.1. Performance of D-A converter

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

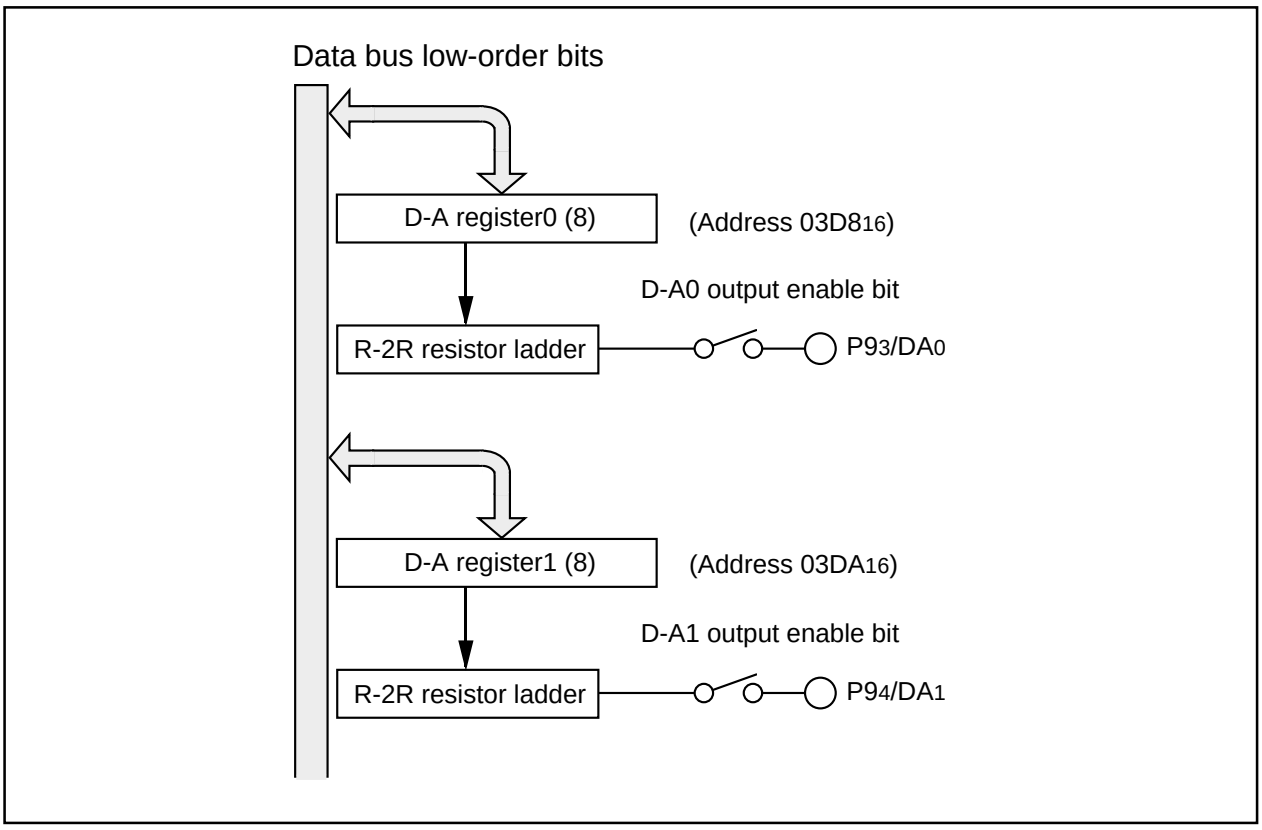


Figure 1.18.1. Block diagram of D-A converter

D-A control register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
DACON	03DC16	0016

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".			–	–

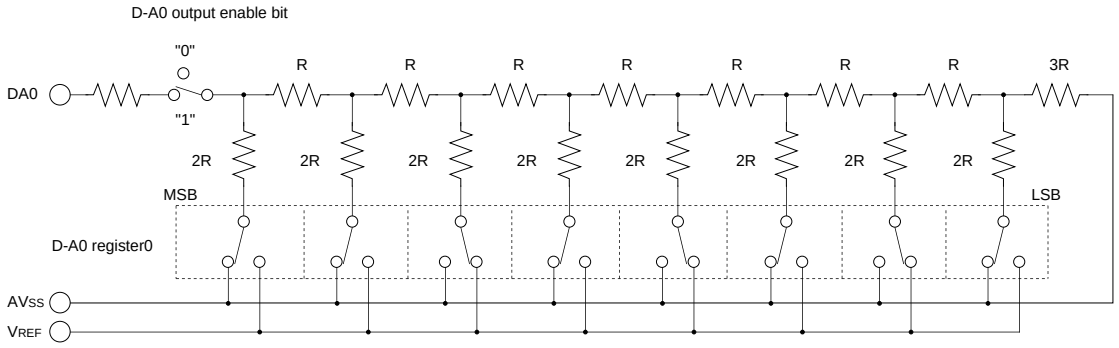
D-A register

b7 b0

Symbol	Address	When reset
DAi (i = 0,1)	03D816, 03DA16	Indeterminate

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

Figure 1.18.2. D-A control register



Note 1: The above diagram shows an instance in which the D-A register is assigned 2A16.
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 "0016" so that no current flows in the resistors Rs and 2Rs.

Figure 1.18.3. 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 1.19.1 shows the block diagram of the CRC circuit. Figure 1.19.2 shows the CRC-related registers. Figure 1.19.3 shows the calculation example using the CRC calculation circuit.

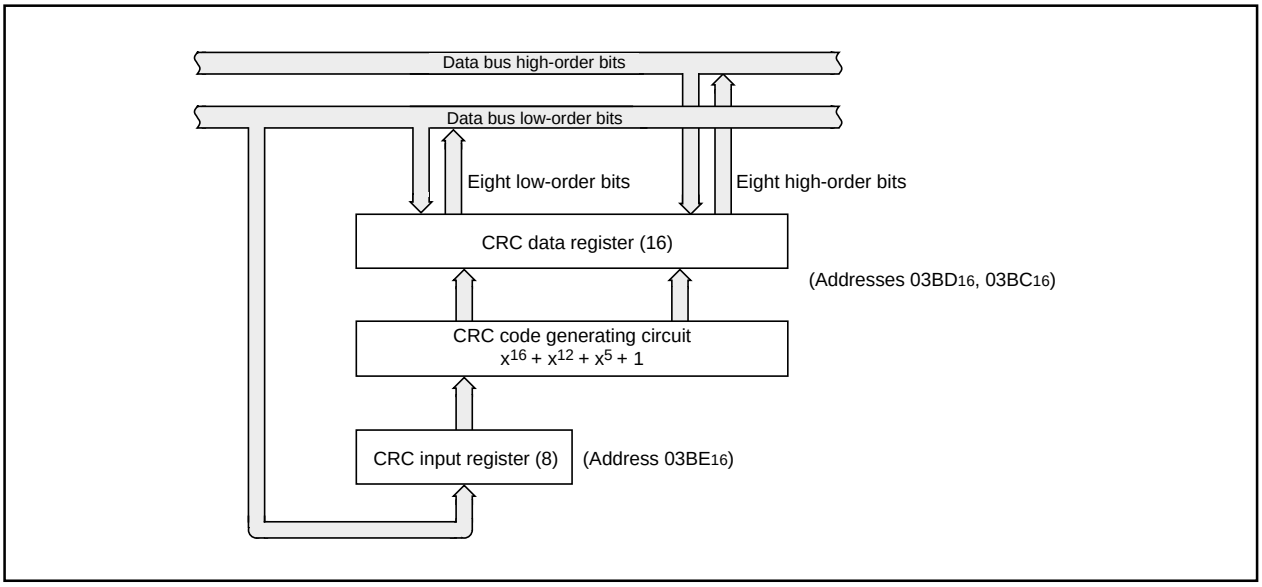


Figure 1.19.1. Block diagram of CRC circuit

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

Symbol	Address	When reset
CRCIN	03BE16	Indeterminate

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

Figure 1.19.2. CRC-related registers

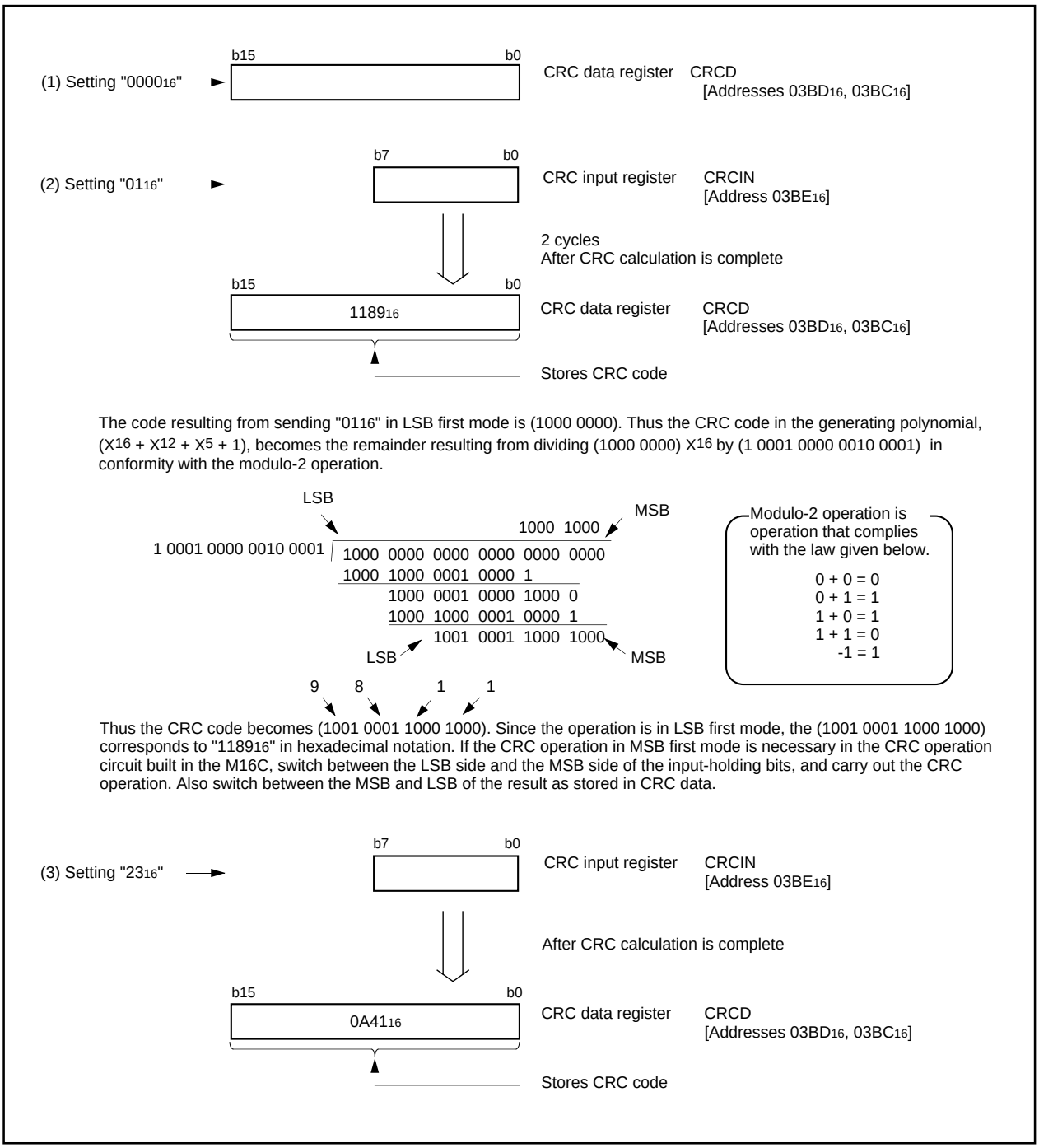
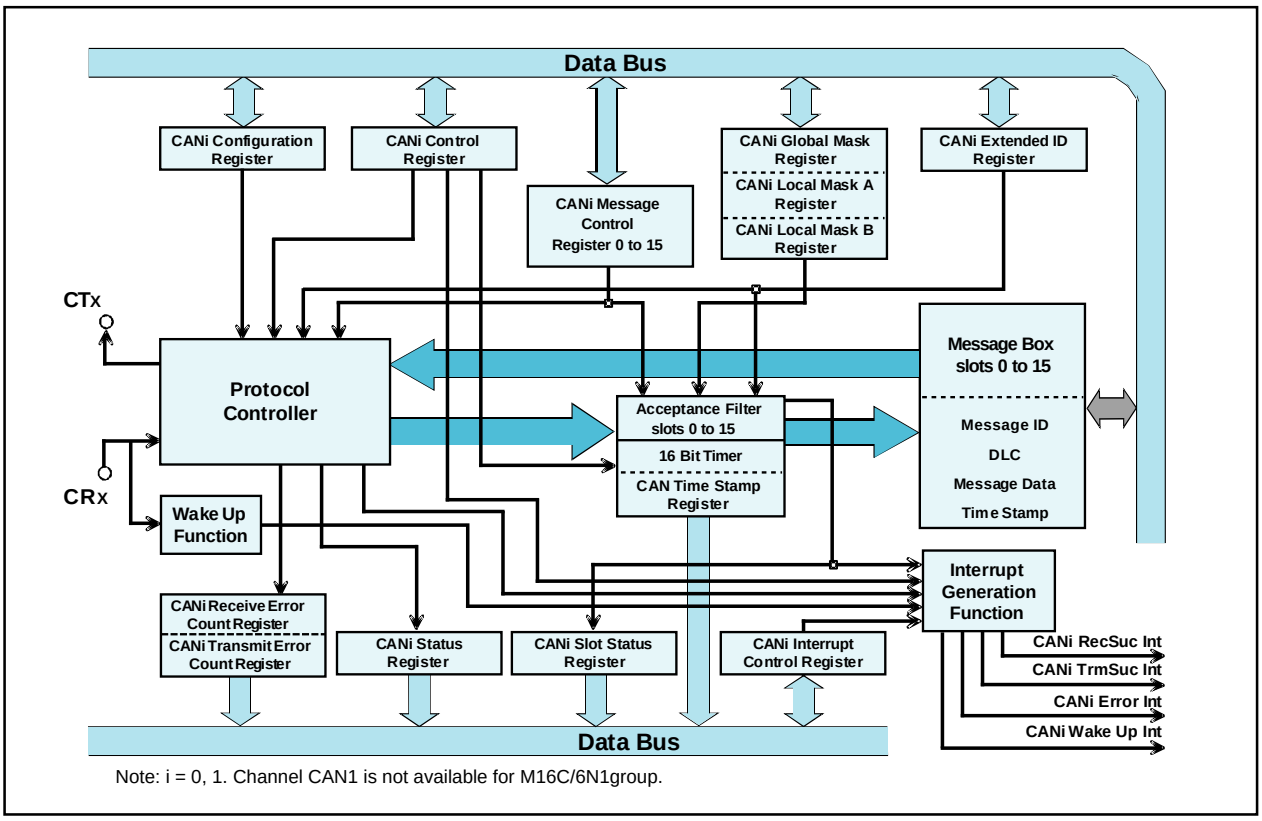


Figure 1.19.3. Calculation example using the CRC calculation circuit

CAN Module

The CAN (Controller Area Network) module for the M16C/6N group of microcomputers is a communication controller implementing the CAN 2.0B protocol as defined in the BOSCH specification. The M16C/6N0 group contains two Full CAN modules and the M16C/6N1 group contains one Full CAN module which can transmit and receive messages in both standard (11 bit) ID and extended (29 bit) ID formats.
Figure 1.20.1 shows a block diagram of the CAN module.
External CAN bus driver and receiver are required.



Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1group.

Figure 1.20.1. Block diagram of the CAN module

CTx/CRx:	CAN I/O pins.
Protocol controller:	This controller handles the bus arbitration and the CAN protocol services, i.e. bit timing, stuffing, error status etc.
Message box:	This memory block consists of 16 slots that can be configured either as transmitter or receiver. Each slot contains an individual ID, data length code, a data field (8 bytes) and a time stamp.
Acceptance filter:	This block performs filtering operation for received messages. For the filtering operation, the CANi global mask register, the CANi local mask A register, or the CANi local mask B register is used.
16 bit timer:	Used for the time stamp function. When the received message is stored in the message memory, the timer value is stored as a time stamp.
Wake up function:	CANi wake up interrupt is generated by a message from the CAN bus.
Interrupt generation function:	The interrupt events are provided by the CAN module. CANi successful reception interrupt, CANi successful transmission interrupt, CAN0/1 error interrupt, and CAN0/1 wake up interrupt.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

Registers Related to the CAN Modules

The CANi modules have the following registers, respectively.

(1) CAN message box

A CAN module is equipped with 16 slots (16 bytes or 8 words each). Slots 14 and 15 can be used as Basic CAN.

- Priority of the slots: The smaller the number of the slot, the higher the priority, in both transmission and reception.
- A program can define whether a slot is defined as transmitter or receiver.

(2) Acceptance mask registers

A CAN module is equipped with 3 masks for the acceptance filter.

- CANi global mask register (6 bytes)
  Configuration of the masking condition for acceptance filtering processing to slots 0 to 13
- CANi local mask A register (6 bytes)
  Configuration of the masking condition for acceptance filtering processing to slot 14
- CANi local mask B register (6 bytes)
  Configuration of the masking condition for acceptance filtering processing to slot 15

(3) CAN SFR Registers

- CANi message control register (8 bits X 16)
  Control of transmission and reception of a corresponding slot
- CANi control register (16 bits)
  Control of the CAN protocol
- CANi status register (16 bits)
  Indication of the protocol status
- CANi slot status register (16 bits)
  Indication of the status of contents of each slot
- CANi interrupt control register (16 bits)
  Selection of "interrupt enabled or disabled" for each slot
- CANi extended ID register (16 bits)
  Selection of ID format (standard or extended) for each slot
- CANi configuration register (16 bits)
  Configuration of the bus timing
- CANi receive error count register (8 bits)
  Indication of the error status of the CAN module in reception: the counter value is incremented or decremented according to the error occurence.
- CANi transmit error count register (8 bits)
  Indication of the error status of the CAN module in transmission: the counter value is incremented or decremented according to the error occurence.
- CANi time stamp register (16 bits)
  Indication of the value of the time stamp counter
- CANi acceptance filter support register
  Decoding the received ID for use by the acceptance filter support unit

Explanation of each register is given below.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

CANi Message Box

Table 1.20.1 shows the memory mapping of the CANi message box.

It is possible to access to the message box in byte or word.

Mapping of the message contents differs from byte access to word access. Byte access or word access can be selected by the MsgOrder bit of the CANi control register.

Table 1.20.1. Memory mapping of the CANi message box (n = 0 to 15: the number of the slot)

Address		Message content (Memory mapping)	
CAN0	CAN1	Byte access (8 bits)	Word access (16 bits)
0060_{16} + n ▪16 + 0	0260_{16} + n ▪16 + 0	SID_{10} to SID_6	SID_5 to SID_0
0060_{16} + n ▪16 + 1	0260_{16} + n ▪16 + 1	SID_5 to SID_0	SID_{10} to SID_6
0060_{16} + n ▪16 + 2	0260_{16} + n ▪16 + 2	EID_{17} to EID_{14}	EID_{13} to EID_6
0060_{16} + n ▪16 + 3	0260_{16} + n ▪16 + 3	EID_{13} to EID_6	EID_{17} to EID_{14}
0060_{16} + n ▪16 + 4	0260_{16} + n ▪16 + 4	EID_5 to EID_0	Data Length Code (DLC)
0060_{16} + n ▪16 + 5	0260_{16} + n ▪16 + 5	Data Length Code (DLC)	EID_5 to EID_0
0060_{16} + n ▪16 + 6	0260_{16} + n ▪16 + 6	Data byte 0	Data byte 1
0060_{16} + n ▪16 + 7 ⋮ 0060_{16} + n ▪16 + 13	0260_{16} + n ▪16 + 7 ⋮ 0260_{16} + n ▪16 + 13	Data byte 1 ⋮ Data byte 7	Data byte 0 ⋮ Data byte 6
0060_{16} + n ▪16 + 14	0260_{16} + n ▪16 + 14	Time stamp high order byte	Time stamp low order byte
0060_{16} + n ▪16 + 15	0260_{16} + n ▪16 + 15	Time stamp low order byte	Time stamp high order byte

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

Figures 1.20.2 and 1.20.3 show the bit mapping in each slot in byte access and word access. The content of each slot remains unchanged unless transmission or reception of a new message is performed.

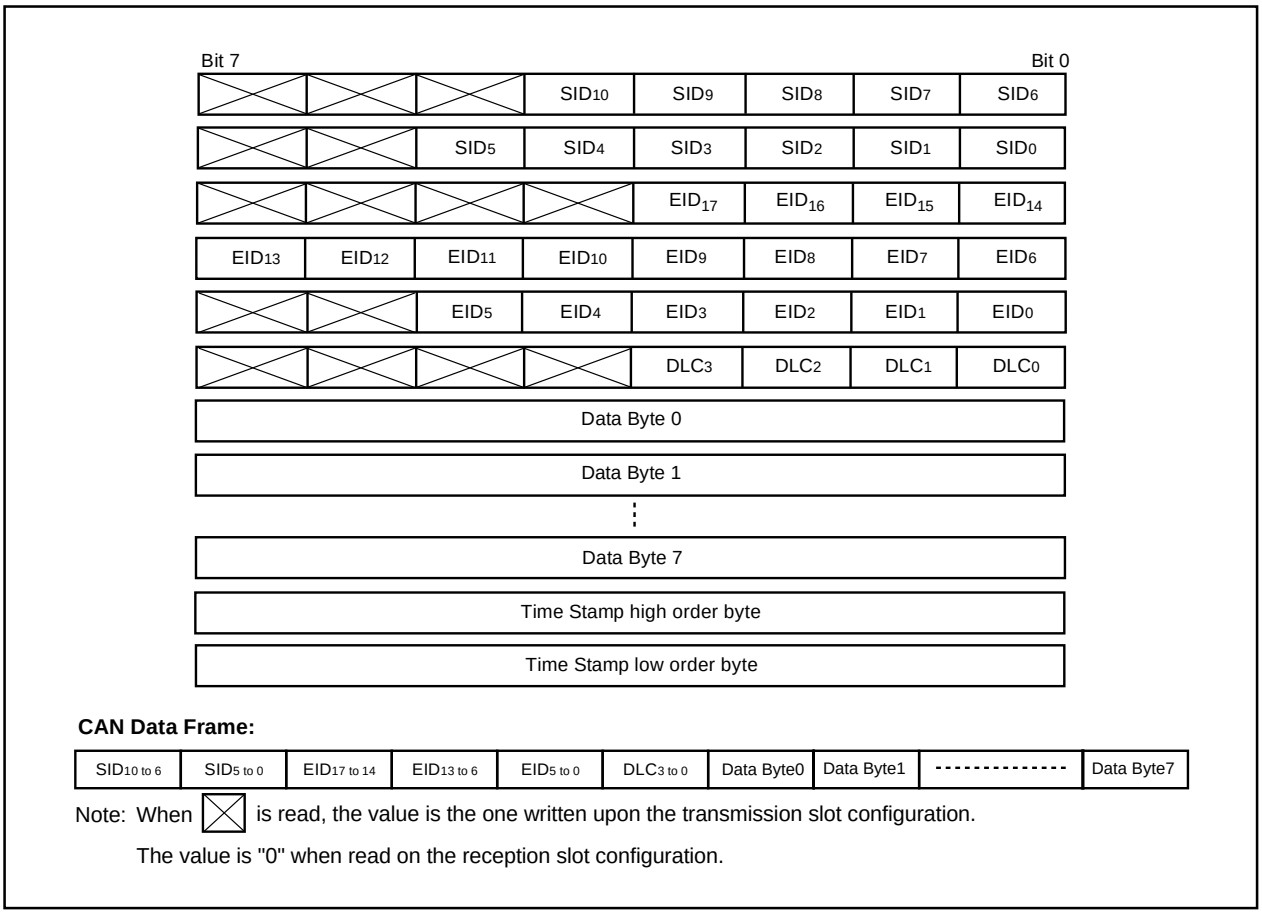


Figure 1.20.2. Bit mapping in byte access

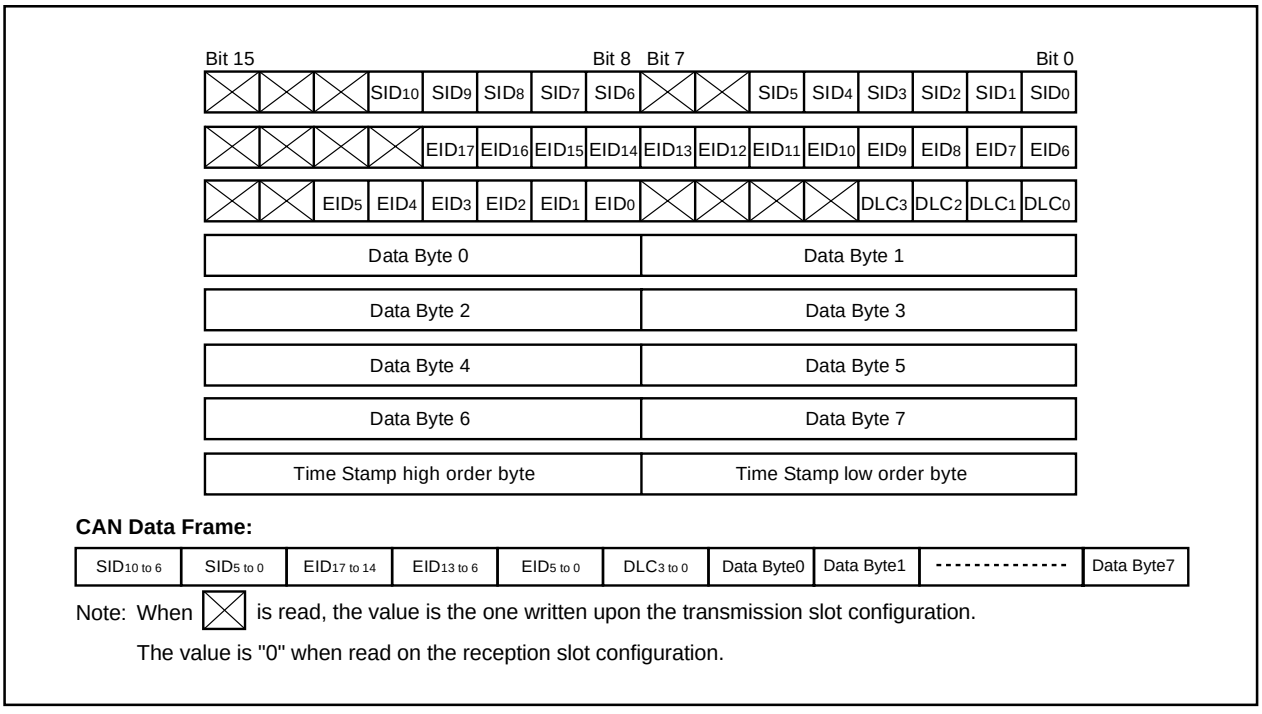


Figure 1.20.3. Bit mapping in word access

Acceptance Mask Registers

Figures 1.20.4 and 1.20.5 show the CANi global mask register, the CANi local mask A register, and the CANi local mask B register, in which bit mapping in byte access and word access are shown.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

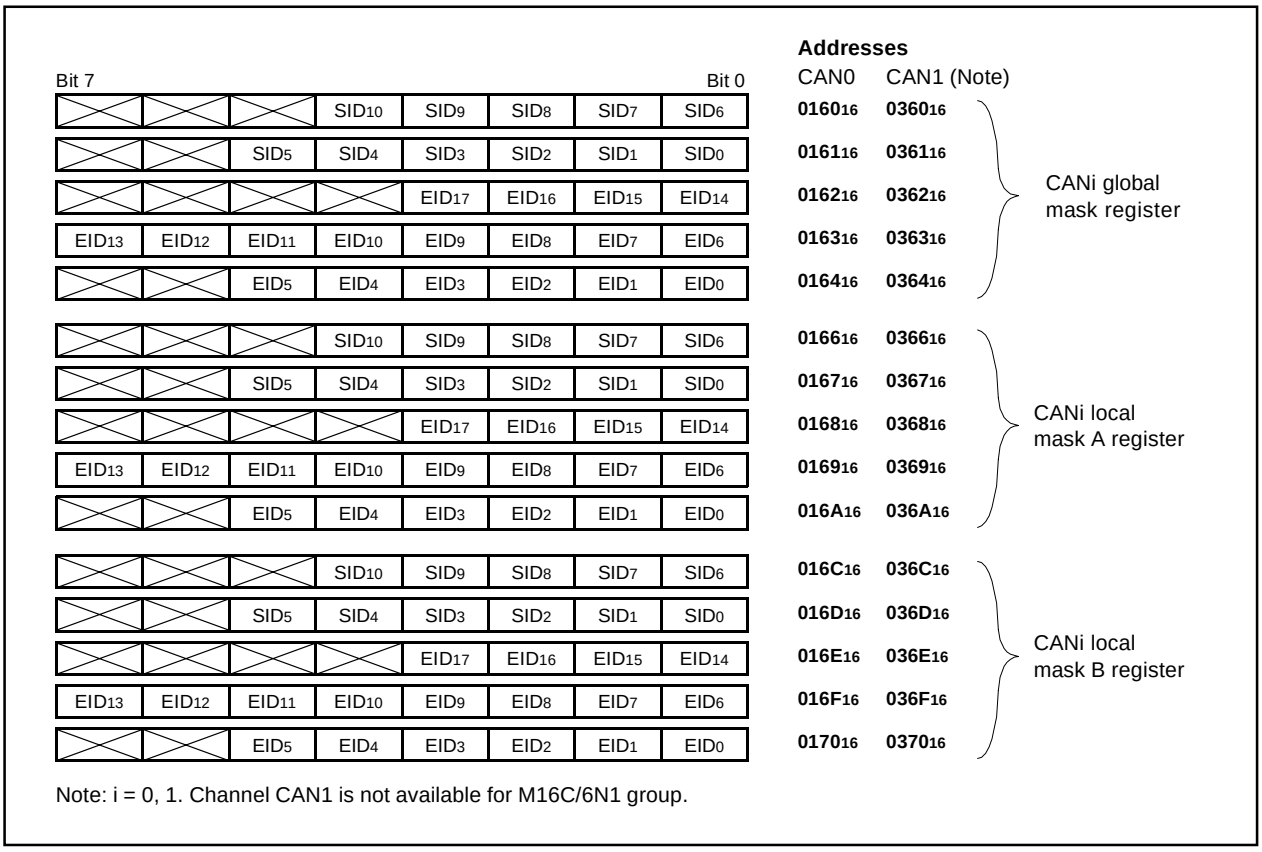


Figure 1.20.4. Bit mapping of the mask registers in byte access

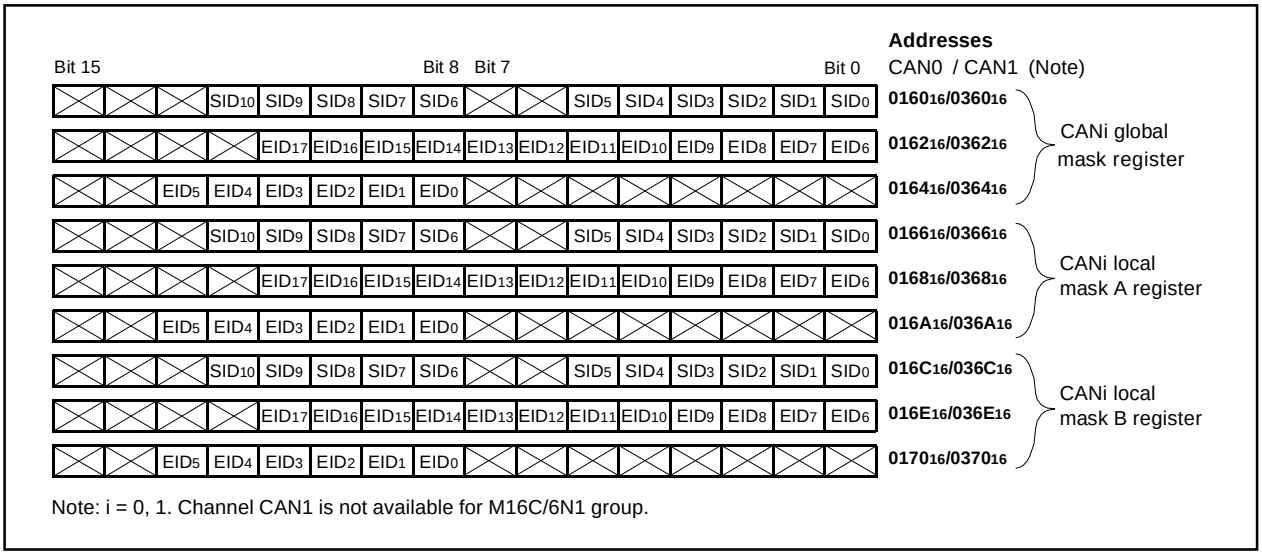


Figure 1.20.5. Bit mapping of the mask registers in word access

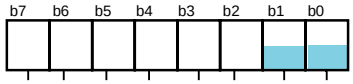
CAN SFR Registers

CANi Message Control Register j (j = 0 to 15)

Figure 1.20.6 shows the CANi message control register.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

CANi message control register j (i = 0, 1, j = 0 to 15)



Symbol	Address	When reset
C0MCTLj (j = 0 to 15)	0200₁₆ to 020F₁₆	00₁₆
C1MCTLj (j = 0 to 15) (Note 1)	0220₁₆ to 022F₁₆	00₁₆

Bit symbol	Bit name	Function	R	W
NewData	Successful reception flag	When set to reception slot 0: The content of the slot is read or still under processing by the CPU. 1: The CAN module has stored new data in the slot.	○	(Note 2)
SentData	Successful transmission flag	When set to transmission slot 0: Transmission is not started or completed yet. 1: Transmission is successfully completed.	○	(Note 2)
InvalData	"Under reception" flag	When set to reception slot 0: The message is valid. 1: The message is invalid. (The message is being updated.).	○	—
TrmActive	"Under transmission" flag	When set to transmission slot 0: Waiting for bus idle or completion of arbitration. 1: Transmitting	○	—
MsgLost	Overwrite flag	When set to reception slot 0: No message has been overwritten in this slot. 1: This slot already contained a message, but it has been overwritten by a new one.	○	(Note 2)
RemActive	Remote frame transmission/ reception status flag (Note 3)	0: Data frame transmission/reception status 1: Remote frame automatic transfer status	○	—
RspLock	Transmission/ reception auto response lock mode select bit	When set to reception remote frame slot 0: After a remote frame is received, it will be answered automatically. 1: After a remote frame is received, no transmission will be started as long as this bit is set to "1". (Not responding)	○	○
Remote	Remote frame corresponding slot select bit	0: Slot not corresponding to remote frame 1: Slot corresponding to remote frame	○	○
RecReq (Note 4)	Reception slot request bit	0: Not reception slot 1: Reception slot	○	○
TrmReq (Note 4)	Transmission slot request bit	0: Not transmission slot 1: Transmission slot	○	○

Note 1: Channel CAN1 is not available for M16C/6N1 group.
Note 2: As for write, only writing "0" is possible. The value of each bit is written when the CAN module enters the respective state.
Note 3: In Basic CAN mode, they serve as data format identification flag. Refer to the section of Basic CAN mode for more details.
Note 4: One slot cannot be defined as reception slot and transmission slot at the same time.

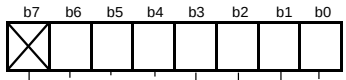
Figure 1.20.6. CANi message control register j

CANi Control Register

Figure 1.20.7 shows the CANi control register.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

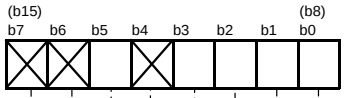
CANi control register (i = 0, 1)



Symbol	Address	When reset
C0CTLR	0210_{16}	$X0000001_2$
C1CTLR (Note 1)	0230_{16}	$X0000001_2$

Bit symbol	Bit name	Function	R	W
Reset	Full CAN module reset bit	0: Operation mode 1: Reset/initialization mode	○	○
LoopBack	Loop back mode select bit	0: Normal operation mode 1: Loop back mode	○	○
MsgOrder	Message order select bit	0: Word access 1: Byte access	○	○
BasicCAN	Basic CAN mode select bit	0: Normal operation mode 1: Basic CAN mode	○	○
BusErrEn	Bus error interrupt enable bit	0: Bus error interrupt disabled 1: Bus error interrupt enabled	○	○
Sleep	Sleep mode select bit	0: Sleep mode disabled 1: Sleep mode enabled; clock supply stopped	○	○
PortEn	CAN port enable bit	0: I/O port function 1: CTx/CRx function (Note 2)	○	○
Nothing is assigned. Write "0" in case of write. The value is indeterminate when read.			—	—

Note 1: The C1CTLR register (addresses 0231_{16}, 0230_{16}) has to be set to "0020_{16}" for M16C/6N1 group.
Note 2: CTx/CRx function regardless of configuration of PD7 and PD9 registers.



Symbol	Address	When reset
C0CTLR	0211_{16}	$XX0X0000_2$
C1CTLR (Note 1)	0231_{16}	$XX0X0000_2$

Bit symbol	Bit name	Function	R	W
TSPreScale Bit1, Bit0	Time stamp prescaler	b1 b0 0 0: Period of 1 bit time 0 1: Period of 1/2 bit time 1 0: Period of 1/4 bit time 1 1: Period of 1/8 bit time	○	○
TSReset	Time stamp counter reset bit (Note 2)	0: Normal operation mode 1: Compulsory reset of the time stamp counter	○	○
RetBusOff	Return from bus off command bit (Note 3)	0: Normal operation mode 1: Compulsory return from bus off	○	○
Nothing is assigned. Write "0" in case of write. The value is indeterminate when read.			—	—
RXOnly	Listen-only mode select bit	0: Normal operation mode 1: Listen-only mode	○	○
Nothing is assigned. Write "0" in case of write. The value is indeterminate when read.			—	—

Note 1: The C1CTLR register (addresses 0231_{16}, 0230_{16}) has to be set to "0020_{16}" for M16C/6N1 group.
Note 2: When the TSReset bit is set to "1", the CiTSR register gets cleared to "0000_{16}". After this, the bit is automatically cleared to "0".
Note 3: When the RetBusOff bit is set to "1", the CiRECR register and the CiTECR register gets cleared to "00_{16}". After this, the bit is automatically cleared to "0".

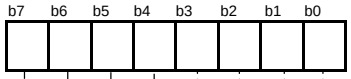
Figure 1.20.7. CANi control register

CANi Status Register

Figure 1.20.8 shows the CANi status register.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

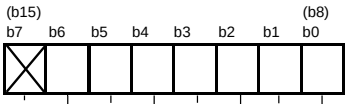
CANi status register (i =0, 1)



Symbol	Address	When reset
C0STR	0212_{16}	00_{16}
C1STR (Note)	0232_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
MBox	Active slot bits	b3 b2 b1 b0 0 0 0 0 : Slot 0 0 0 0 1 : Slot 1 0 0 1 0 : Slot 2 ... 1 1 1 0 : Slot 14 1 1 1 1 : Slot 15	○	—
TrmSucc	Successful transmission flag	0: No [successful] transmission 1: The CAN module has transmitted a message successfully.	○	—
RecSucc	Successful reception flag	0: No [successful] reception 1: CAN module received a message successfully.	○	—
TrmState	Transmission flag (Transmitter)	0: CAN module is idle or receiver. 1: CAN module is transmitter.	○	—
RecState	Reception flag (Receiver)	0: CAN module is idle or transmitter. 1: CAN module is receiver.	○	—

Note: Channel CAN1 is not available for M16C/6N1 group.



Symbol	Address	When reset
C0STR	0213_{16}	$X0000001_2$
C1STR (Note)	0233_{16}	$X0000001_2$

Bit symbol	Bit name	Function	R	W
State_Reset	Reset state flag	0: Operation mode 1: Reset mode	○	—
State_ LoopBack	Loop back state flag	0: Normal operation mode 1: Loop back mode	○	—
State_ MsgOrder	Message order state flag	0: Word access 1: Byte access	○	—
State_ BasicCAN	Basic CAN mode state flag	0: Normal operation mode 1: BasicCAN mode	○	—
State_ BusError	Bus error state flag	0: No error has occurred. 1: A CAN bus error has occurred.	○	—
State_ ErrPass	Error passive state flag	0: The CAN module is not in error passive state. 1: The CAN module is in error passive state.	○	—
State_ BusOff	Error bus off state flag	0: The CAN module is not in error bus off state. 1: The CAN module is in error bus off state.	○	—
Nothing is assigned. Write "0" in case of write. The value is indeterminate when read.			—	—

Note: Channel CAN1 is not available for M16C/6N1 group.

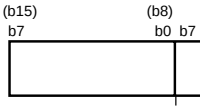
Figure 1.20.8. CANi status register

CANi Slot Status Register

Figure 1.20.9 shows the CANi slot status register.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

CANi slot status register (i = 0, 1)



Symbol	Address	When reset
C0SSTR	0215_{16}, 0214_{16}	0000_{16}
C1SSTR (Note)	0235_{16}, 0234_{16}	0000_{16}

Function	Values	R	W
Slot status bits Each bit corresponds to the slot with the same number.	0: Reception slot The message has been read. Transmission slot Transmission is not completed. 1: Reception slot The message has not been read. Transmission slot Transmission is completed.	○	–

Note: Channel CAN1 is not available for M16C/6N1 group.

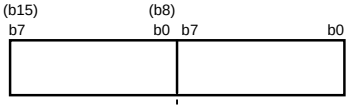
Figure 1.20.9. CANi slot status register

CANi Interrupt Control Register

Figure 1.20.10 shows the CANi interrupt control register.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

CANi interrupt control register (i = 0, 1)



Symbol	Address	When reset
C0ICR	0217_{16}, 0216_{16}	0000_{16}
C1ICR (Note 1)	0237_{16}, 0236_{16}	0000_{16}

Function	Values (Note 2)	R	W
Interrupt enable bits: Each bit corresponds with a slot with the same number. Enabled/disabled of successful transmission interrupt or successful reception interrupt can be selected.	0: Interrupt disabled 1: Interrupt enabled	○	○

Note 1: Channel CAN1 is not available for M16C/6N1 group.
Note 2: These bits can not be set in CAN reset/initialization and CAN sleep modes of the CAN module.

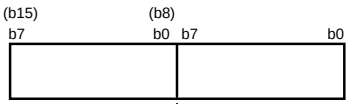
Figure 1.20.10. CANi interrupt control register

CANi Extended ID Register

Figure 1.20.11 shows the CANi extended ID register.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

CANi extended ID register (i = 0, 1)



Symbol	Address	When reset
C0IDR	0219_{16}, 0218_{16}	0000_{16}
C1IDR (Note 1)	0239_{16}, 0238_{16}	0000_{16}

Function	Values (Note 2)	R	W
Extended ID bits: Each bit corresponds with a slot with the same number. Selection of the ID format that each slot handles.	0: Standard ID 1: Extended ID	○	○

Note 1: Channel CAN1 is not available for M16C/6N1 group.
Note 2: These bits can not be set in CAN reset/initialization and CAN sleep modes of the CAN module.

Figure 1.20.11. CANi extended ID register

CANi Configuration Register

Figure 1.20.12 shows the CANi configuration register.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

CANi configuration register (i = 0, 1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
C0CONR	021A16	Indeterminate
C1CONR (Note 1)	023A16	Indeterminate

Bit symbol	Bit name	Function	R	W
BRP	Prescaler division ratio select bits	b3 b2 b1 b0 0 0 0 0 : fCAN/1 0 0 0 1 : fCAN/2 0 0 1 0 : fCAN/3 ⋮ 1 1 1 0 : fCAN/15 1 1 1 1 : fCAN/16 (Note 2)	○	○
SAM	Sampling control bit	0 : One time sampling 1 : Three times sampling	○	○
PTS	Propagation time segment control bits	b7 b6 b5 0 0 0 : 1Tq 0 0 1 : 2Tq 0 1 0 : 2Tq ⋮ 1 1 0 : 7Tq 1 1 1 : 8Tq	○	○

Note 1: Channel CAN1 is not available for M16C/6N1 group.
Note 2: fCAN serves for the CAN clock. The period is decided by configuration of the CCLKi bits (i = 0, 1, 2, 3, 4, 5, 6).

(b15) (b8)
b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
C0CONR	021B16	Indeterminate
C1CONR (Note)	023B16	Indeterminate

Bit symbol	Bit name	Function	R	W
PBS1	Phase buffer segment 1 control bits	b2 b1b0 0 0 0 : Inhibited 0 0 1 : 2Tq 0 1 0 : 3Tq ⋮ 1 1 0 : 7Tq 1 1 1 : 8Tq	○	○
PBS2	Phase buffer segment 2 control bits	b5 b4 b3 0 0 0 : Inhibited 0 0 1 : 2Tq 0 1 0 : 3Tq ⋮ 1 1 0 : 7Tq 1 1 1 : 8Tq	○	○
SJW	Resynchronization jump width control bits	b7 b6 0 0 : 1Tq 0 1 : 2Tq 1 0 : 3Tq 1 1 : 4Tq	○	○

Note: Channel CAN1 is not available for M16C/6N1 group.

Figure 1.20.12. CANi configuration register

CANi Receive Error Count Register

Figure 1.20.13 shows the CANi receive error count register.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

CANi receive error count register (i = 0, 1)

b7 b0

Symbol	Address	When reset
C0RECR	$021C_{16}$	00_{16}
C1RECR (Note 1)	$023C_{16}$	00_{16}

Function	Range of the value	R	W
Reception error counting function The value is incremented or decremented according to the CAN module's error status.	00_{16} to FF_{16} (Note 2)	○	–

Note 1: Channel CAN1 is not available for M16C/6N1 group.
Note 2: The value is indeterminate in bus off state.

Figure 1.20.13. CANi receive error count register

CANi Transmit Error Count Register

Figure 1.20.14 shows the CANi transmit error count register.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

CANi transmit error count register (i = 0, 1)

b7 b0

Symbol	Address	When reset
C0TECR	$021D_{16}$	00_{16}
C1TECR (Note)	$023D_{16}$	00_{16}

Function	Range of the value	R	W
Transmission error counting function The value is incremented or decremented according to the CAN module's error status.	00_{16} to FF_{16}	○	–

Note: Channel CAN1 is not available for M16C/6N1 group.

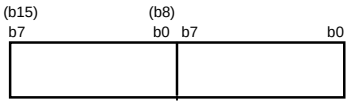
Figure 1.20.14. CANi transmit error count register

CANi Time Stamp Register

Figure 1.20.15 shows the CANi time stamp register.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

CANi time stamp register (i = 0, 1)



Symbol	Address	When reset
C0TSR	$021F_{16}$, $021E_{16}$	0000_{16}
C1TSR (Note)	$023F_{16}$, $023E_{16}$	0000_{16}

Function	Range of the value	R	W
Time stamp function	0000_{16} to $FFFF_{16}$	○	○

Note: Channel CAN1 is not available for M16C/6N1 group.

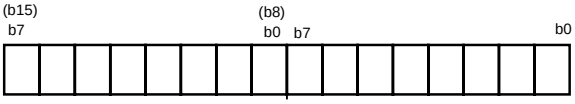
Figure 1.20.15. CANi time stamp register

CANi Acceptance Filter Support Register

Figure 1.20.16 shows the CANi acceptance filter support register.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

CANi acceptance filter support register (i = 0, 1)



Symbol	Address	When reset
C0AFS	0243_{16}, 0242_{16}	Indeterminate
C1AFS (Note)	0245_{16}, 0244_{16}	Indeterminate

Function	Values	R	W
Write the content equivalent to the standard frame ID of the received message. The value is "converted standard ID" when read.	Standard frame ID	○	○

Note: Channel CAN1 is not available for M16C/6N1 group.

Figure 1.20.16. CANi acceptance filter support register

Operational Modes

The CAN module has the following three operational modes.

- CAN Reset/Initialization Mode
- CAN Sleep Mode
- CAN Operation Mode

Figure 1.20.17 shows transition between operational modes.

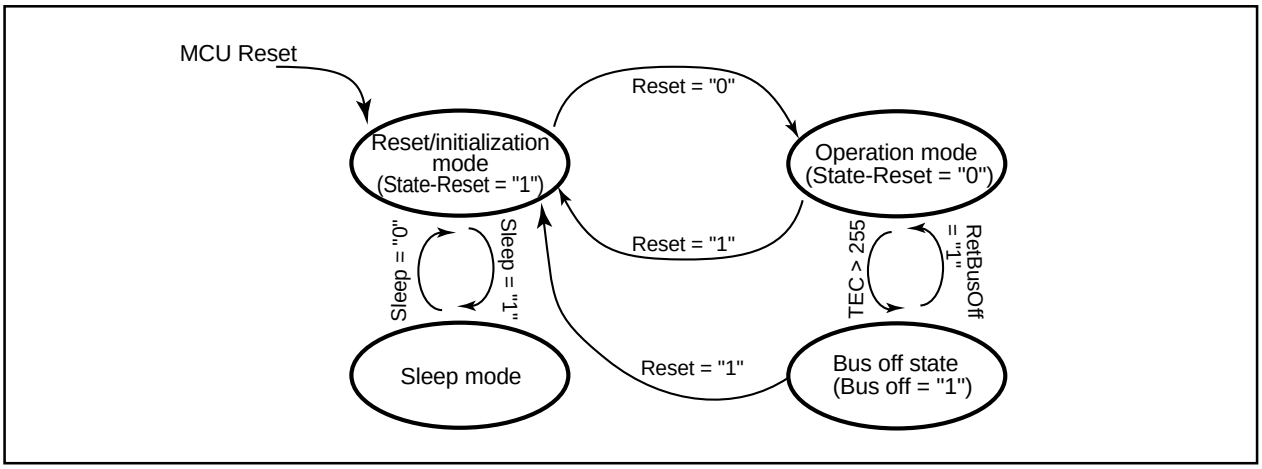


Figure 1.20.17. Transition between operational modes

CAN Reset/Initialization Mode

The CAN reset/initialization mode is activated upon MCU reset or by setting the Reset bit of the CANi control register. It can be observed by reading the State-Reset bit of the CANi status register. Entering the CAN reset/initialization mode initiates the following functions by the module:

- Suspend all communication functions. When the CAN reset/initialization mode is activated during an ongoing transmission in operation mode, the module suspends the mode transition until completion of the transmission (successful, arbitration loss, or error detection) and then sets the State-Reset bit.
- Initialization of CANi extended ID, CANi message control, CANi interrupt control, CANi status, CANi time stamp, CANi receive error count and CANi transmit error count registers to their reset values. All these registers are locked to prevent CPU modification.
- The CANi configuration and CANi control registers and the message box retain their contents and are available for CPU access.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

CAN Operation Mode

The CAN operation mode is activated by clearing the Reset bit of the CANi control register. Entering the operation mode initiates the following functions by the module:

- The module's communication functions are released and it becomes an active node on the network and may transmit and receive CAN messages.
- Release the internal fault confinement logic including receive and transmit error counters. The module may leave the CAN operation mode depending on the error counts.

Within the CAN operation mode the module may be in three different sub modes, depending on which type of communication functions are performed:

- Module idle: The modules receive and transmit sections are inactive.
- Module receives: The module receives a CAN message sent by another node.
- Module transmits: The module transmits a CAN message. The module may receive its own message simultaneously when the loopback function is enabled.

Figure 1.20.18 shows sub modes of the CAN operation mode.

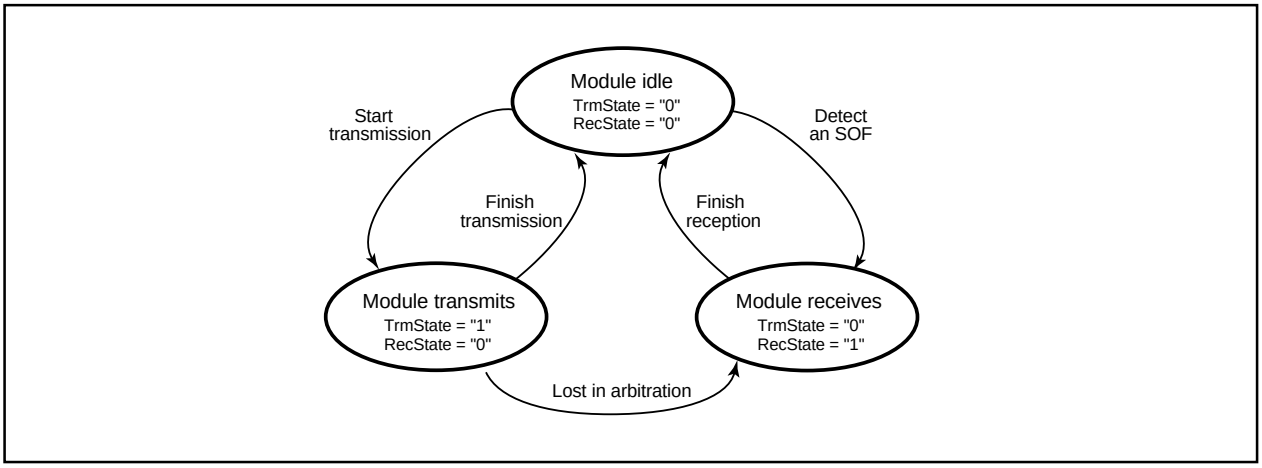


Figure 1.20.18. Sub modes of the CAN operation mode

CAN Sleep Mode

The CAN sleep mode is activated by setting the Sleep bit of the CANi control register. It should never be activated from the CAN operation mode but only via the CAN reset/initialization mode. Entering the CAN sleep mode instantly stops the modules clock supply and thereby reduces power dissipation.

Bus off State

The bus off state is entered according to the fault confinement rules of the CAN specification. It can be quit instantly to error active state by setting the RetBusOff bit of the CANi control register. This does not alter any CAN registers, except CANi receive error count and CANi transmit error count registers.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

Configuration of the CAN Module System Clock

The M16C/6N group has a CAN module system clock select circuit dedicated to each channel. Configuration of the CAN module system clock can be done through manipulating the CAN0/1 clock select register (address $025F_{16}$) and the BRP bits of the CANi configuration registers (addresses $021A_{16}$ and $023A_{16}$). For the CAN0/1 clock select register, refer to the section of the clock generating circuit. Figure 1.20.19 shows a block diagram of the clock generating circuit of the CAN module system.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

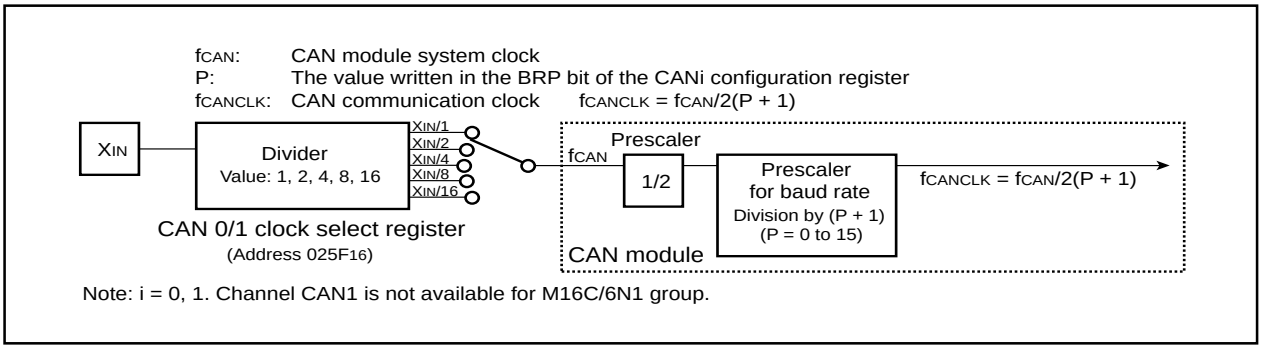


Figure 1.20.19. Block diagram of the CAN module system clock generating circuit

CAN Bus Timing Control

Bit Timing Configuration

The bit time consists of the following four segments:

- Synchronization segment (SS)
  This serves for monitoring a falling edge for synchronization.
- Propagation time segment (PTS)
  This segment absorbs physical delay on the CAN network which amounts to double the total sum of delay on the CAN bus, the input comparator delay, and the output driver delay.
- Phase buffer segment 1 (PBS1)
  This serves for compensating the phase error. When the falling edge of the bit falls later than expected, the segment can become longer by the maximum of the value defined in SJW.
- Phase buffer segment 2 (PBS2)
  This segment has the same function as the phase buffer segment 1. When the falling edge of the bit falls earlier than expected, the segment can become shorter by the maximum of the value defined in SJW.

Figure 1.20.20 shows the bit timing.

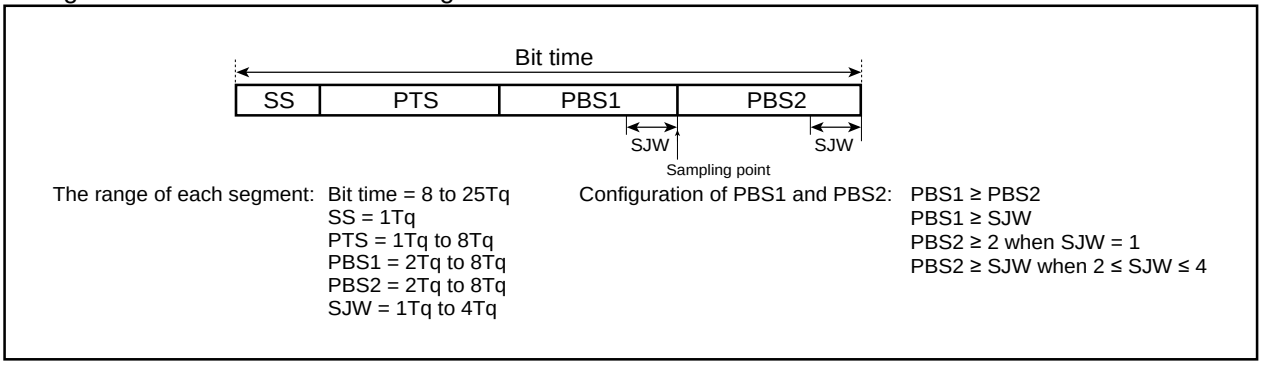


Figure 1.20.20. Bit timing

Baud Rate

Baud rate depends on XIN, the division value of the CAN module system clock, the division value of the prescaler for baud rate, and the number of Tq of one bit.

Table 1.20.2 shows the examples of baud rate.

Table 1.20.2. Examples of baud rate

Baud rate	16MHz	10MHz	8MHz
1Mbps	8Tq (1)	———	———
500kbps	8Tq (2) 16Tq (1)	10Tq (1) ———	8Tq (1) ———
125kbps	8Tq (8) 16Tq (4)	10Tq (4) 20Tq (2)	8Tq (4) 16Tq (2)
83.3kbps	8Tq (12) 18Tq (6)	10Tq (6) 20Tq (3)	8Tq (6) 16Tq (3)
33.3kbps	8Tq (30) 16Tq (15)	10Tq (15) ———	8Tq (15) ———

Note: The number in ( ) indicates a value of "fCAN division value" multiplied by "division value of the prescaler for baud rate".

■Calculation of baud rate

$$\frac{X_{IN}}{2 \times \text{"}f_{CAN}\text{ division value (Note 1)"} \times \text{"division value of prescaler for baud rate (Note 2)"} \times \text{"number of Tq of one bit"}}$$

Note 1: fCAN division value = 1, 2, 4, 8, 16
fCAN division value: a value selected in the CAN0/1 clock select register

Note 2: Division value of prescaler for baud rate = P + 1 (P: 0 to 15)
P: a value selected in the BRP bit of the CANi configuration register

Acceptance Filtering Function and Masking Function

These functions serve the users to select and receive a facultative message. The CANi global mask register, the CANi local mask A register, and the CANi local mask B register can perform masking to the standard ID and the extended ID of 29 bits. The CANi global mask register corresponds to slots 0 to 13, the CANi local mask A register corresponds to slot 14, and the CANi local mask B register corresponds to slot 15. The masking function becomes valid to 11 bits or 29 bits of a received ID according to the value in the corresponding slot of the extended ID register upon acceptance filtering operation. When the masking function is employed, it is possible to receive a certain range of IDs.
Figure 1.20.21 shows correspondence of the mask registers and slots, Figure 1.20.22 shows the acceptance function.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

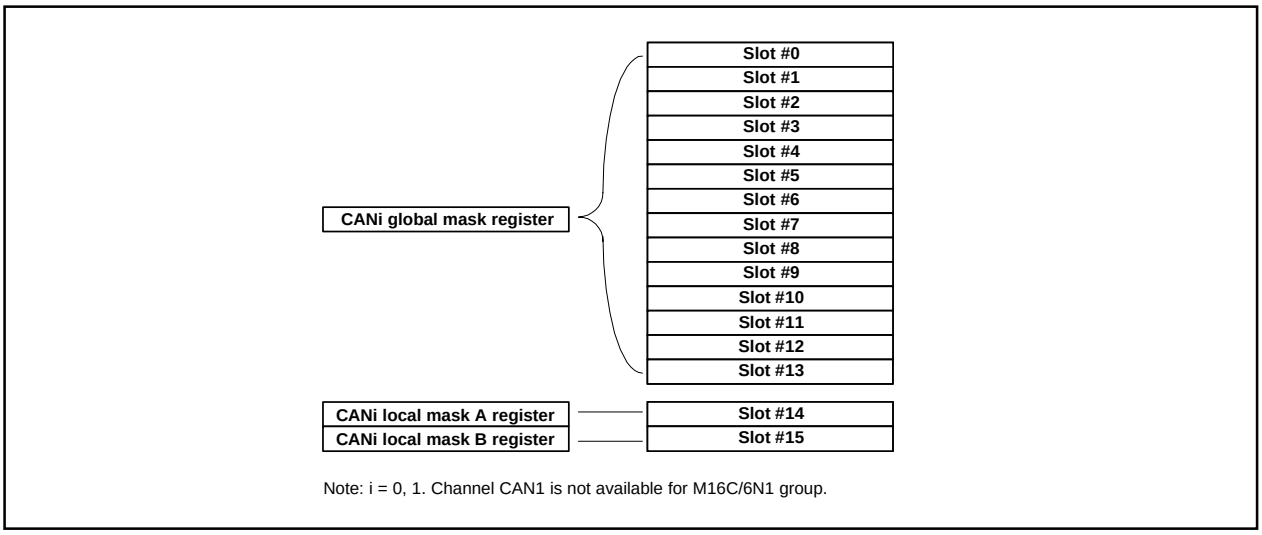


Figure 1.20.21. Correspondence of the mask registers to slots

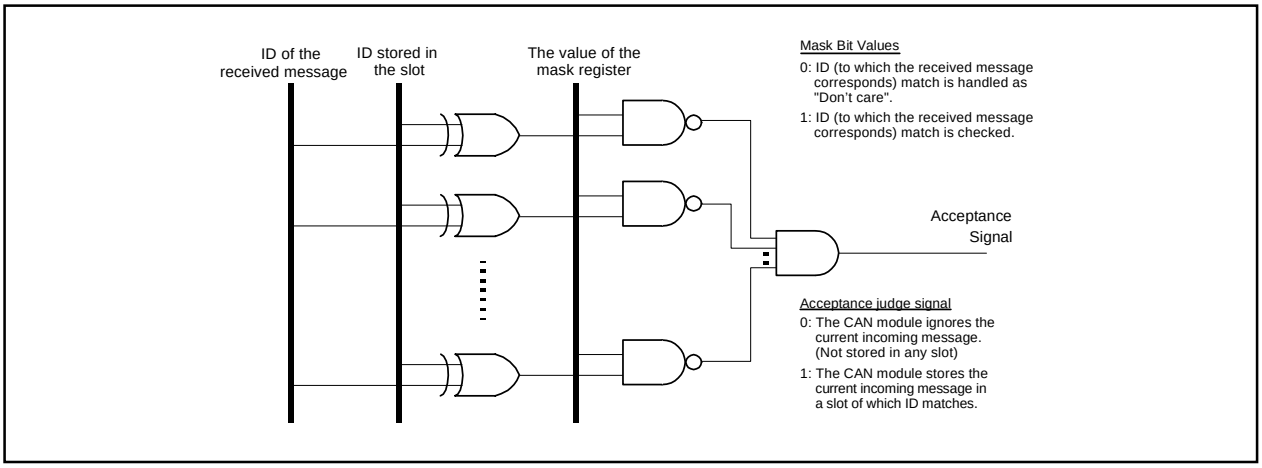


Figure 1.20.22. Acceptance function

When using the acceptance function, note the following points.

(1) When one ID is defined in two slots, the one with a smaller number alone is valid.

(2) When it is configured that slots 14 and 15 receive all IDs with Basic CAN mode, slots 14 and 15 receive all IDs which are not stored into slots 0 to 13.

Acceptance Filter Support Unit (ASU)

The acceptance filter support unit has a function to judge valid/invalid of a received ID through table search. The IDs to receive are registered in the data table; a received ID is stored in the CANi acceptance filter support register, and table search is performed with a decoded received ID. The acceptance filter support unit can be used for the IDs of the standard frame only.

The acceptance filter support unit is valid in the following cases.

- When the ID to receive cannot be masked by the acceptance filter.
  (Example) IDs to receive: 078_{16}, 087_{16}, 111_{16}
- When there are too many IDs to receive; it would take too much time to filter them by software.

Figure 1.20.23 shows the write and read of CANi acceptance filter support register in word access.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

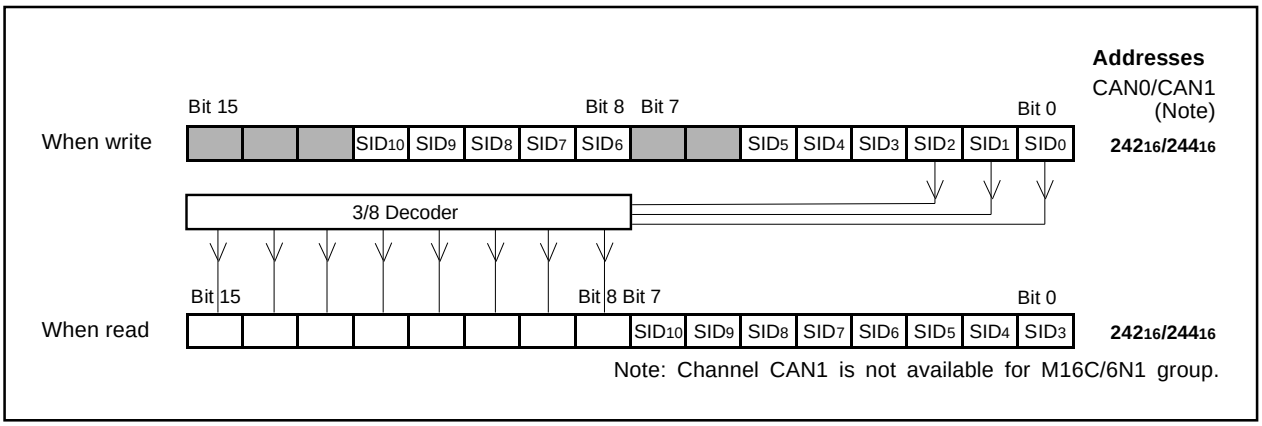


Figure 1.20.23. Write/read of acceptance filter support register in word access

Basic CAN Mode

When the BasicCAN bit of the CANi control register is set to "1", slots 14 and 15 correspond to Basic CAN mode. When slots 14 and 15 are defined as reception slots in Basic CAN mode, received messages are stored in slots 14 and 15 alternately.

Figure 1.20.24 shows the operation of slots 14 and 15 in Basic CAN mode.

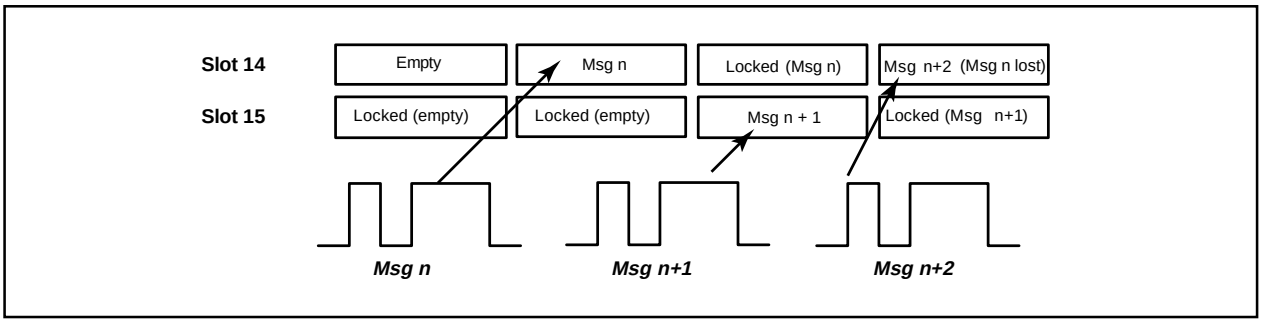


Figure 1.20.24. Operation of slots 14 and 15 in Basic CAN mode

When configuring Basic CAN mode, note the following points.

(1) Selection of Basic CAN mode has to be done in reset/initialization mode.
(2) Select the same ID for slots 14 and 15. Also, configuration of the CANi local mask A register and that of the local mask B register has to be the same.
(3) Define slots 14 and 15 as reception slot only.
(4) There is no protection available against message overwrite. A message can be overwritten by a new message.
(5) Slots 0 to 13 can be used in the same way as in normal CAN operation mode.

Concerning the CANi message control registers and communication environment configuration, Basic CAN mode is different from normal CAN operation mode in the following points:

(1) In normal CAN operation mode each slot can handle either data frame or remote frame, while in Basic CAN mode each slot can handle both frames. Namely, in Basic CAN mode slots 14 and 15 can receive both data frame and remote frame.
(2) For the above (1), the data format of a received message should be identified. In Basic CAN mode, the data is judged by the RemActive bit of the CANi message control register. The bit is cleared to "0" when the corresponding slot has received a data frame; set to "1" when the slot has received a remote frame.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

Return from Bus off Function

When the protocol controller enters bus off state, it is possible to make it compulsorily return from bus off state by the return from bus off function of the CANi control register. At this time, the error state changes from bus off state to error active state. Implementation of this function initializes the protocol controller. However, registers of the CAN module such as CANi configuration register and the content of each slot are not initialized.

Time Stamp Counter and Time Stamp Function

When the time stamp register is read, the value of the time stamp counter at the moment is read. The period of the time stamp counter reference clock is the same as that of 1 bit time that is configured by the CANi configuration register. The time stamp counter functions as a free run counter.

1/1, 1/2, 1/4, and 1/8 can be selected for the time stamp counter reference clock by configuration of the TSPreScale bits 1 and 0 of the CANi control register.

The time stamp counter is equipped with a register that captures the counter value when the protocol controller regards it as a successful reception. The captured value is stored when a time stamp value is stored in a reception slot.

Listen-Only Mode

When the RXOnly bit of the CANi control register is set to "1", the module enters listen-only mode.

In listen-only mode, no transmission -- data frames, error frames, and ACK response -- is performed to bus.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

Reception and Transmission

Configuration of CAN Reception and Transmission Mode

Table 1.20.3 shows configuration of CAN reception and transmission mode.

Table 1.20.3. Configuration of CAN reception and transmission mode

TrmReq	RecReq	Remote	RspLock	Communication mode of the slot
0	0	——	——	Communication environment configuration mode: configure the communication mode of the slot.
0	1	0	0	Configured as a reception slot for a data frame.
1	0	1	0	Configured as a transmission slot for a remote frame. (At this time the RemActive bit is "1".) After completion of transmission, this functions as a reception slot for a data frame. (At this time the RemActive bit is "0".) However, when an ID that matches on the CAN bus is detected before remote frame transmission, this immediately functions as a reception slot for a data frame.
1	0	0	0	Configured as a transmission slot for a data frame.
0	1	1	1/0	Configured as a reception slot for a remote frame. (At this time the RemActive bit is "1".) After completion of reception, this functions as a transmission slot for a data frame. (At this time the RemActive bit is "0".) However, transmission does not start as long as RspLock bit remains "1"; thus no automatic remote frame response. Response (transmission) starts when RspLock bit is cleared to "0".

When configuring a slot as a reception slot, note the following points.

(1) Before configuring a slot as a reception slot, be sure to clear the CANi message control register.

(2) A received message is stored in a slot that matches the condition first according to the result of reception mode configuration and acceptance filtering operation. Upon deciding in which slot to store, the smaller the number of the slot is, the higher priority it has.

(3) In normal CAN operation mode, when a CAN module transmits a message of which ID matches, the CAN module never receives the transmitted data. In loop back mode, however, the CAN module receives back the transmitted data. In this case, the module does not return ACK.

When configuring a slot as a transmission slot, note the following points.

(1) Before configuring a slot as a transmission slot, be sure to clear the CANi message control register.

(2) Clear the TrmReq bit without fail before rewriting a transmission slot.

(3) A transmission slot cannot be rewritten when the TrmActive bit is "1".

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

Reception

Figure 1.20.25 shows the behavior of the module when receiving two consecutive CAN messages, that fit into the slot of the shown CANi message a control register and leads to losing/overwriting of the first message.

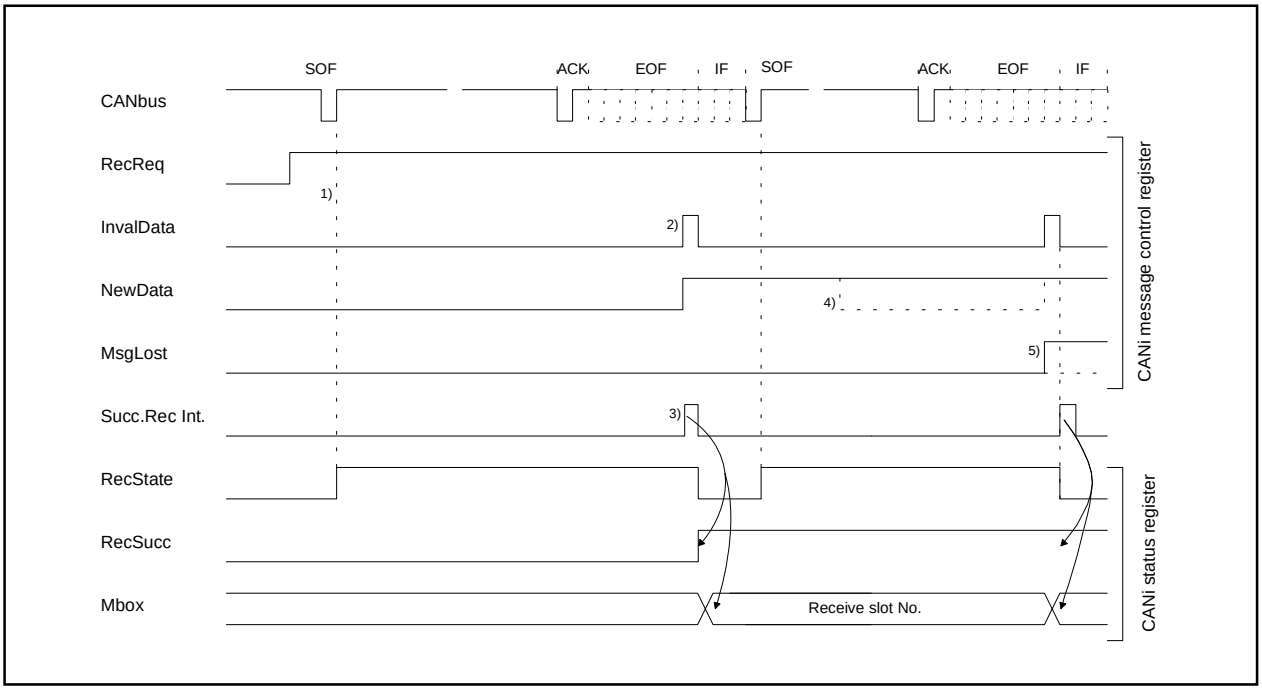


Figure 1.20.25. Timing of receive data frame sequence

1) On monitoring a SOF on the CAN bus the RecState bit becomes active immediately, given the module has no transmission pending (see section "Transmission" below).
2) After successful reception of the message the NewData bit of the receiving slot becomes active. The InvalData bit becomes active at the same time and becomes inactive again after the complete message was transferred to the slot.
3) When the bit in the CANi interrupt control register of the receiving slot is active the receive successful interrupt is requested and the CANi status register changes. It shows the slot number where the message was stored and the RecSucc bit is active.
4) After reading out the message out of the slot the CPU should clear the NewData bit to signal this to the module.
5) If the NewData bit is not cleared by the CPU and the Receive request for the slot is not disabled before the next successful reception of a CAN message that is fitting in this slot the MsgLost bit becomes acitive. The new received message is transferred to the slot. The interrupt request and change of the status register is same as in 3).

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

Transmission

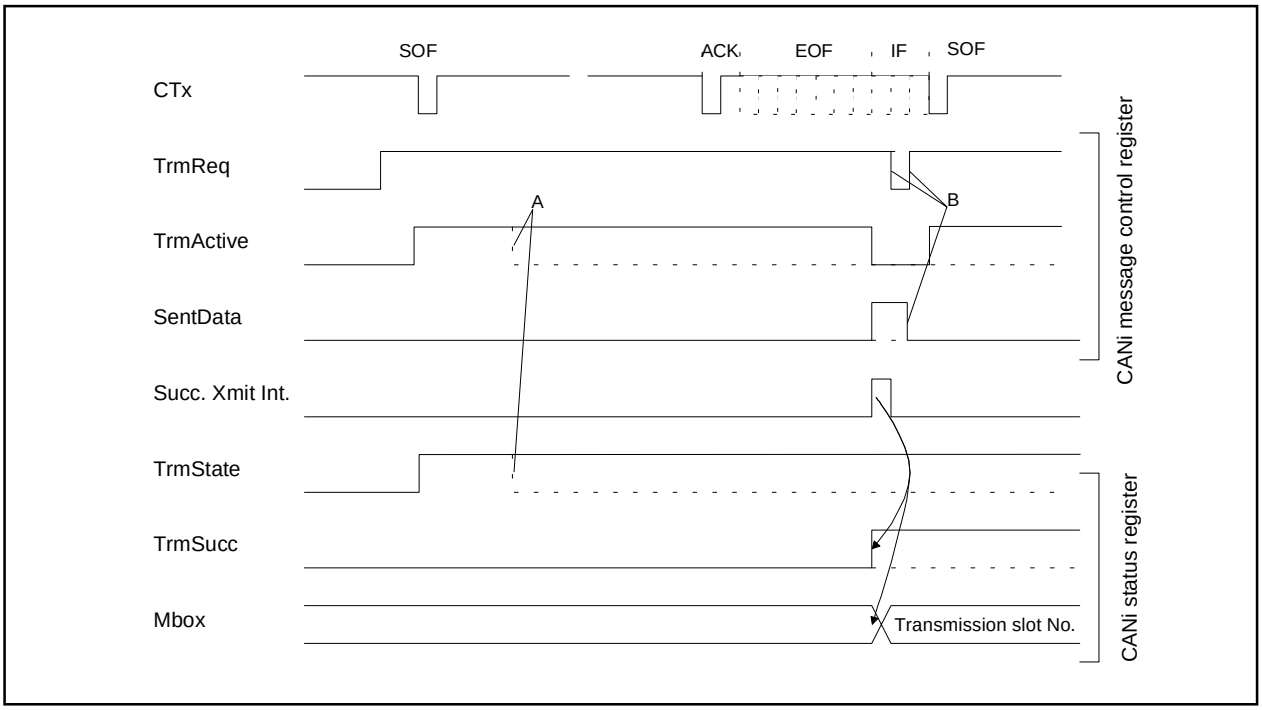


Figure 1.20.26. Timing of transmit sequence

1) If one or more of the slots of a module has a request for transmission, the module attempts to start the transmission at the next possible time (depending on the bus condition).
2) The TrmActive bit of the lowest slot with transmit request is set. Also the TrmState bit is set. If the arbitration is lost against another CAN node both bits are cleared again (A).
3a) When the arbitration was won, but the transmission was not successful, the module will attempt to re-transmit.
3b) When the arbitration was won and the transmission has been successful the SentData bit is set together with TrmSucc bit and the transmit successfull interrupt is activated, if the according bit in the CANi interrupt control register is active. The number of the slot that was transmitted can be found in Mbox bit.
4) After a successful transmission the module will not attempt to send the slot again until it is reactivated. To reactivate a slot for transmission, first the TrmReq bit has to be cleared. Then the SentData bit can be cleared and the TrmReq bit set again (B). Note that the SentData bit is locked and cannot be cleared as long as TrmReq bit is active.

Note: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

CAN Interrupts

The CAN module provides the following CAN interrupts. (Note 1)

- CAN0 Successful Reception Interrupt
- CAN0 Successful Transmission Interrupt
- CAN1 Successful Reception Interrupt
- CAN1 Successful Transmission Interrupt
- CAN0/1 Error Interrupt
  - Error Passive State
  - Error BusOff State
  - Bus Error (this feature can be disabled separately)
- CAN0/1 Wake Up Interrupt

When the CPU detects a successful reception/transmission interrupt, the CANi status register must be read to determine which slot has issued the interrupt. (Note 2)

Note 1: M16C/6N1 group provides 4 interrupts. Interrupts relating to channel CAN1 are invalid for M16C/6N1 group.

Note 2: i = 0, 1. Channel CAN1 is not available for M16C/6N1 group.

Programmable I/O Port

There are 87 programmable I/O ports: P0 to P10 (excluding P85). 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. P71 and P91 are Nch open drain ports and have no built-in pull-up resistance. P85 is an input-only port and has no built-in pull-up resistance.

Figures 1.21.1 to 1.21.4 show the programmable I/O ports. Figure 1.21.5 shows the input pins.

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 the 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.

(1) Direction registers

Figure 1.21.6 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. (Note)

In memory expansion and microprocessor mode, the contents of corresponding direction register of pins A0 to A19, D0 to D15, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RD}$, $\overline{WRL}$/$\overline{WR}$, $\overline{WRH}$/$\overline{BHE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$ and BCLK cannot be modified.

Note: There is no direction register bit for P85.

(2) Port registers

Figure 1.21.7 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.

In memory expansion and microprocessor mode, the contents of corresponding direction register of pins A0 to A19, D0 to D15, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RD}$, $\overline{WRL}$/$\overline{WR}$, $\overline{WRH}$/$\overline{BHE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$ and BCLK cannot be modified.

(3) Pull-up control registers

Figure 1.21.8 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.

However, in memory expansion mode and microprocessor mode, the pull-up control register of P0 to P3, P40 to P43, and P5 is invalid.

The contents of register can be changed, but the pull-up resistance is not connected.

(4) Port control register

Figure 1.21.9 shows the port control register.

The bit 0 of port control register is used to read port P1 as follows:

0 : When port P1 is input port, port input level is read.
When port P1 is output port , the contents of port P1 register is read.

1 : The contents of port P1 register is read though port P1 is input/output port.

This register is valid in the following:

- External bus width is 8 bits in microprocessor mode or memory expansion mode.
- Port P1 can be used as a port in multiplexed bus for the entire space.

Programmable I/O Port

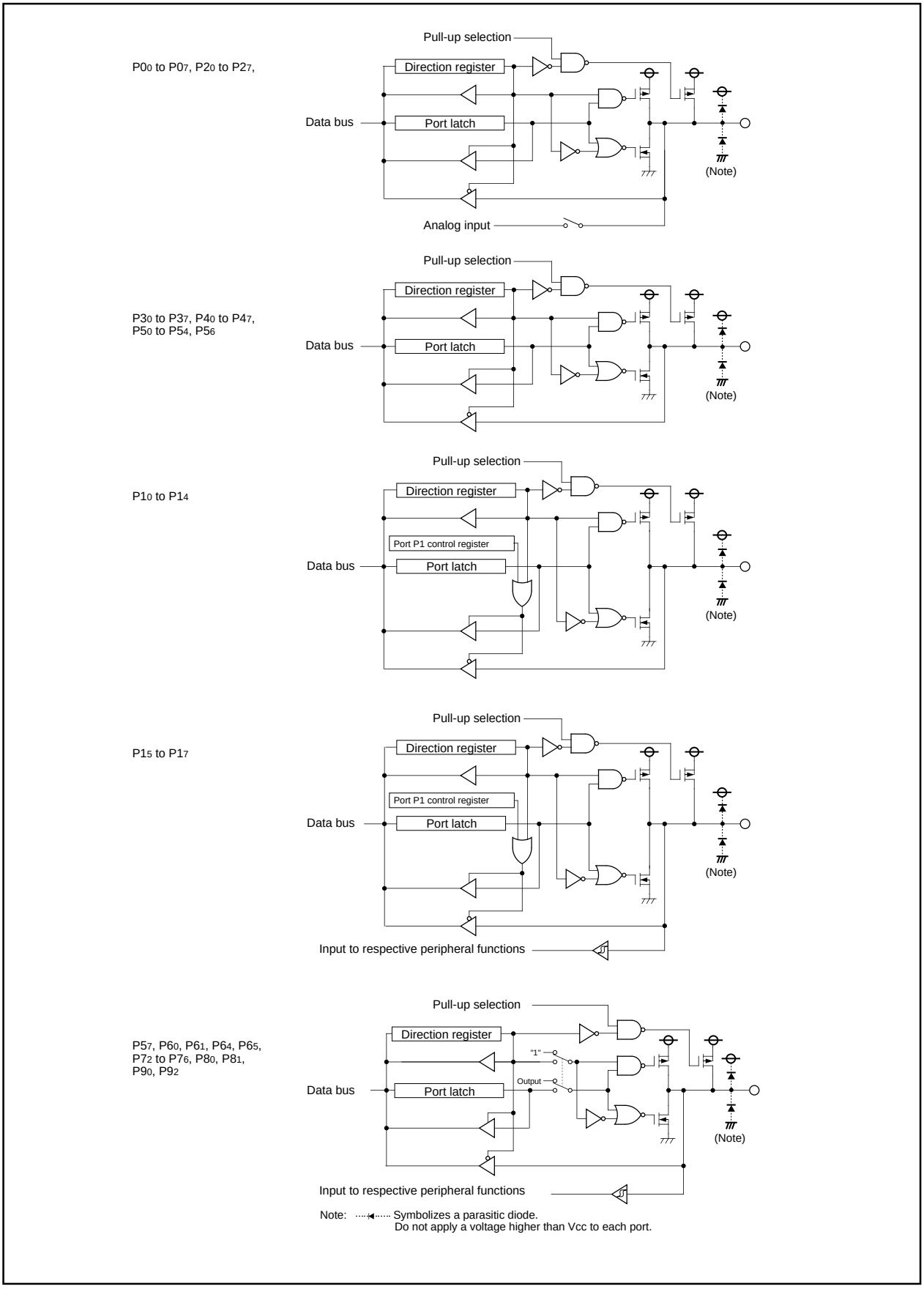


Figure 1.21.1. Programmable I/O ports (1)

Programmable I/O Port

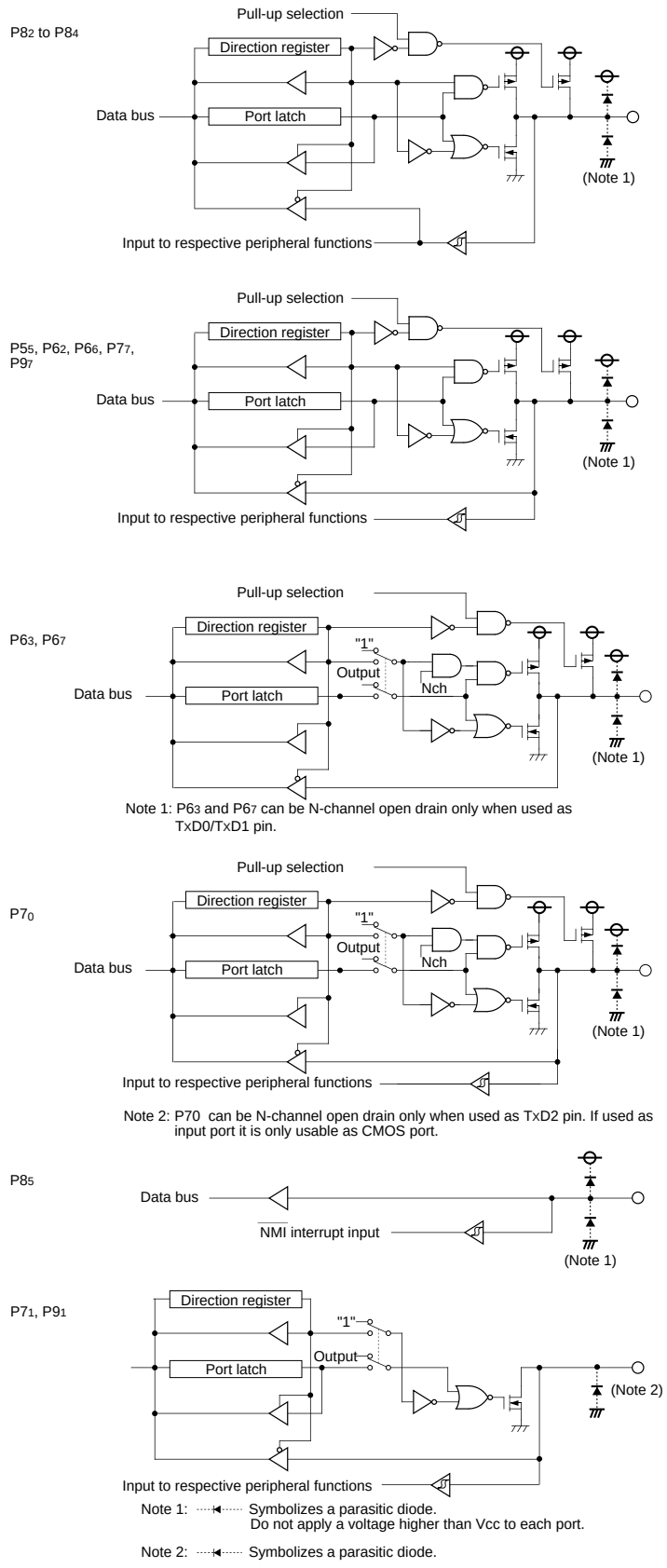


Figure 1.21.2. Programmable I/O ports (2)

Programmable I/O Port

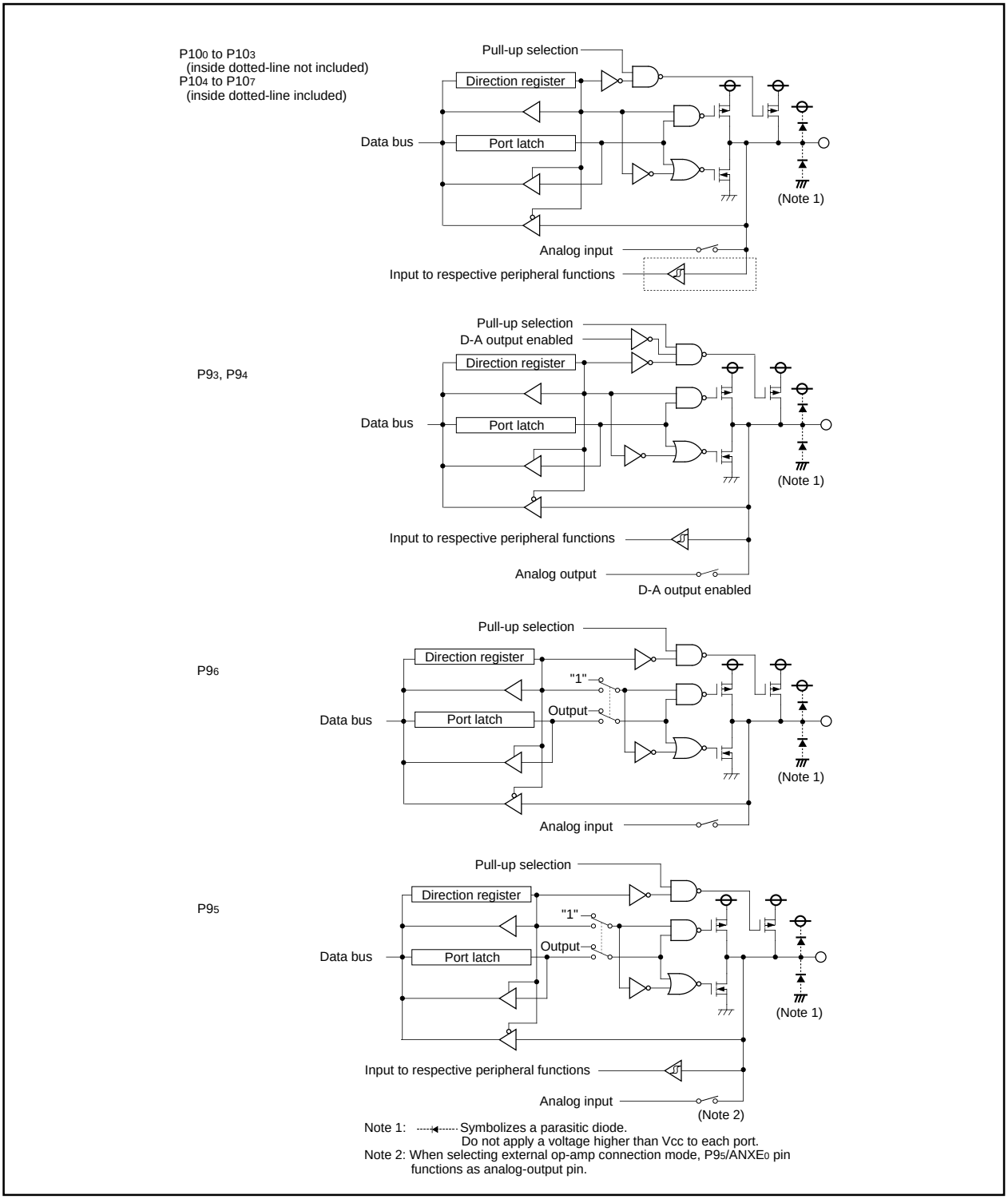


Figure 1.21.3. Programmable I/O ports (3)

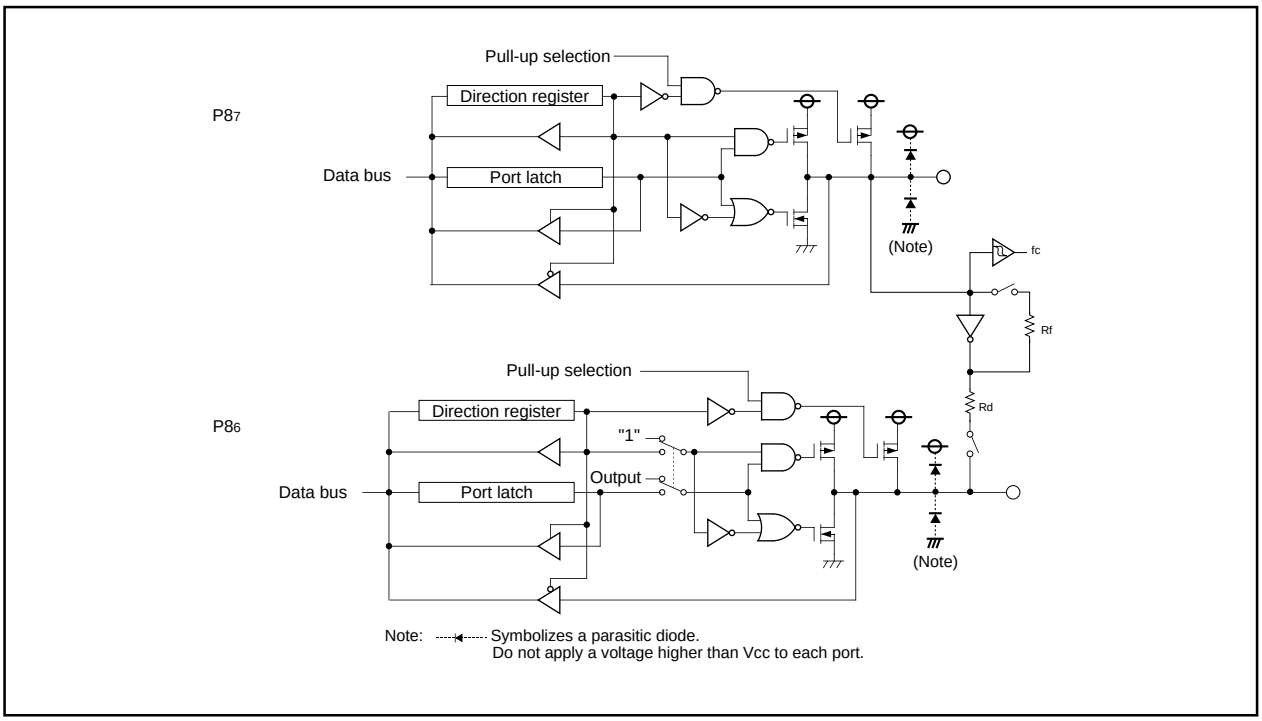


Figure 1.21.4. Programmable I/O ports (4)

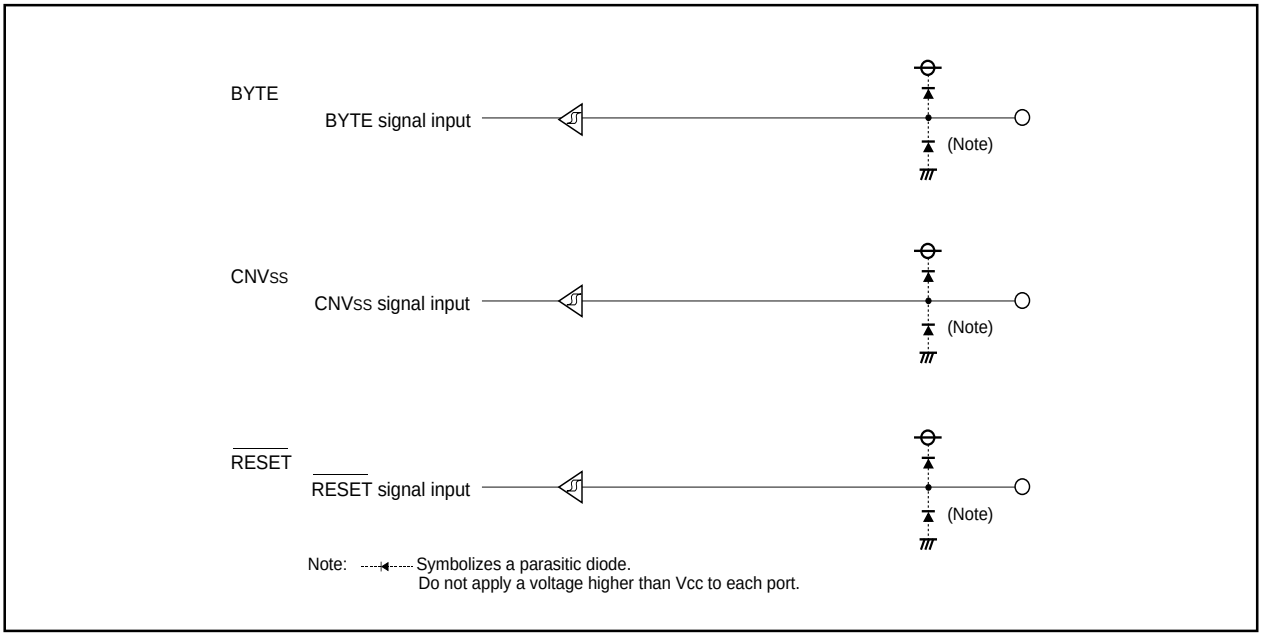
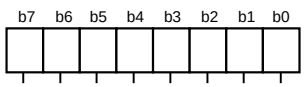


Figure 1.21.5. Input pins

Port Pi direction register (Note1, 2)



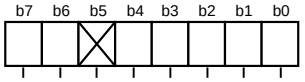
Symbol	Address	When reset
PDi (i = 0 to 10, except 8)	03E2$_{16}$, 03E3$_{16}$, 03E6$_{16}$, 03E7$_{16}$, 03EA$_{16}$ 03EB$_{16}$, 03EE$_{16}$, 03EF$_{16}$, 03F3$_{16}$, 03F6$_{16}$	00$_{16}$

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 = 0 to 10 except 8)	○	○
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		○	○

Note 1: Set bit 2 of protect register (address 000A$_{16}$) to "1" before rewriting to the port P7 and P9 direction register.

Note 2: In memory expansion and microprocessor mode, the contents of corresponding port Pi register of pins A$_0$ to A$_{19}$, D$_0$ to D$_{15}$, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RD}$, $\overline{WRL}$/$\overline{WR}$, $\overline{WRH}$/$\overline{BHE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$, and BCLK cannot be modified.

Port P8 direction register

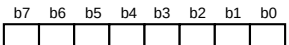


Symbol	Address	When reset
PD8	03F2$_{16}$	00X00000$_2$

Bit symbol	Bit name	Function	R	W
PD8_0	Port P8$_0$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○	○
PD8_1	Port P8$_1$ direction register		○	○
PD8_2	Port P8$_2$ direction register		○	○
PD8_3	Port P8$_3$ direction register		○	○
PD8_4	Port P8$_4$ direction register		○	○
Nothing is assigned. This bit can either be set nor reset. When read, its content is indeterminate.			–	–
PD8_6	Port P8$_6$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○	○
PD8_7	Port P8$_7$ direction register		○	○

Figure 1.21.6. Direction register

Port Pi register (Note1, 2)



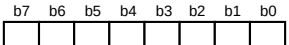
Symbol	Address	When reset
Pi (i = 0 to 10, except 8)	03E0$_{16}$, 03E1$_{16}$, 03E4$_{16}$, 03E5$_{16}$, 03E8$_{16}$ 03E9$_{16}$, 03EC$_{16}$, 03ED$_{16}$, 03F1$_{16}$, 03F4$_{16}$	Indeterminate Indeterminate

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 except 8)	○	○
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		○	○

Note 1: Since P7$_0$ and P9$_1$ are N-channel open drain ports, the data is high-impedance.

Note 2: In memory expansion and microprocessor mode, the contents of corresponding port Pi register of pins A$_0$ to A$_{19}$, D$_0$ to D$_{15}$, $\overline{CS0}$ to $\overline{CS3}$, $\overline{RD}$, $\overline{WRL}$/$\overline{WR}$, $\overline{WRH}$/$\overline{BHE}$, ALE, $\overline{RDY}$, $\overline{HOLD}$, $\overline{HLDA}$, and BCLK cannot be modified.

Port P8 register



Symbol	Address	When reset
P8	03F0$_{16}$	Indeterminate

Bit symbol	Bit name	Function	R	W
P8_0	Port P8$_0$ register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit (except for P8$_5$) 0 : "L" level data 1 : "H" level data	○	○
P8_1	Port P8$_1$ register		○	○
P8_2	Port P8$_2$ register		○	○
P8_3	Port P8$_3$ register		○	○
P8_4	Port P8$_4$ register		○	○
P8_5	Port P8$_5$ register		○	×
P8_6	Port P8$_6$ register		○	○
P8_7	Port P8$_7$ register		○	○

Figure 1.21.7. Port register

Pull-up control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
PUR0	03FC16	0016

Bit symbol	Bit name	Function	R	W
PU00	P00 to P03 pull-up select bit	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU01	P04 to P07 pull-up select bit		○	○
PU02	P10 to P13 pull-up select bit		○	○
PU03	P14 to P17 pull-up select bit		○	○
PU04	P20 to P23 pull-up select bit		○	○
PU05	P24 to P27 pull-up select bit		○	○
PU06	P30 to P33 pull-up select bit		○	○
PU07	P34 to P37 pull-up select bit		○	○

Note: In memory expansion and microprocessor mode, the content of this register can be changed, but the pull-up resistance is not connected.

Pull-up control register 1

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
PUR1	03FD16	0016 (Note 2)

Bit symbol	Bit name	Function	R	W
PU10	P40 to P43 pull-up select bit (Note 3)	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU11	P44 to P47 pull-up select bit		○	○
PU12	P50 to P53 pull-up select bit (Note 3)		○	○
PU13	P54 to P57 pull-up select bit		○	○
PU14	P60 to P63 pull-up select bit		○	○
PU15	P64 to P67 pull-up select bit		○	○
PU16	P70 to P73 pull-up select bit (Note 1)		○	○
PU17	P74 to P77 pull-up select bit		○	○

Note 1: Since P71 is an N-channel open drain port, pull-up control is not available for it.
Note 2: When the VCC level is being impressed to the CNVSS terminal, this register becomes to 0216 when reset (PU11 becomes to "1").
Note 3: In memory expansion and microprocessor mode, the content of this register can be changed, but the pull-up resistance is not connected.

Pull-up control register 2

b7	b6	b5	b4	b3	b2	b1	b0
✕	✕						

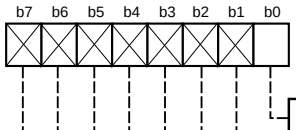
Symbol	Address	When reset
PUR2	03FE16	0016

Bit symbol	Bit name	Function	R	W
PU20	P80 to P83 pull-up select bit	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU21	P84 to P87 pull-up select bit (Except P85)		○	○
PU22	P90, P92, P93 pull-up select bit (Note)		○	○
PU23	P94 to P97 pull-up select bit		○	○
PU24	P100 to P103 pull-up select bit		○	○
PU25	P104 to P107 pull-up select bit		○	○
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are "0".			—	—

Note: Since P91 is an N-channel open drain port, pull-up control is not available for it.

Figure 1.21.8. Pull-up control register

Port control register



Symbol	Address	When reset
PCR	$03FF_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
PCR0	Port P1 control register	0 : When input port, read port input level. When output port, read the contents of port P1 register. 1 : Read the contents of port P1 register though input/output port.	○	○
Nothing is assigned. Theses bits can neither be set nor reset. When read, their contents are "0".			—	—

Figure 1.21.9. Port control register

Table 1.21.1. Example connection of unused pins in single-chip mode

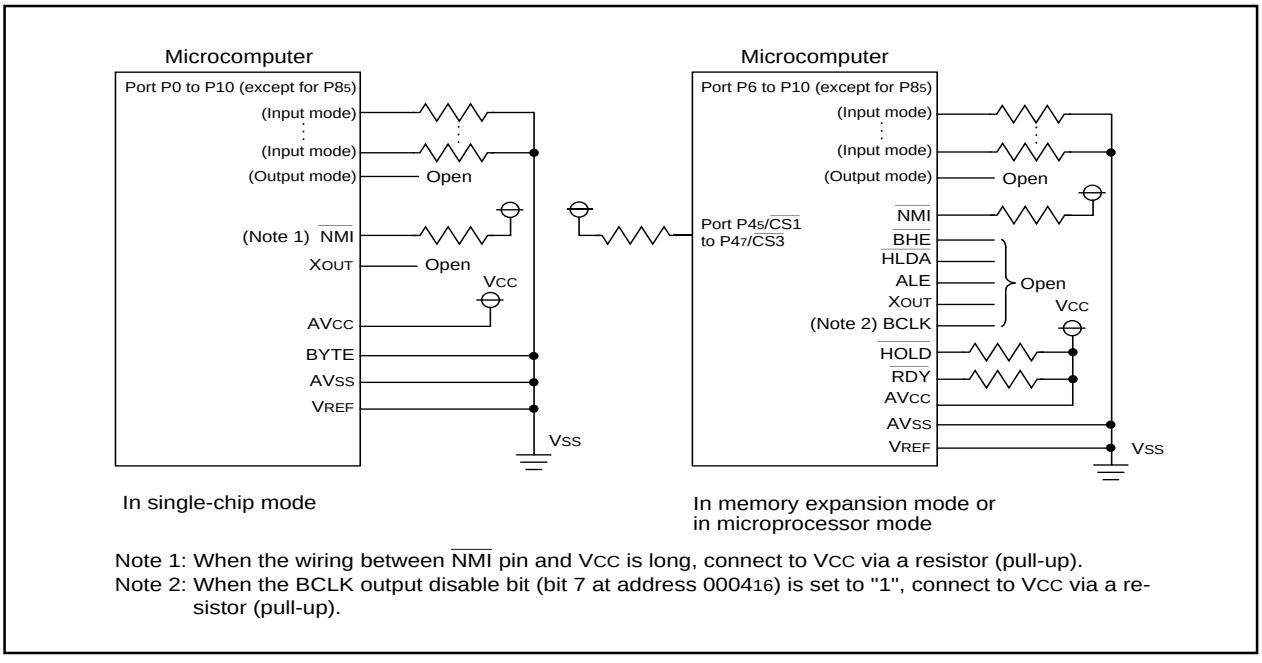
Pin name	Connection
Ports P0 to P10 (excluding P85)	After setting for input mode, connect every pin to VSS via a resistor (pull-down); or after setting for output mode, leave these pins open.
XOUT (Note)	Open
$\overline{NMI}$	Connect via a resistor to VCC (pull-up)
AVCC	Connect to VCC
AVSS, VREF, BYTE	Connect to VSS

Note: With external clock input to XIN pin.

Table 1.21.2. Example connection of unused pins in memory expansion mode and microprocessor mode

Pin name	Connection
Ports P6 to P10 (excluding P85)	After setting for input mode, connect every pin to VSS via a resistor (pull-down); or after setting for output mode, leave these pins open.
P45/$\overline{CS1}$ to P47/$\overline{CS3}$	Sets ports to input mode, set output enable bits of $\overline{CS1}$ through $\overline{CS3}$ to "0", and connect to VCC via resistors (pull-up).
$\overline{BHE}$, ALE, $\overline{HLDA}$, XOUT(Note 1), BCLK(Note 2)	Open
$\overline{HOLD}$, $\overline{RDY}$, $\overline{NMI}$	Connect via a resistor to VCC (pull-up)
AVCC	Connect to VCC
AVSS, VREF	Connect to VSS

Note 1: With external clock input to XIN pin.
Note 2: When the BCLK output disable bit (bit 7 at address 000416) is set to "1", connect to VCC via a resistor (pull-up).



Note 1: When the wiring between $\overline{NMI}$ pin and VCC is long, connect to VCC via a resistor (pull-up).
Note 2: When the BCLK output disable bit (bit 7 at address 000416) is set to "1", connect to VCC via a resistor (pull-up).

Figure 1.21.10. Example connection of unused pins

Table 1.22.1. Absolute maximum ratings

Symbol	Parameter		Condition	Rated value	Unit
V_{CC}	Supply voltage		V_{CC}=AV$_{CC}$	-0.3 to 6.5	V
AVcc	Analog supply voltage		V_{CC}=AV$_{CC}$	-0.3 to 6.5	V
V_I	Input voltage	$\overline{RESET}$, CNV$_{SS}$, BYTE, P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P2$_0$ to P2$_7$, P3$_0$ to P3$_7$, P4$_0$ to P4$_7$, P5$_0$ to P5$_7$, P6$_0$ to P6$_7$, P7$_0$, P7$_2$ to P7$_7$, P8$_0$ to P8$_7$, P9$_0$, P9$_2$ to P9$_7$, P10$_0$ to P10$_7$, V_{REF}, X_{IN}		-0.3 to Vcc+0.3	V
		P7$_1$, P9$_1$		-0.3 to 6.5	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$_7$, P5$_0$ to P5$_7$, P6$_0$ to P6$_7$, P7$_0$, P7$_2$ to P7$_7$, P8$_0$ to P8$_4$, P8$_6$, P8$_7$, P9$_0$, P9$_2$ to P9$_7$, P10$_0$ to P10$_7$, X_{OUT}		-0.3 to Vcc+0.3	V
		P7$_1$, P9$_1$,		-0.3 to 6.5	V
P_d	Power dissipation		Topr=25 ˚C	700	mW
T_{opr}	Operating ambient temperature			-40 to 125 (Note)	˚C
T_{stg}	Storage temperature			-65 to 150	˚C

Note: In case of 85 ˚C guaranted version, -40 to 85 ˚C.
In case of 125 ˚C guaranted version, -40 to 125 ˚C.

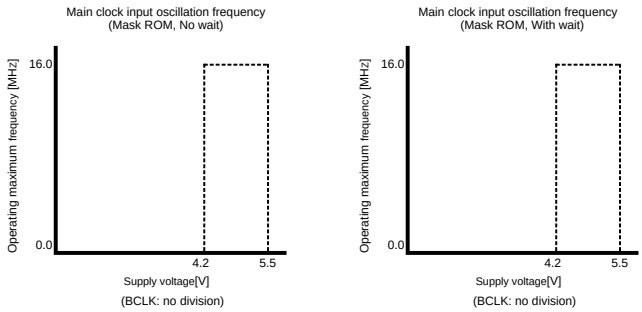
Table 1.22.2. Recommended operating conditions (referenced to Vcc = 4.2 V to 5.5 V at Topr = –40 to 125 °C unless otherwise specified)

Symbol	Parameter		Standard			Unit
			Min	Typ.	Max.	
Vcc	Supply voltage		4.2	5.0	5.5	V
AVcc	Analog supply voltage			Vcc		V
Vss	Supply voltage			0		V
AVss	Analog supply voltage			0		V
V_{IH}	HIGH input voltage	$P3_1$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$, $P7_2$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$, $P9_2$ to $P9_7$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	0.8Vcc		Vcc	V
		$P7_1$, $P9_1$	0.8Vcc		6.5	V
		$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ (during single-chip mode)	0.8Vcc		Vcc	V
		$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ (data input function during memory expansion and microprocessor modes)	0.5Vcc		Vcc	V
V_{IL}	LOW input voltage	$P3_1$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNV_{SS}, BYTE	0		0.2Vcc	V
		$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ (during single-chip mode)	0		0.2Vcc	V
		$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ (data input function during memory expansion and microprocessor modes)	0		0.16Vcc	V
$I_{OH (peak)}$	HIGH peak output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$, $P7_2$ to $P7_7$, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, $P9_0$, $P9_2$ to $P9_7$, $P10_0$ to $P10_7$			-10.0	mA
$I_{OH (avg)}$	HIGH average output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$, $P2_2$ to $P7_7$, $P8_0$ to $P8_4$, $P8_0$, $P8_7$, $P9_0$, $P9_2$ to $P3_7$, $P10_0$ to $P10_7$			-5.0	mA
$I_{OL (peak)}$	LOW peak output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$			10.0	mA
$I_{OL (avg)}$	LOW average output current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$			5.0	mA
f (X_{IN})	Main clock input oscillation frequency	No wait	0		16	MHz
		With wait	0		16	MHz
f (X_{CIN})	Subclock oscillation frequency			32.768	50	kHz

Note 1: The mean output current is the mean value within 100ms.

Note 2: The total I_{OL} (peak) for ports P0, P1, P2, $P8_6$, $P8_7$, P9, and P10 must be 80mA max. The total I_{OH} (peak) for ports P0, P1, P2, $P8_6$, $P8_7$, P9, and P10 must be 80mA max. The total I_{OL} (peak) for ports P3, P4, P5, P6, P7, and $P8_0$ to $P8_4$ must be 80mA max. The total I_{OH} (peak) for ports P3, P4, P5, P6, $P7_0$, $P7_2$ to $P7_7$, and $P8_0$ to $P8_4$ must be 80mA max.

Note 3: Relationship between main clock oscillation frequency and supply voltage.



Note 4: Execute case without wait, program/erase of flash memory by Vcc = 4.2V to 5.5V and f(BCLK) ≤ 6.25 MHz. Execute case with wait, program/erase of flash memory by Vcc = 4.2V to 5.5V and f(BCLK) ≤ 12.5 MHz.

Table 1.22.3. Electrical characteristics (referenced to Vcc = 5V, Vss = AVss = 0 V at Topr = –40 to 125 ˚C, f(XIN) = 16MHz 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 $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$, $P7_2$ to $P7_7$, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, $P9_0$, $P9_2$ to $P9_7$, $P10_0$ to $P10_7$	I_{OH}=-5mA	0.6Vcc			V
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 $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$, $P7_2$ to $P7_7$, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, $P9_0$, $P9_2$ to $P9_7$, $P10_0$ to $P10_7$	I_{OH}=-200µA	0.9Vcc			V
V_{OH}	HIGH output voltage	X_{OUT} HIGHPOWER	I_{OH}=-1mA	3.0			V
		X_{OUT} LOWPOWER	I_{OH}=-0.5mA	3.0			
	HIGH output voltage	X_{COUT} HIGHPOWER	With no load applied		3.0		V
		X_{COUT} LOWPOWER	With no load applied		1.6		
V_{OL}	LOW 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 $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$	I_{OL}=5mA			0.4Vcc	V
V_{OL}	LOW 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 $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$	I_{OL}=200µA			0.1Vcc	V
V_{OL}	LOW output voltage	X_{OUT} HIGHPOWER	I_{OL}=1mA			2.0	V
		X_{OUT} LOWPOWER	I_{OL}=0.5mA			2.0	
	LOW output voltage	X_{COUT} HIGHPOWER	With no load applied		0		V
		X_{COUT} LOWPOWER	With no load applied		0		
V_{T+} -V_{T-}	Hysteresis	$\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$ to $TA4_{IN}$, $TB0_{IN}$ to $TB5_{IN}$, $\overline{INT0}$ to $\overline{INT5}$, $\overline{ADTRG}$, $\overline{CTS0}$, $\overline{CTS1}$, $CLK0$ to $CLK3$, $TA2_{OUT}$ to $TA4_{OUT}$, $\overline{NMI}$, $\overline{KI0}$ to $\overline{KI3}$, $RXD0$ to $RXD2$, $SIN3$		0.2		0.8	V
V_{T+} -V_{T-}	Hysteresis	$\overline{RESET}$, CNV_{SS}, BYTE		0.2		1.8	V
I_{IH}	HIGH input current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNVss, BYTE	V_I=5V			5.0	µA
I_{IL}	LOW input current	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_7$, $P10_0$ to $P10_7$, X_{IN}, $\overline{RESET}$, CNVss, BYTE	V_I=0V			-5.0	µA
R_{PULLUP}	Pull-up resistance	$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_7$, $P4_0$ to $P4_7$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$, $P7_2$ to $P7_7$, $P8_0$ to $P8_4$, $P8_6$, $P8_7$, $P9_0$, $P9_2$ to $P9_7$, $P10_0$ to $P10_7$	V_I=0V	30.0	50.0	167.0	kΩ
R_{fXIN}	Feedback resistance X_{IN}				1.0		MΩ
R_{fCXIN}	Feedback resistance X_{CIN}				6.0		MΩ
V_{RAM}	RAM retention voltage		When clock is stopped	2.0			V

Table 1.22.4. Electrical characteristics (referenced to Vcc = AVcc = VREF = 5V, Vss = AVss = 0 V at Topr = –40 to 125 ˚C, f(XIN) = 16MHz unless otherwise specified)

Symbol	Parameter	Measuring condition			Standard			Unit
					Min.	Typ.	Max.	
Icc	Power supply current	In single-chip mode, the output pins are open and other pins are Vss.	f(XIN)=16MHz Square wave, no division	6NBMC		50	65	mA
				6NBFC		55	70	mA
				6NAMG		60	75	mA
				6NAMC		60	75	mA
				6NAFG		65	80	mA
			f(XCIN)=32kHz, square wave (Note)			200		µA
			f(XCIN)=32kHz, square wave, wait. Timer A operates with fc32.			4.0		µA
			Topr=25°C Clock is stopped.				1.0	µA
			Topr=85°C Clock is stopped.				20.0	µA
			Topr=125°C Clock is stopped.				50.0	µA

Note: For devices with flash memory the program resides in the internal RAM and FMR13 is set to "1".

Vcc = 5 V

Table 1.22.5. A-D conversion characteristics (referenced to Vcc = AVcc = VREF = 5V, Vss = AVss = 0 V at Topr = –40 to 125 ˚C, f(XIN) = 16MHz unless otherwise specified)

Symbol	Parameter		Measuring condition (Note1, 2, 3)		Standard Min.	Standard Typ.	Standard Max.	Unit
–	Resolution		VREF = VCC = 5V				10	Bits
–	Absolute accuracy (8bit)		VREF = AVCC = VCC = 5V, øAD ≤ 10MHz				±2	LSB
–	Absolute accuracy (10bit)	Sample & hold function disabled	VREF = AVCC = VCC = 5V, øAD ≤ 10MHz				±3	LSB
		Sample & hold function enabled	VREF = AVCC = VCC = 5V, øAD ≤ 10MHz	AN0 to AN7 input, AN00 to AN07 input, AN20 to AN27 input, ANEX0, ANEX1 input			±3	LSB
				External op-amp connection mode			±7	LSB
RLADDER	Ladder resistance		VREF = VCC = 5V		10		40	kΩ
tCONV	Conversion time (10bit)		f(XIN) = 16MHz, øAD = f2AD/2 = 8MHz (Note 4)		4.125			µs
			f(XIN) = 10MHz, øAD = f2AD = 10MHz (Note 4)		3.3			
tCONV	Conversion time (8bit)		f(XIN) = 16MHz, øAD = f2AD/2 = 8MHz (Note 4)		3.5			µs
			f(XIN) = 10MHz, øAD = f2AD = 10MHz (Note 4)		2.8			
tSAMP	Sampling time		f(XIN) = 16MHz, øAD = f2AD/2 = 8MHz (Note 4)		0.375			µs
			f(XIN) = 10MHz, øAD = f2AD = 10MHz (Note 4)		0.3			
VREF	Reference voltage				2		VCC	V
VIA	Analog input voltage				0		VREF	V

Note 1: Do f(XIN) in range of main clock input oscillation frequency prescribed with recommended operating conditions of table 1.22.2. Divide the f2AD if f(XIN) exceeds 10MHz, and make AD operation clock frequency (øAD) equal to or lower than 10MHz.

Note 2: A case without sample & hold function turn AD operation clock frequency (øAD) into 250kMHz or more in addition to a limit of Note 2.
A case with sample & hold function turn AD operation clock frequency (øAD) into 1MHz or more in addition to a limit of Note 2.

Note 3: Connect AVCC pin to VCC pin and apply the same potential.

Note 4: This applies when f2AD is selected no division mode and PCLK0 is "1".

Table 1.22.6. D-A conversion characteristics (referenced to Vcc = 5 V, VREF = 5V, Vss = AVss = 0 V at Topr = –40 to 125 ˚C, f(XIN) = 16MHz unless otherwise specified)

Symbol	Parameter	Measuring condition	Standard Min.	Standard Typ.	Standard Max.	Unit
–	Resolution				8	Bits
–	Absolute accuracy				1.0	%
tsu	Setup time				3	µs
RO	Output resistance		4	10	20	kΩ
IVREF	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, IVREF is passed.

Table 1.22.7. Flash memory version electrical characteristics (referenced to Vcc = 4.2 to 5.5V, Vss = AVss = 0 V at Topr = 0 to 60 ˚C, f(XIN) = 16MHz unless otherwise specified)

Parameter	Standard Min.	Standard Typ.	Standard Max.	Unit
Page program time		6	120	ms
Block erase time		50	600	ms
Erase all unlocked blocks time		50 X n (Note)	600 X n (Note)	ms
Lock bit program time		6	120	ms

Note: "n" denotes the number of block erase.

Vcc = 5 V

Timing requirements (Vcc = 5 V, Vss = 0 V at Topr = –40 to 125 ˚C unless otherwise specified)

Table 1.22.8. External clock input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	62.5		ns
$t_{w(H)}$	External clock input HIGH pulse width	25		ns
$t_{w(L)}$	External clock input LOW pulse width	25		ns
t_r	External clock rise time		15	ns
t_f	External clock fall time		15	ns

Table 1.22.9. Memory expansion- and microprocessor modes

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{ac1(RD-DB)}$	Data input access time (no wait)		(Note)	ns
$t_{ac2(RD-DB)}$	Data input access time (with wait)		(Note)	ns
$t_{ac3(RD-DB)}$	Data input access time (when accessing multiplex bus area)		(Note)	ns
$t_{su(DB-RD)}$	Data input setup time	40		ns
$t_{su(RDY-BCLK)}$	$\overline{RDY}$ input setup time	30		ns
$t_{su(HOLD-BCLK)}$	$\overline{HOLD}$ input setup time	40		ns
$t_{h(RD-DB)}$	Data input hold time	0		ns
$t_{h(BCLK-RDY)}$	$\overline{RDY}$ input hold time	0		ns
$t_{h(BCLK-HOLD)}$	$\overline{HOLD}$ input hold time	0		ns
$t_{d(BCLK-HLDA)}$	$\overline{HLDA}$ output delay time		40	ns

Note: Calculated according to the BCLK frequency as follows:

$$t_{ac1(RD-DB)} = \frac{10^9}{f(BCLK) \times 2} - 45 \quad [ns]$$

$$t_{ac2(RD-DB)} = \frac{3 \times 10^9}{f(BCLK) \times 2} - 45 \quad [ns]$$

$$t_{ac3(RD-DB)} = \frac{3 \times 10^9}{f(BCLK) \times 2} - 45 \quad [ns]$$

Vcc = 5 V

Timing requirements (referenced to Vcc = 5 V, Vss = 0 V at Topr = –40 to 125 °C unless otherwise specified)

Table 1.22.10. 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 1.22.11. 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 1.22.12. 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 1.22.13. 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 1.22.14. 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

Vcc = 5 V

Timing requirements (referenced to Vcc = 5 V, Vss = 0 V at Topr = –40 to 125 °C unless otherwise specified)

Table 1.22.15. Timer B input (counter input in event counter mode)

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

Table 1.22.16. Timer B input (pulse period measurement mode)

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

Table 1.22.17. Timer B input (pulse width measurement mode)

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

Table 1.22.18. A-D trigger input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{AD_{TRG}}$ input cycle time (trigger able minimum)	1000		ns
$t_{w(ADL)}$	$\overline{AD_{TRG}}$ input LOW pulse width	125		ns

Table 1.22.19. 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 1.22.20. 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

Vcc = 5 V

Switching characteristics (referenced to Vcc = 5 V, Vss = 0 V at Topr = –40 to 125 °C, CM15 ="1" unless otherwise specified)

Table 1.22.21. Memory expansion mode and microprocessor mode (no wait)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.22.1		25	ns
$t_{h(BCLK-AD)}$	Address output hold time (BCLK standard)		4		ns
$t_{h(RD-AD)}$	Address output hold time (RD standard)		0		ns
$t_{h(WR-AD)}$	Address output hold time (WR standard)		0		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			25	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (BCLK standard)		4		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			25	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		– 4		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			25	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			25	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (BCLK standard)			40	ns
$t_{h(BCLK-DB)}$	Data output hold time (BCLK standard)		4		ns
$t_{d(DB-WR)}$	Data output delay time (WR standard)		(Note1)		ns
$t_{h(WR-DB)}$	Data output hold time (WR standard)(Note2)		0		ns

Note 1: Calculated according to the BCLK frequency as follows:

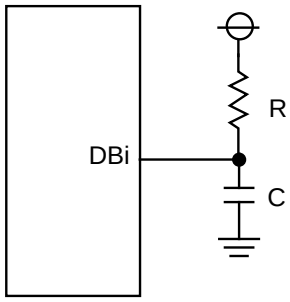
$$td(DB - WR) = \frac{10^9}{f(BCLK) \times 2} - 40 \quad [ns]$$

Note 2: This is standard value shows the timing when the output is off, and doesn't show hold time of data bus.
Hold time of data bus is different by capacitor volume and pull-up (pull-down) resistance value.
Hold time of data bus is expressed in

$$t = -CR \times \ln (1 - V_{OL} / V_{CC})$$

by a circuit of the right figure.
For example, when $V_{OL} = 0.2V_{CC}$, C = 30pF, R = 1kΩ, hold time of output "L" level is

$$t = -30pF \times 1k\Omega \times \ln (1 - 0.2V_{CC} / V_{CC})$$
$$= 6.7ns.$$



Vcc = 5 V

Switching characteristics (referenced to Vcc = 5 V, Vss = 0 V at Topr = –40 to 125 °C, CM15 ="1" unless otherwise specified)

Table 1.22.22. Memory expansion mode and microprocessor mode (with wait, accessing external memory)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.22.1		25	ns
$t_{h(BCLK-AD)}$	Address output hold time (BCLK standard)		4		ns
$t_{h(RD-AD)}$	Address output hold time (RD standard)		0		ns
$t_{h(WR-AD)}$	Address output hold time (WR standard)		0		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			25	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (BCLK standard)		4		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			25	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		– 4		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			25	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			25	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (BCLK standard)			40	ns
$t_{h(BCLK-DB)}$	Data output hold time (BCLK standard)		4		ns
$t_{d(DB-WR)}$	Data output delay time (WR standard)		(Note1)		ns
$t_{h(WR-DB)}$	Data output hold time (WR standard)(Note2)		0		ns

Note 1: Calculated according to the BCLK frequency as follows:

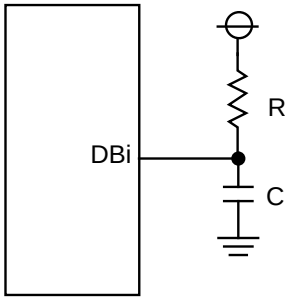
$$td(DB - WR) = \frac{10^9}{f(BCLK)} - 40 \quad [ns]$$

Note 2: This is standard value shows the timing when the output is off, and doesn't show hold time of data bus.
Hold time of data bus is different by capacitor volume and pull-up (pull-down) resistance value.
Hold time of data bus is expressed in

$$t = -CR \times \ln (1 - V_{OL} / V_{CC})$$

by a circuit of the right figure.
For example, when $V_{OL} = 0.2V_{CC}$, C = 30pF, R = 1kΩ, hold time of output "L" level is

$$t = -30pF \times 1k\Omega \times \ln (1 - 0.2V_{CC} / V_{CC})$$
$$= 6.7ns.$$



Vcc = 5 V

Switching characteristics (referenced to Vcc = 5 V, Vss = 0 V at Topr = –40 to 125 °C, CM15 ="1" unless otherwise specified)

Table 1.22.23. Memory expansion mode and microprocessor mode (with wait, accessing external memory, multiplex bus area selected)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.22.1		25	ns
$t_{h(BCLK-AD)}$	Address output hold time (BCLK standard)		4		ns
$t_{h(RD-AD)}$	Address output hold time (RD standard)		(Note)		ns
$t_{h(WR-AD)}$	Address output hold time (WR standard)		(Note)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			25	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (BCLK standard)		4		ns
$t_{h(RD-CS)}$	Chip select output hold time (RD standard)		(Note)		ns
$t_{h(WR-CS)}$	Chip select output hold time (WR standard)		(Note)		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			25	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		0		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			25	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(BCLK-DB)}$	Data output delay time (BCLK standard)			40	ns
$t_{h(BCLK-DB)}$	Data output hold time (BCLK standard)		4		ns
$t_{d(DB-WR)}$	Data output delay time (WR standard)		(Note)		ns
$t_{h(WR-DB)}$	Data output hold time (WR standard)		(Note)		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time (BCLK standard)			25	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time (BCLK standard)		– 4		ns
$t_{d(AD-ALE)}$	ALE signal output delay time (Address standard)		(Note)		ns
$t_{h(ALE-AD)}$	ALE signal output hold time (Adderss standard)		30		ns
$t_{d(AD-RD)}$	Post-address RD signal output delay time		0		ns
$t_{d(AD-WR)}$	Post-address WR signal output delay time		0		ns
$t_{dZ(RD-AD)}$	Address output floating start time			8	ns

Note: Calculated according to the BCLK frequency as follows:

$$th(RD - AD) = \frac{10^9}{f(BCLK) \times 2} \quad [ns]$$

$$th(WR - AD) = \frac{10^9}{f(BCLK) \times 2} \quad [ns]$$

$$th(RD - CS) = \frac{10^9}{f(BCLK) \times 2} \quad [ns]$$

$$th(WR - CS) = \frac{10^9}{f(BCLK) \times 2} \quad [ns]$$

$$td(DB - WR) = \frac{10^9 \times 3}{f(BCLK) \times 2} - 40 \quad [ns]$$

$$th(WR - DB) = \frac{10^9}{f(BCLK) \times 2} \quad [ns]$$

$$td(AD - ALE) = \frac{10^9}{f(BCLK) \times 2} - 25 \quad [ns]$$

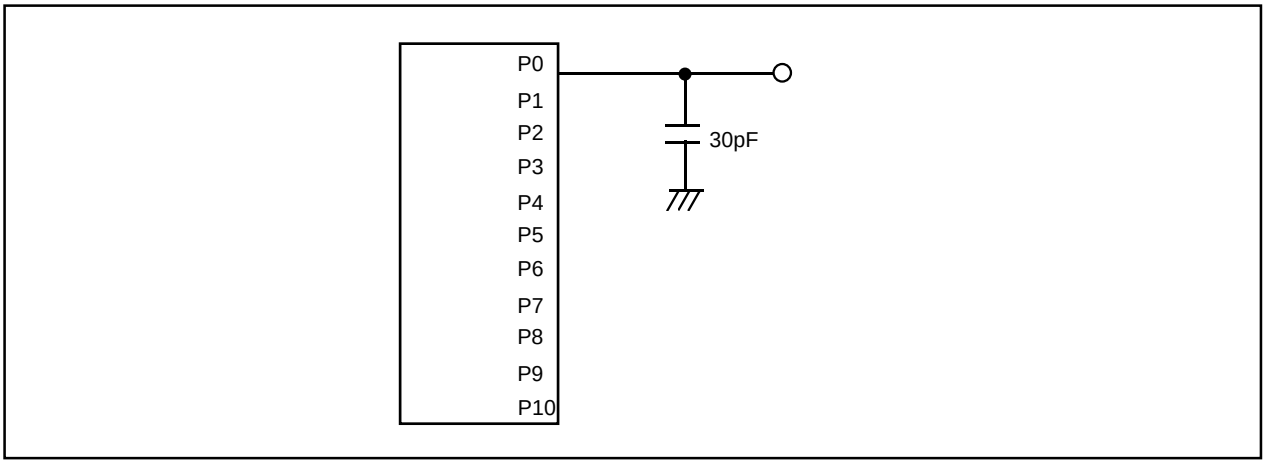


Figure 1.22.1. Port P0 to P10 measurement circuit

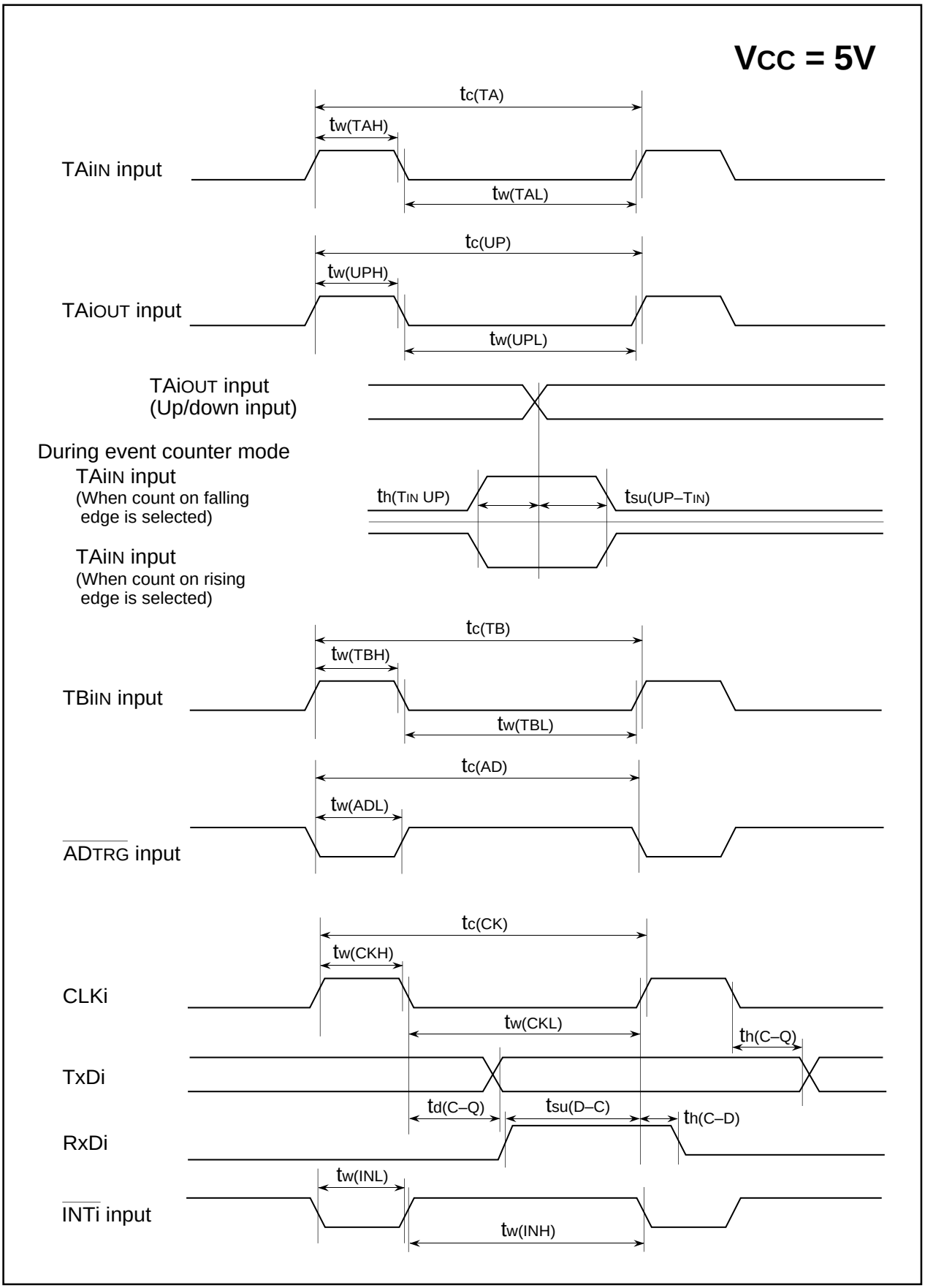
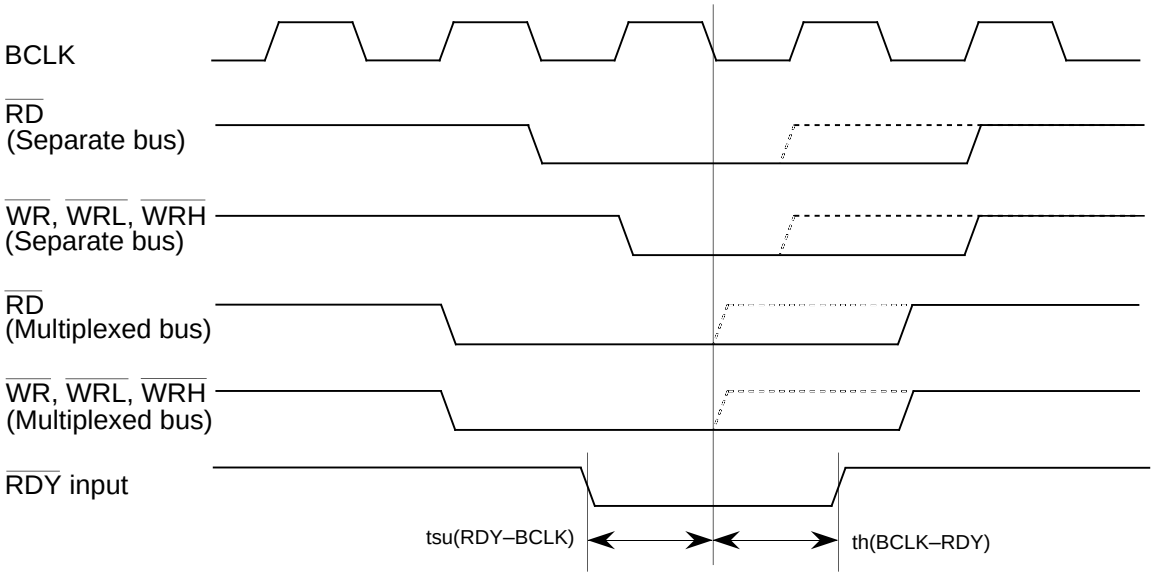


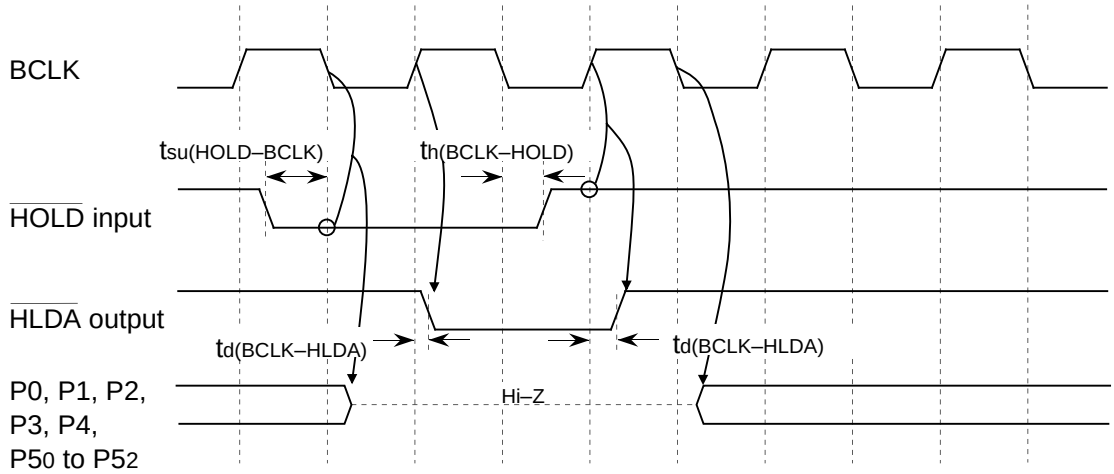
Figure 1.22.2. Vcc = 5V timing diagram (1)

Vcc = 5V

Memory expansion mode and microprocessor mode
(valid only with wait)



(Valid with or without wait)



Note: The above pins are set to high-impedance regardless of the input level of the BYTE pin and bit (PM06) of processor mode register 0 selects the function of ports P40 to P43.

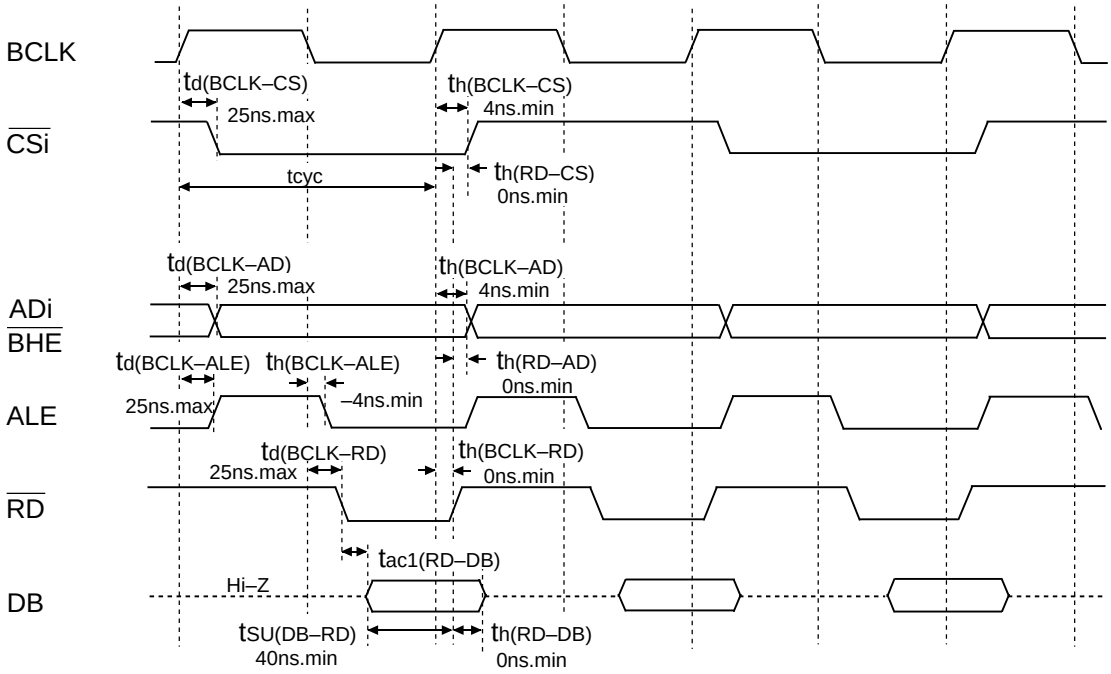
Measuring conditions :
- VCC=5V
- Input timing voltage : Determined with VIL=1.0V, VIH=4.0V
- Output timing voltage : Determined with VOL=2.5V, VOH=2.5V

Figure 1.22.3. Vcc = 5V timing diagram (2)

Memory expansion mode and microprocessor mode (with no wait)

Vcc = 5V

Read timing



Write timing

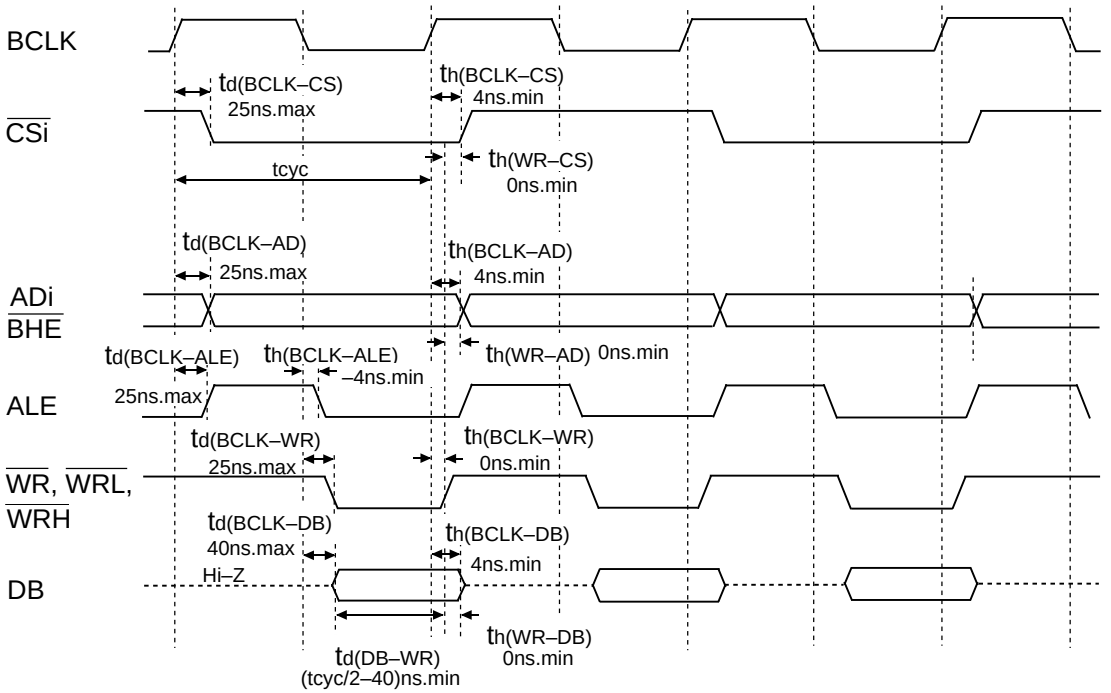
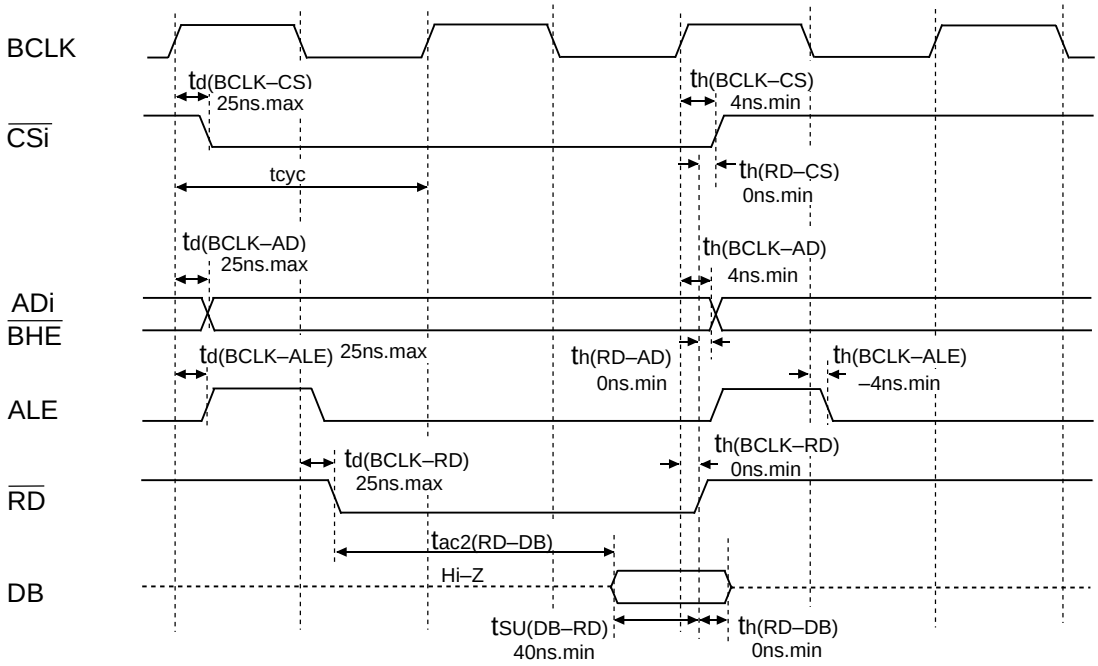


Figure 1.22.4. Vcc = 5V timing diagram (3)

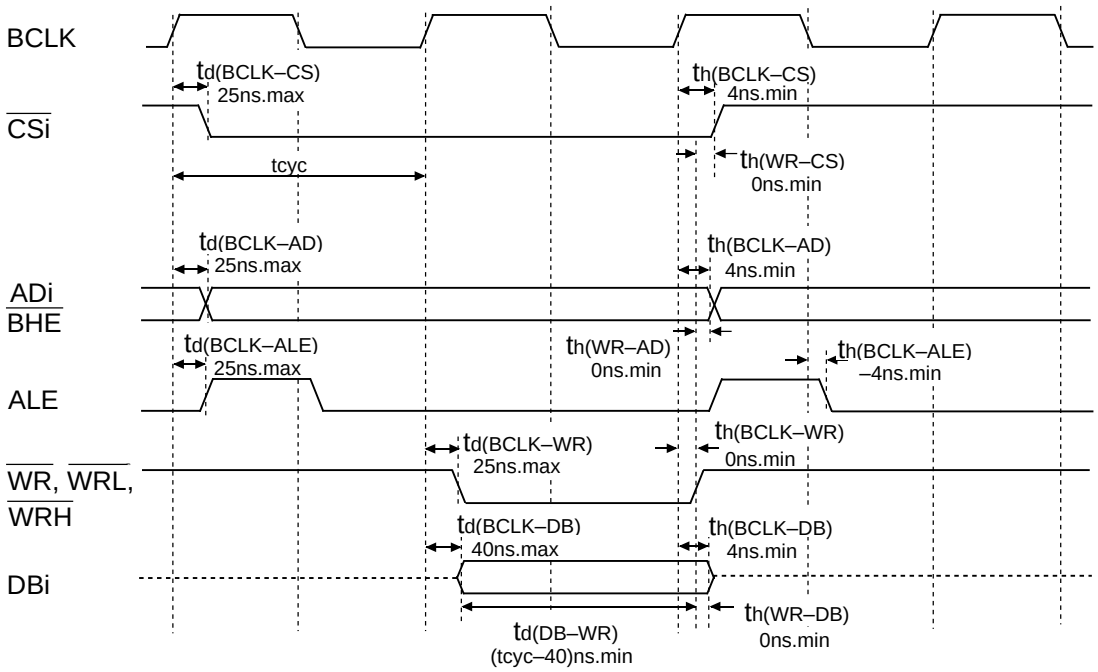
Vcc = 5V

Memory expansion mode and microprocessor mode
(when accessing external memory area with wait)

Read timing



Write timing



Measuring conditions :
- V_{CC}=5V
- Input timing voltage : Determined with: V_{IL}=0.8V, V_{IH}=2.5V
- Output timing voltage : Determined with: V_{OL}=0.8V, V_{OH}=2.0V

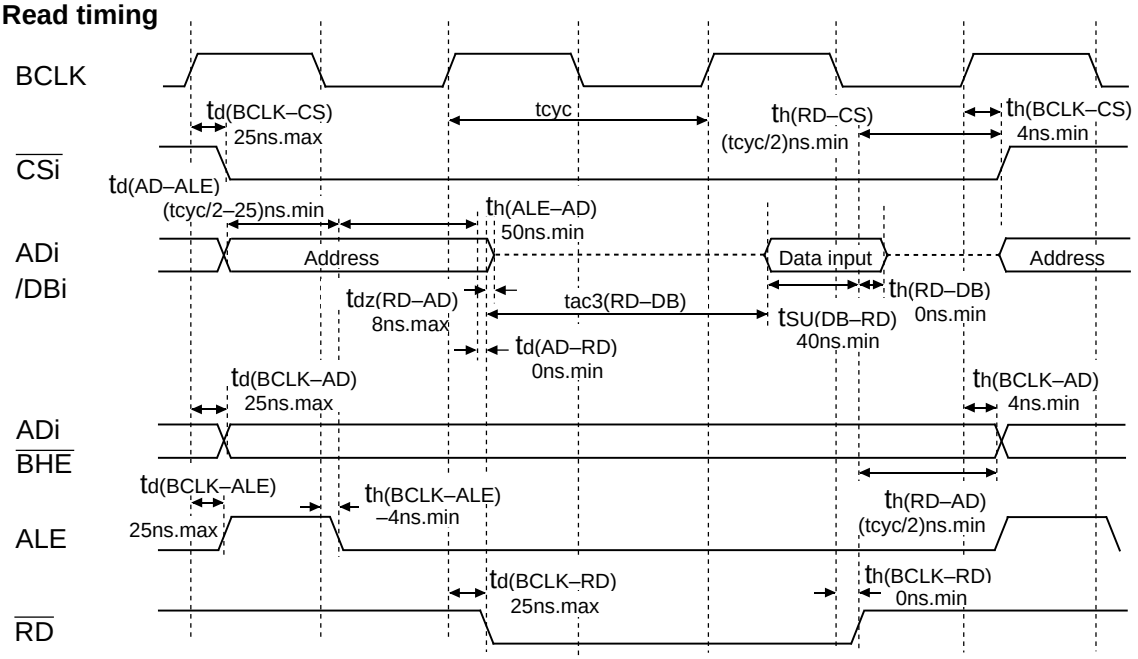
Figure 1.22.5. Vcc = 5V timing diagram (4)

Memory expansion mode and microprocessor mode

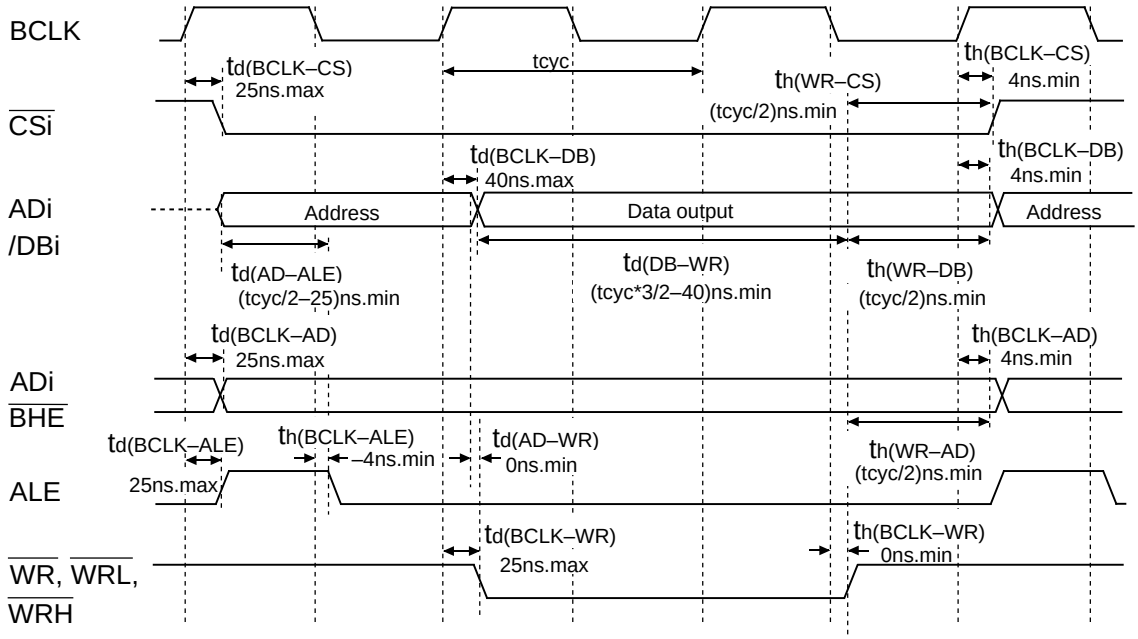
Vcc = 5V

(when accessing external memory area with wait, and select multiplexed bus)

Read timing



Write timing



Measuring conditions :

- Vcc=5V
- Input timing voltage : Determined with VIL=0.8V, VIH=2.5V
- Output timing voltage : Determined with VOL=0.8V, VOH=2.0V

Figure 1.22.6. Vcc = 5V timing diagram (5)

Outline Performance

Table 1.23.1 shows the outline performance of the M16C/6N group (flash memory version).

Table 1.23.1. Outline performance of the M16C/6N group (flash memory version)

Item		Performance
Flash memory operation mode		Four modes (parallel I/O, standard serial I/O, CPU rewrite and CAN I/O)
Erase block division	User ROM area	See Figure 1.23.1
	Boot ROM area	One division (8 Kbytes) (Note)
Program method		In units of pages (in units of 256 bytes)
Erase method		Collective erase/block erase
Program/erase control method		Program/erase control by software command
Protect method		Protected for each block by lock bit
Number of commands		8 commands
Program/erase count		100 times
Data retention		10 years
ROM code protect		Parallel I/O, standard serial I/O and CAN I/O modes are supported.

Note: The boot ROM area contains a standard serial I/O and CAN I/O modes 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 M16C/6N group (flash memory version) contains the flash memory that can be rewritten with a single voltage. For this flash memory, four flash memory modes are available in which to read, program, and erase: parallel I/O, standard serial I/O and CAN 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.

The flash memory is divided into several blocks as shown in Figure 1.23.1, so that memory can be erased one block at a time. Each block has a lock bit to enable or disable execution of an erase or program operation, allowing for data in each block to be protected.

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, standard serial I/O and CAN I/O modes. This boot ROM area has had a standard serial I/O and CAN I/O modes 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.

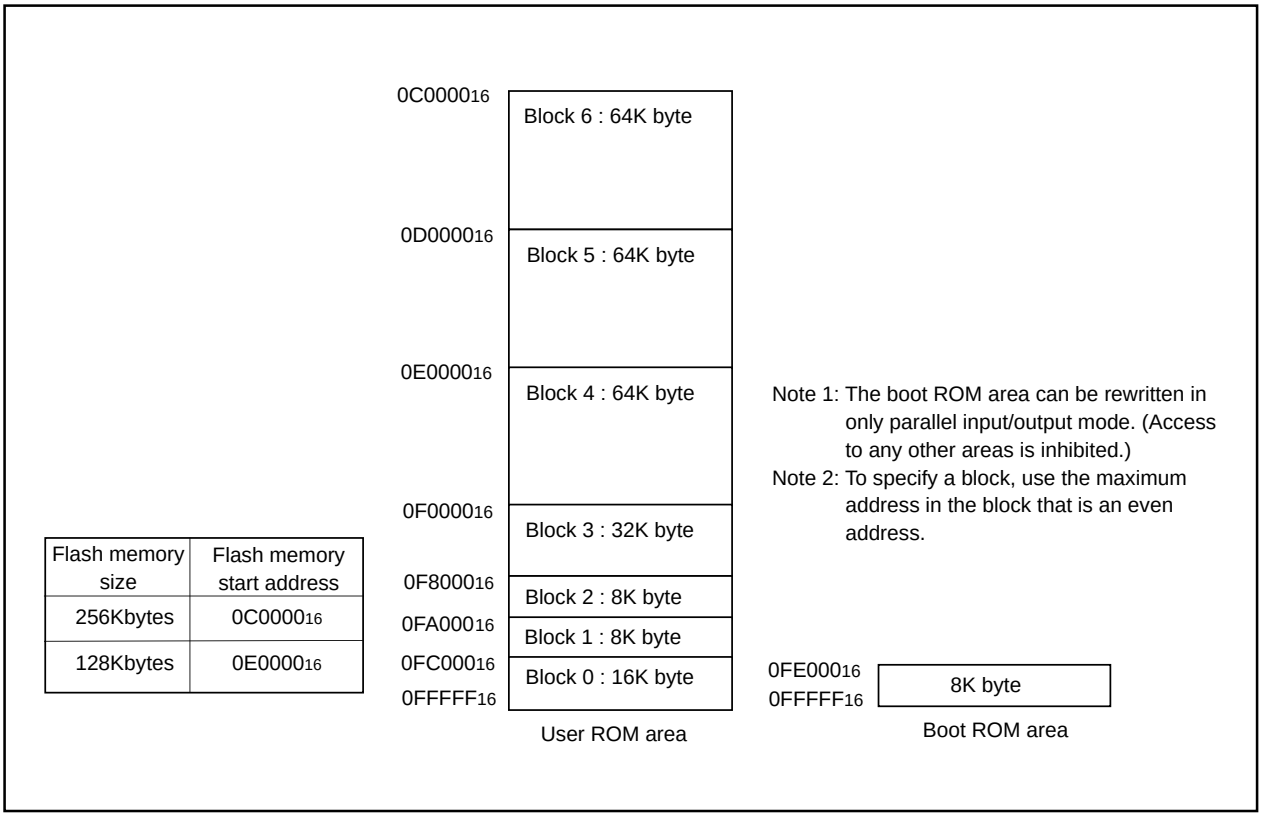


Figure 1.23.1. Block diagram of flash memory version

CPU Rewrite Mode

In CPU rewrite mode, the internal flash memory can be operated on (read, program, or erase) under control of the central processing unit (CPU).

In CPU rewrite mode, only the user ROM area shown in Figure 1.23.1 can be rewritten, the boot ROM area cannot be rewritten. Make sure the program and block erase commands are issued for only the user ROM area and each block 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 any area other than the internal flash memory before it can be executed.

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 and CAN I/O modes become unusable.)

See Figure 1.23.1 for details about the boot ROM area.

Normal microcomputer mode is entered when the microcomputer is reset with pulling CNV_{SS} pin low. In this case, the CPU starts operating using the control program in the user ROM area.

When the microcomputer is reset by pulling the $P5_5$ pin low, the CNV_{SS} pin high, and the $P5_0$ pin high, 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.

Block Address

Block addresses refer to the maximum even address of each block. These addresses are used in the block erase command, lock bit program command, and read lock status command.

Outline Performance (CPU Rewrite Mode)

In the CPU rewrite mode, the CPU erases, programs and reads the internal flash memory as instructed by software commands. Operations must be executed from a memory other than the internal flash memory, such as the internal RAM.

When the CPU rewrite mode select bit (bit 1 at address $03B7_{16}$) is set to "1", transition to CPU rewrite mode occurs and software commands can be accepted.

In the CPU rewrite mode, write to and read from software commands and data into even numbered address ("0" for byte address A_0) in 16 bit units. Always write 8 bit software commands into even numbered address. Commands are ignored with odd numbered addresses.

Use software commands to control program and erase operations. Whether a program or erase operation has terminated normally or in error can be verified by reading the status register.

Figure 1.24.1 shows the flash memory control register 0 and the flash memory control register 1.

Bit 0 of the flash memory control register 0 is the RY/$\overline{BY}$ status flag used exclusively to read the operating status of the flash memory. During programming and erase operations, it is "0". Otherwise, it is "1".

Bit 1 of the flash memory control register 0 is the CPU rewrite mode select bit. The CPU rewrite mode is entered by setting this bit to "1", so that software commands become acceptable. In CPU rewrite mode, the CPU becomes unable to access the internal flash memory directly. Therefore, write bit 1 in an area other than the internal flash memory. Also only when $\overline{NMI}$ pin is "H" level. To set this bit to "1", it is necessary to write "0" and then write "1" in succession. The bit can be set to "0" by only writing a "0".

Bit 2 of the flash memory control register 0 is a lock bit disable select bit. By setting this bit to "1", it is possible to disable erase and write protect (block lock) effectuated by the lock bit data. The lock bit disable select bit only disables the lock bit function; it does not change the lock data bit value. However, if an erase operation is performed when this bit is "1", the lock bit data that is "0" (locked) is set to "1" (unlocked) after erasure. To set this bit to "1", it is necessary to write "0" and then write "1" in succession. This bit can be manipulated only when the CPU rewrite mode select bit is "1".

Bit 3 of the flash memory control register 0 is the flash memory reset bit used to reset the control circuit of the internal flash memory. This bit is used when exiting CPU rewrite mode and when flash memory access has failed. When the CPU rewrite mode select bit is "1", writing "1" for this bit resets the control circuit. To release the reset, it is necessary to set this bit to "0".

Bit 5 of the flash memory control register 0 is a user ROM area select bit which is effective in only boot mode. If this bit is set to "1" in boot mode, the area to be accessed is switched from the boot ROM area to the user ROM area. When the CPU rewrite mode needs to be used in boot mode, set this bit to "1". Note that if the microcomputer is booted from the user ROM area, it is always the user ROM area that can be accessed and this bit has no effect. When in boot mode, the function of this bit is effective regardless of whether the CPU rewrite mode is on or off. Write to this bit only when executing out of an area other than the internal flash memory.

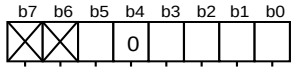
Bit 3 of the flash memory control register 1 turns power supply to the internal flash memory on/off. When this bit is set to "1", power is not supplied to the internal flash memory, thus power dissipation can be reduced. However, in this state, the internal flash memory cannot be accessed. To set this bit to "1", it is necessary to write "0" and then write "1" in succession. Use this bit mainly in the low speed mode (when X_{CIN} is the count source of BCLK).

When the CPU is shifted to the stop or wait modes, power to the internal flash memory is automatically shut off. It is reconnected automatically when CPU operation is restored. Therefore, it is not particularly necessary to set flash memory control register 1.

CPU Rewrite Mode (Flash Memory Version)

Figure 1.24.2 shows a flowchart for setting/releasing the CPU rewrite mode. Figure 1.24.3 shows a flowchart for shifting to the low speed mode. Always perform operation as indicated in these flowcharts.

Flash memory control register 0

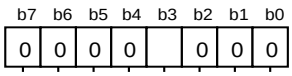


Symbol	Address	When reset
FMR0	$03B7_{16}$	$XX000001_2$

Bit symbol	Bit name	Function	R	W
FMR00	RY/$\overline{BY}$ status flag	0: Busy (being written or erased) 1: Ready	○	×
FMR01	CPU rewrite mode select bit (Note 1)	0: Normal mode (Software commands invalid) 1: CPU rewrite mode (Software commands acceptable)	○	○
FMR02	Lock bit disable select bit (Note 2)	0: Block lock by lock bit data is enabled 1: Block lock by lock bit data is disabled	○	○
FMR03	Flash memory reset bit (Note 3)	0: Normal operation 1: Reset	○	○
Reserved bit		Must always be set to "0"	○	○
FMR05	User ROM area select bit (Note 4) (Effective in only boot mode)	0: Boot ROM area is accessed 1: User ROM area is accessed	○	○
Nothing is assigned. When write, set "0". When read, values are indeterminate.			—	—

Note 1: For this bit to be set to "1", the user needs to write a "0" and then a "1" to it in succession. When it is not this procedure, it is not enacted in "1". This is necessary to ensure that no interrupt or DMA transfer will be executed during the interval. Write to this bit only when executing out of an area other than the internal flash memory. Also only when NMI pin is "H" level. Clear this bit to "0" after read array command.
Note 2: For this bit to be set to "1", the user needs to write a "0" and then a "1" to it in succession when the CPU rewrite mode select bit = "1". When it is not this procedure, it is not enacted in "1". This is necessary to ensure that no interrupt or DMA transfer will be executed during the interval.
Note 3: Effective only when the CPU rewrite mode select bit = "1". Set this bit to "0" subsequently after setting it to "1" (reset).
Note 4: Write to this bit only when executing out of an area other than the internal flash memory.

Flash memory control register 1



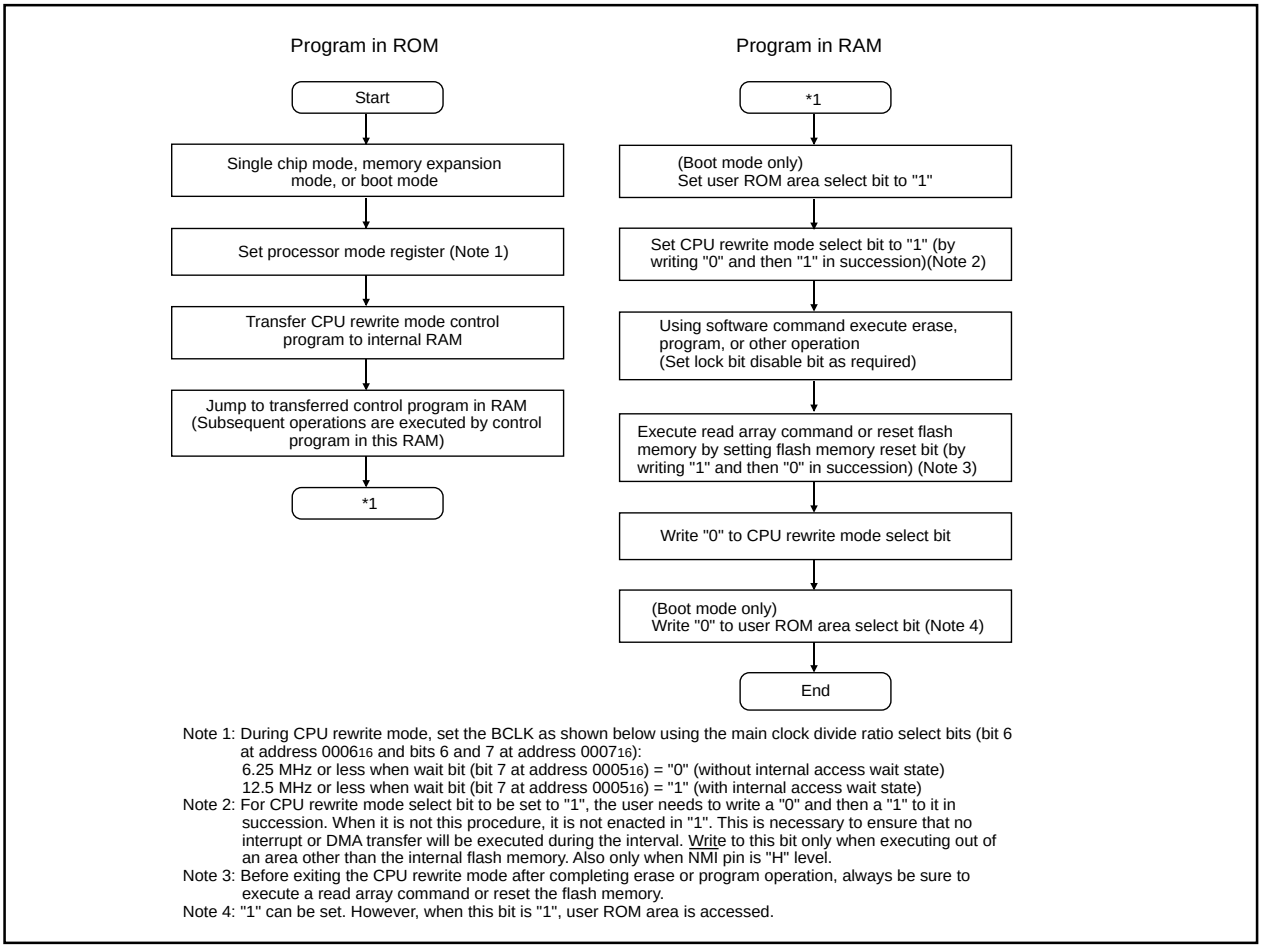
Symbol	Address	When reset
FMR1	$03B6_{16}$	$XXXX0XXX_2$

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to "0"	—	○
FMR13	Flash memory power supply off bit (Note)	0: Flash memory power supply is connected 1: Flash memory power supply off	○	○
Reserved bits		Must always be set to "0"	—	○

Note : For this bit to be set to "1", the user needs to write a "0" and then a "1" to it in succession. When it is not this procedure, it is not enacted in "1". This is necessary to ensure that no interrupt or DMA transfer will be executed during the interval. During parallel I/O mode, programming, erase or read of flash memory is not controlled by this bit,only by external pins. Write to this bit only when executing out of an area other than the internal flash memory.

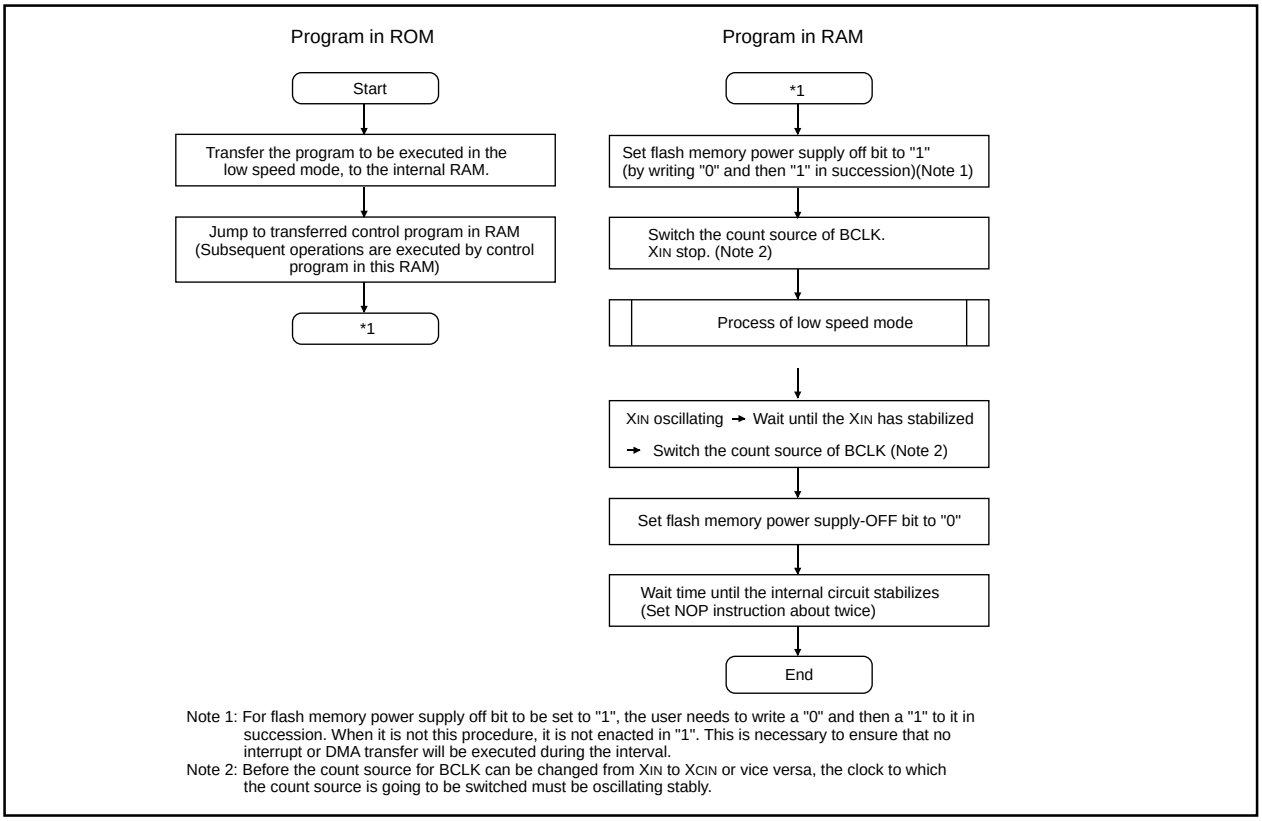
Figure 1.24.1. Flash memory control registers

CPU Rewrite Mode (Flash Memory Version)



Note 1: During CPU rewrite mode, set the BCLK as shown below using the main clock divide ratio select bits (bit 6 at address 0006_{16} and bits 6 and 7 at address 0007_{16}):
6.25 MHz or less when wait bit (bit 7 at address 0005_{16}) = "0" (without internal access wait state)
12.5 MHz or less when wait bit (bit 7 at address 0005_{16}) = "1" (with internal access wait state)
Note 2: For CPU rewrite mode select bit to be set to "1", the user needs to write a "0" and then a "1" to it in succession. When it is not this procedure, it is not enacted in "1". This is necessary to ensure that no interrupt or DMA transfer will be executed during the interval. Write to this bit only when executing out of an area other than the internal flash memory. Also only when NMI pin is "H" level.
Note 3: Before exiting the CPU rewrite mode after completing erase or program operation, always be sure to execute a read array command or reset the flash memory.
Note 4: "1" can be set. However, when this bit is "1", user ROM area is accessed.

Figure 1.24.2. CPU rewrite mode set/reset flowchart



Note 1: For flash memory power supply off bit to be set to "1", the user needs to write a "0" and then a "1" to it in succession. When it is not this procedure, it is not enacted in "1". This is necessary to ensure that no interrupt or DMA transfer will be executed during the interval.
Note 2: Before the count source for BCLK can be changed from XIN to XCIN or vice versa, the clock to which the count source is going to be switched must be oscillating stably.

Figure 1.24.3. Shifting to the low speed mode flowchart

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 CPU rewrite mode, set the BCLK as shown below using the main clock divide ratio select bit (bit 6 at address 0006_{16} and bits 6 and 7 at address 0007_{16}):

6.25 MHz or less when wait bit (bit 7 at address 0005_{16}) = "0" (without internal access wait state)

12.5 MHz or less when wait bit (bit 7 at address 0005_{16}) = "1" (with internal access wait state)

(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

The address match interrupt cannot be used during CPU rewrite mode because they refer to the internal data of the flash memory. If interrupts have their vector in the variable vector table, they can be used by transferring the vector into the RAM area. The $\overline{\text{NMI}}$ and watchdog timer interrupts can be used because the flash memory conterol register 0 and 1 is forcibly initialized and return to normal mode when each interrupt occurs. But it is needed that the jump addresses for each interrupt are set in the fixed vector table and there is an interrupt program. Since the rewrite operation is halted when the $\overline{\text{NMI}}$ and watchdog timer interrupts occur, it is needed that CPU rewriting mode select bit is set to "1" and the erase/program operation is performed over again.

(4) Internal reserved area expansion bit (bit 3 at address 0005_{16})

The reserved area of the internal memory can be changed by using the internal reserved area expansion bit (bit 3 at address 0005_{16}). However, if the CPU rewrite mode select bit (bit 1 at address $03B7_{16}$) is set to "1", the internal reserved area expansion bit (bit 3 at address 0005_{16}) also is set to "1" automatically. Similarly, if the CPU rewrite mode select bit (bit 1 at address $03B7_{16}$) is set to "0", the internal reserved area expansion bit (bit 3 at address 0005_{16}) also is set to "0" automatically.

The precautions above apply to the products which flash memory size is over 192 Kbytes.

(5) Reset

Reset input is always accepted. After a reset, the addresses $0C0000_{16}$ through $0CFFFF_{16}$ are made a reserved area and cannot be accessed. Therefore, if your product has this area in the user ROM area, do not write any address of this area to the reset vector. This area is made accessible by changing the internal reserved area expansion bit (bit 3 at address 0005_{16}) in a program.

(6) Access disable

Write CPU rewrite mode select bit, flash memory power supply off bit and user ROM area select bit only when executing out of an area other than the internal flash memory.

(7) How to access

For CPU rewrite mode select bit, lock bit disable select bit, and flash memory power supply off bit to be set to "1", the user needs to write a "0" and then a "1" to it in succession. When it is not this procedure, it is not enacted in "1". This is necessary to ensure that no interrupt or DMA transfer will be executed during the interval.

Write CPU rewrite mode select bit only when executing out of an area other than the internal flash memory. Also only when $\overline{\text{NMI}}$ pin is "H" level.

(8) Writing in the user ROM area

If power is lost while rewriting blocks that contain the flash rewrite program with the CPU rewrite mode, those blocks may not be correctly rewritten and it is possible that the flash memory can no longer be rewritten after that. Therefore, it is recommended to use the standard serial I/O mode, CAN I/O mode or parallel I/O mode to rewrite these blocks.

(9) Using the lock bit

To use the CPU rewrite mode, use a boot program that can set and cancel the lock command.

Software Commands

Table 1.24.1 lists the software commands available with the M16C/6N group (flash memory version). After setting the CPU rewrite mode select bit to "1", write a software command to specify an erase or program operation. Note that when entering a software command, the upper byte (D_8 to D_{15}) is ignored. The content of each software command is explained below.

Table 1.24.1. List of software commands (CPU rewrite mode)

Command	1st bus cycle			2nd bus cycle			3rd bus cycle		
	Mode	Address	Data (D_0 to D_7)	Mode	Address	Data (D_0 to D_7)	Mode	Address	Data (D_0 to D_7)
Read array	Write	X (Note 6)	FF_{16}						
Read status register	Write	X	70_{16}	Read	X	SRD (Note 2)			
Clear status register	Write	X	50_{16}						
Page program (Note 3)	Write	X	41_{16}	Write	WA0 (Note 3)	WD0 (Note 3)	Write	WA1	WD1
Block erase	Write	X	20_{16}	Write	BA (Note 4)	$D0_{16}$			
Erase all unlocked blocks	Write	X	$A7_{16}$	Write	X	$D0_{16}$			
Lock bit program	Write	X	77_{16}	Write	BA (Note 4)	$D0_{16}$			
Read lock bit status	Write	X	71_{16}	Read	BA (Note 4)	D_6 (Note 5)			

Note 1: When a software command is input, the high order byte of data (D_8 to D_{15}) is ignored.
Note 2: SRD = Status Register Data
Note 3: WA = Write Address, WD = Write Data
WA and WD must be set sequentially from 00_{16} to FE_{16} (byte address; however, an even address). The page size is 256 bytes.
Note 4: BA = Block Address (Enter the maximum address of each block that is an even address.)
Note 5: D_6 corresponds to the block lock status. Block not locked when D_6 = "1", block locked when D_6 = "0".
Note 6: X denotes a given address in the user ROM area (that is an even address).

Read Array Command (FF_{16})

The read array mode is entered by writing the command code "FF_{16}" in the 1st bus cycle. When an even 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 to D_{15}), 16 bits at a time.

The read array mode is retained intact until another command is written.

Read Status Register Command (70_{16})

When the command code "70_{16}" is written in the 1st bus cycle, the content of the status register is read out at the data bus (D_0 to D_7) by a read in the 2nd bus cycle.

The status register is explained in the next section.

Clear Status Register Command (50_{16})

This command is used to clear the bits SR3 to 5 of the status register after they have been set. These bits indicate that operation has ended in an error. To use this command, write the command code "50_{16}" in the 1st bus cycle.

Page Program Command (41$_{16}$)

Page program allows for high speed programming in units of 256 bytes. Page program operation starts when the command code "41$_{16}$" is written in the 1st bus cycle. In the 2nd bus cycle through the 129th bus cycle, the write data is sequentially written 16 bits at a time. At this time, the addresses A$_0$ to A$_7$ need to be incremented by 2 from "00$_{16}$" to "FE$_{16}$." When the system finishes loading the data, it starts an auto write operation (data program and verify operation).

Whether the auto write operation is completed can be confirmed by reading the status register or the flash memory control register 0. At the same time the auto write operation starts, the read status register mode is automatically entered, so the content of the status register can be read out. The status register bit 7 (SR7) is set to "0" at the same time the auto write operation starts and is returned to "1" upon completion of the auto write operation. In this case, the read status register mode remains active until the Read Array command (FF$_{16}$) or Read Lock Bit Status command (71$_{16}$) is written or the flash memory is reset using its reset bit.

The RY/$\overline{BY}$ status flag of the flash memory control register 0 is "0" during auto write operation and "1" when the auto write operation is completed as is the status register bit 7.

After the auto write operation is completed, the status register can be read out to know the result of the auto write operation. For details, refer to the section where the status register is detailed.

Figure 1.24.4 shows an example of a page program flowchart.

Each block of the flash memory can be write protected by using a lock bit. For details, refer to the section where the data protect function is detailed.

Additional writes to the already programmed pages are prohibited.

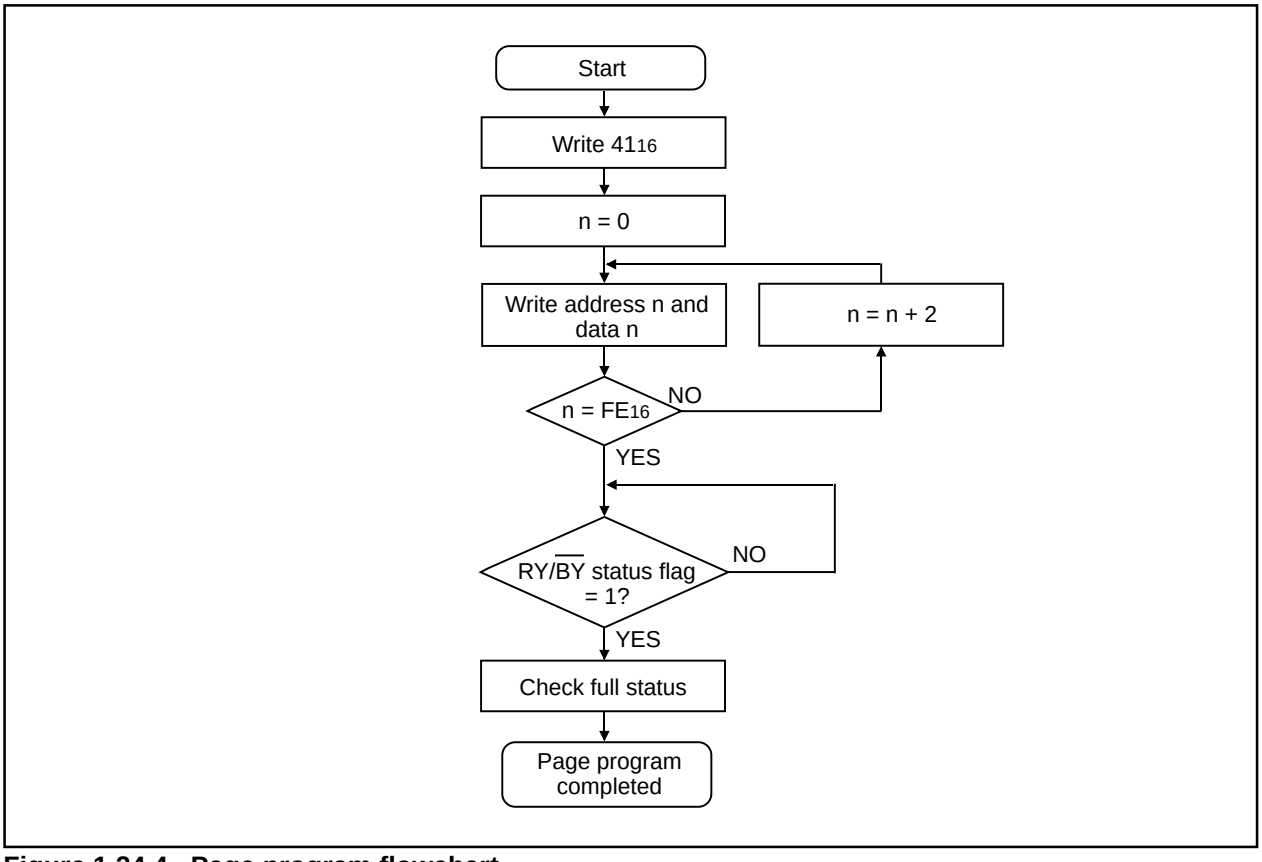


Figure 1.24.4. Page program flowchart

Block Erase Command (20_{16}/$D0_{16}$)

By writing the command code "20_{16}" in the 1st bus cycle and the confirmation command code "$D0_{16}$" in the 2nd bus cycle that follows to the block address of a flash memory block, the system initiates an auto erase (erase and erase verify) operation.

Whether the auto erase operation is completed can be confirmed by reading the status register or the flash memory control register 0. At the same time the auto erase operation starts, the read status register mode is automatically entered, so the content of the status register can be read out. The status register bit 7 (SR7) is set to "0" at the same time the auto erase operation starts and is returned to "1" upon completion of the auto erase operation. In this case, the read status register mode remains active until the read array command (FF_{16}) or the read lock bit status command (71_{16}) is written or the flash memory is reset using its reset bit.

The RY/$\overline{BY}$ status flag of the flash memory control register 0 is "0" during auto erase operation and "1" when the auto erase operation is completed as is the status register bit 7.

After the auto erase operation is completed, the status register can be read out to know the result of the auto erase operation. For details, refer to the section where the status register is detailed.

Figure 1.24.5 shows an example of a block erase flowchart.

Each block of the flash memory can be protected against erasure by using a lock bit. For details, refer to the section where the data protect function is detailed.

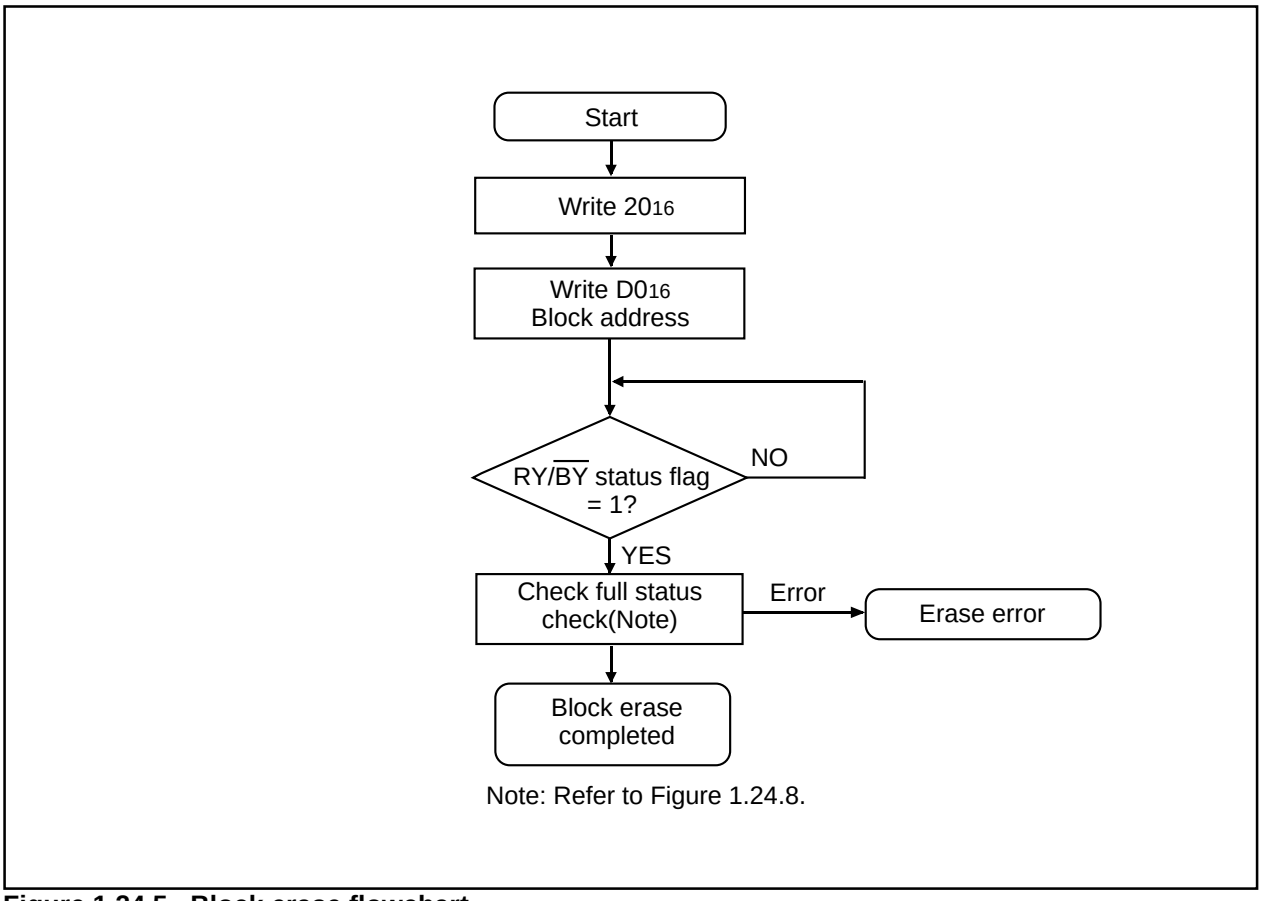


Figure 1.24.5. Block erase flowchart

Erase All Unlocked Blocks Command (A7$_{16}$/D0$_{16}$)

By writing the command code "A7$_{16}$" in the 1st bus cycle and the confirmation command code "D0$_{16}$" in the 2nd bus cycle that follows, the system starts erasing blocks successively.

Whether the erase all unlock blocks command is terminated can be confirmed by reading the status register or the flash memory control register 0, in the same way as for block erase. Also, the status register can be read out to know the result of the auto erase operation.

When the lock bit disable select bit of the flash memory control register 0 is "1", all blocks are erased no matter how the lock bit is set. On the other hand, when the lock bit disable select bit is "0", the function of the lock bit is effective and only nonlocked blocks (where lock bit data is "1") are erased.

Lock Bit Program Command (77$_{16}$/D0$_{16}$)

By writing the command code "77$_{16}$" in the 1st bus cycle and the confirmation command code "D0$_{16}$" in the 2nd bus cycle that follows to the block address of a flash memory block, the system sets the lock bit for the specified block to "0" (locked).

Figure 1.24.6 shows an example of a lock bit program flowchart. The status of the lock bit (lock bit data) can be read out by a read lock bit status command.

Whether the lock bit program command is terminated can be confirmed by reading the status register or the flash memory control register 0, in the same way as for page program.

For details about the function of the lock bit and how to reset the lock bit, refer to the section where the data protect function is detailed.

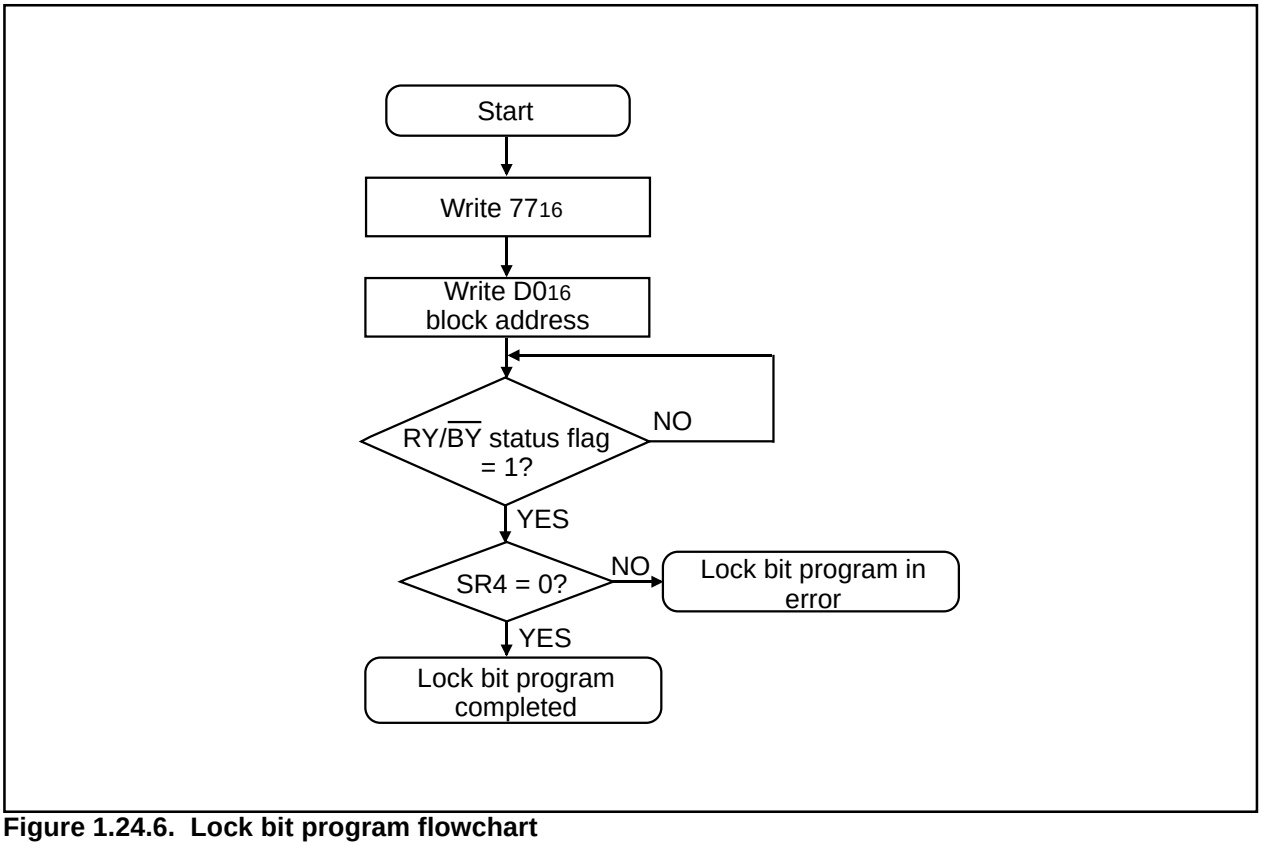


Figure 1.24.6. Lock bit program flowchart

Read Lock Bit Status Command (71_{16})

By writing the command code "71_{16}" in the 1st bus cycle and then reading the block address of a flash memory block in the 2nd bus cycle that follows, the system reads out the status of the lock bit of the specified block on to the data bus (D_6).

Figure 1.24.7 shows an example of a read lock bit program flowchart.

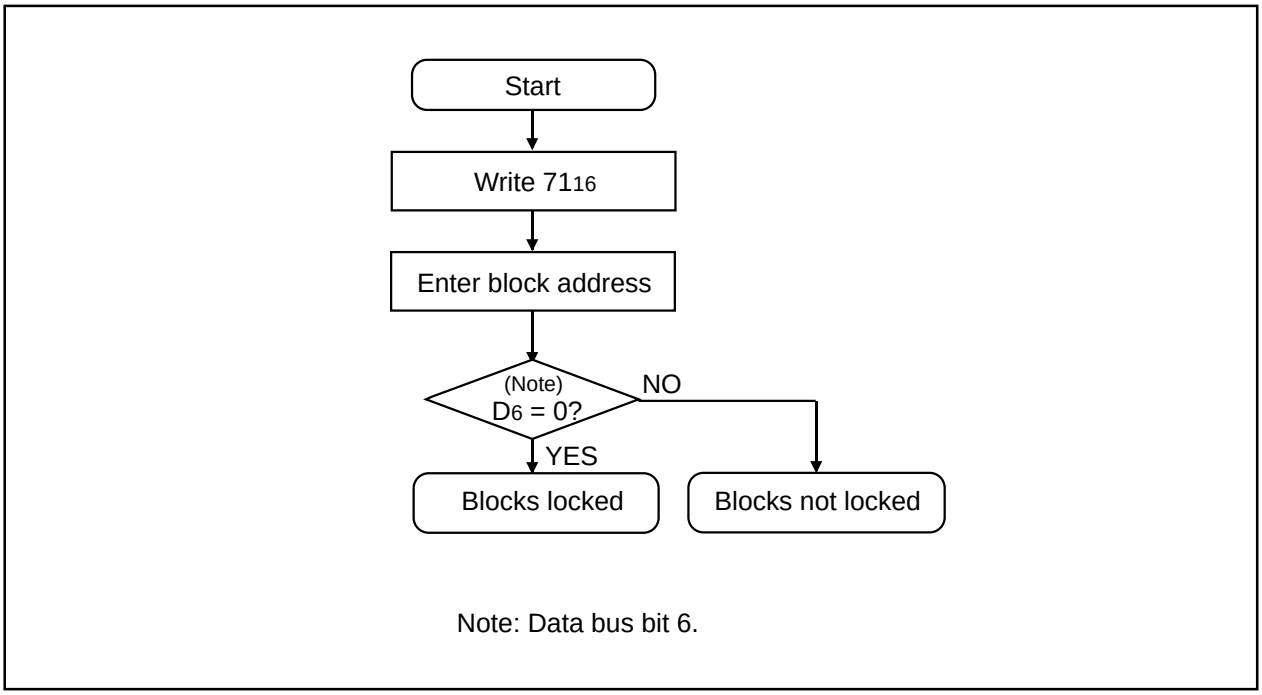


Figure 1.24.7. Read lock bit status flowchart

Data Protect Function (Block Lock)

Each block in Figure 1.23.1 has a nonvolatile lock bit to specify that the block be protected (locked) against erase/write. The lock bit program command is used to set the lock bit to "0" (locked). The lock bit of each block can be read out using the read lock bit status command.

Whether block lock is enabled or disabled is determined by the status of the lock bit and how the flash memory control register 0's lock bit disable select bit is set.

(1) When the lock bit disable select bit is "0", a specified block can be locked or unlocked by the lock bit status (lock bit data). Blocks whose lock bit data is "0" are locked, so they are disabled against erase/write. On the other hand, the blocks whose lock bit data is "1" are not locked, so they are enabled for erase/write.

(2) When the lock bit disable select bit is "1", all blocks are nonlocked regardless of the lock bit data, so they are enabled for erase/write. In this case, the lock bit data that is "0" (locked) is set to "1" (nonlocked) after erasure, so that the lock bit actuated lock is removed.

Status Register

The status register indicates the operating status of the flash memory and whether an erase or program operation has terminated normally or in an error. The content of this register can be read out by only writing the read status register command (70_{16}). Table 1.24.2 details the status register.

The status register is cleared by writing the clear status register command (50_{16}).

After a reset, the status register is set to "80_{16}."

Each bit in this register is explained below.

Write State Machine (WSM) Status (SR7)

After power on, the write state machine (WSM) status is set to "1".

The write state machine (WSM) status indicates the operating status of the device, as for output on the RY/$\overline{BY}$ pin. This status bit is set to "0" during auto write or auto erase operation and is set to "1" upon completion of these operations.

Erase Status (SR5)

The erase status informs the operating status of auto erase operation to the CPU. When an erase error occurs, it is set to "1".

The erase status is reset to "0" when cleared.

Program Status (SR4)

The program status informs the operating status of auto write operation to the CPU. When a write error occurs, it is set to "1".

The program status is reset to "0" when cleared.

When an erase command is in error (which occurs if the command entered after the block erase command (20_{16}) is not the confirmation command ($D0_{16}$), both the program status and erase status (SR5) are set to "1".

When the program status or erase status is "1", only the following flash commands will be accepted: Read Array, Read Status Register, and Clear Status Register.

Also, in one of the following cases, both SR4 and SR5 are set to "1" (command sequence error):

(1) When the valid command is not entered correctly

(2) When the data entered in the 2nd bus cycle of lock bit program (77_{16}/$D0_{16}$), block erase (20_{16}/$D0_{16}$), or erase all unlock blocks ($A7_{16}$/$D0_{16}$) is not the $D0_{16}$ or FF_{16}. However, if FF_{16} is entered, read array is assumed and the command that has been set up in the 1st bus cycle is canceled.

Block Status After Program (SR3)

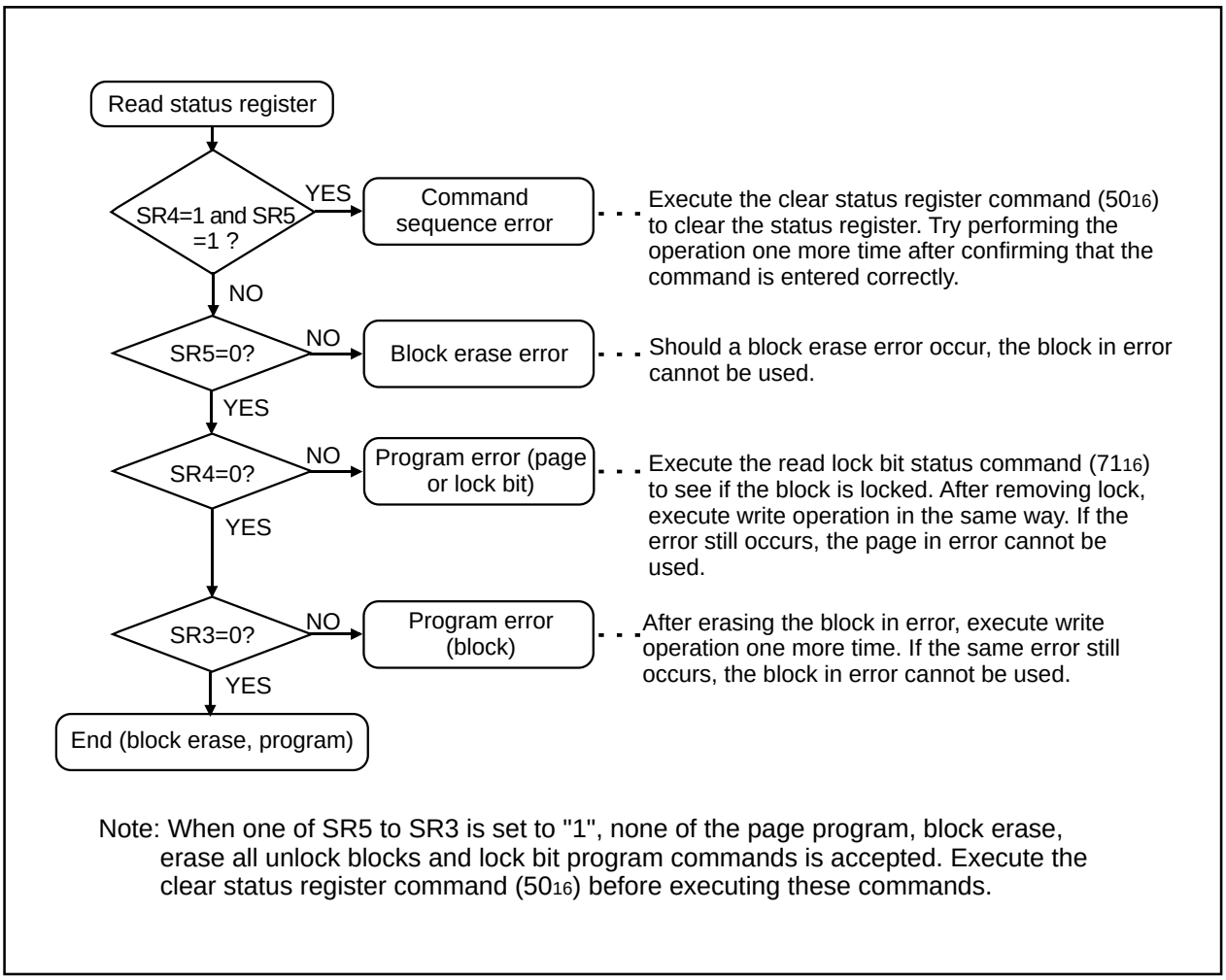
If excessive data is written (phenomenon whereby the memory cell becomes depressed which results in data not being read correctly), "1" is set for the program status after program at the end of the page write operation. In other words, when writing ends successfully, "80_{16}" is output; when writing fails, "90_{16}" is output; and when excessive data is written, "88_{16}" is output.

Table 1.24.2. Definition of each bit in status register

Each bit of SRD	Status name	Definition	
		"1"	"0"
SR7 (bit7)	Write state machine (WSM) status	Ready	Busy
SR6 (bit6)	Reserved	-	-
SR5 (bit5)	Erase status	Terminated in error	Terminated normally
SR4 (bit4)	Program status	Terminated in error	Terminated normally
SR3 (bit3)	Block status after program	Terminated in error	Terminated normally
SR2 (bit2)	Reserved	-	-
SR1 (bit1)	Reserved	-	-
SR0 (bit0)	Reserved	-	-

Full Status Check

By performing full status check, it is possible to know the execution results of erase and program operations. Figure 1.24.8 shows a full status check flowchart and the action to be taken when each error occurs.



Note: When one of SR5 to SR3 is set to "1", none of the page program, block erase, erase all unlock blocks and lock bit program commands is accepted. Execute the clear status register command (50_{16}) before executing these commands.

Figure 1.24.8. Full status check flowchart and remedial procedure for errors

Functions to Inhibit Rewriting Flash Memory Version

To prevent the contents of the flash memory version from being read out or rewritten easily, the device incorporates a ROM code protect function for use in parallel I/O mode and an ID code check function for use in standard serial I/O mode.

ROM Code Protect Function

The ROM code protect function is used to prohibit reading out or modifying the contents of the flash memory during parallel I/O mode and is set by using the ROM code protect control address register ($0FFFFF_{16}$). Figure 1.25.1 shows the ROM code protect control address ($0FFFFF_{16}$). (This address exists in the user ROM area.)

If one of the pair of ROM code protect bits is set to 0, ROM code protect is turned on, so that the contents of the flash memory version are protected against readout and modification. ROM code protect is implemented in two levels. If level 2 is selected, the flash memory is protected even against readout by a shipment inspection LSI tester, etc. When an attempt is made to select both level 1 and level 2, level 2 is selected by default.

If both of the two ROM code protect reset bits are set to "00", ROM code protect is turned off, so that the contents of the flash memory version can be read out or modified. Once ROM code protect is turned on, the contents of the ROM code protect reset bits cannot be modified in parallel I/O mode. Use the serial I/O or some other mode to rewrite the contents of the ROM code protect reset bits.

ROM code protect control address

b7	b6	b5	b4	b3	b2	b1	b0
						1	1

Symbol: ROMCP
Address: $0FFFFF_{16}$
When reset: FF_{16}

Bit symbol	Bit name	Function
Reserved bits		Always set this bit to "1".
ROMCP2	ROM code protect level 2 set bit (Note 1, 2)	b3 b2 00: Protect enabled 01: Protect enabled 10: Protect enabled 11: Protect disabled
ROMCR	ROM code protect reset bit (Note 3)	b5 b4 00: Protect removed 01: Protect set bit effective 10: Protect set bit effective 11: Protect set bit effective
ROMCP1	ROM code protect level 1 set bit (Note 1)	b7 b6 00: Protect enabled 01: Protect enabled 10: Protect enabled 11: Protect disabled

Note 1: When ROM code protect is turned on, the internal memory is protected against readout or modification in parallel input/output mode.
Note 2: When ROM code protect level 2 is turned on, ROM code readout by a shipment inspection LSI tester, etc. also is inhibited.
Note 3: The ROM code protect reset bits can be used to turn off ROM code protect level 1 and ROM code protect level 2. However, since these bits cannot be changed in parallel input/output mode, they need to be rewritten in serial input/output or some other mode.

Figure 1.25.1. ROM code protect control address

ID Code Check Function

Use this function in standard serial I/O and CAN I/O modes. When the contents of the flash memory are not blank, the ID code sent from the peripheral unit is compared with the ID code written in the flash memory to see if they match. If the ID codes do not match, the commands sent from the peripheral unit are not accepted. The ID code consists of 8 bit data, the areas of which, beginning with the 1st byte, are 0FFFDF16, 0FFFE316, 0FFFEB16, 0FFFEF16, 0FFFF316, 0FFFF716, and 0FFFFB16. Write a program which has had the ID code preset at these addresses to the flash memory.

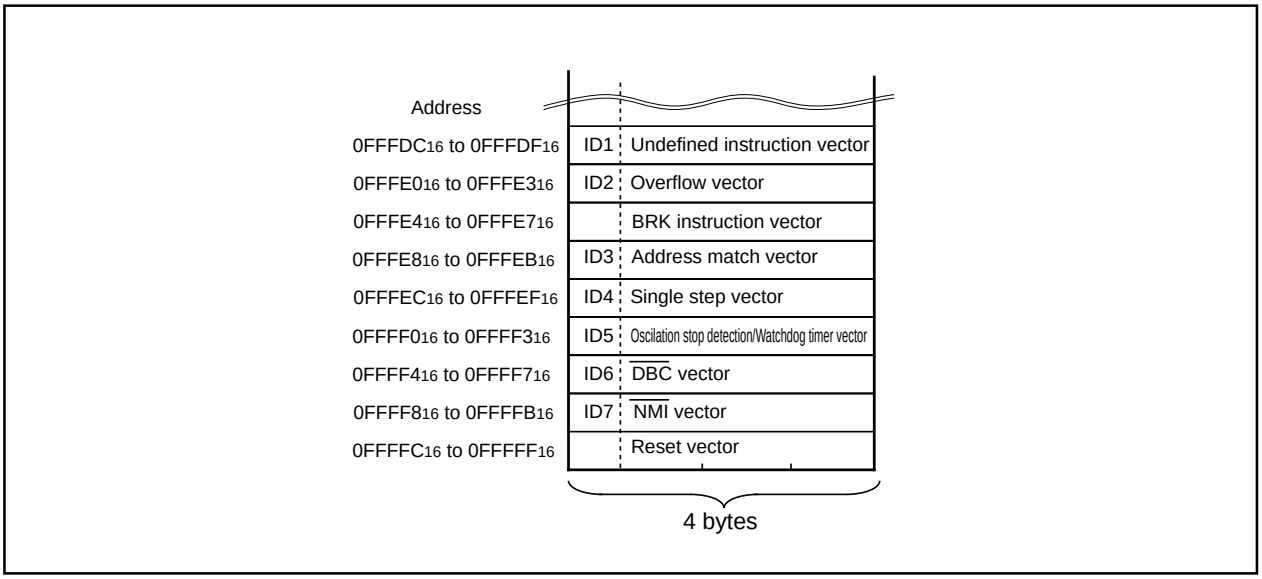


Figure 1.25.2. ID code store addresses

Parallel I/O Mode

The parallel I/O mode inputs and outputs the software commands, addresses and data needed to operate (read, program, erase, etc.) the internal flash memory. This I/O is parallel.
Use an exclusive programmer supporting M16C/6N group (flash memory version).
Refer to the instruction manual of each programer maker for the datails of use.

User ROM and Boot Rom Areas

In parallel I/O mode, the user ROM and boot ROM areas shown in Figure 1.23.1 can be rewritten. Both areas of flash memory can be operated on in the same way.
Program and block erase operations can be performed in the user ROM area. The user ROM area and its blocks are shown in Figure 1.23.1.
The boot ROM area is 8 Kbytes in size. In parallel I/O mode, it is located at addresses $0FE000_{16}$ through $0FFFFF_{16}$. Make sure program and block erase operations are always performed within this address range. (Access to any location outside this address range is prohibited.)
In the boot ROM area, an erase block operation is applied to only one 8 Kbyte block. The boot ROM area has had a standard serial I/O and CAN I/O modes. control program stored in it when shipped from the Mitsubishi factory. Therefore, using the device in standard serial input/output and CAN I/O modes, you do not need to write to the boot ROM area.

Pin Functions (Flash memory standard serial I/O mode)

Pin	Name	I/O	Description
V_{CC},V_{SS}	Power input		Apply program/erase protection voltage to V_{CC} pin and 0 V to V_{SS} pin.
CNV_{SS}	CNV_{SS}	I	Connect to V_{CC} pin.
$\overline{RESET}$	Reset input	I	Reset input pin. While reset is "L" level, more than 20cycles of clock must be input to X_{IN} pin.
X_{IN}	Clock input	I	Connect a ceramic resonator or a quartz 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	
BYTE	BYTE	I	Connect to V_{SS} or V_{CC}.
AV_{CC}, AV_{SS}	Analog power supply input		Connect AV_{SS} to V_{SS} and AV_{CC} to V_{CC}, respectively.
V_{REF}	Reference voltage input	I	Reference voltage input pin for AD converter.
$P0_0$ to $P0_7$	Input port P0	I	Input "H" or "L" level or open.
$P1_0$ to $P1_7$	Input port P1	I	Input "H" or "L" level or open.
$P2_0$ to $P2_7$	Input port P2	I	Input "H" or "L" level or open.
$P3_0$ to $P3_7$	Input port P3	I	Input "H" or "L" level or open.
$P4_0$ to $P4_7$	Input port P4	I	Input "H" or "L" level or open.
$P5_1$ to $P5_4$, $P5_6$, $P5_7$	Input port P5	I	Input "H" or "L" level or open.
$P5_0$	$\overline{CE}$ input	I	Input "H" level.
$P5_5$	$\overline{EPM}$ input	I	Input "L" level.
$P6_0$ to $P6_3$	Input port P6	I	Input "H" or "L" level or open.
$P6_4$	BUSY output	O	Standard serial I/O mode 1: BUSY signal output pin. Standard serial I/O mode 2: Monitors the boot program operation. check signal output pin.
$P6_5$	SCLK input	I	Standard serial I/O mode 1: Serial clock input pin. Standard serial I/O mode 2: Input "L" level.
$P6_6$	RxD input	I	Serial data input pin.
$P6_7$	TxD output	O	Serial data output pin.
$P7_0$ to $P7_7$	Input port P7	I	Input "H" or "L" level or open.
$P8_0$ to $P8_4$, $P8_6$, $P8_7$	Input port P8	I	Input "H" or "L" level or open.
$P8_5$	$\overline{NMI}$ input	I	Connect to V_{CC}.
$P9_0$ to $P9_4$, $P9_7$	Input port P9	I	Input "H" or "L" level or open.
$P9_5$	CAN input	I	Connect to a CAN transceiver or input "H" or "L" level.
$P9_6$	CAN output	O	Connect to a CAN transceiver, connect to V_{CC} via a resister or open.
$P10_0$ to $P10_7$	Input port P10	I	Input "H" or "L" level or open.

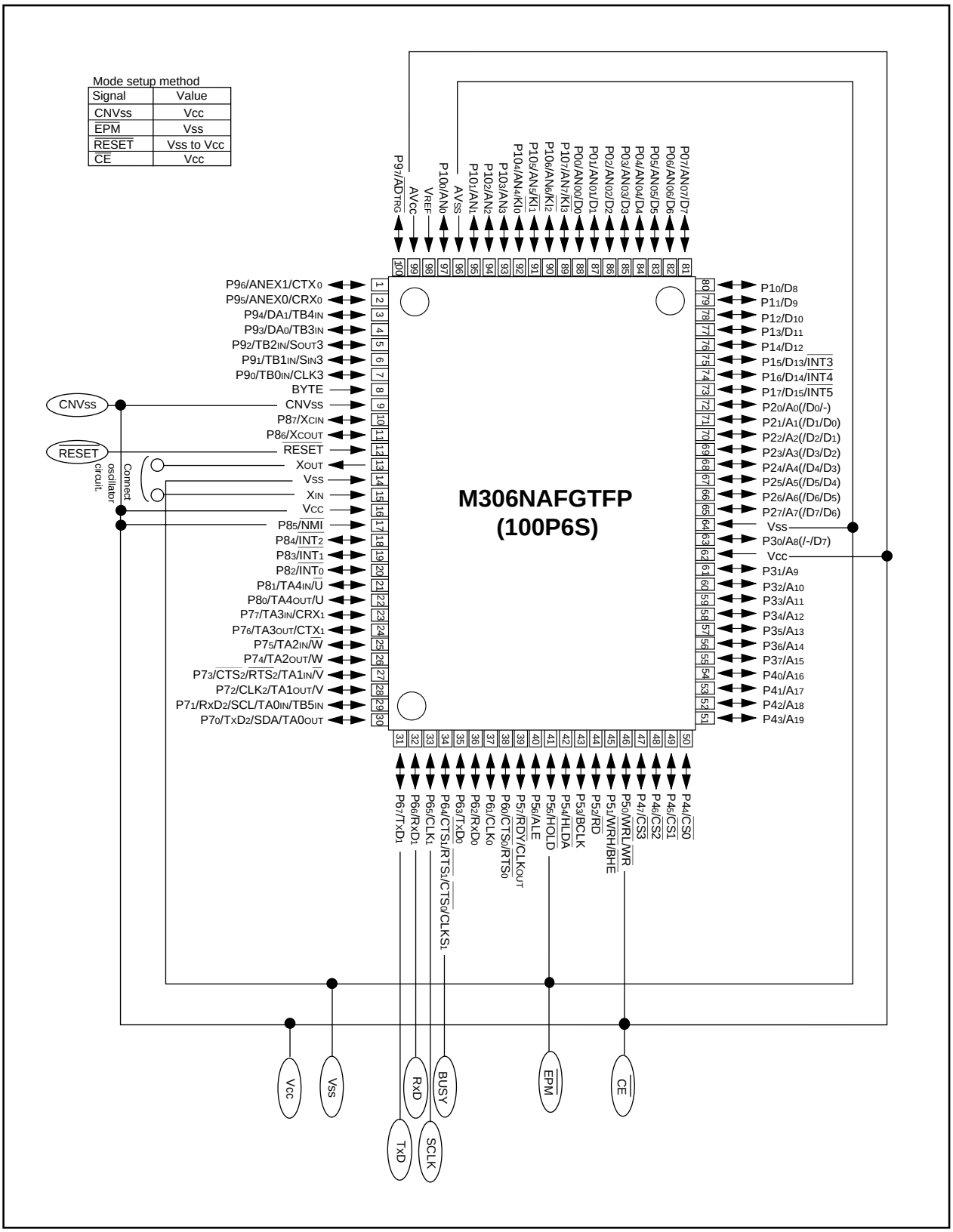


Figure 1.27.1. Pin connections for serial I/O mode

Standard Serial I/O Mode

The standard serial I/O mode inputs and outputs the software commands, addresses and data needed to operate (read, program, erase, etc.) the internal flash memory. This I/O is serial. There are actually two standard serial I/O modes: mode 1, which is clock synchronized, and mode 2, which is asynchronized. Both modes require a purpose specific peripheral unit.

The standard serial I/O mode is different from the parallel I/O mode in that the CPU controls flash memory rewrite (uses the CPU's rewrite mode), rewrite data input and so forth. It is started when the reset is released, which is done when the $P5_0$ ($\overline{CE}$) pin is "H" level, the $P5_5$ ($\overline{EPM}$) pin "L" level and the CNVss pin "H" level. (In the ordinary command mode, set CNVss pin to "L" level.)
This control program is written in the boot ROM area when the product is shipped from Mitsubishi. Accordingly, make note of the fact that the standard serial I/O mode cannot be used if the boot ROM area is rewritten in the parallel I/O mode. Figure 1.27.1 shows the pin connections for the standard serial I/O mode. Serial data I/O uses UART1 and transfers the data serially in 8 bit units. Standard serial I/O switches between mode 1 (clock synchronized) and mode 2 (clock asynchronized) according to the level of CLK_1 pin when the reset is released.

To use standard serial I/O mode 1 (clock synchronized), set the CLK_1 pin to "H" level and release the reset. The operation uses the four UART1 pins CLK_1, RxD_1, TxD_1 and RTS_1 (BUSY). The CLK_1 pin is the transfer clock input pin through which an external transfer clock is input. The TxD_1 pin is for CMOS output. The RTS_1 (BUSY) pin outputs an "L" level when ready for reception and an "H" level when reception starts.

To use standard serial I/O mode 2 (clock asynchronized), set the CLK_1 pin to "L" level and release the reset. The operation uses the two UART1 pins RxD_1 and TxD_1.

In the standard serial I/O mode, only the user ROM area indicated in Figure 1.28.17 can be rewritten. The boot ROM cannot.
In the standard serial I/O mode, a 7 byte ID code is used. When there is data in the flash memory, commands sent from the peripheral unit are not accepted unless the ID code matches.

Overview of Standard Serial I/O Mode 1 (Clock Synchronized)

In standard serial I/O mode 1, software commands, addresses and data are input and output between the MCU and peripheral units (serial programer, etc.) using 4 wire clock synchronized serial I/O (UART1).

Standard serial I/O mode 1 is engaged by releasing the reset with the $P6_5$ (CLK_1) pin "H" level.

In reception, software commands, addresses and program data are synchronized with the rise of the transfer clock that is input to the CLK_1 pin, and are then input to the MCU via the RxD_1 pin. In transmission, the read data and status are synchronized with the fall of the transfer clock, and output from the TxD_1 pin.

The TxD_1 pin is for CMOS output. Transfer is in 8 bit units with LSB first.

When busy, such as during transmission, reception, erasing or program execution, the RTS_1 (BUSY) pin is "H" level. Accordingly, always start the next transfer after the RTS_1 (BUSY) pin is "L" level.

Also, data and status registers in memory can be read after inputting software commands. Status, such as the operating state of the flash memory or whether a program or erase operation ended successfully or not, can be checked by reading the status register. Here following are explained software commands, status registers, etc.

Software Commands

Table 1.28.1 lists software commands. In the standard serial I/O mode 1, erase operations, programs and reading are controlled by transferring software commands via the RxD_1 pin. Software commands are explained here below.

Table 1.28.1. Software commands (Standard serial I/O mode 1)

	Control command	1st byte transfer	2nd byte	3ne byte	4th byte	5th byte	6th byte		When ID is not verified
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	Block erase	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 lock bit status	71_{16}	Address (middle)	Address (high)	Lock bit data output				Not acceptable
8	Lock bit program	77_{16}	Address (middle)	Address (high)	D016				Not acceptable
9	Lock bit enable	$7A_{16}$							Not acceptable
10	Lock bit disable	75_{16}							Not acceptable
11	ID check function	$F5_{16}$	Address (low)	Address (middle)	Address (high)	ID size	ID 1	To ID 7	Acceptable
12	Download function	FA_{16}	Size (low)	Size (high)	Check-sum	Data input	To required number of times		Not acceptable
13	Version information 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
14	Boot ROM area output function	FC_{16}	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
15	Read check data	FD_{16}	Check data (low)	Check data (high)					Not acceptable

Note 1: Shading indicates transfer from flash memory microcomputer to peripheral unit. All other data is transferred from the peripheral unit to the flash memory microcomputer.

Note 2: SRD refers to status register data. SRD1 refers to status register 1 data.

Note 3: 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) Transfer the "FF16" command code with the 1st byte.

(2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, data (D0 to 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 fall of the clock.

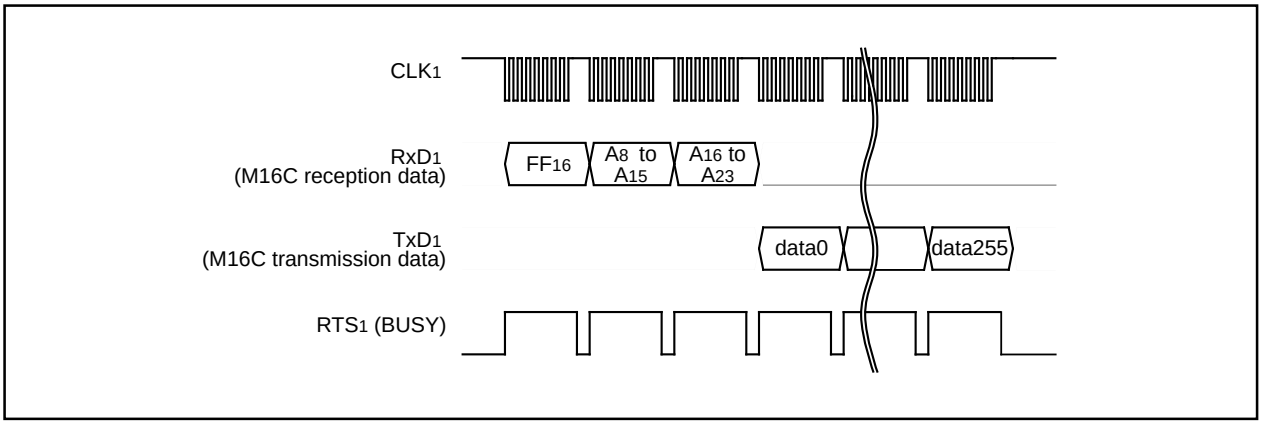


Figure 1.28.1. Timing for page read

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) Transfer the "4116" command code with the 1st byte.

(2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, as write data (D0 to D7) for the page (256 bytes) specified with addresses A8 to A23 is input sequentially from the smallest address first, that page is automatically written.

When reception setup for the next 256 bytes ends, the RTS1 (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.

Each block can be write-protected with the lock bit. For more information, see the section on the data protection function. Additional writing is not allowed with already programmed pages.

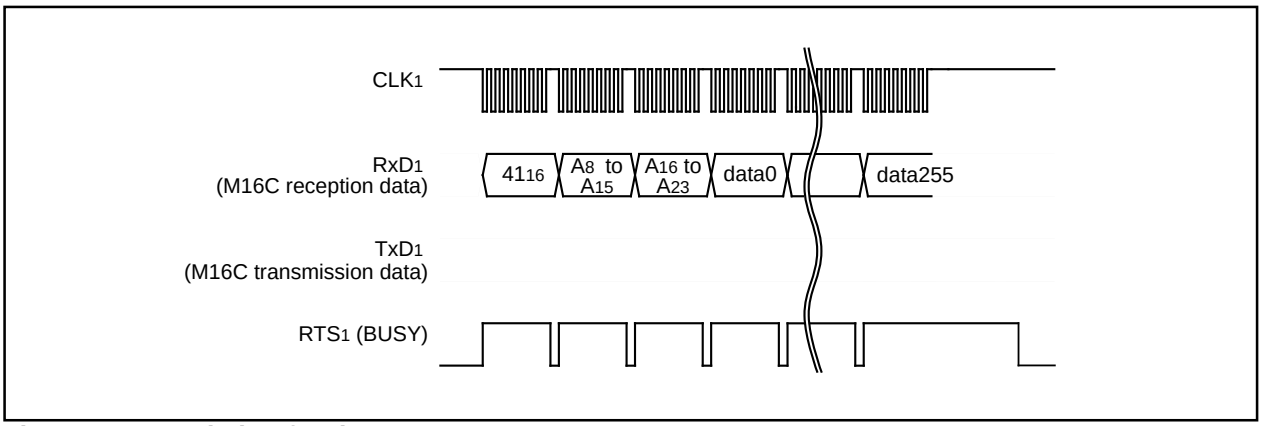


Figure 1.28.2. Timing for the page program

Block Erase Command

This command erases the data in the specified block. Execute the block erase command as explained here following.

(1) Transfer the "20_{16}" command code with the 1st byte.
(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.
(3) Transfer the verify command code "$D0_{16}$" with the 4th byte. 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_8 to A_{23}.

When block erasing ends, the RTS_1 (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.

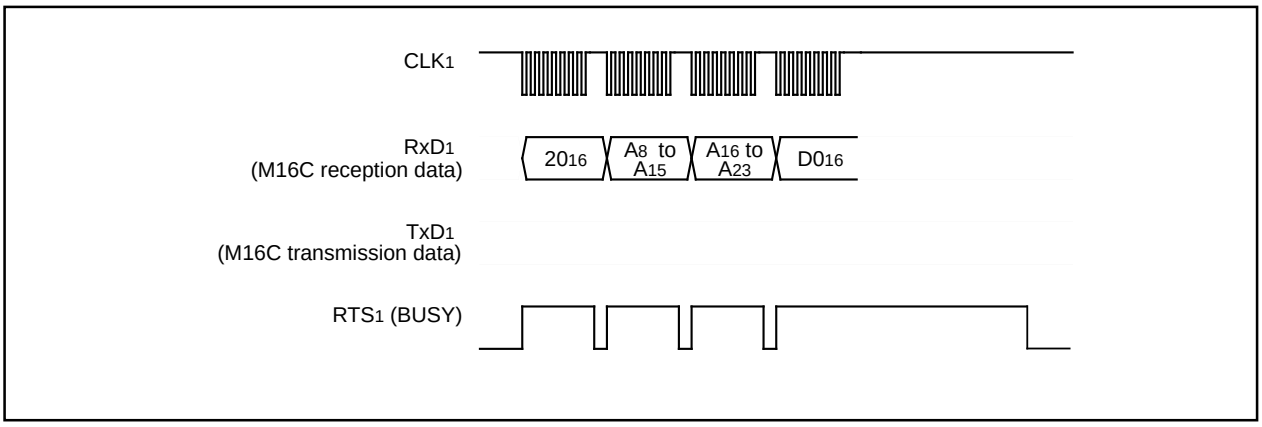


Figure 1.28.3. Timing for block erasing

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) Transfer the "$A7_{16}$" command code with the 1st byte.
(2) Transfer the verify command code "$D0_{16}$" with the 2nd byte. 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_1 (BUSY) signal changes from the "H" to the "L" level. The result of the erase operation can be known by reading the status register. Each block can be erase protected with the lock bit. For more information, see the section on the data protection function.

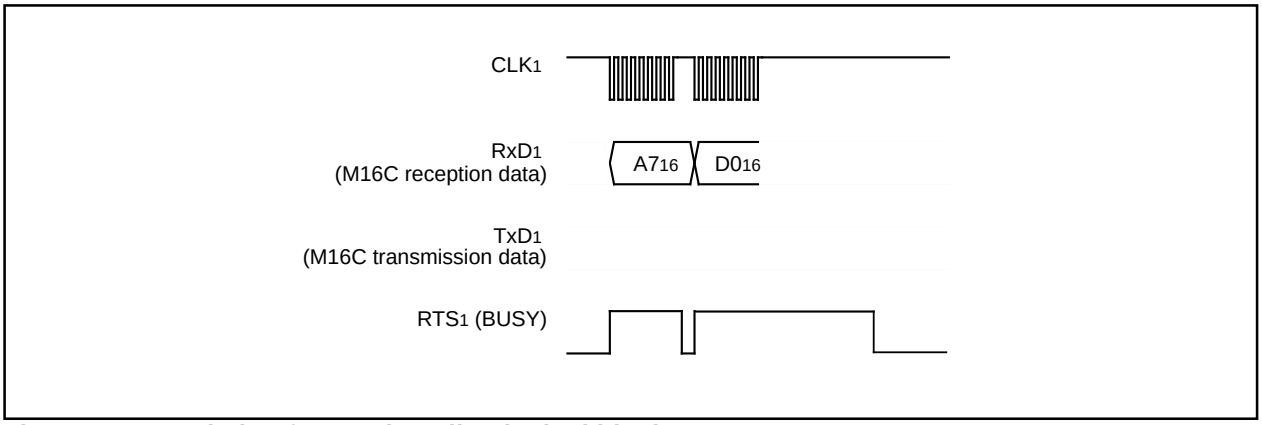


Figure 1.28.4. Timing for erasing all unlocked blocks

Read Status Register Command

This command reads status information. When the "70_{16}" command code is sent with the 1st byte, the contents of the status register (SRD) specified with the 2nd byte and the contents of status register 1 (SRD1) specified with the 3rd byte are read.

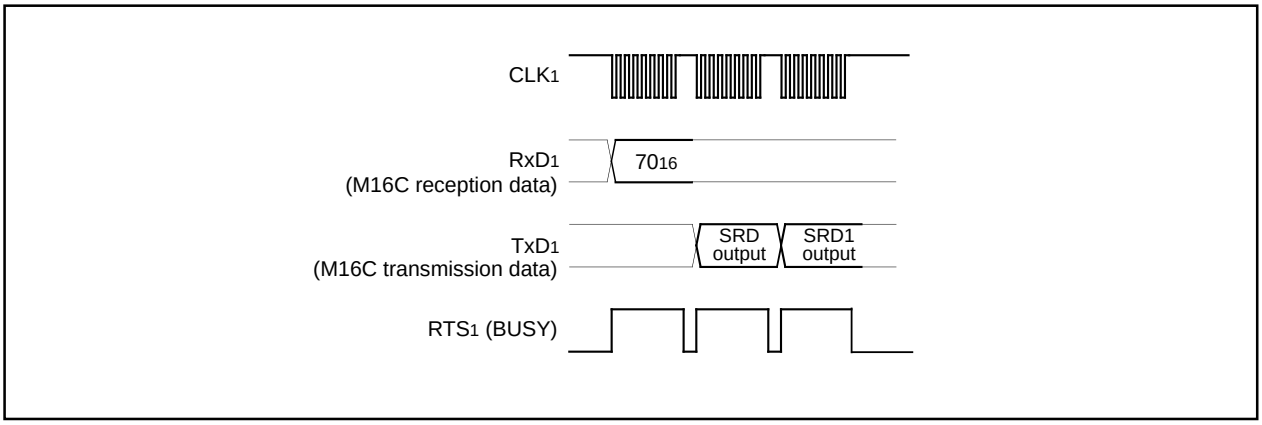


Figure 1.28.5. Timing for reading the status register

Clear Status Register Command

This command clears the bits (SR3 to SR5) which are set when the status register operation ends in error. When the "50_{16}" command code is sent with the 1st byte, the aforementioned bits are cleared. When the clear status register operation ends, the RTS1 (BUSY) signal changes from the "H" to the "L" level.

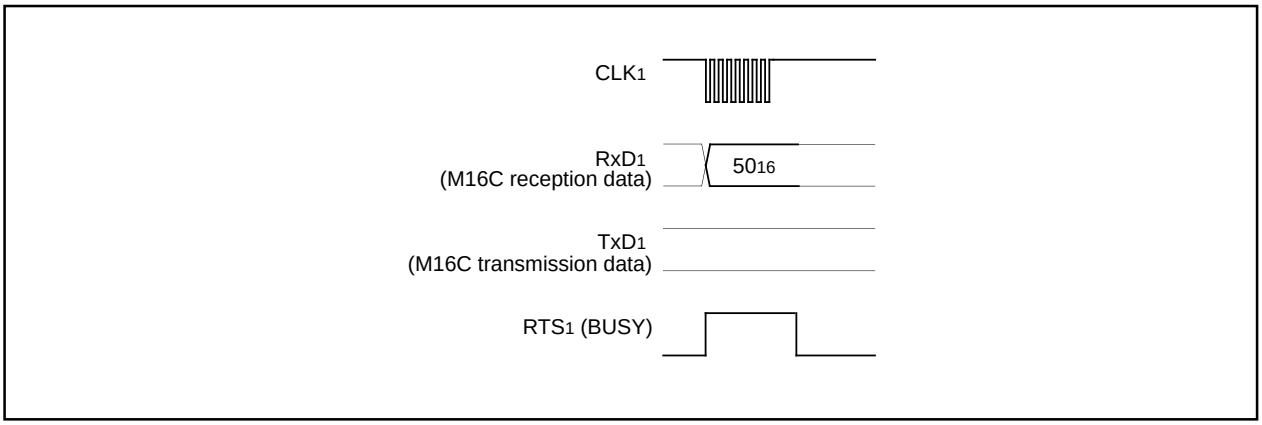


Figure 1.28.6. Timing for clearing the status register

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) Transfer the "71_{16}" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) The lock bit data of the specified block is output with the 4th byte. The lock bit data is the 6th bit(D_6) of the output data. Write the highest address of the specified block for addresses A_8 to A_{23}.

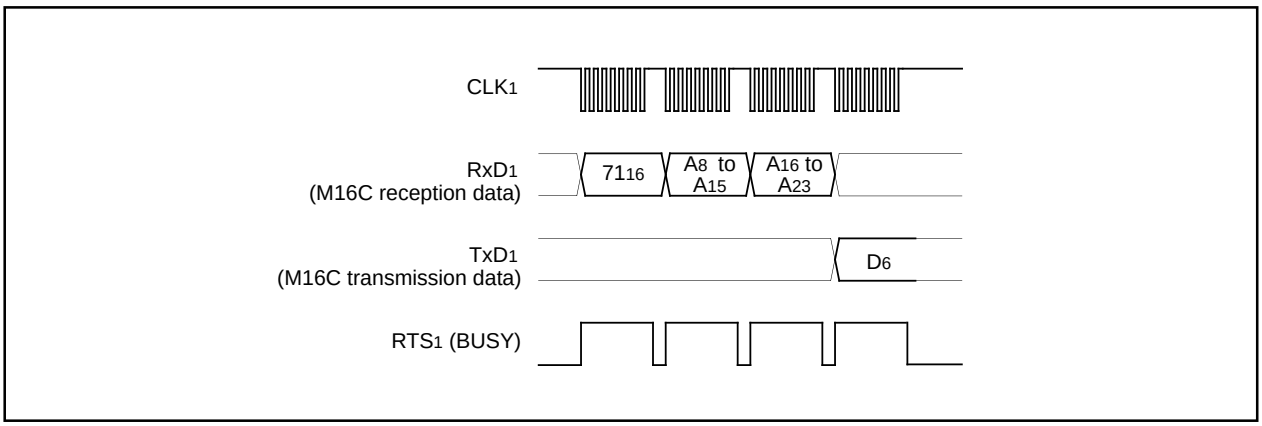


Figure 1.28.7. Timing for reading lock bit status

Lock Bit Program Command

This command writes "0" (lock) for the lock bit of the specified block. Execute the lock bit program command as explained here following.

(1) Transfer the "77_{16}" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) Transfer the verify command code "$D0_{16}$" with the 4th byte. With the verify command code, "0" is written for the lock bit of the specified block. Write the highest address of the specified block for addresses A_8 to A_{23}.

When writing ends, the RTS1 (BUSY) signal changes from the "H" to the "L" level. Lock bit status can be read with the read lock bit status command. For information on the lock bit function, reset procedure and so on, see the section on the data protection function.

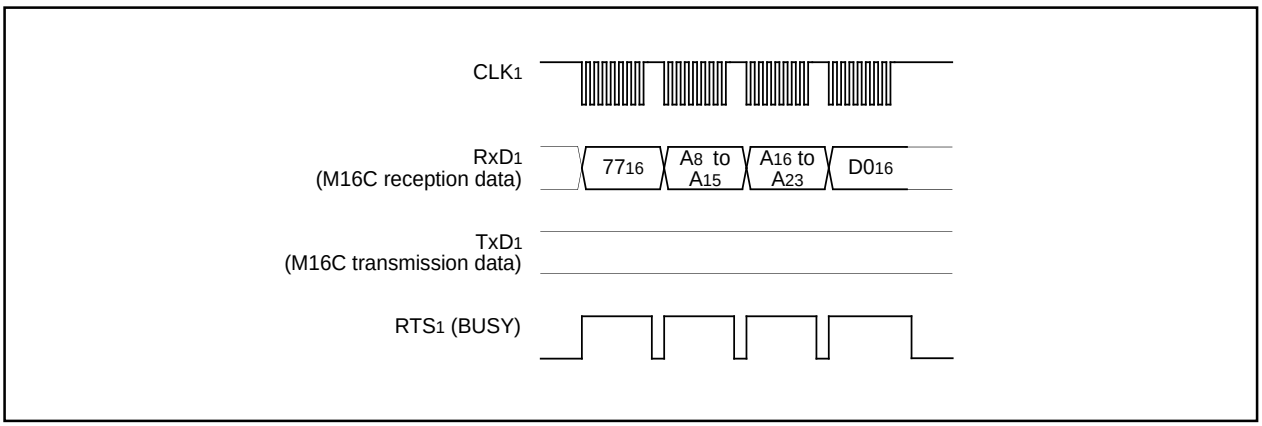


Figure 1.28.8. Timing for the lock bit program

Lock Bit Enable Command

This command enables the lock bit in blocks whose bit was disabled with the lock bit disable command. The command code "7A16" is sent with the 1st byte of the serial transmission. This command only enables the lock bit function; it does not set the lock bit itself.

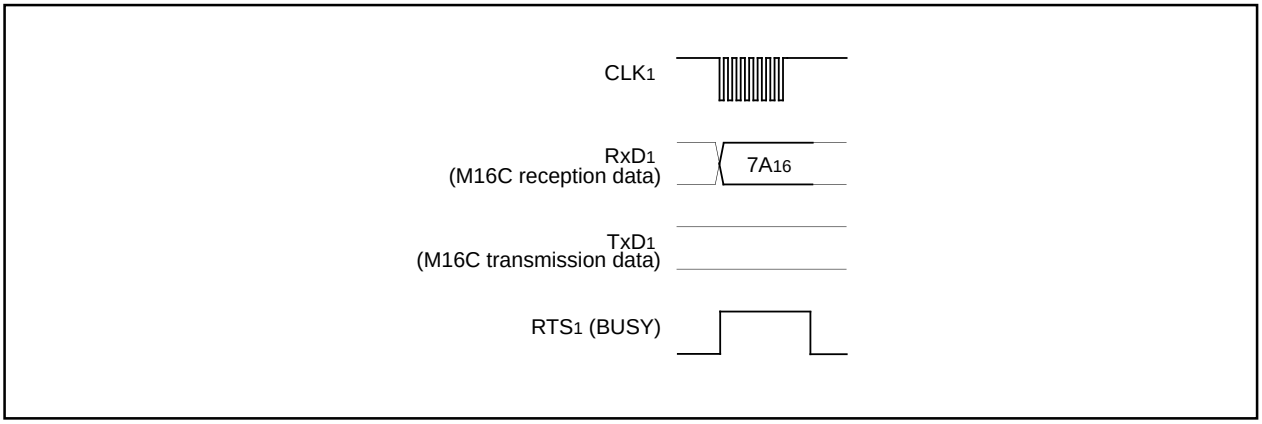


Figure 1.28.9. Timing for enabling the lock bit

Lock Bit Disable Command

This command disables the lock bit. The command code "7516" is sent with the 1st byte of the serial transmission. This command only disables the lock bit function; it does not set the lock bit itself. However, if an erase command is executed after executing the lock bit disable command, "0" (locked) lock bit data is set to "1" (unlocked) after the erase operation ends. In any case, after the reset is cancelled, the lock bit is enabled.

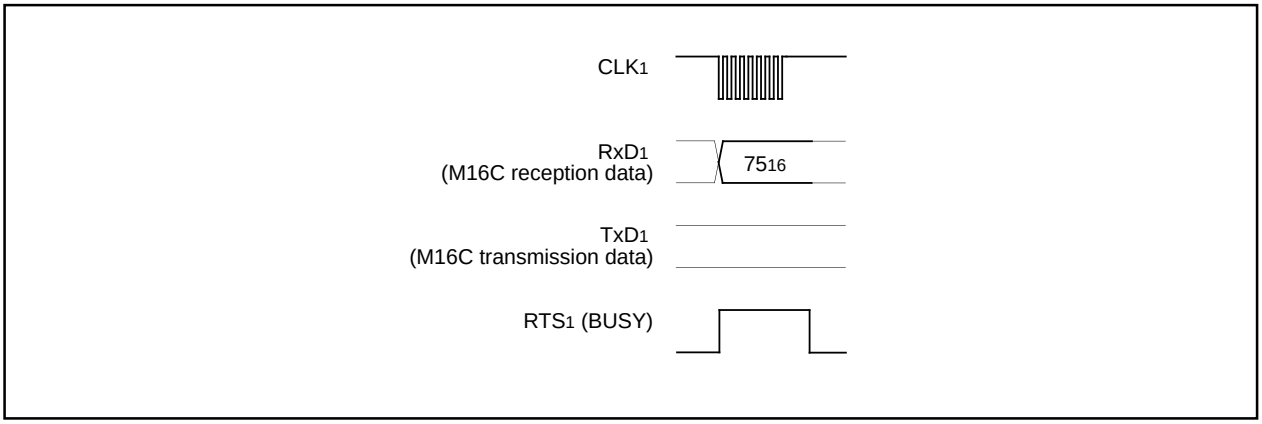


Figure 1.28.10. Timing for disabling the lock bit

ID Check Function Command

This command checks the ID code. Execute the boot ID check command as explained here following.

(1) Transfer the "F5$_{16}$" command code with the 1st byte.

(2) Transfer 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 with the 2nd, 3rd and 4th bytes respectively.

(3) Transfer the number of data sets of the ID code with the 5th byte.

(4) The ID code is sent with the 6th byte onward, starting with the 1st byte of the code.

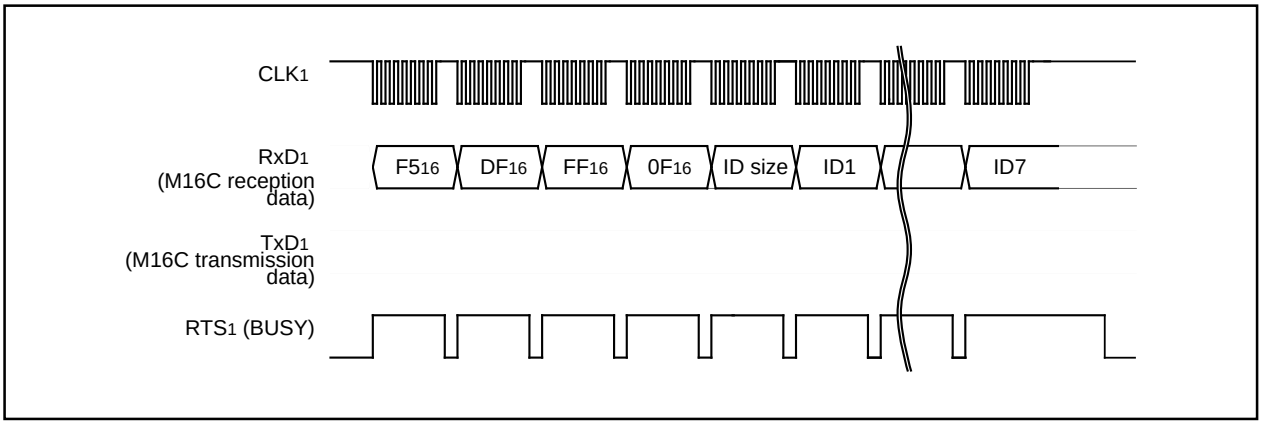


Figure 1.28.11. Timing for the ID check

Download Function Command

This command downloads a program to the RAM for execution. Execute the download command as explained here following.

(1) Transfer the "FA$_{16}$" command code with the 1st byte.

(2) Transfer the program size with the 2nd and 3rd bytes.

(3) Transfer the check sum with the 4th byte. The check sum is added to all data sent with the 5th byte onward.

(4) The program to execute is sent with 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.

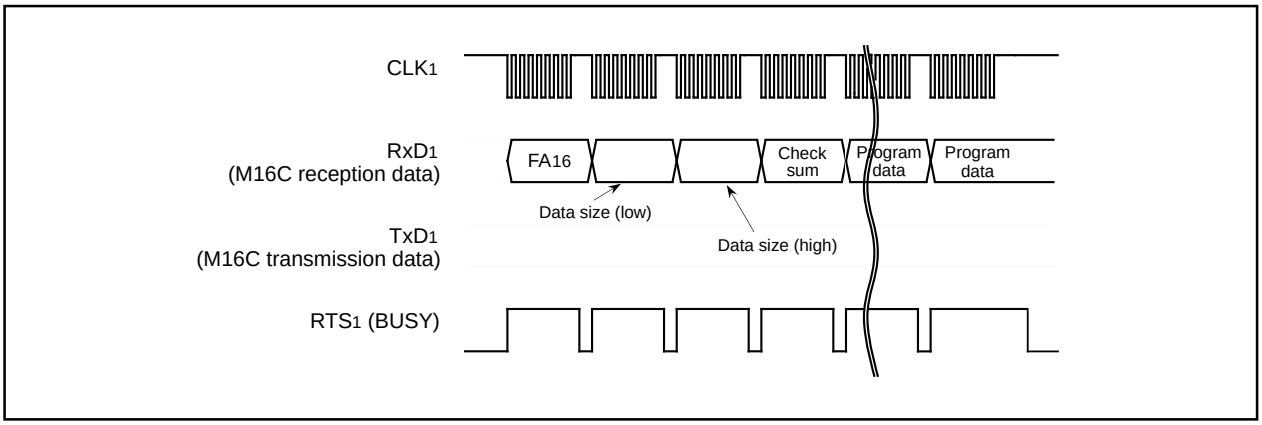


Figure 1.28.12. Timing for download

Version Information Output Function 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) Transfer the "FB16" command code with the 1st byte.

(2) The version information will be output from the 2nd byte onward. This data is composed of 8 ASCII code characters.

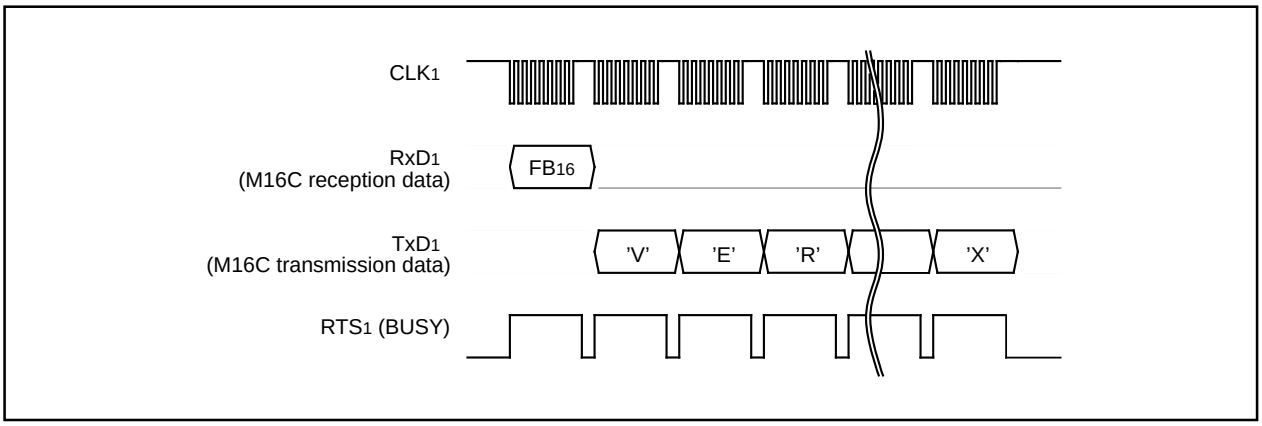


Figure 1.28.13. Timing for version information output

Boot ROM Area Output Function Command

This command outputs the control program stored in the boot ROM area in one page blocks (256 bytes). Execute the boot ROM area output command as explained here following.

(1) Transfer the "FC16" command code with the 1st byte.

(2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, data (D0 to 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 fall of the clock.

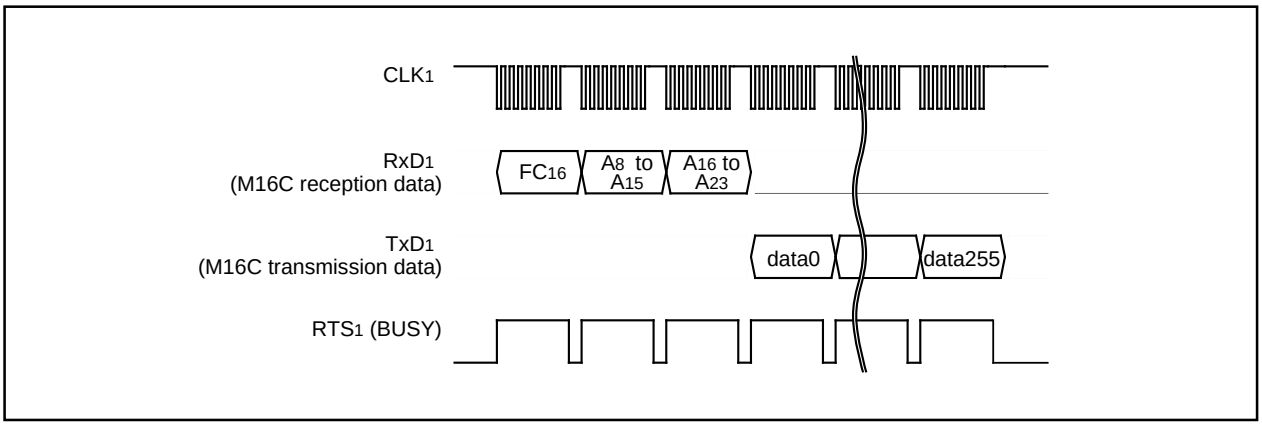


Figure 1.28.14. Timing for boot ROM area output

Read Check Data Command

This command reads the check data that confirms that the write data, which was sent with the page program command, was successfully received.

(1) Transfer the "FD16" command code with the 1st byte.

(2) The check data (low) is received with the 2nd byte and the check data (high) with the 3rd.

To use this read check data command, first execute the command and then initialize the check data. Next, execute the page program command the required number of times. After that, when the read check command is executed again, the check data for all of the read data that was sent with the page program command during this time is read. The check data is the result of CRC operation of write data.

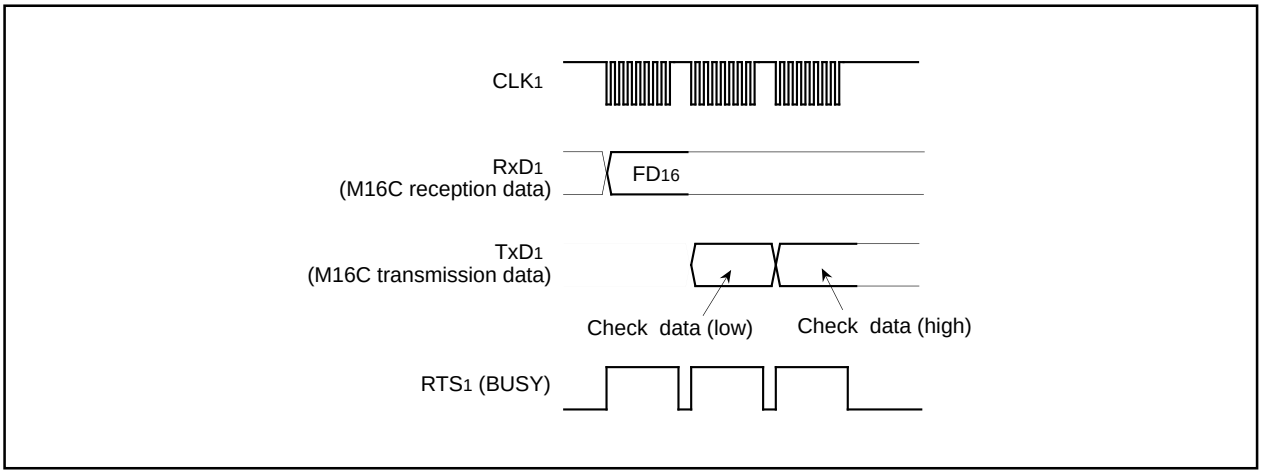


Figure 1.28.15. Timing for the read check data

ID Code

When the flash memory is not blank, the ID code sent from the peripheral units 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 peripheral units is not accepted. An ID code contains 8 bits of data. Area is, from the 1st byte, addresses 0FFFDF16, 0FFFE316, 0FFFEB16, 0FFFEF16, 0FFFF316, 0FFFF716 and 0FFFFB16. Write a program into the flash memory, which already has the ID code set for these addresses.

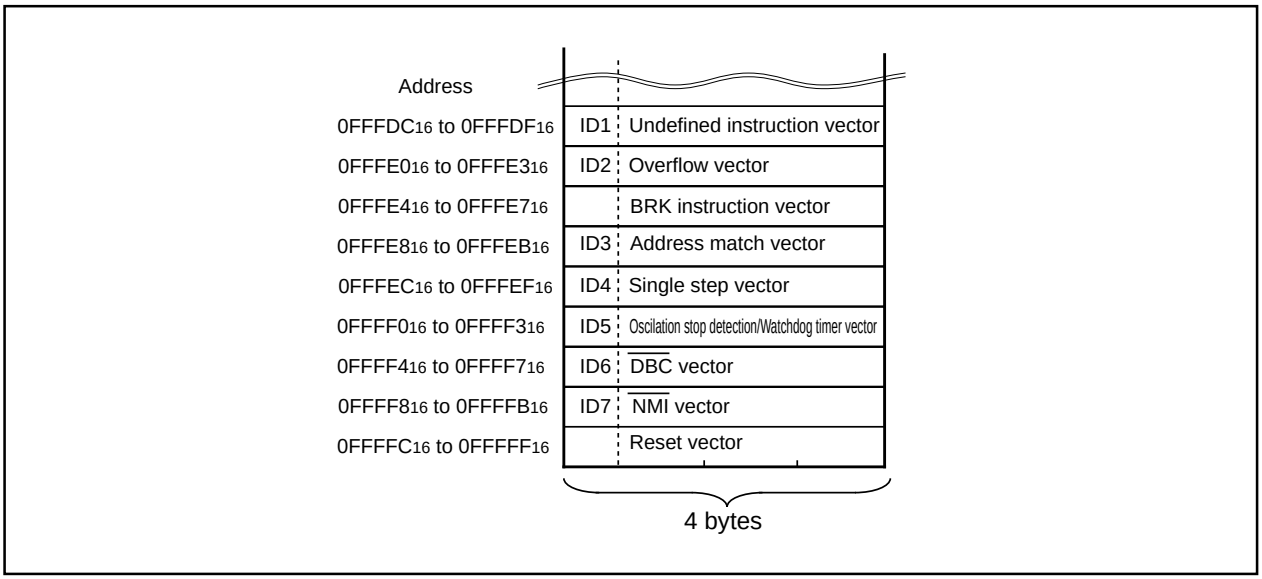


Figure 1.28.16. ID code storage addresses

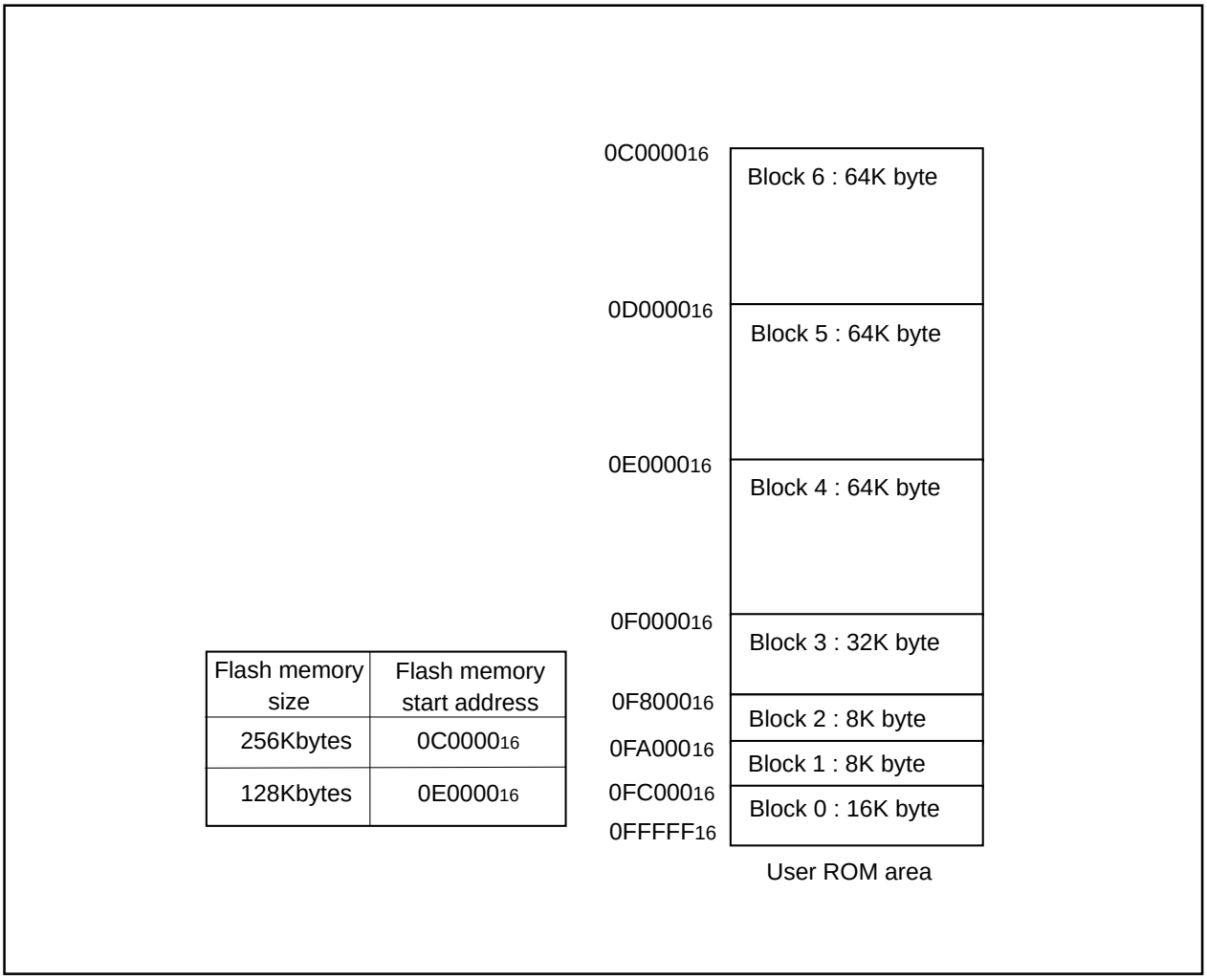
Data Protection Function (Block Lock)

Each of the blocks in Figure 1.28.17 have a nonvolatile lock bit that specifies protection (block lock) against erasing/writing. A block is locked (writing "0" for the lock bit) with the lock bit program command. Also, the lock bit of any block can be read with the read lock bit status command.

Block lock disable/enable is determined by the status of the lock bit itself and execution status of the lock bit disable and lock enable bit commands.

(1) After the reset has been cancelled and the lock bit enable command executed, the specified block can be locked/unlocked using the lock bit (lock bit data). Blocks with a "0" lock bit data are locked and cannot be erased or written in. On the other hand, blocks with a "1" lock bit data are unlocked and can be erased or written in.

(2) After the lock bit disable command has been executed, all blocks are unlocked regardless of lock bit data status and can be erased or written in. In this case, lock bit data that was "0" (locked) before the block was erased is set to "1" (unlocked) after erasing, therefore the block is actually unlocked with the lock bit.



Flash memory size	Flash memory start address
256Kbytes	0C000016
128Kbytes	0E000016

Figure 1.28.17. Blocks in the user area

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 1.28.2 gives the definition of each status register bit. After clearing the reset, the status register outputs "80_{16}".

Table 1.28.2. Status register (SRD)

SRD bits	Status name	Definition	
		"1"	"0"
SR7 (bit7)	Write state machine (WSM) status	Ready	Busy
SR6 (bit6)	Reserved	-	-
SR5 (bit5)	Erase status	Terminated in error	Terminated normally
SR4 (bit4)	Program status	Terminated in error	Terminated normally
SR3 (bit3)	Block status after program	Terminated in error	Terminated normally
SR2 (bit2)	Reserved	-	-
SR1 (bit1)	Reserved	-	-
SR0 (bit0)	Reserved	-	-

Write State Machine (WSM) Status (SR7)

The write state machine (WSM) status 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 Status (SR5)

The erase status 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 Status (SR4)

The program status 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".

Block Status After Program (SR3)

If excessive data is written (phenomenon whereby the memory cell becomes depressed which results in data not being read correctly), "1" is set for the block status after program at the end of the page write operation. In other words, when writing ends successfully, "80_{16}" is output; when writing fails, "90_{16}" is output; and when excessive data is written, "88_{16}" is output.

If "1" is written for any of the SR5, SR4 or SR3 bits, the page program, block erase, erase all unlocked blocks and lock bit program commands are not accepted. Before executing these commands, execute the clear status register command (50_{16}) and clear the status register.

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 1.28.3 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 1.28.3. 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)	Check sum 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 Match 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)

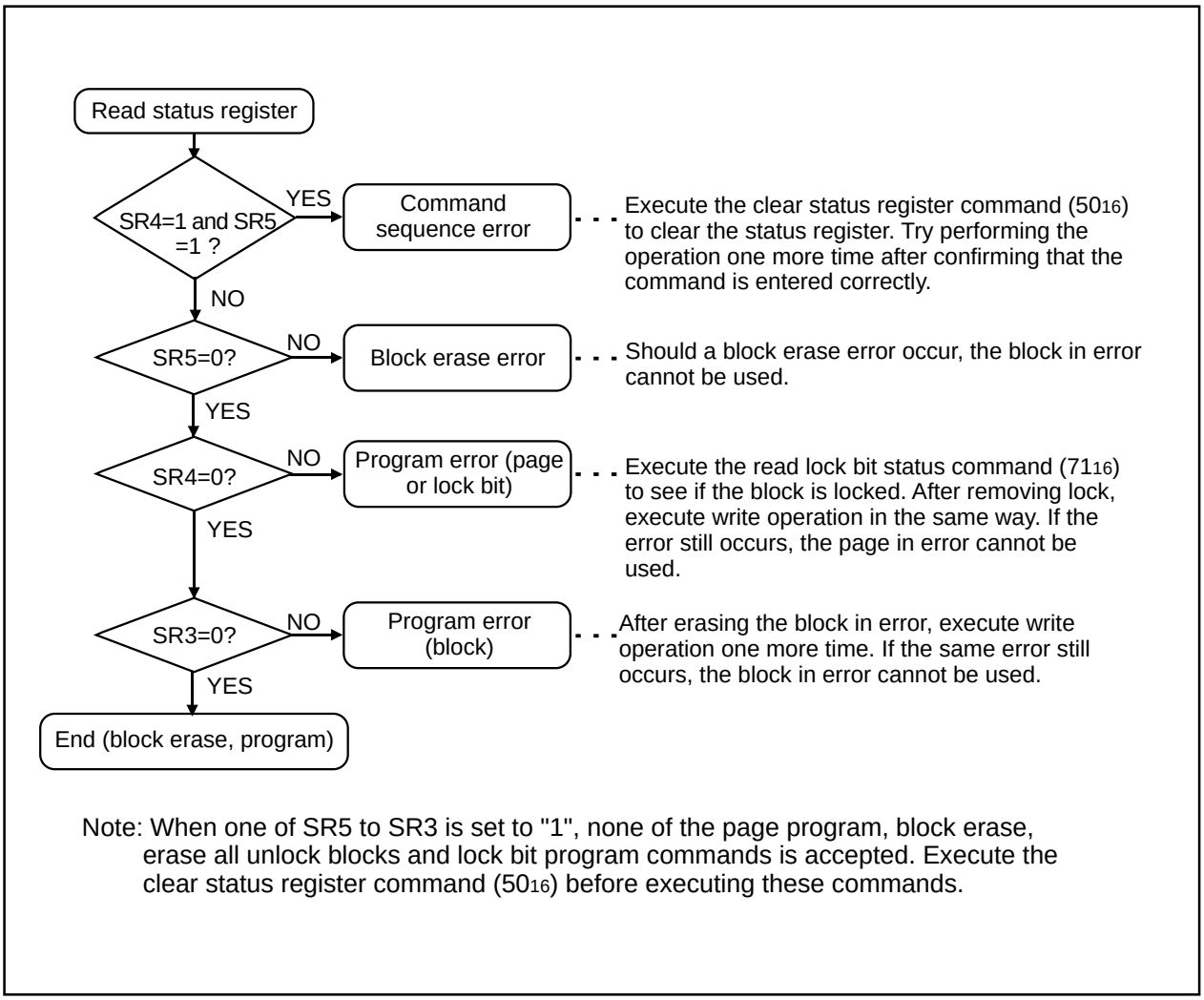
These flags indicate the result of ID checks. Some commands cannot be accepted without an ID check.

Data Receive 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.

Full Status Check

Results from executed erase and program operations can be known by running a full status check. Figure 1.28.18 shows a flowchart of the full status check and explains how to remedy errors which occur.

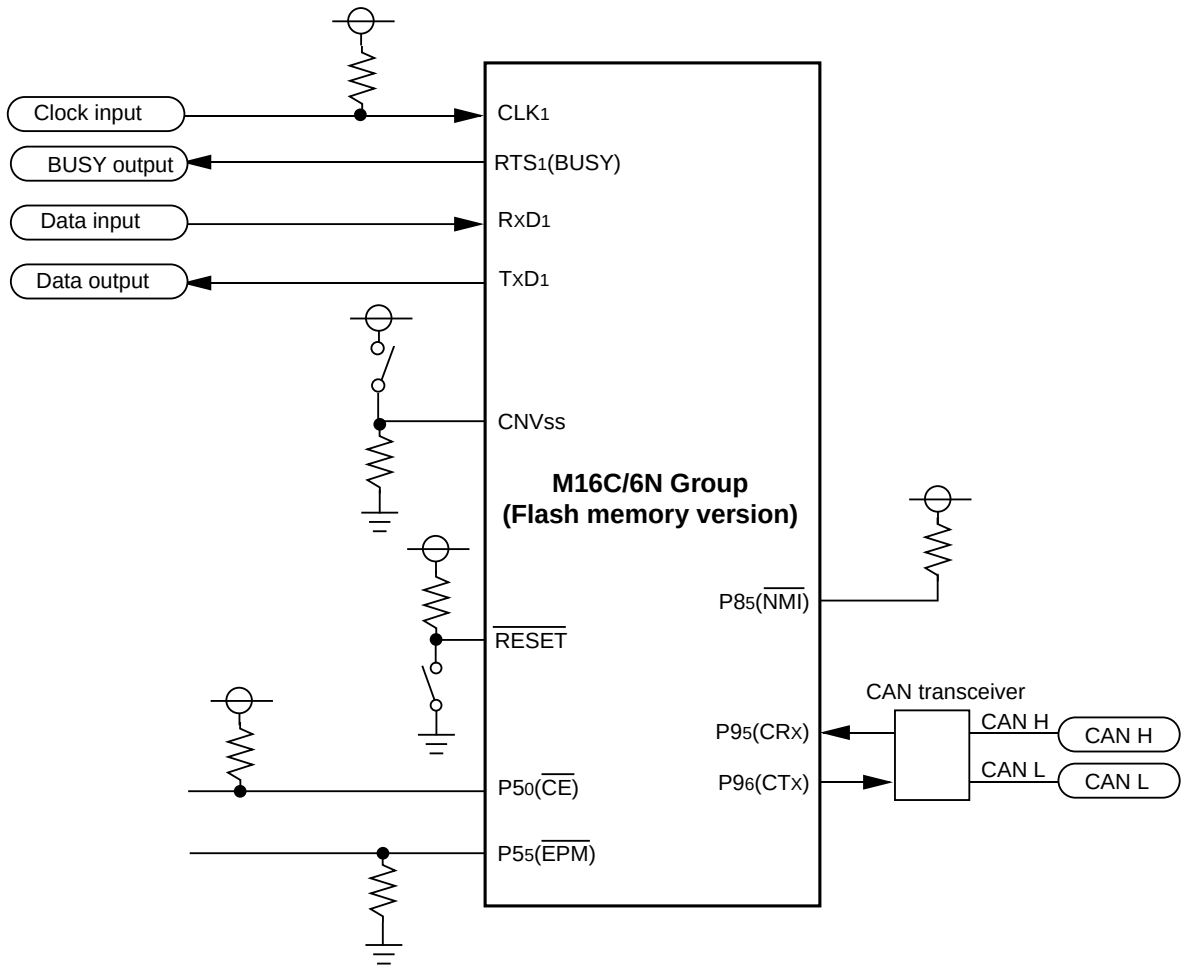


Note: When one of SR5 to SR3 is set to "1", none of the page program, block erase, erase all unlock blocks and lock bit program commands is accepted. Execute the clear status register command (5016) before executing these commands.

Figure 1.28.18. Full status check flowchart and remedial procedure for errors

Example Circuit Application for the Standard Serial I/O Mode 1

The below figure shows a circuit application for the standard serial I/O mode 1. Control pins will vary according to programmer, therefore see the peripheral unit manual for more information.



(1) Control pins and external circuitry will vary according to peripheral unit.
For more information, see the peripheral unit manual.
(2) In this example, the microprocessor mode and standard serial I/O mode are switched via a switch.

Figure 1.28.19. Example circuit application for the standard serial I/O mode 1

Overview of Standard Serial I/O Mode 2 (Clock Asynchronized)

In standard serial I/O mode 2, software commands, addresses and data are input and output between the MCU and peripheral units (serial programer, etc.) using 2 wire clock asynchronized serial I/O (UART1).
Standard serial I/O mode 2 is engaged by releasing the reset with the $P6_5$ (CLK_1) pin "L" level.
The TxD_1 pin is for CMOS output. Data transfer is in 8 bit units with LSB first, 1 stop bit and parity off.
After the reset is released, connections can be established at 9,600 bps when initial communications are made with a peripheral unit. However, this requires a main clock with 10 MHz or 16 MHz input oscillation frequency. Baud rate can also be changed from 9,600 bps, 19,200 bps or 38,400 bps by executing software commands. However, communication errors may occur because of the oscillation frequency of the main clock. If errors occur, change the main clock's oscillation frequency and the baud rate.
After executing commands from a peripheral unit that requires time to erase and write data, as with erase and program commands, allow a sufficient time interval or execute the read status command and check how processing ended, before executing the next command.
Data and status registers in memory can be read after transmitting software commands. Status, such as the operating state of the flash memory or whether a program or erase operation ended successfully or not, can be checked by reading the status register. Here following are explained the operation frequency and the baud rate, how frequency is identified and software commands.

Table 1.29.1. Operation frequency and the baud rate

Operation frequency (MHz)	Baud rate 9,600bps	Baud rate 19,200bps	Baud rate 38,400bps
16MHz	√	√	√
10MHz	√	√	—

√ : Communications possible
— : Communications not possible

Software Commands

Table 1.29.2 lists software commands. In the standard serial I/O mode 2, erase operations, programs and reading are controlled by transferring software commands via the RxD_1 pin. Standard serial I/O mode 2 adds three transmission speed commands 9,600, 19,200 and 38,400 bps to the software commands of standard serial I/O mode 1. Software commands are explained here below.

Table 1.29.2. Software commands (Standard serial I/O mode 2)

	Control command	1st byte transfer	2nd byte	3ne byte	4th byte	5th byte	6th byte		When ID is not verified
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	Block erase	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 lock bit status	71_{16}	Address (middle)	Address (high)	Lock bit data output				Not acceptable
8	Lock bit program	77_{16}	Address (middle)	Address (high)	D016				Not acceptable
9	Lock bit enable	$7A_{16}$							Not acceptable
10	Lock bit disable	75_{16}							Not acceptable
11	ID check function	$F5_{16}$	Address (low)	Address (middle)	Address (high)	ID size	ID 1	To ID 7	Acceptable
12	Download function	FA_{16}	Size (low)	Size (high)	Check-sum	Data input	To required number of times		Not acceptable
13	Version information 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
14	Boot ROM area output function	FC_{16}	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
15	Read check data	FD_{16}	Check data (low)	Check data (high)					Not acceptable
16	Baud rate 9600	$B0_{16}$	$B0_{16}$						Acceptable
17	Baud rate 19200	$B1_{16}$	$B1_{16}$						Acceptable
18	Baud rate 38400	$B2_{16}$	$B2_{16}$						Acceptable

Note 1: Shading indicates transfer from flash memory microcomputer to peripheral unit. All other data is transferred from the peripheral unit to the flash memory microcomputer.

Note 2: SRD refers to status register data. SRD1 refers to status register 1 data.

Note 3: 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) Transfer the "FF_{16}" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, data (D_0 to D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} will be output sequentially from the smallest address first.

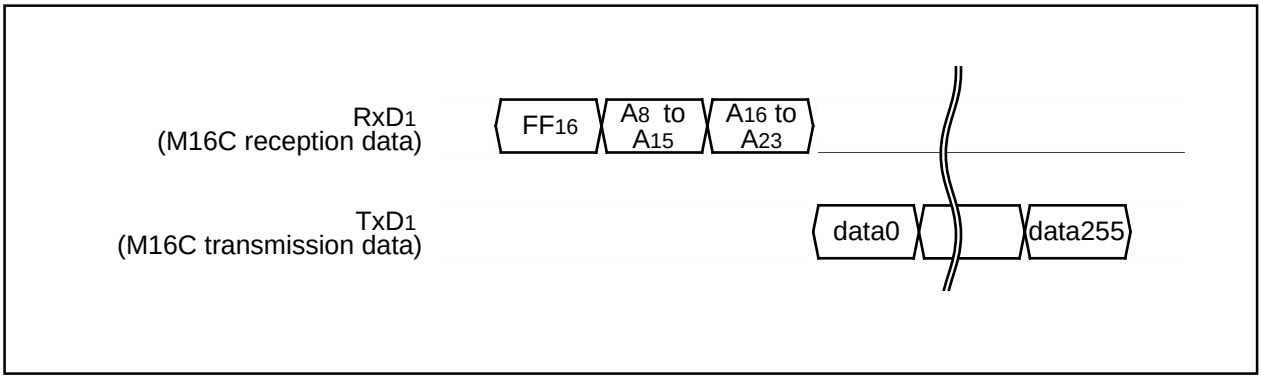


Figure 1.29.1. Timing for page read

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) Transfer the "41_{16}" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, as write data (D_0 to 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.

The result of the page program can be known by reading the status register. For more information, see the section on the status register.

Each block can be write protected with the lock bit. For more information, see the section on the data protection function. Additional writing is not allowed with already programmed pages.

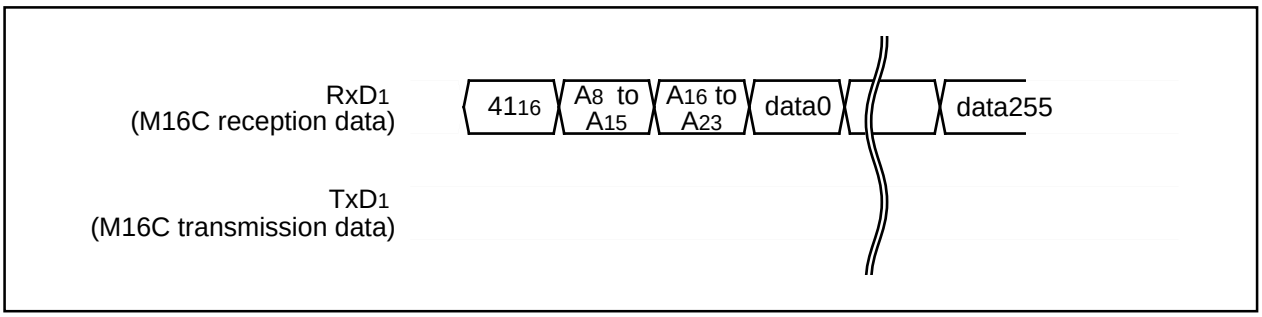


Figure 1.29.2. Timing for the page program

Block Erase Command

This command erases the data in the specified block. Execute the block erase command as explained here following.

(1) Transfer the "20_{16}" command code with the 1st byte.
(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.
(3) Transfer the verify command code "$D0_{16}$" with the 4th byte. 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_8 to A_{23}.

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.

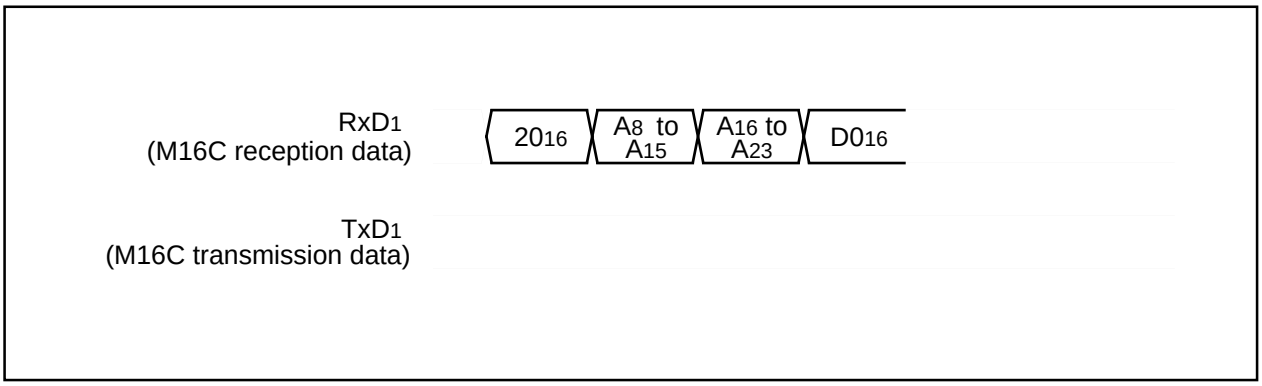


Figure 1.29.3. Timing for block erasing

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) Transfer the "$A7_{16}$" command code with the 1st byte.
(2) Transfer the verify command code "$D0_{16}$" with the 2nd byte. With the verify command code, the erase operation will start and continue for all blocks in the flash memory.

The result of the erase operation can be known by reading the status register. Each block can be erase-protected with the lock bit. For more information, see the section on the data protection function.

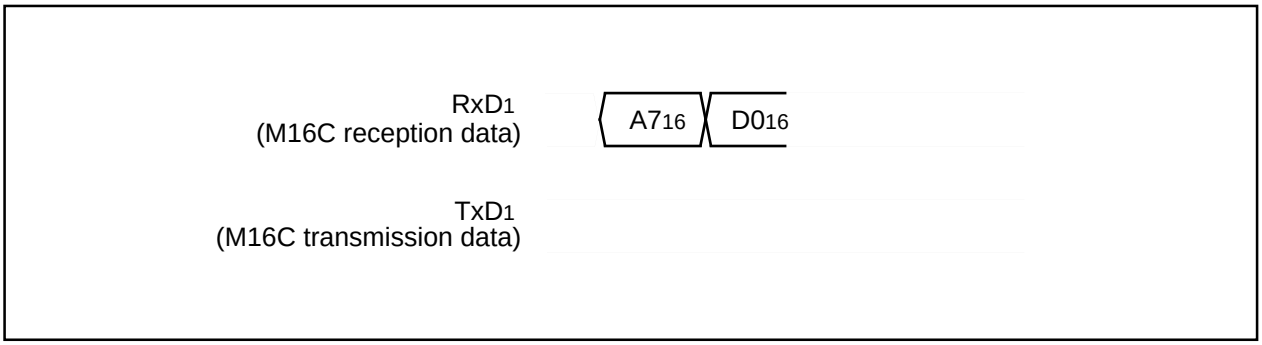


Figure 1.29.4. Timing for erasing all unlocked blocks

Read Status Register Command

This command reads status information. When the "70_{16}" command code is sent with the 1st byte, the contents of the status register (SRD) specified with the 2nd byte and the contents of status register 1 (SRD1) specified with the 3rd byte are read.

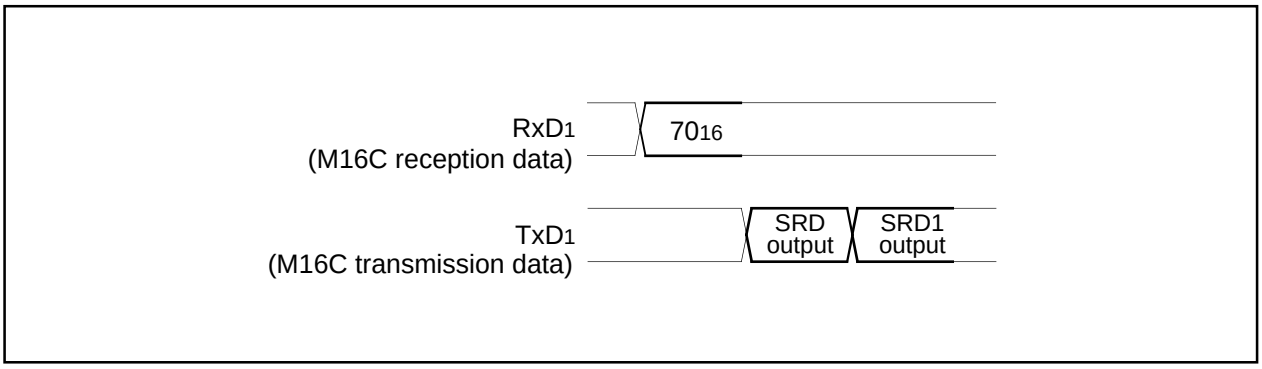


Figure 1.29.5. Timing for reading the status register

Clear Status Register Command

This command clears the bits (SR3 to SR5) which are set when the status register operation ends in error. When the "50_{16}" command code is sent with the 1st byte, the aforementioned bits are cleared.

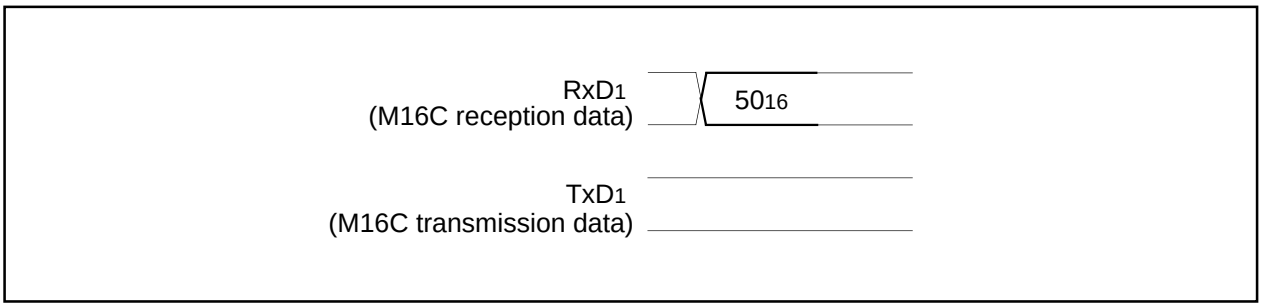


Figure 1.29.6. Timing for clearing the status register

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) Transfer the "71_{16}" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) The lock bit data of the specified block is output with the 4th byte. The lock bit data is the 6th bit(D_6) of the output data. Write the highest address of the specified block for addresses A_8 to A_{23}.

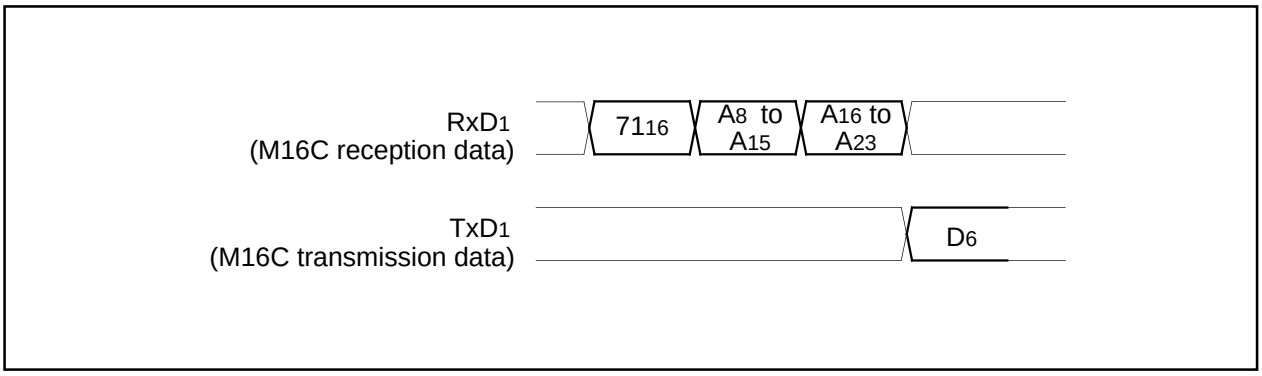


Figure 1.29.7. Timing for reading lock bit status

Lock Bit Program Command

This command writes "0" (lock) for the lock bit of the specified block. Execute the lock bit program command as explained here following.

(1) Transfer the "77_{16}" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) Transfer the verify command code "$D0_{16}$" with the 4th byte. With the verify command code, "0" is written for the lock bit of the specified block. Write the highest address of the specified block for addresses A_8 to A_{23}.

Lock bit status can be read with the read lock bit status command. For information on the lock bit function, reset procedure and so on, see the section on the data protection function.

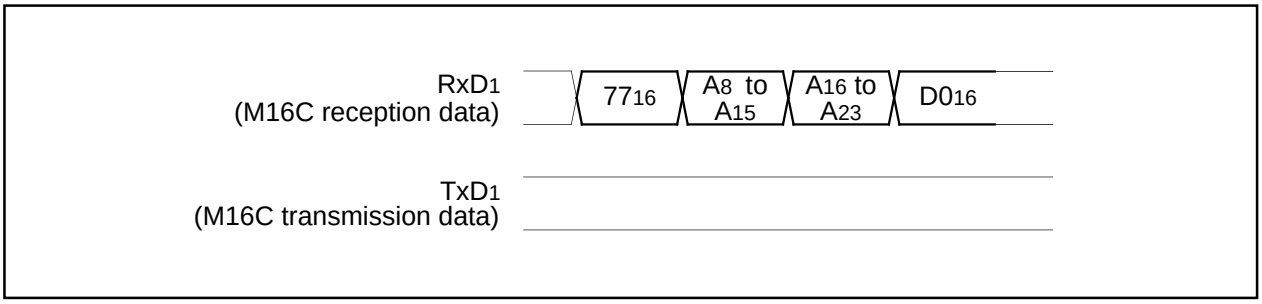


Figure 1.29.8. Timing for the lock bit program

Lock Bit Enable Command

This command enables the lock bit in blocks whose bit was disabled with the lock bit disable command. The command code "7A16" is sent with the 1st byte of the serial transmission. This command only enables the lock bit function; it does not set the lock bit itself.

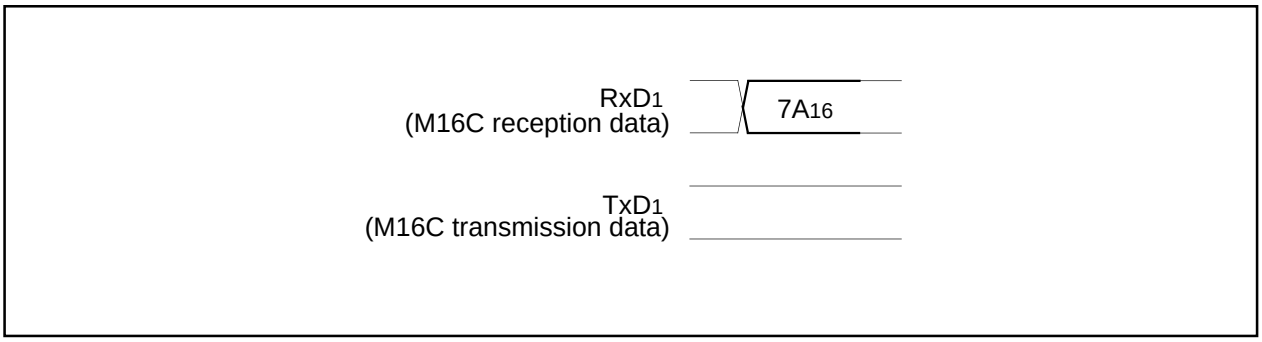


Figure 1.29.9. Timing for enabling the lock bit

Lock Bit Disable Command

This command disables the lock bit. The command code "7516" is sent with the 1st byte of the serial transmission. This command only disables the lock bit function; it does not set the lock bit itself. However, if an erase command is executed after executing the lock bit disable command, "0" (locked) lock bit data is set to "1" (unlocked) after the erase operation ends. In any case, after the reset is cancelled, the lock bit is enabled.

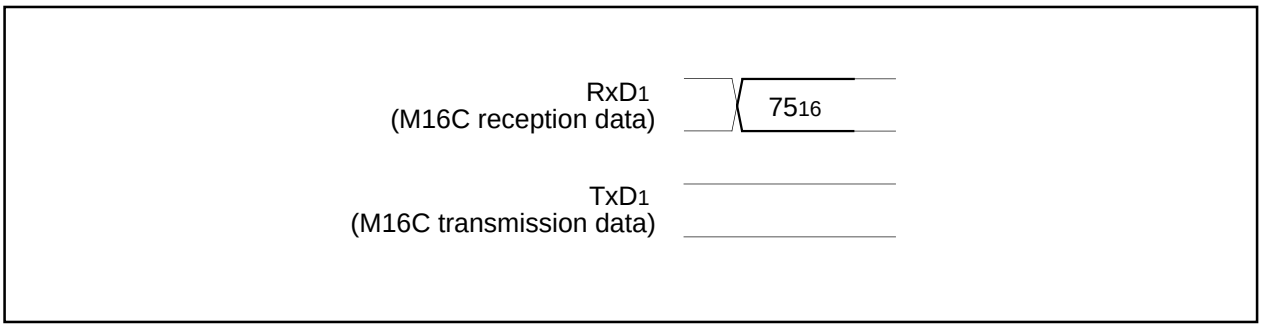


Figure 1.29.10. Timing for disabling the lock bit

ID Check Function Command

This command checks the ID code. Execute the boot ID check command as explained here following.

(1) Transfer the "F5$_{16}$" command code with the 1st byte.

(2) Transfer 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 with the 2nd, 3rd and 4th bytes respectively.

(3) Transfer the number of data sets of the ID code with the 5th byte.

(4) The ID code is sent with the 6th byte onward, starting with the 1st byte of the code.

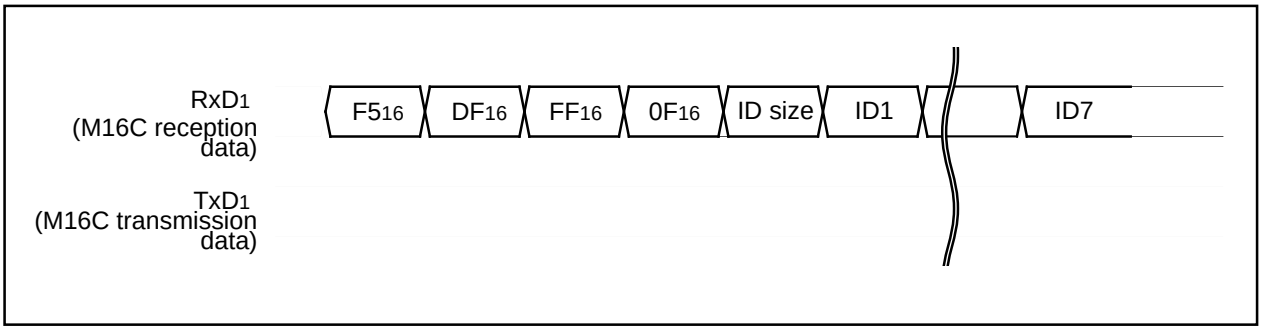


Figure 1.29.11. Timing for the ID check

Download Function Command

This command downloads a program to the RAM for execution. Execute the download command as explained here following.

(1) Transfer the "FA$_{16}$" command code with the 1st byte.

(2) Transfer the program size with the 2nd and 3rd bytes.

(3) Transfer the check sum with the 4th byte. The check sum is added to all data sent with the 5th byte onward.

(4) The program to execute is sent with 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.

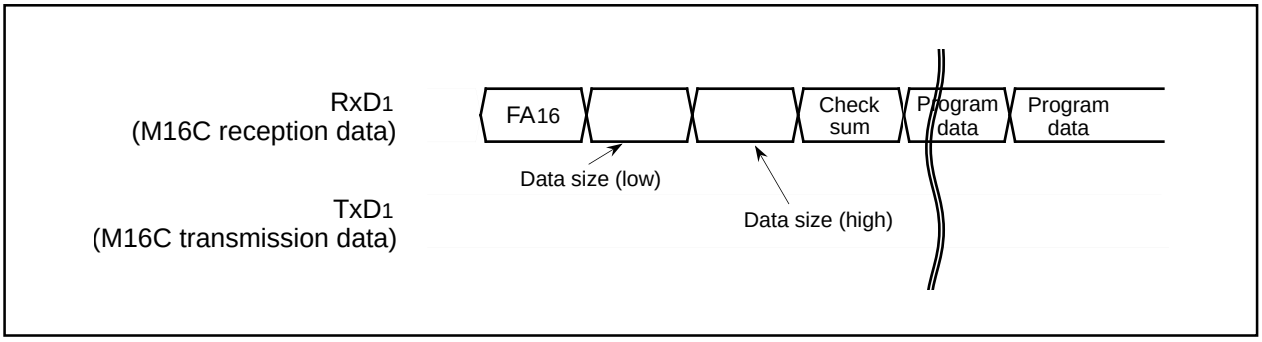


Figure 1.29.12. Timing for download

Version Information Output Function 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) Transfer the "FB16" command code with the 1st byte.

(2) The version information will be output from the 2nd byte onward. This data is composed of 8 ASCII code characters.

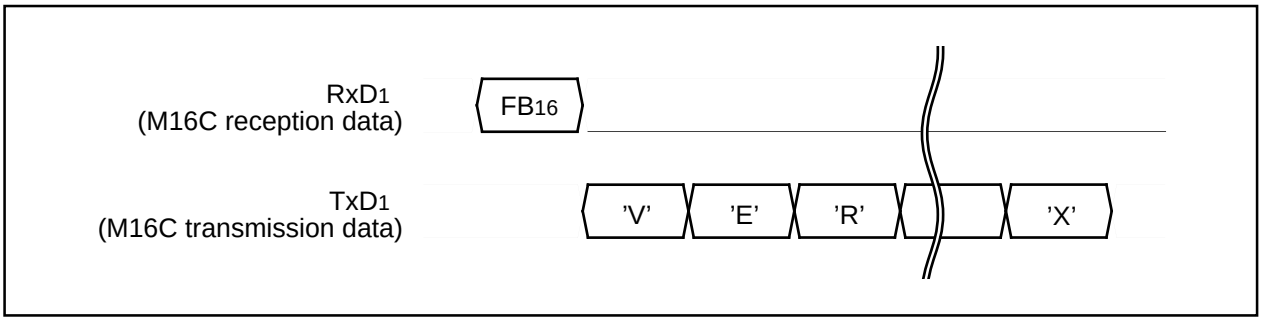


Figure 1.29.13. Timing for version information output

Boot ROM Area Output Function Command

This command outputs the control program stored in the boot ROM area in one page blocks (256 bytes). Execute the boot ROM area output command as explained here following.

(1) Transfer the "FC16" command code with the 1st byte.

(2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, data (D0 to D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first.

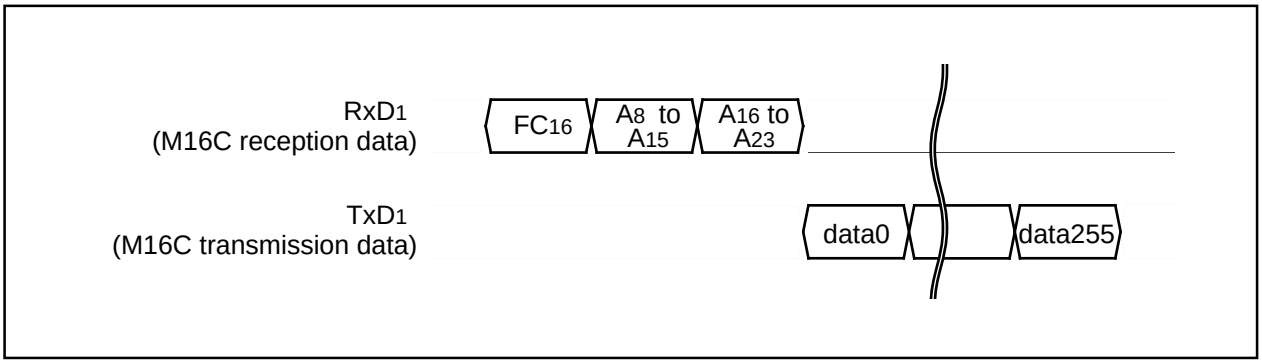


Figure 1.29.14. Timing for boot ROM area output

Read Check Data Command

This command reads the check data that confirms that the write data, which was sent with the page program command, was successfully received.

(1) Transfer the "FD16" command code with the 1st byte.

(2) The check data (low) is received with the 2nd byte and the check data (high) with the 3rd.

To use this read check data command, first execute the command and then initialize the check data. Next, execute the page program command the required number of times. After that, when the read check command is executed again, the check data for all of the read data that was sent with the page program command during this time is read. The check data is the result of CRC operation of write data.

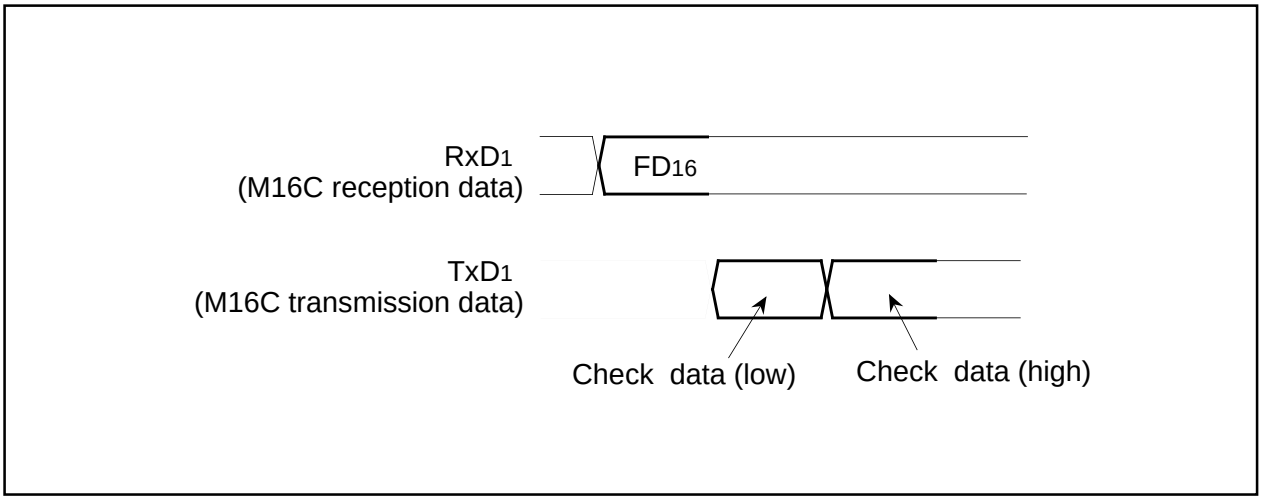


Figure 1.29.15. Timing for the read check data

Baud Rate 9600 Command

This command changes baud rate to 9,600 bps. Execute it as follows.

(1) Transfer the "B016" command code with the 1st byte.

(2) After the "B016" check code is output with the 2nd byte, change the baud rate to 9,600 bps.

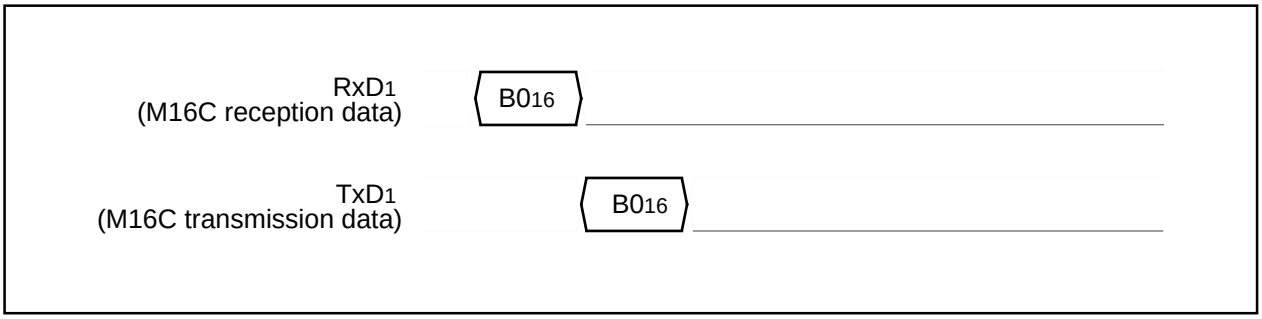


Figure 1.29.16. Timing of baud rate 9600

Baud Rate 19200 Command

This command changes baud rate to 19,200 bps. Execute it as follows.

(1) Transfer the "B1$_{16}$" command code with the 1st byte.

(2) After the "B1$_{16}$" check code is output with the 2nd byte, change the baud rate to 19,200 bps.

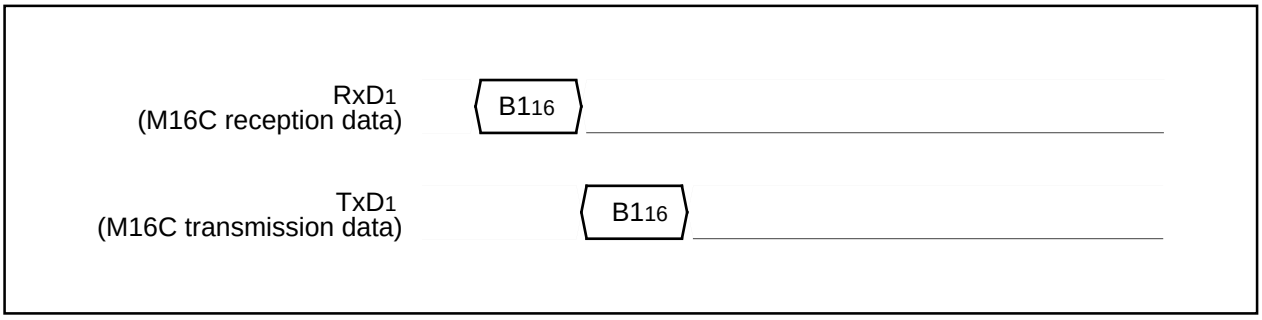


Figure 1.29.17. Timing of baud rate 19200

Baud Rate 38400 Command

This command changes baud rate to 38,400 bps. Execute it as follows.

(1) Transfer the "B2$_{16}$" command code with the 1st byte.

(2) After the "B2$_{16}$" check code is output with the 2nd byte, change the baud rate to 38,400 bps.

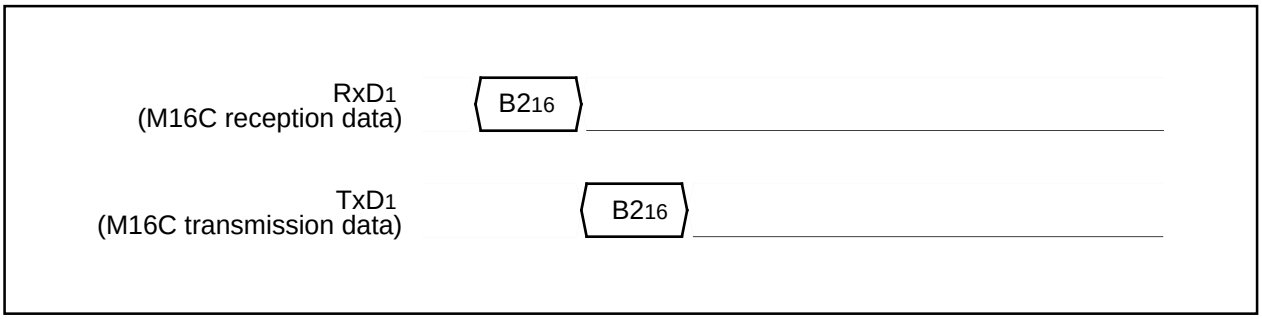


Figure 1.29.18. Timing of baud rate 38400

ID Code

When the flash memory is not blank, the ID code sent from the peripheral units 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 peripheral units is not accepted. An ID code contains 8 bits of data. Area is, from the 1st byte, addresses 0FFFDF16, 0FFFE316, 0FFFEB16, 0FFFEF16, 0FFFF316, 0FFFF716 and 0FFFFB16. Write a program into the flash memory, which already has the ID code set for these addresses.

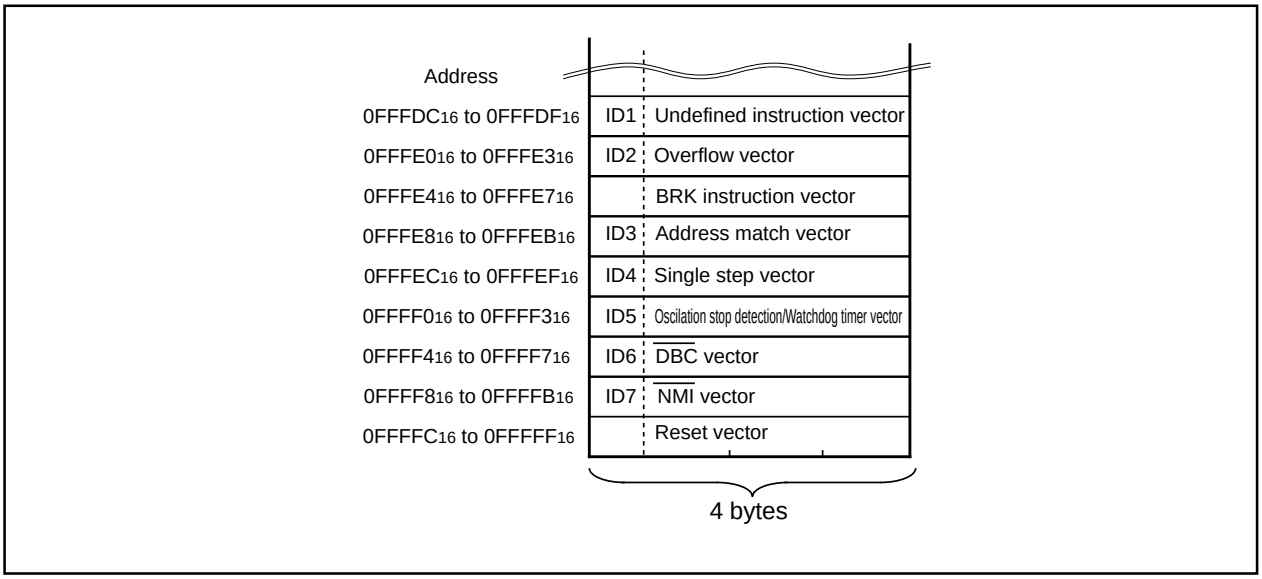
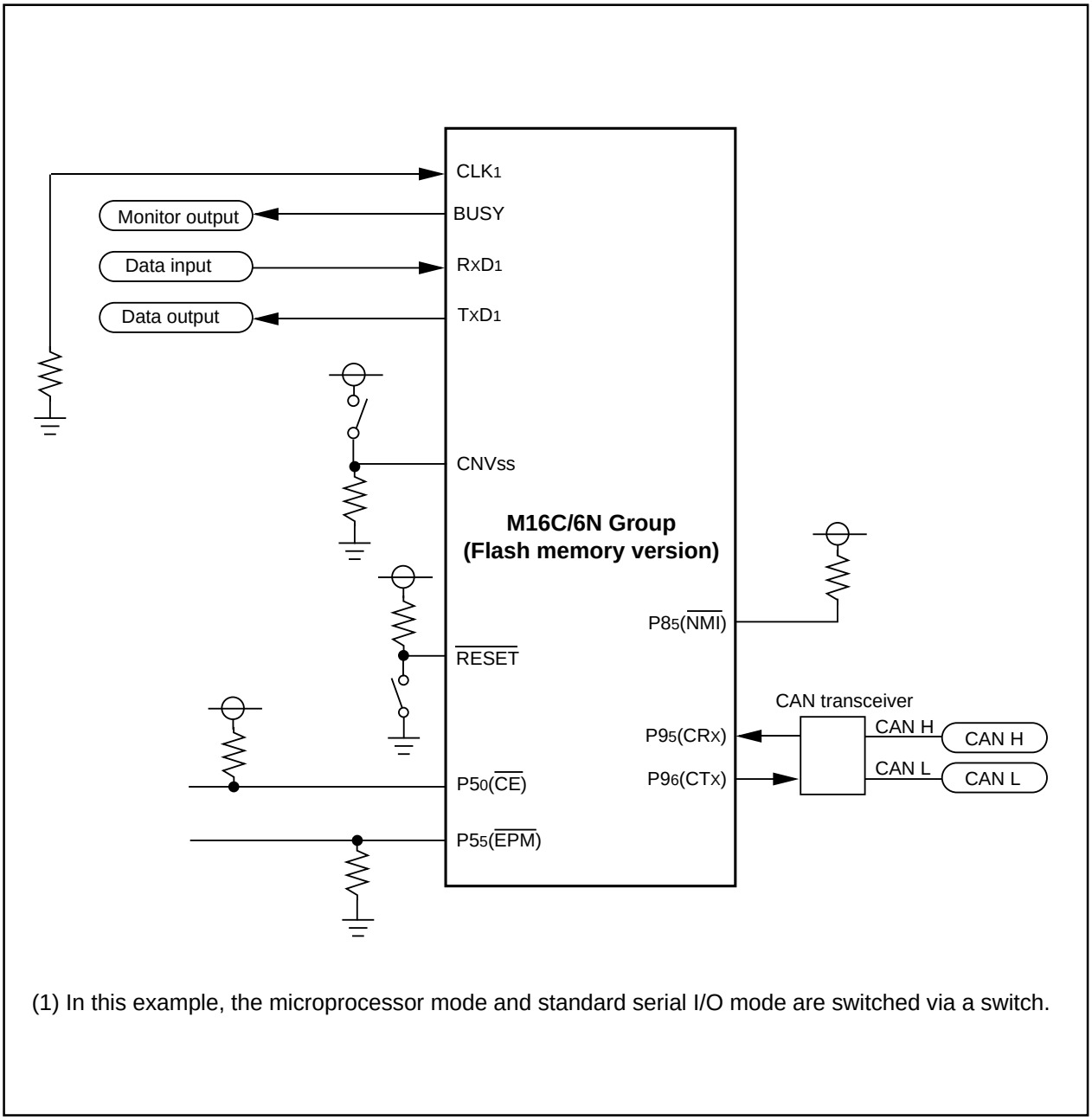


Figure 1.29.19. ID code storage addresses

Example Circuit Application for the Standard Serial I/O Mode 2

The below figure shows a circuit application for the standard serial I/O mode 2.



(1) In this example, the microprocessor mode and standard serial I/O mode are switched via a switch.

Figure 1.29.20. Example circuit application for the standard serial I/O mode 2

CAN I/O Mode

In CAN I/O mode, output and input of the software command, address, and data required for the operations (read, program, erase, etc.) are performed to the internal flash memory. An exclusive programmer is used for this purpose.

CAN I/O mode, unlike parallel I/O mode, the CPU controls operations such as rewrite (in CPU rewrite mode) in the flash memory and CAN input for rewrite data. CAN I/O mode is started when the CNVss pin and the $P5_0$ ($\overline{CE}$) pin are pulled up to "H" level, the $P5_5$ ($\overline{EPM}$) pin is pulled down to "L" level, and reset is released. (In normal microprocessor mode, pull the CNVss pin down to "L" level.)

The control program is written in the boot ROM area when the device is shipped from Mitsubishi. Note, if the boot ROM area is rewritten in the parallel I/O mode, the CAN I/O mode cannot be used.

Figure 1.30.1 shows the pin connections for CAN I/O mode.

Two CAN pins are used for input and output of the CAN data: $P9_6$ (CTx) pin and $P9_5$ (CRx) pin. The $P9_6$ (CTx) pin and the $P9_5$ (CRx) pin have to be connected to the CAN transceiver IC.

In CAN I/O mode, only the user ROM area shown in Figure 1.28.17 can be rewritten; the boot ROM area cannot.

CAN I/O mode has a 7 byte ID code. When the flash memory is not blank and the ID code does not match, the command sent from the external equipment (programmer) cannot be accepted.

Pin Functions

Pin	Name	I/O	Description
V_{CC},V_{SS}	Power input		Apply program/erase guaranteed voltage to Vcc pin and 0 V to Vss pin.
CNV_{SS}	CNV_{SS}	I	Connect this pin to Vcc pin.
$\overline{RESET}$	Reset input	I	Reset input pin: While reset is at "L" level, 20 cycles or more clock must be input to X_{IN} pin.
X_{IN}	Clock input	I	Connect a ceramic resonator or a quartz crystal oscillator between X_{IN} and X_{OUT} pins. To input an externally generated clock, input it to X_{IN} pin and leave X_{OUT} pin open.
X_{OUT}	Clock output	O	
BYTE	BYTE	I	Connect to Vcc or Vss.
AV_{CC}, AV_{SS}	Analog power supply input	I	Connect AVss to Vss and AVcc to Vcc, respectively
V_{REF}	Reference voltage input	I	Reference voltage input pin for AD converter.
$P0_0$ to $P0_7$	Input port P0	I	Input "H" or "L" level or open.
$P1_0$ to $P1_7$	Input port P1	I	Input "H" or "L" level or open.
$P2_0$ to $P2_7$	Input port P2	I	Input "H" or "L" llevel or open.
$P3_0$ to $P3_7$	Input port P3	I	Input "H" or "L" level or open.
$P4_0$ to $P4_7$	Input port P4	I	Input "H" or "L" level or open.
$P5_0$	$\overline{CE}$ input	I	Input "H" level.
$P5_1$ to $P5_4$, $P5_6$, $P5_7$	Input port P5	I	Input "H" or "L" level or open.
$P5_5$	$\overline{EPM}$ input	I	Input "L" level.
$P6_0$ to $P6_4$, $P6_6$	Input port P6	I	Input "H" or "L" level or open.
$P6_5$	SCLK input	I	Connect to Vss.
$P6_7$	TxD output	O	Connect to Vcc or open.
$P7_0$ to $P7_7$	Input port P7	I	Input "H" or "L" level or open.
$P8_0$ to $P8_4$ $P8_6$, $P8_7$	Input port P8	I	Input "H" or "L" level or open.
$P8_5$	$\overline{NMI}$ input	I	Connect to Vcc.
$P9_0$ to $P9_4$, $P9_7$	Input port P9	I	Input "H" or "L" level or open.
$P9_5$	CRx input	I	Connect to a CAN transceiver.
$P9_6$	CTx Output	O	Connect to a CAN transceiver.
$P10_0$ to $P10_7$	Input port P10	I	Input "H" or "L" level or open.

Appendix CAN I/O Mode (Flash Memory Version)

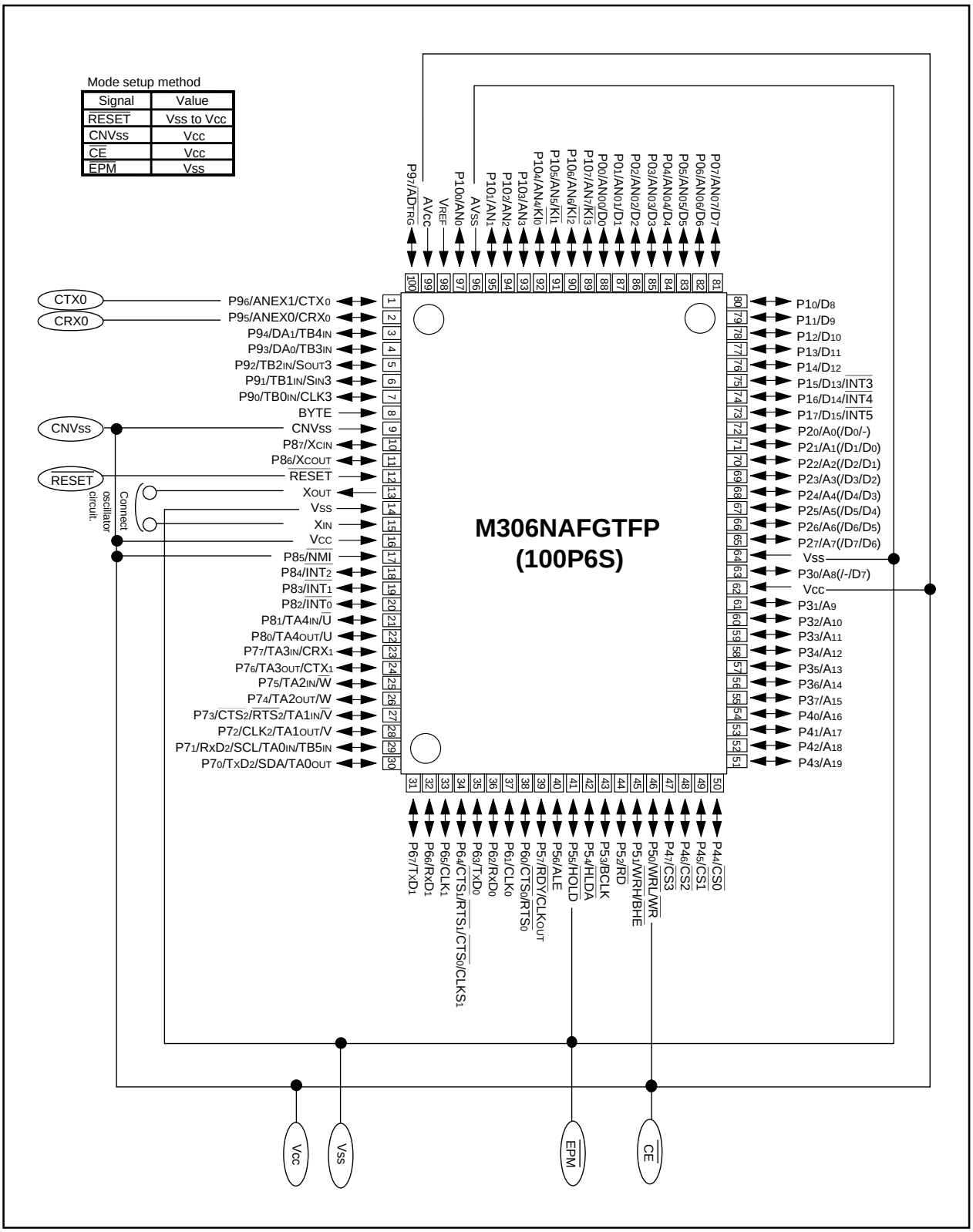


Figure 1.30.1. Pin connections for CAN I/O mode

Upon entering CAN I/O mode, the microcomputer with the internal flash memory selects automatically the CAN baud rate. Table 1.30.1 lists baud rate that can be automatically selected.

Table 1.30.1. List of selectable CAN baudrate

Baud rate	Clock frequency of M16C/6N group microcomputer with the internal flash memory						
	16MHz	10MHz	8MHz	5MHz	4MHz	2.5MHz	2MHz
1000kbps	√	—	—	—	—	—	—
500kbps	√	√	√	—	—	—	—
250kbps	√	√	√	√	√	—	—
125kbps	√	√	√	√	√	√	√
100kbps	√	√	√	√	√	—	√
83.3kbps	√	√	√	√	√	√	√
80kbps	√	—	√	—	√	—	—
40kbps	√	√	√	—	√	—	√
20kbps	√	√	√	√	√	—	√
10kbps	√	√	√	√	√	√	√

√ : Selectable

When a configuration remote frame (see Table 1.30.2) transmitted from an external equipment (programmer) is received and CAN baud rate configuration is completed, the M16C/6N group microcomputer with the internal flash memory transmits a configuration data frame.

Table 1.30.2. List of CAN frame at the time of baudrate selection

Frame	Type	Format	ID	DLC (Note 1)	Data contents
Configuration remote frame	Remote	Standard	7F0$_{16}$	0	Transmit at the time of baud rate selection
Configuration data frame	Data	Standard	7F0$_{16}$	0	Transmit when M16C/6N group completes automatic baud rate selection

Note 1: DLC indicates the number of the transfer data byte.
Note 2: The shadowed part is a transfer frame from the microcomputer with the internal flash memory to an external equipment. Otherwise a transfer frame from the external equipment to the microcomputer with the internal flash memory.

Table 1.30.3 lists IDs for the CAN data transfer that the external equipment uses. Here, all are standard data frames.

Table 1.30.3. ID for CAN data transfer that the external equipment uses

Frame	ID	DLC (Note 1)	Data contents
Command	7FE$_{16}$	1 to 5	Command that an external equipment issues
Write data	7FF$_{16}$	0 to 8	Data that an external equipment transmits
Busy	7F1$_{16}$	0	To transmit when M16C/6N group accepts a command
Ready	7F2$_{16}$	0	To transmit when M16C/6N group is waiting for a command
Read data	7F3$_{16}$	0 to 8	Data that M16C/6N group transmits

Note 1: DLC indicates the number of the transfer data byte.
Note 2: The shadowed part is a transfer frame from the microcomputer with the internal flash memory to an external equipment. Otherwise a transfer frame from the external equipment to the microcomputer with the internal flash memory.

In CAN I/O mode, input and output of software commands, address, and data are performed between the microcomputer and the external equipment using a CAN interface.
Moreover, the data in a memory, status register, etc. can be read by read operation after a software command input. Status, such as operating status of the flash memory and whether program operation or erase operation is completed successfully or ended up in error, can be checked by reading the status register. An explanation of the software commands, status register, etc. are given below.

Software Command

Table 1.30.4 lists software commands and I/O CAN frames. In CAN I/O mode, erase operation, program, and reading are controlled by transferring software commands through the CAN bus. Software commands are explained here below.

Table 1.30.4. List of software commands and I/O frames

	Control command	Frame 1 (Note 1)	Frame 2	Frame 3 ~	The last frame	When ID is not verified
1	Page read	Command	Busy	Write data 32 times	Ready	Not accepted
2	Page program	Command	Busy	Write data 33 times	Ready	Not accepted
3	Block erase	Command	Busy	—	Ready	Not accepted
4	Erase all unlocked blocks	Command	Busy	—	Ready	Not accepted
5	Read status register	Command	Busy	Read data (SRD, SRD1)(Note 2)	Ready	Accepted
6	Clear status register	Command	Busy	—	Ready	Not accepted
7	Read lock bit status	Command	Busy	Read data	Ready	Not accepted
8	Lock bit program	Command	Busy	—	Ready	Not accepted
9	Lock bit enable	Command	Busy	—	Ready	Not accepted
10	Lock bit disable	Command	Busy	—	Ready	Not accepted
11	ID check function	Command	Busy	Write data	Ready	Accepted
12	Download function	Command	Busy	Write data n times	Ready	Not accepted
13	Version information output function	Command	Busy	Read data	Ready	Accepted
14	Boot ROM area output function	Command	Busy	Read data 32 times	Ready	Not accepted

Note 1: See Table 1.30.5.
Note 2: SRD signifies Status Register Data, SRD1 signifies Status Register Data 1.
Note 3: The shadowed part is transfer frame from the microcomputer with the internal flash memory to the external equipment. Otherwise transfer frame from the external equipment to the microcomputer with the internal flash memory
Note 4: All the commands can be accepted on the blank devices.

Table 1.30.5. Contents of software command frame

Software command		Contents of command frame				
		1st byte	2nd byte	3rd byte	4th byte	5th byte
1	Page read	FF_{16}	Address (middle)	Address (high)	—	—
2	Page program	41_{16}	Address (middle)	Address (high)	—	—
3	Block erase	20_{16}	Address (middle)	Address (high)	$D0_{16}$	—
4	Erase all unlocked blocks	$A7_{16}$	$D0_{16}$	—	—	—
5	Read status register	70_{16}	—	—	—	—
6	Clear status register	50_{16}	—	—	—	—
7	Read lock bit status	71_{16}	Address (middle)	Address (high)	—	—
8	Lock bit program	77_{16}	Address (middle)	Address (high)	$D0_{16}$	—
9	Lock bit enable	$7A_{16}$	—	—	—	—
10	Lock bit disable	75_{16}	—	—	—	—
11	ID check function	$F5_{16}$	Address (low)	Address (middle)	Address (high)	ID size
12	Download function	FA_{16}	Size (low)	Size (high)	Check sum	—
13	Version information output function	FB_{16}	—	—	—	—
14	Boot ROM area output function	FC_{16}	Address (middle)	Address (high)	—	—

Page Read Command

A specified page (256 bytes) of the flash memory is read in unit of 8 byte by turns. Execute the command in the following procedure:

(1) Receive data of 3 bytes in the 1st frame: "FF$_{16}$", address A$_8$ to A$_{15}$, and A$_{16}$ to A$_{23}$.
(2) Transmit a busy frame on the 2nd frame.
(3) Read sequentially from address 0XXX00$_{16}$ on the 3rd to 34th frame, and transmit data at every 8 bytes.
(4) Transmit a ready frame on the 35th frame.

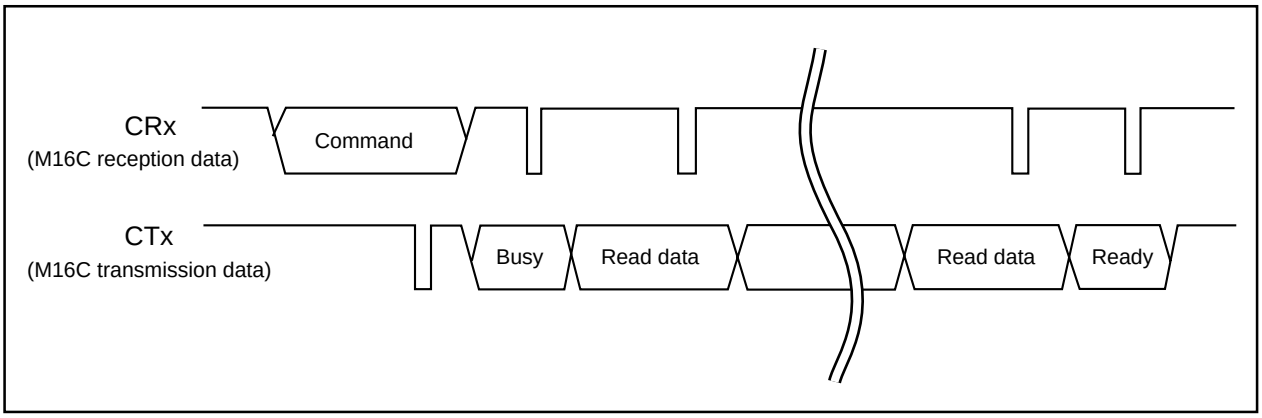


Figure 1.30.2. Timing of page read command

Page Program Command

This command serves for writing data in unit of 256 bytes, to the page (256 bytes) specified in the flash memory. The command should be executed in the following procedure:

(1) Receive data of 3 bytes in the 1st frame: "41$_{16}$", address A$_8$ to A$_{15}$, and A$_{16}$ to A$_{23}$.
(2) Transmit a busy frame on the 2nd frame.
(3) Receive the write data sequentially from address 0XXX00$_{16}$ on the 3rd to 34th frame.
(4) Receive DLC= "0" on the 35th frame.
(5) Transmit a ready frame on the 36th frame after completion of flash write operation.

The result of the page program can be known by reading the status register. Refer to the section of the status register for details.

In addition, write protect on each block can be realized by the lock bit. For further details, refer to the section of the data protection function. Program cannot be done again to the page which is already programmed.

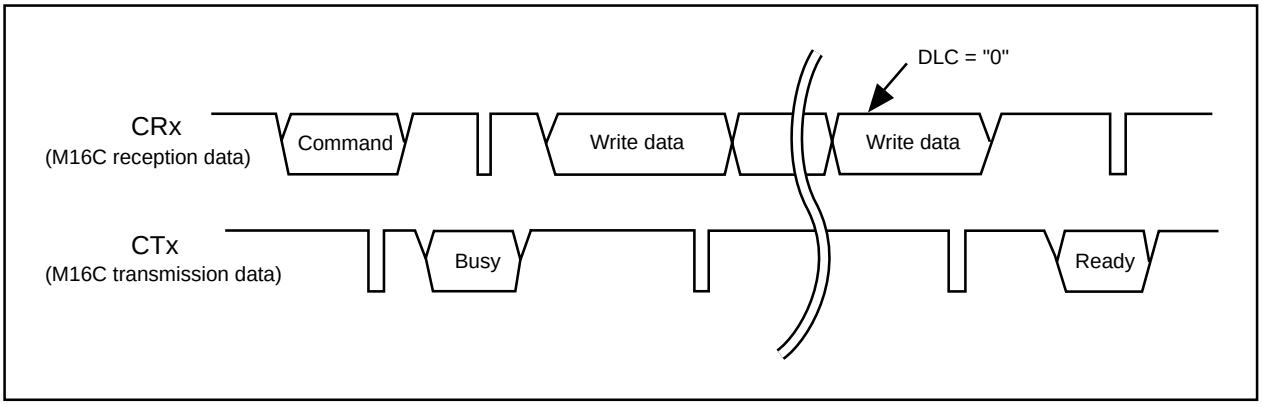


Figure 1.30.3. Timing of page program command

Block Erase Command

This command serves for erasing the data in the specified block. Execute the command in the following procedure:

(1) Receive data of 4 bytes in the 1st frame: "20_{16}", address A_8 to A_{15}, A_{16} to A_{23}, and "$D0_{16}$".
(2) Transmit a busy frame on the 2nd frame.
(3) Transmit a ready frame on the 3rd frame after completion of flash erase operation.

The result of block erase can be known by reading the status register after completion of block erase. Refer to the section of the status register for details. In addition, erase protect on each block can be realized by the lock bit. For details, refer to the section of data protection function.

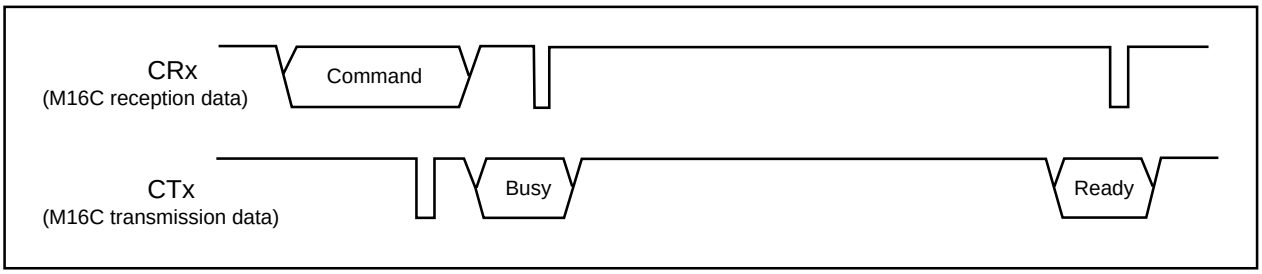


Figure 1.30.4. Timing of block erase command

Erase All Unlocked Blocks Command

This command serves for erasing the contents of all blocks. Execute the command in the following procedures.

(1) Receive data of 2 bytes in the 1st frame: "$A7_{16}$" and "$D0_{16}$".
(2) Transmit a busy frame on the 2nd frame.
(3) Transmit a ready frame on the 3rd frame after completion of erase all unlocked blocks operation.

The result of erase all unlocked blocks can be known by reading the status register after completion of erase all unlocked blocks. Refer to the section of the status register for details.

In addition, erase protect on each block can be realized by the lock bit. For details, refer to the section of data protection function.

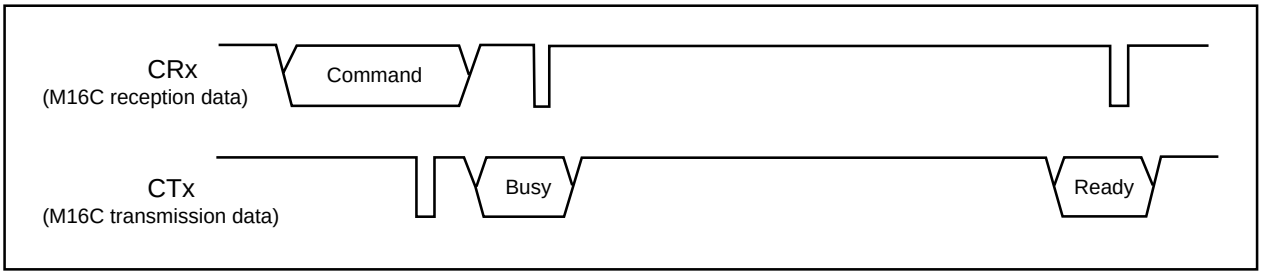


Figure 1.30.5. Timing of erase all unlocked blocks command

Read Status Register Command

This command serves for confirmation of the operating status of the flash memory. Execute the command in the following procedure:

(1) Receive data of 1 byte in the 1st frame: "70_{16}".
(2) Transmit a busy frame on the 2nd frame.
(3) Transmit 2 bytes on the 3rd frame: SRD and SRD1.
(4) Transmit a ready frame on the fourth frame.

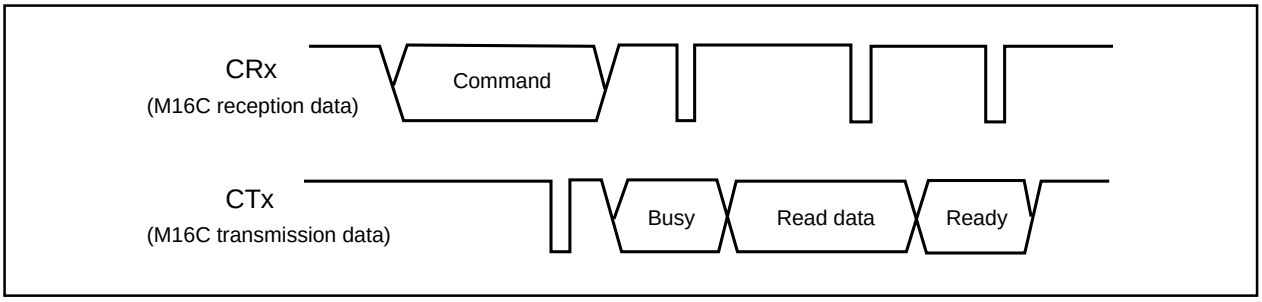


Figure 1.30.6. Timing of read status register command

Clear Status Register Command

This command serves for clearing the bits (SR4, SR5), that show the status register has been ended in error. Execute the command in the following procedure:

(1) Receive 1 byte of "50_{16}" in the 1st frame.
(2) Transmit a busy frame on the 2nd frame.
(3) Transmit a ready frame on the 3rd frame after completion of status register reset operation.

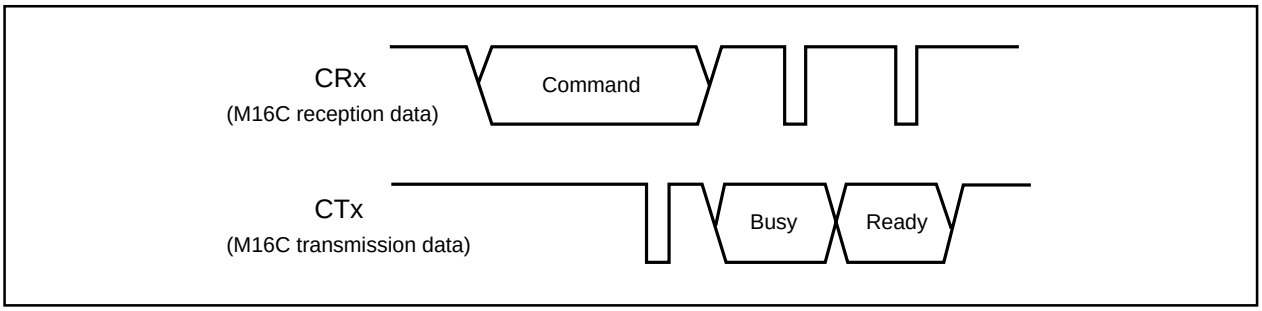


Figure 1.30.7. Timing of clear status register command

Read Lock Bit Status Command

This command serves for reading the state of the lock bit of a specified block. Execute the command in the following procedure:

(1) Receive 3 bytes in the 1st frame: "71_{16}", address A_8 to A_{15}, and A_{16} to A_{23}.
(2) Transmit a busy frame on the 2nd frame.
(3) Transmit 1 byte of lock bit data on the 3rd frame. A state is displayed at the 6th bit of the data. "1" means that the block is not locked, "0" means that the block is locked.
(4) Transmit a ready frame on the 4th frame.

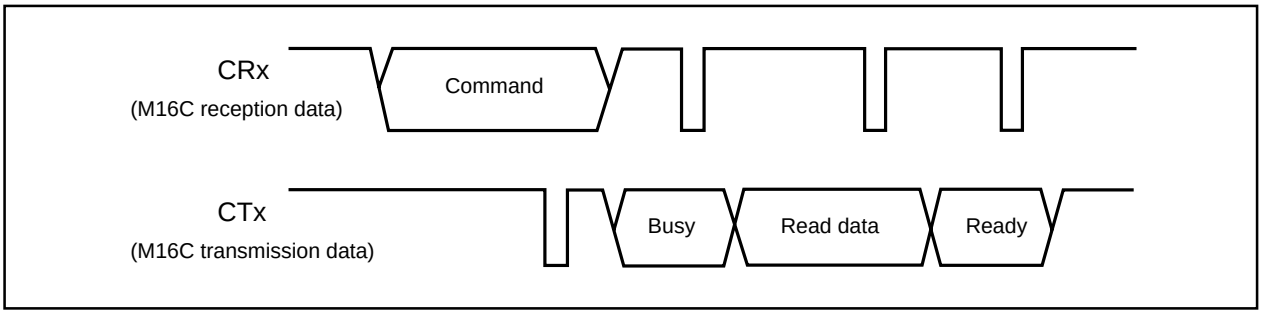


Figure 1.30.8. Timing of read lock bit status command

Lock Bit Program Command

This program serves for writing "0" (locked state) to the lock bit of a specified block. Execute the program in the following procedure:

(1) Receive 4 bytes in the 1st frame: "77_{16}", address A_8 to A_{15}, A_{16} to A_{23}, and "$D0_{16}$".
(2) Transmit a busy frame on the 2nd frame.
(3) Transmit a ready frame on the 3rd frame after completion of lock operation (clearing the lock bit to "0") to the specified block.

The state of a lock bit can be read by the read lock bit status command.

Refer to the section of data protection function about the function of a lock bit, reset method and others.

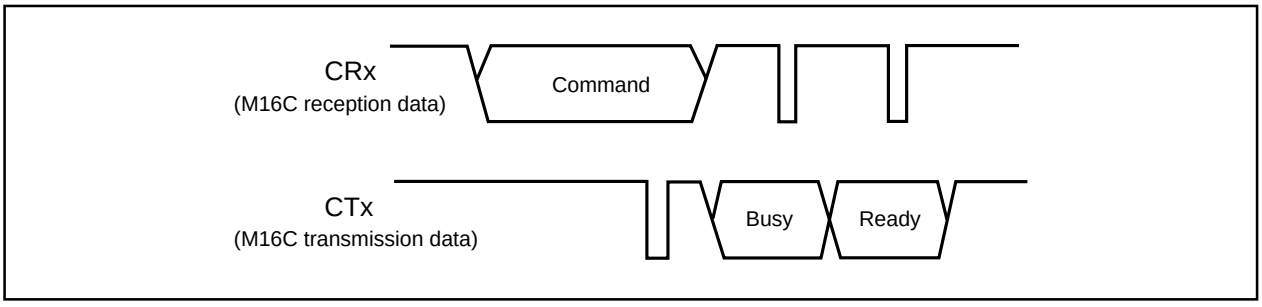


Figure 1.30.9. Timing of lock bit program command

Lock Bit Enable Command

This command enables again the lock bit in blocks whose bit has been disabled by the lock bit disable command. Execute the command in the following procedure:

(1) Receive 1 byte in the 1st frame: "$7A_{16}$".
(2) Transmit a busy frame on the 2nd frame.
(3) Transmit a ready frame on the 3rd frame after setting to enable the lock bit.

This command only enables the lock bit function, however, it does not either set or clear the lock bit itself.

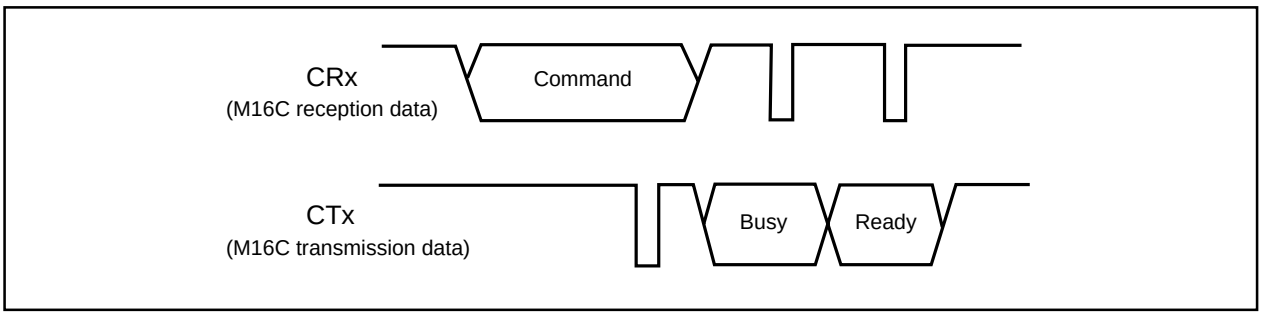


Figure 1.30.10. Timing of lock bit enable command

Lock Bit Disable Command

This command disables the block lock. Execute the command in the following procedure:

(1) Receive 1 byte in the 1st frame: "75_{16}".
(2) Transmit a busy frame on the 2nd frame.
(3) Transmit a ready frame on the 3rd frame after completion of lock bit disable selection.

This command only disables the lock bit function, however, it does not either set or clear the lock bit itself. The lock bit data which has been "0" (locked state) is set to "1" (unlocked sate) after completion of erase operation, when erase is performed after lock bit disable command execution. The lock bit enables after reset release.

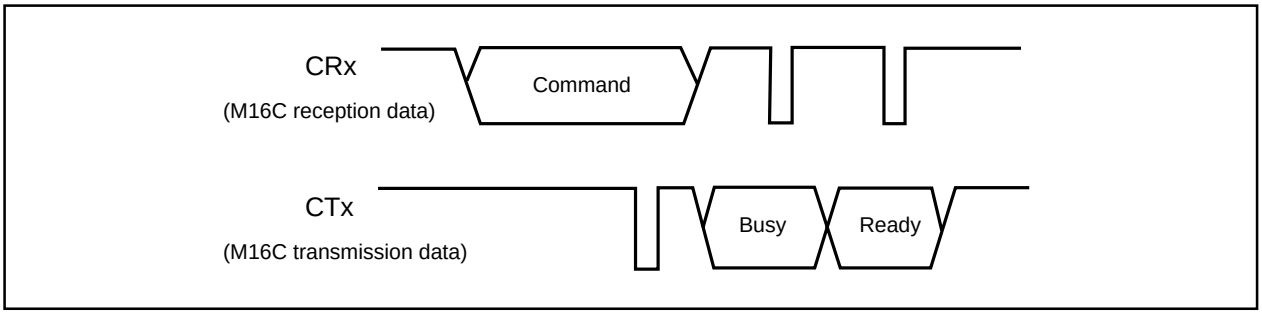


Figure 1.30.11. Timing of lock bit disable command

ID Check Function Command

This command serves for judging the ID code. Execute the command in the following procedure:

(1) Receive 5 bytes in the 1st frame: "F5$_{16}$", address A$_0$ to A$_7$, A$_8$ to A$_{15}$, A$_{16}$ to A$_{23}$, and the number of ID data.
(2) Transmit a busy frame on the 2nd frame.
(3) Receive the ID data in the 3rd frame and on.
(4) Receive, by DLC = "0" after completion of ID data transmission.
(5) Transmit a ready frame on the last frame after completion of ID check.

It takes up to a maximum of 1 sec., from the transmission of the ID data on the 3rd frame to the reception of the next ready frame.

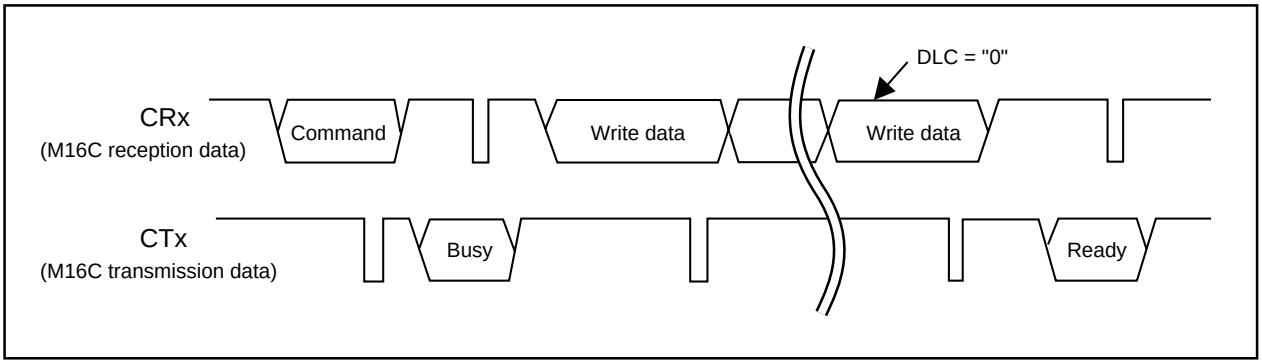


Figure 1.30.12. Timing of ID check function command

Download Function Command

This command serves for download of an execution program to RAM. Execute the command in the following procedure:

(1) Transmit 4 bytes in the 1st frame: "FA$_{16}$", program size (low), program size (high), and check sum.
(2) Transmit a busy frame on the 2nd frame.
(3) Receive an execution program as a write data on the 3rd frame and after.
(4) Transmit a ready frame on the last frame after completion of check sum comparison.

A transfer program will be executed after completion of ready frame transmission, if a check sum matches. The size of the transmission program varies according to the internal RAM.

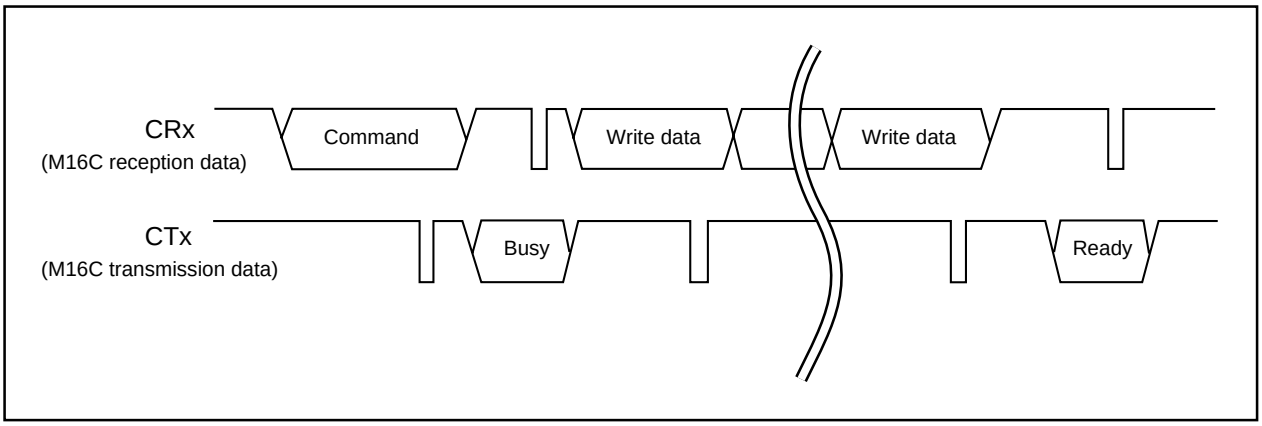


Figure 1.30.13. Timing of download function command

Version Information Output Function Command

The version information on the control program stored in the boot ROM area is output. Please perform it in the following procedures.

(1) Receive 1 byte in the 1st frame: "FB16".
(2) Transmit a busy frame on the 2nd frame.
(3) Transmit 8 bytes of version information on the 3rd frame.
(4) Transmit a ready frame on the 4th frame.

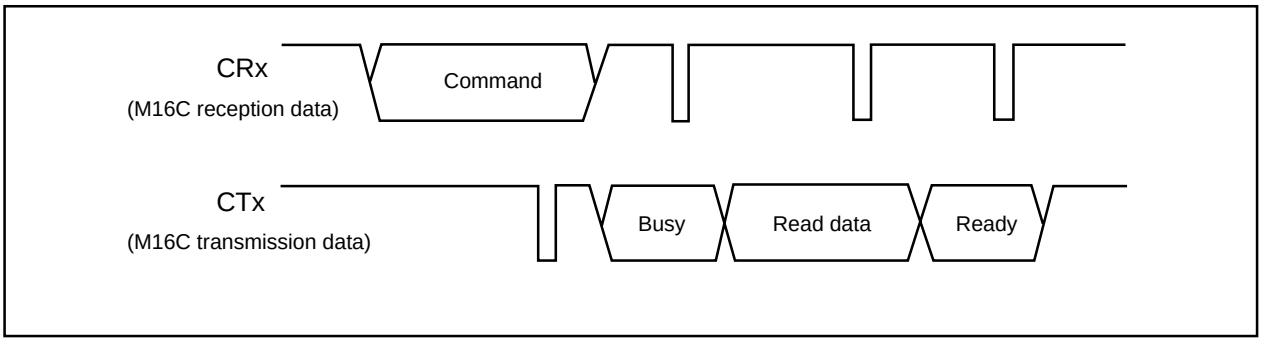


Figure 1.30.14. Timing of version information output function command

Boot ROM Area Output Function Command

This function serves for reading the control program stored in the boot ROM area in unit of page (256 bytes). Please perform it in the following procedure:

(1) Receive 3 bytes in the 1st frame: "FC16", address A8 to A15 and A16 to A23.
(2) Transmit a busy frame on the 2nd frame.
(3) Transmit the read data sequentially, starting from address 0XXX0016 from the 3rd frame to the 34th frame.
(4) Receive a ready frame at the 35th frame.

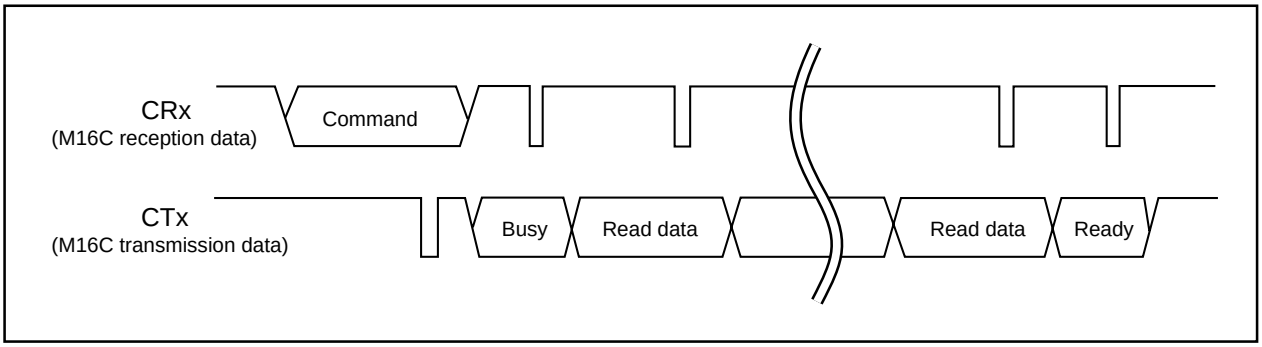


Figure 1.30.15. Timing of boot ROM area output function command

ID Code

If the contents of a flash memory are not blank, the microcomputer judges whether the ID code currently written in the flash memory matches the one that is sent from the external equipment. If they do not match, the command sent from the external equipment cannot be accepted. Each of the ID codes consists of 8 bit data, and the area spans, from the first byte and on, address 0FFFDF16, 0FFFE316, 0FFFEB16, 0FFFEF16, 0FFFF316, 0FFFF716, and 0FFFFB16 respectively. Write in the flash memory a program in which the ID codes are defined beforehand.

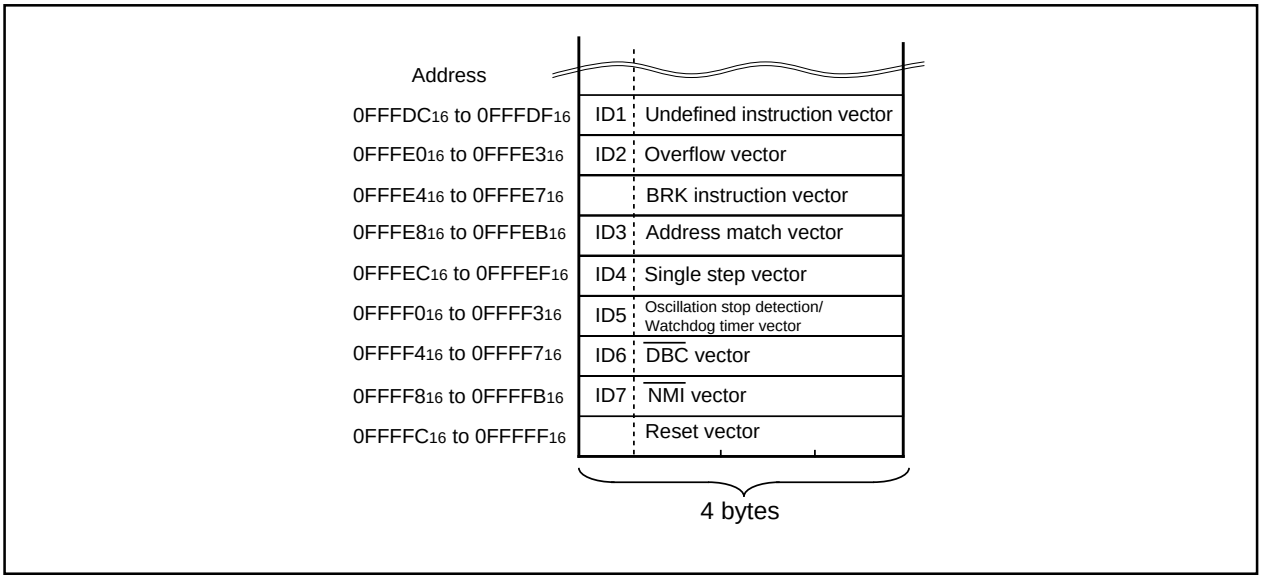


Figure 1.30.16. Storing address of ID code

Data Protection Function (block lock)

Each block shown in Figure 1.28.17 has a nonvolatile lock bit that selects the protection (block lock) against erase/write. Writing "0" (locked state) into a lock bit is done by lock bit program command. The lock bit of each block can be read by the read lock bit status command.

Disable or enable block lock is defined by the state of the lock bit and execution status of the lock bit disable command and the lock bit enable command.

(1) After reset release and after execution of the lock bit enable command, lock or unlock to the specified block can be selected, according to the lock bit state (lock bit data). A block of a lock bit data "0" is in locked state, and erase/write is disabled. On the other hand, a block of a lock bit data "1" is in unlocked state, and erase/write is enabled.

(2) After lock bit disable command execution, all the blocks are in unlocked state regardless of the lock bit data, and erase/write is enabled. At this time, the lock bit data which has been "0" (locked state) is set to "1" (unlocked state) after completion of erase, and the lock by the lock bit is released.

Status Register (SRD)

The status register indicates operating status of the flash memory and status such as whether an erase or a program operation has been 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}). The status register outputs "80_{16}" after reset release.

Table 1.30.6 shows the definition of each status register bit, definition of each bit is given below.

Table 1.30.6. Status register (SRD)

SRD bits	Status name	Definition	
		"1"	"0"
SR7 (bit7)	Write state machine (WSM) status	Ready	Busy
SR6 (bit6)	Reserved	-	-
SR5 (bit5)	Erase status	Terminated in error	Terminated normally
SR4 (bit4)	Program status	Terminated in error	Terminated normally
SR3 (bit3)	Block status after program	Terminated in error	Terminated normally
SR2 (bit2)	Reserved	-	-
SR1 (bit1)	Reserved	-	-
SR0 (bit0)	Reserved	-	-

Write State Machine (WSM) Status (SR7)

The write state machine (WSM) status 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 Status (SR5)

The erase status indicates the operating status of an auto erase operation. It is set to "1" when an erase error occurs. When the erase status is cleared, the value becomes "0".

Program Status (SR4)

The program status indicates the operating status of an auto write operation. It is set to "1" when an program error occurs. When the program status is cleared, the value becomes "0".

Block Status After Program (SR3)

This is set to "1" if excessive write (phenomenon whereby the memory cell enters depletion state which results in data not being read correctly) occurs at completion of page write. Namely, when write is completed successfully, the status register is at "80_{16}"; the status register is at "90_{16}" when write fails; it is at "88_{16}" when excessive write occurs.

If any of SR5, SR4 and SR3 is set to "1", the page program, block erase, erase all unlocked blocks and lock bit program commands are not accepted. Before executing these commands, execute clear status register command (50_{16}) to clear the status.

Status Register 1 (SRD1)

Status register 1 indicates the status of serial communication, results of ID checks, and result of checksum 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}).

The value of the register is "00_{16}" when power is turned on and the flag status is maintained even after the reset.

Table 1.30.7 shows the status register 1, definition of each bit is given below.

Table 1.30.7. Status register 1 (SRD1)

SRD1 bits	Status name	Definition	
		"1"	"0"
SR15 (bit7)	Boot updating completed bit	Updated	Not updated
SR14 (bit6)	Reserved	—	—
SR13 (bit5)	Reserved	—	—
SR12 (bit4)	Checksum match bit	Matched	Not matched
SR11 (bit3) SR10 (bit2)	ID check completed bits	b3 b2 0 0: Not verified 0 1: Verified as mismatched 1 0: Reserved 1 1: Verified	
SR9 (bit1)	Data reception time out	Time out	Successful operation
SR8 (bit0)	Reserved	—	—

Boot Update Completed Bit (SR15)

This bit indicates whether the control program has been downloaded to the RAM by the download function or not.

Checksum Match Bit (SR12)

This bit indicates whether the checksum matches or not when an execution program is downloaded by the download function.

ID Check Completed Bits (SR11, SR10)

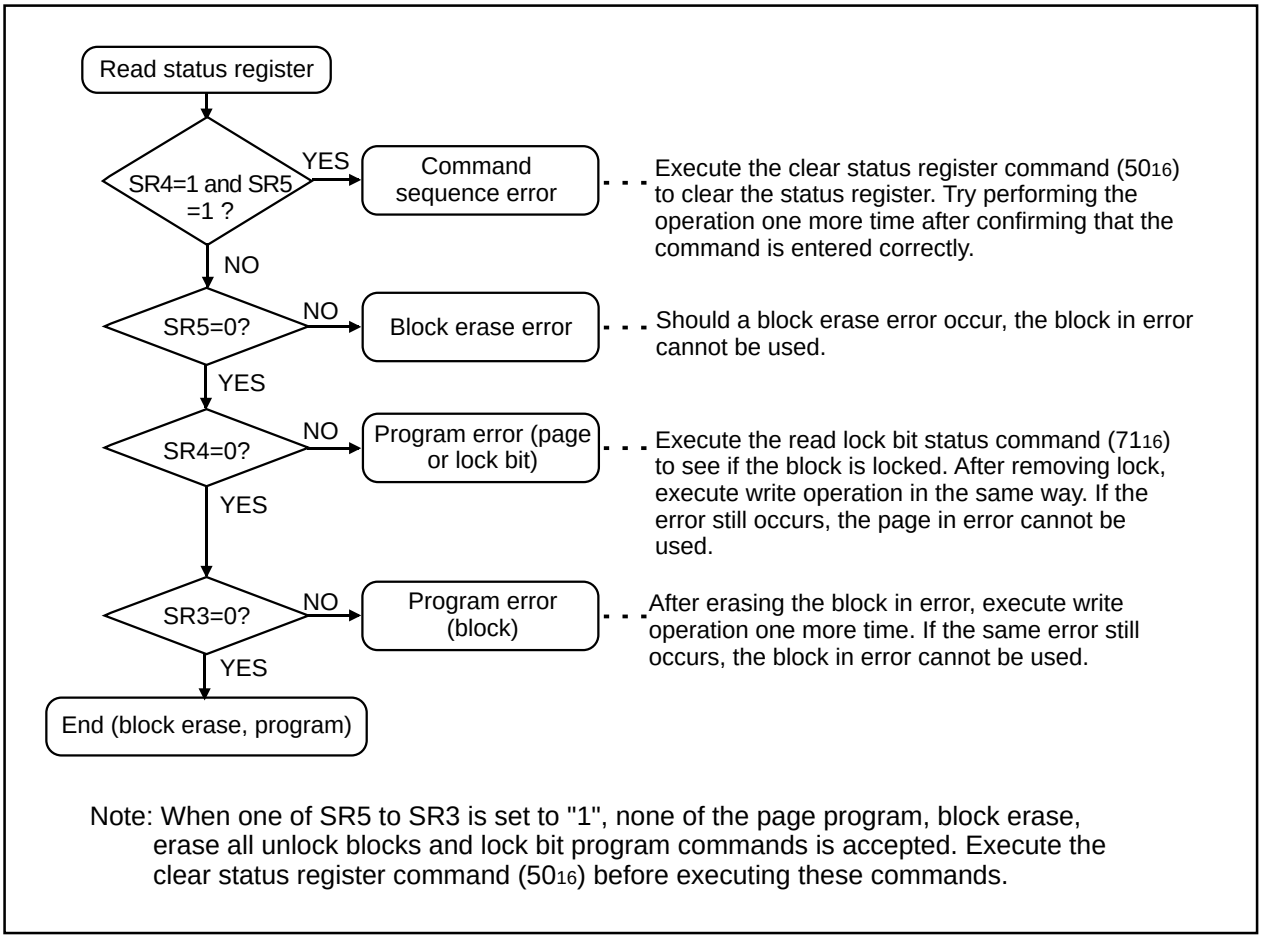
These bits indicate the result of ID checks. Some commands cannot be accepted without an ID check.

Data Reception Time Out (SR9)

This bit indicates a time out error during data reception. If this flag is set during data reception, the received data is discarded and the microcomputer returns to the command wait state.

Full Status Check

Execution results of erase and program operations can be known by full status check. Figure 1.30.17 shows a flowchart of the full status check and explains how to handle errors when they occur.

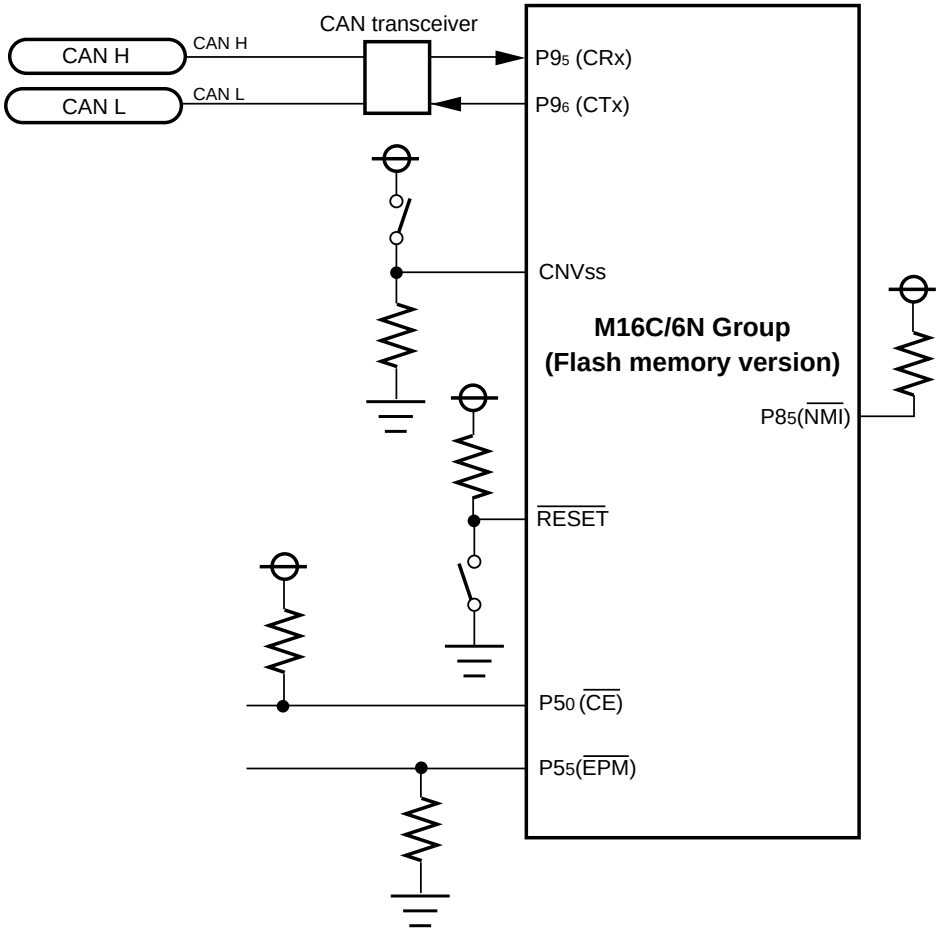


Note: When one of SR5 to SR3 is set to "1", none of the page program, block erase, erase all unlock blocks and lock bit program commands is accepted. Execute the clear status register command (5016) before executing these commands.

Figure 1.30.17. Full status check flowchart and remedial procedure for errors

Example Circuit Application for the CAN I/O Mode

The figure blow shows a circuit application for the CAN I/O mode. Control pins may vary according to programmer, therefore see the programmer manual for more information.



Note 1: Control pins and external circuitry may vary according to the external equipment (writer). Refer to the using manual of the exclusive programmer for more details.

Note 2: In this example, switching between microprocessor mode and CAN I/O mode is done by a switch.

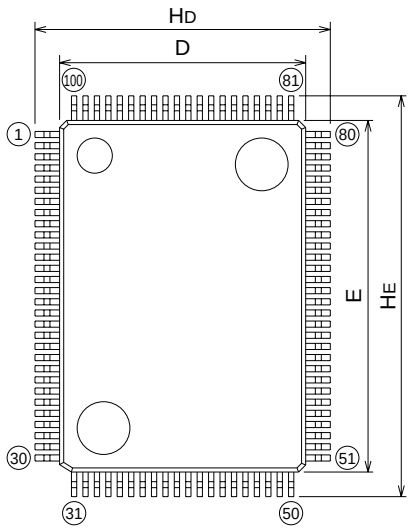
Figure 1.30.18. Example circuit application for the CAN I/O mode

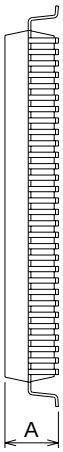
Package Outline

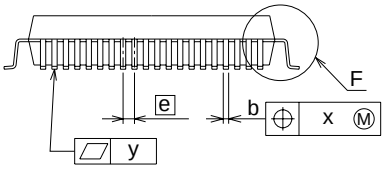
100P6S-A (MMP)

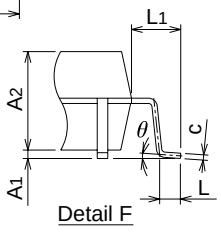
Plastic 100pin 14×20mm body QFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP100-P-1420-0.65	–	1.58	Alloy 42

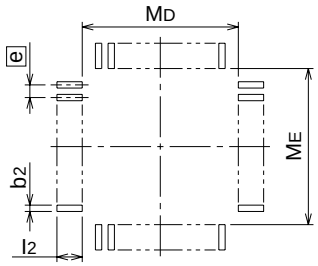








Detail F



Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	3.05
A1	0	0.1	0.2
A2	–	2.8	–
b	0.25	0.3	0.4
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	19.8	20.0	20.2
e	–	0.65	–
HD	16.5	16.8	17.1
HE	22.5	22.8	23.1
L	0.4	0.6	0.8
L1	–	1.4	–
x	–	–	0.13
y	–	–	0.1
θ	0°	–	10°
b2	–	0.35	–
l2	1.3	–	–
MD	–	14.6	–
ME	–	20.6	–

Renesas Technology Corp.
Nippon Bldg.,6-2,Otemachi 2-chome,Chiyoda-ku,Tokyo,100-0004 Japan

RENESAS
Renesas Technology Corp.

REVISON HISTORY	M16C/6N0/6N1 GROUP DATA SHEET

REV.	Date	Description	
		Page	Summary
1.0	11/15/02		Full-fledged revision