

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



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## Table of contents

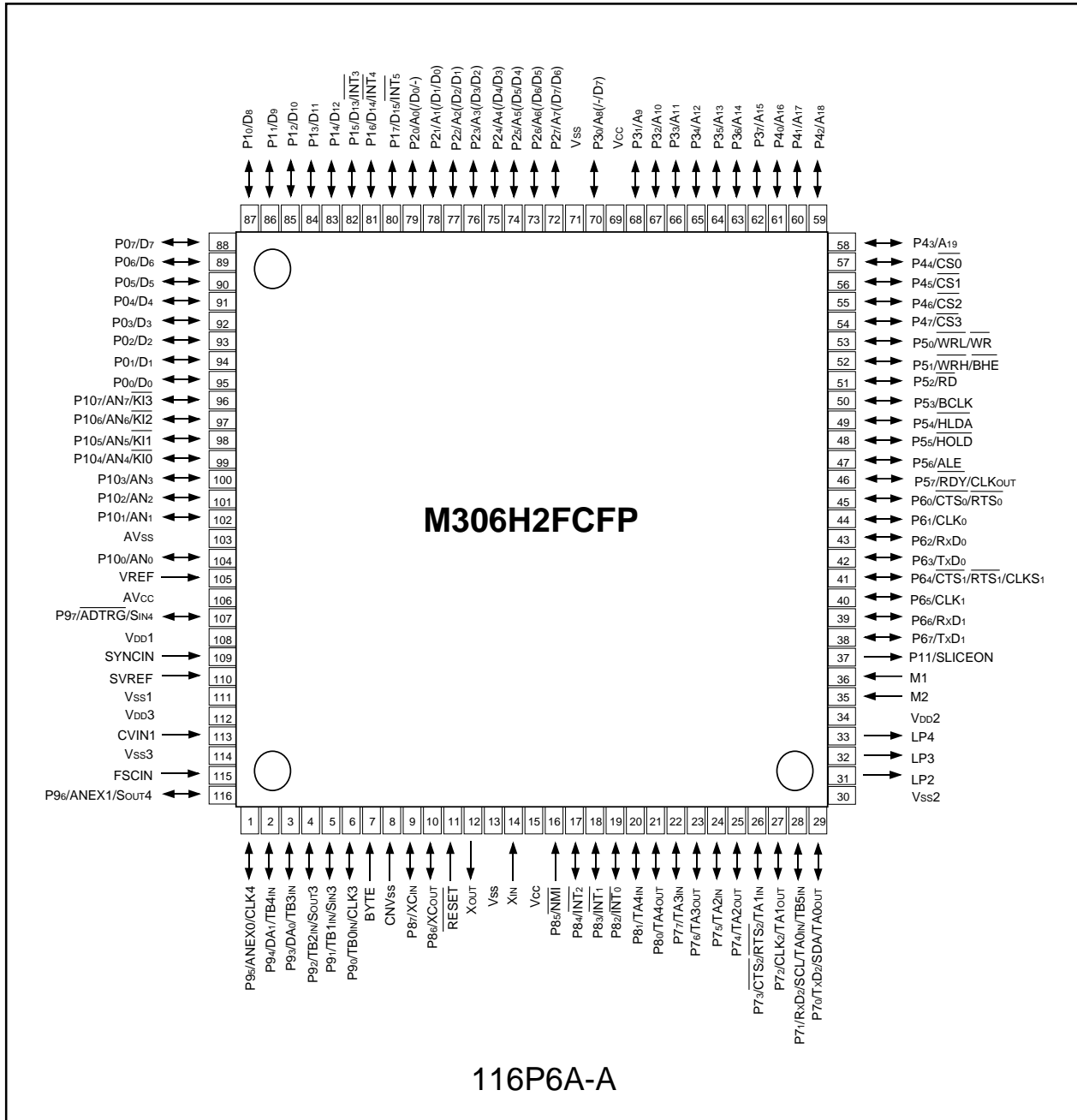
1. DESCRIPTION .....	1
1.1 Features .....	1
1.2 Applications .....	1
1.3 Pin Configuration .....	3
1.4 Block Diagram .....	4
1.5 Performance Outline .....	5
2. OPERATION OF FUNCTIONAL BLOCKS .....	9
2.1 Memory .....	9
2.2 Central Processing Unit (CPU) .....	13
2.3 Reset .....	16
2.4 Processor Mode .....	20
2.5 Clock Generating Circuit .....	31
2.6 Protection .....	40
2.7 Interrupt .....	41
2.8 Watchdog Timer .....	61
2.9 DMAC .....	63
2.10 Timer .....	73
2.11 Serial I/O .....	91
2.12 A-D Converter .....	132
2.13 D-A Converter .....	142
2.14 CRC Calculation Circuit .....	144
2.15 Expansion Function .....	146
2.16 Programmable I/O Ports .....	170
3. USAGE PRECAUTION .....	180
4. ELECTRICAL CHARACTERISTICS .....	185
5. FLASH MEMORY .....	203
5.1 Outline Performance .....	203
5.2 Flash Memory mode .....	204
5.3 CPU Rewrite Mode .....	205
5.4 Functions To Inhibit Rewriting Flash Memory Version .....	215
5.5 Parallel I/O Mode .....	217
5.6 Standard serial I/O mode .....	218
5.7 Absolute maximum ratings .....	246
6. PACKAGE OUTLINE .....	248
7. DIFFERENCES BETWEEN M306H2MC-XXXFP AND M306H2FCFP .....	249

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

### 1.3 Pin Configuration

Figures 1.3.1 shows the pin configuration (top view).



**Figure 1.3.1 Pin configuration (top view)**

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 1.4 Block Diagram

Figure 1.4.1 is a block diagram of the M306H2FCFP.

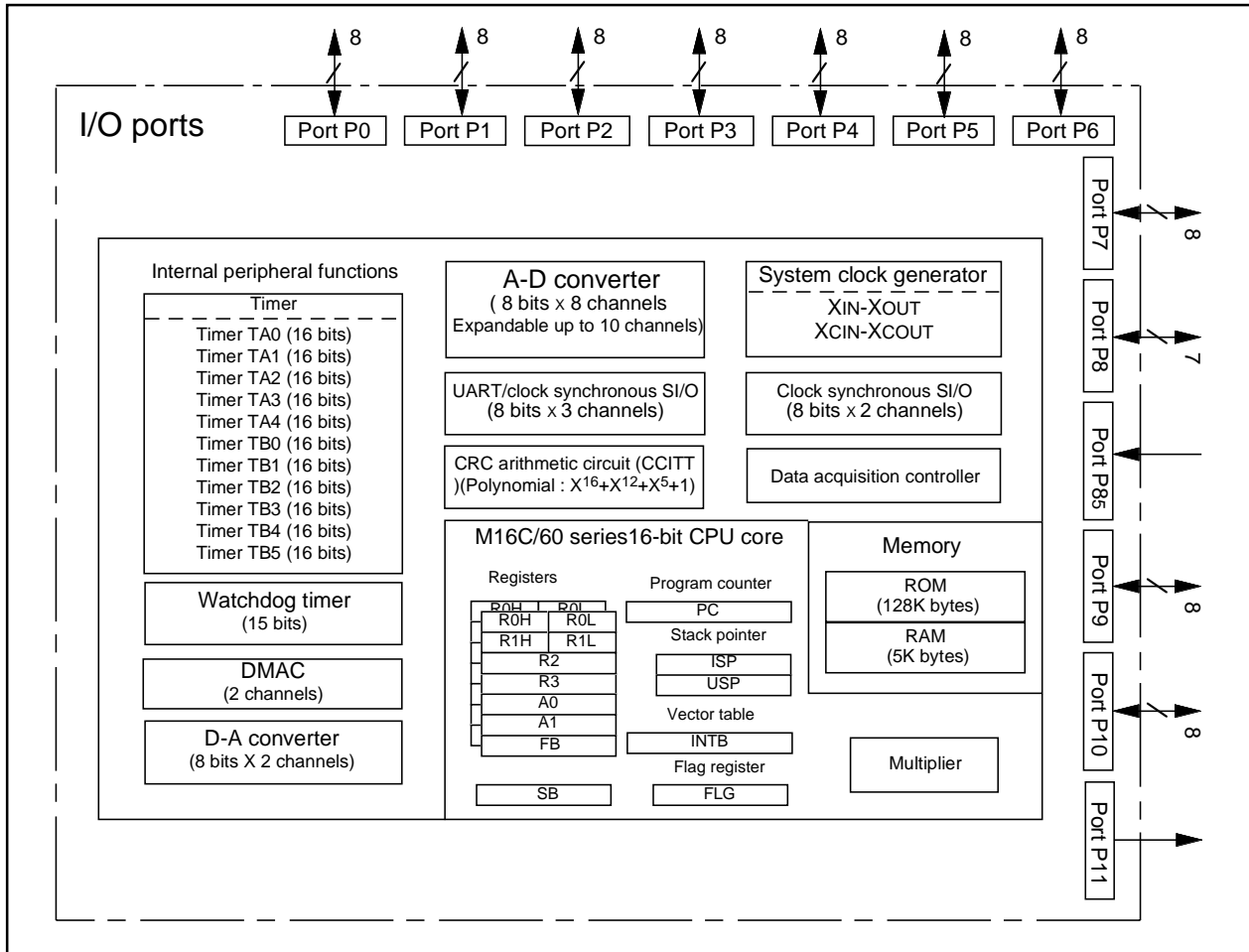


Figure 1.4.1 Block diagram of M306H2FCFP

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 1.5 Performance Outline

Table 1.5.1 is a performance outline of M306H2FCFP.

**Table 1.5.1 Performance outline of M306H2FCFP**

Item		Performance
Number of basic instructions		91 instructions
Shortest instruction execution time		100ns (f(XIN)=10MHz)
Memory capacity	ROM	128K bytes
	RAM	5K bytes
I/O port	P0 to P10 (except P85)	8 bits X 10, 7 bits X 1
Input port	P85	1 bit X 1
Output port	P11	1 bit X 1
Multifunction timer	TA0, TA1, TA2, TA3, TA4	16 bits X 5
	TB0, TB1, TB2, TB3, TB4, TB5	16 bits X 6
Serial I/O	UART0, UART1, UART2	(UART or clock synchronous) x 3
	SI/O3, SI/O4	(Clock synchronous) x 2
A-D converter		8 bits X (8 + 2) channels
D-A converter		8 bits X 2 channels
DMAC		2 channels (trigger: 24 sources)
CRC calculation circuit		CRC-CCITT
Watchdog timer		15 bits X 1 (with prescaler)
Interrupt		25 internal and 8 external sources, 4 software sources, 7 levels
Clock generating circuit		2 built-in clock generation circuits (built-in feedback resistor, and external ceramic or crystal oscillator)
Supply voltage		4.75V to 5.25V (at f(XIN)=10MHz) 2.80V to 5.25V(at f(XCIN)=32kHz, Only low power dissipation mode)
Device configuration		CMOS high performance silicon gate
Package		116-pin plastic mold QFP
Data acquisition	Slice RAM	864 Bytes (48 X 18 X 8-bit)
	Data acquisition circuit	for PDC, VPS, EPG-J, XDS and WSS

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

**Table 1.5.2 Pin Description**

Pin name	Signal name	I/O type	Function
VCC, VSS	Power supply input		Supply 4.75 to 5.25 V to the Vcc pin. Supply 0 V to the Vss pin.
CNVss	CNVss	Input	Connect this pin to Vss.
RESET	Reset input	Input	A "L" on this input resets the microcomputer.
XIN XOUT	Clock input Clock output	Input Output	These pins are provided for the main clock generating circuit. Connect a ceramic resonator or crystal between the XIN and the XOUT pins. To use an externally derived clock, input it to the XIN pin and leave the XOUT pin open.
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". Connect this pin to the Vss pin when not using external data bus.
AVCC	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to VCC.
AVSS	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to Vss.
VREF	Reference voltage input	Input	This pin is a reference voltage input for the A-D converter.
P00 to P07	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 used for input in signal-chip mode, the port can be set to have or not have a pull-up resistor in units of four bits by software. In memory expansion mode, selection of the internal pull-resistor is not available.
D0 to D7		Input/output	When set as a separate bus, these pins input and output data (D0–D7).
P10 to P17	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.
D8 to D15		Input/output	When set as a separate bus, these pins input and output data (D8–D15).
P20 to P27	I/O port P2	Input/output	This is an 8-bit I/O port equivalent to P0.
A0 to A7		Output	These pins output 8 low-order address bits (A0–A7).
A0/D0 to A7/D7		Input/output	If the external bus is set as an 8-bit wide multiplexed bus, these pins input and output data (D0–D7) and output 8 low-order address bits (A0–A7) separated in time by multiplexing.
A0, A1/D0 to A7/D6		Output Input/output	If the external bus is set as a 16-bit wide multiplexed bus, these pins input and output data (D0–D6) and output address (A1–A7) separated in time by multiplexing. They also output address (A0).
P30 to P37	I/O port P3	Input/output	This is an 8-bit I/O port equivalent to P0.
A8 to A15		Output	These pins output 8 middle-order address bits (A8–A15).
A8/D7, A9 to A15		Input/output Output	If the external bus is set as a 16-bit wide multiplexed bus, these pins input and output data (D7) and output address (A8) separated in time by multiplexing. They also output address (A9–A15).
P40 to P47	I/O port P4	Input/output	This is an 8-bit I/O port equivalent to P0.
CS0 to CS3, A16 to A19		Output Output	These pins output CS0–CS3 signals and A16–A19. CS0–CS3 are chip select signals used to specify an access space. A16–A19 are 4 high-order address bits.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

**Table 1.5.3 Pin Description**

Pin name	Signal name	I/O type	Function
P50 to P57	I/O port P5	Input/output	This is an 8-bit I/O port equivalent to P0. In single-chip mode, P57 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.
$\overline{\text{WRL}}$ / $\overline{\text{WR}}$ , $\overline{\text{WRH}}$ / $\overline{\text{BHE}}$ , $\overline{\text{RD}}$ , $\overline{\text{BCLK}}$ , $\overline{\text{HLDA}}$ , $\overline{\text{HOLD}}$ , $\overline{\text{ALE}}$ , $\overline{\text{RDY}}$		Output Output Output Output Input Output Input	<p>Output <math>\overline{\text{WRL}}</math>, <math>\overline{\text{WRH}}</math> (<math>\overline{\text{WR}}</math> and <math>\overline{\text{BHE}}</math>), <math>\overline{\text{RD}}</math>, <math>\overline{\text{BCLK}}</math>, <math>\overline{\text{HLDA}}</math>, and <math>\overline{\text{ALE}}</math> signals. <math>\overline{\text{WRL}}</math> and <math>\overline{\text{WRH}}</math>, and <math>\overline{\text{BHE}}</math> and <math>\overline{\text{WR}}</math> can be switched using software control.</p> <p>■ <math>\overline{\text{WRL}}</math>, <math>\overline{\text{WRH}}</math>, and <math>\overline{\text{RD}}</math> selected  With a 16-bit external data bus, data is written to even addresses when the <math>\overline{\text{WRL}}</math> signal is "L" and to the odd addresses when the <math>\overline{\text{WRH}}</math> signal is "L". Data is read when <math>\overline{\text{RD}}</math> is "L".</p> <p>■ <math>\overline{\text{WR}}</math>, <math>\overline{\text{BHE}}</math>, and <math>\overline{\text{RD}}</math> selected  Data is written when <math>\overline{\text{WR}}</math> is "L". Data is read when <math>\overline{\text{RD}}</math> is "L". Odd addresses are accessed when <math>\overline{\text{BHE}}</math> is "L". Use this mode when using an 8-bit external data bus.</p> <p>While the input level at the <math>\overline{\text{HOLD}}</math> pin is "L", the microcomputer is placed in the hold state. While in the hold state, <math>\overline{\text{HLDA}}</math> outputs a "L" level. <math>\overline{\text{ALE}}</math> is used to latch the address. While the input level of the <math>\overline{\text{RDY}}</math> pin is "L", the microcomputer is in the ready state.</p>
P60 to P67	I/O port P6	Input/output	This is an 8-bit I/O port equivalent to P0. When used input in singlechip and memory expansion 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.
P70 to P77	I/O port P7	Input/output	This is an 8-bit I/O port equivalent to P6 (P70 and P71 are N channel open-drain output). Pins in this port also function as timer A0–A3, timer B5 or UART2 I/O pins as selected by software.
P80 to P84, P86, P87, P85	I/O port P8  I/O port P85	Input/output Input/output Input/output Input	P80 to P84, P86, and P87 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. P86 and P87 can be set using software to function as the I/O pins for a sub clock generation circuit. In this case, connect a quartz oscillator between P86 ( $\overline{\text{XCOUT}}$ pin) and P87 ( $\overline{\text{XCIN}}$ pin). P85 is an input-only port that also functions for NMI. The NMI interrupt is generated when the input at this pin changes from "H" to "L". The NMI function cannot be cancelled using software. The pull-up cannot be set for this pin.
P90 to P97	I/O port P9	Input/output	This is an 8-bit I/O port equivalent to P6. Pins in this port also function as SI/O3, 4 I/O pins, Timer B0–B4 input pins, D-A converter output pins, A-D converter extended input pins, or A-D trigger input pins as selected by software.
P100 to P107	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, P104–P107 also function as input pins for the key input interrupt function.
P11	Output port P11	Output	This is a 1-bit output-only port. Pins in this port also function as SLICEON output pins as selected by software.



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

**Table 1.5.4 Pin Description**

Pin name	Signal name	I/O type	Function
VDD1, Vss1	Power supply input		Digital power supply pin. Supply 4.75 to 5.25 V to the VDD1 pin. Supply 0 V to the Vss1 pin.
VDD2, Vss2	Power supply input		Analog power supply pin. Supply 4.75 to 5.25 V to the VDD2 pin. Supply 0 V to the Vss2 pin.
VDD3, Vss3	Power supply input		Analog power supply pin. Supply 4.75 to 5.25 V to the VDD3 pin. Supply 0 V to the Vss3 pin.
SVREF	Synchronous slice level input	Input	When slice the vertical synchronous signal, input slice power.
CVIN1	Composite video signal input 1	Input	This pin inputs the external composite video signal. Data slices this signal internally by setting.
SYNCIN	Composite video signal input 2	Input	This pin inputs the external composite video signal. Synchronous divides this signal internally.
M1	Chip mode setting input	Input	Usually, supply 0 V to the pin.
LP2	Filter output 1	Output	This is a filter output pin 1 (for fsc).
LP3	Filter output 2	Output	This is a filter output pin 2 (for VPS).
LP4	Filter output 3	Output	This is a filter output pin 3 (for PDC).
FSCIN	fsc input pin for synchronous signal generation	Input	Sub-carrier (fsc) input pin for synchronous signal generation.
M2	Power supply input for flash rewriting	Input	It is power supply input pin for rewriting of built-in flash memory. Usually supply 0V to M2 pin, and supply 4.75-5.25V at the time of rewriting of built-in flash memory.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## 2. OPERATION OF FUNCTIONAL BLOKS

The M306H2MC-XXXFP accommodates certain units in a single chip. These units include RAM to store instructions and data and the central processing unit (CPU) to execute arithmetic/logic operations. Also included are peripheral units such as timers, serial I/O, D-A converter, DMAC, CRC calculation circuit, A-D converter, Data slicer circuit and I/O ports.

The following explains each unit.

### 2.1 Memory

Figure 2.1.1 is a memory map of the M306H2MC-XXXFP. The address space extends the 1M bytes from address 00000<sub>16</sub> to FFFFF<sub>16</sub>. From address FFFFF<sub>16</sub> down is ROM. In the M306H2MC-XXXFP, can use from address from E0000<sub>16</sub> to FFFFF<sub>16</sub> as 128K bytes internal ROM area. The vector table for fixed interrupts such as the reset and  $\overline{\text{NMI}}$  are mapped to from address FFFDC<sub>16</sub> to FFFFF<sub>16</sub>. The starting address of the interrupt routine is stored here. The address of the vector table for timer interrupts, etc., can be set as desired using the internal register (INTB). See the section on interrupts for details.

5K bytes of internal RAM is mapped to from address 00400<sub>16</sub> to 017FF<sub>16</sub>. 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 from address 00000<sub>16</sub> to 003FF<sub>16</sub>. This area accommodates the control registers for peripheral devices such as I/O ports, A-D converter, serial I/O, and timers, etc. Figures 2.1.2 to 2.1.4 are location 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 from address FFE00<sub>16</sub> to FFFDB<sub>16</sub>. If the starting addresses of subroutines or the destination addresses of jumps are stored here, subroutine call instructions and jump instructions can be used as 2-byte instructions, reducing the number of program steps.

In memory expansion mode, a part of the spaces are reserved and cannot be used. The following spaces cannot be used.

- The space between 01800<sub>16</sub> and 03FFF<sub>16</sub> (memory expansion mode)
- The space between D0000<sub>16</sub> and DFFFF<sub>16</sub> (memory expansion mode)

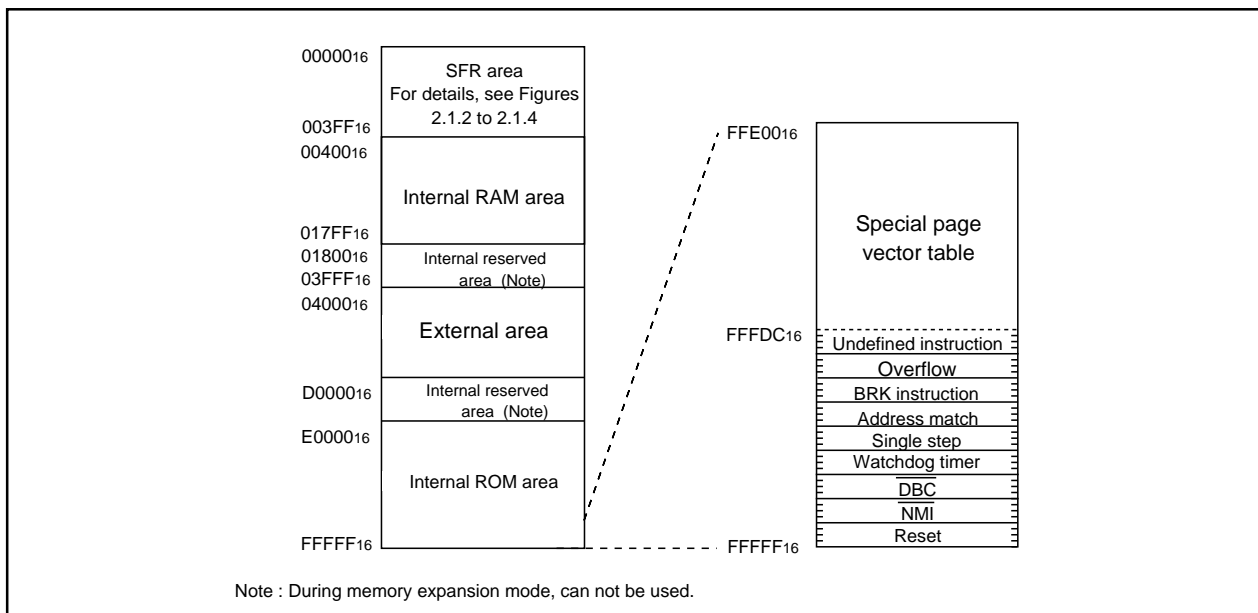


Figure 2.1.1 Memory map

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

0000 <sub>16</sub>		0044 <sub>16</sub>	INT3 interrupt control register(INT3IC)
0001 <sub>16</sub>		0045 <sub>16</sub>	Timer B5 interrupt control register (TB5IC)
0002 <sub>16</sub>		0046 <sub>16</sub>	Timer B4 interrupt control register (TB4IC)
0003 <sub>16</sub>		0047 <sub>16</sub>	Timer B3 interrupt control register (TB3IC)
0004 <sub>16</sub>	Processor mode register 0 (PM0)	0048 <sub>16</sub>	SI/O4 interrupt control register (S4IC)
0005 <sub>16</sub>	Processor mode register 1(PM1)		INT5 interrupt control register(INT5IC)
0006 <sub>16</sub>	System clock control register 0 (CM0)	0049 <sub>16</sub>	SI/O3 interrupt control register (S3IC)
0007 <sub>16</sub>	System clock control register 1 (CM1)		INT4 interrupt control register(INT4IC)
0008 <sub>16</sub>	Chip select control register (CSR)	004A <sub>16</sub>	Bus collision detection interrupt control register (BCNIC)
0009 <sub>16</sub>	Address match interrupt enable register (AIER)	004B <sub>16</sub>	DMA0 interrupt control register (DM0IC)
000A <sub>16</sub>	Protect register (PRCR)	004C <sub>16</sub>	DMA1 interrupt control register (DM1IC)
000B <sub>16</sub>		004D <sub>16</sub>	Key input interrupt control register (KUPIC)
000C <sub>16</sub>		004E <sub>16</sub>	A-D conversion interrupt control register (ADIC)
000D <sub>16</sub>		004F <sub>16</sub>	UART2 transmit interrupt control register (S2TIC)
000E <sub>16</sub>	Watchdog timer start register (WDTS)	0050 <sub>16</sub>	UART2 receive interrupt control register (S2RIC)
000F <sub>16</sub>	Watchdog timer control register (WDC)	0051 <sub>16</sub>	UART0 transmit interrupt control register (S0TIC)
0010 <sub>16</sub>		0052 <sub>16</sub>	UART0 receive interrupt control register (S0RIC)
0011 <sub>16</sub>	Address match interrupt register 0 (RMAD0)	0053 <sub>16</sub>	UART1 transmit interrupt control register (S1TIC)
0012 <sub>16</sub>		0054 <sub>16</sub>	UART1 receive interrupt control register (S1RIC)
0013 <sub>16</sub>		0055 <sub>16</sub>	Timer A0 interrupt control register (TA0IC)
0014 <sub>16</sub>		0056 <sub>16</sub>	Timer A1 interrupt control register (TA1IC)
0015 <sub>16</sub>	Address match interrupt register 1 (RMAD1)	0057 <sub>16</sub>	Timer A2 interrupt control register (TA2IC)
0016 <sub>16</sub>		0058 <sub>16</sub>	Timer A3 interrupt control register (TA3IC)
0017 <sub>16</sub>		0059 <sub>16</sub>	Timer A4 interrupt control register (TA4IC)
0018 <sub>16</sub>		005A <sub>16</sub>	Timer B0 interrupt control register (TB0IC)
0019 <sub>16</sub>		005B <sub>16</sub>	Timer B1 interrupt control register (TB1IC)
001A <sub>16</sub>		005C <sub>16</sub>	Timer B2 interrupt control register (TB2IC)
001B <sub>16</sub>		005D <sub>16</sub>	INT0 interrupt control register (INT0IC)
001C <sub>16</sub>		005E <sub>16</sub>	INT1 interrupt control register (INT1IC)
001D <sub>16</sub>		005F <sub>16</sub>	INT2 interrupt control register (INT2IC)
001E <sub>16</sub>		0060 <sub>16</sub>	
001F <sub>16</sub>			
0020 <sub>16</sub>	DMA0 source pointer (SAR0)	0200 <sub>16</sub>	
0021 <sub>16</sub>		0201 <sub>16</sub>	
0022 <sub>16</sub>		0202 <sub>16</sub>	
0023 <sub>16</sub>		0203 <sub>16</sub>	
0024 <sub>16</sub>		0204 <sub>16</sub>	
0025 <sub>16</sub>	DMA0 destination pointer (DAR0)	0205 <sub>16</sub>	
0026 <sub>16</sub>		0206 <sub>16</sub>	
0027 <sub>16</sub>		0207 <sub>16</sub>	
0028 <sub>16</sub>	DMA0 transfer counter (TCR0)	0208 <sub>16</sub>	
0029 <sub>16</sub>		0209 <sub>16</sub>	
002A <sub>16</sub>		020A <sub>16</sub>	
002B <sub>16</sub>		020B <sub>16</sub>	
002C <sub>16</sub>	DMA0 control register (DM0CON)	020C <sub>16</sub>	
002D <sub>16</sub>		020D <sub>16</sub>	
002E <sub>16</sub>		020E <sub>16</sub>	Slice RAM address control register
002F <sub>16</sub>		020F <sub>16</sub>	
0030 <sub>16</sub>		0210 <sub>16</sub>	Slice RAM data control register
0031 <sub>16</sub>	DMA1 source pointer (SAR1)	0211 <sub>16</sub>	
0032 <sub>16</sub>		0212 <sub>16</sub>	
0033 <sub>16</sub>		0213 <sub>16</sub>	
0034 <sub>16</sub>		0214 <sub>16</sub>	
0035 <sub>16</sub>	DMA1 destination pointer (DAR1)	0215 <sub>16</sub>	
0036 <sub>16</sub>		0216 <sub>16</sub>	Address control register for expansion register
0037 <sub>16</sub>		0217 <sub>16</sub>	
0038 <sub>16</sub>		0218 <sub>16</sub>	Data control register for expansion register
0039 <sub>16</sub>	DMA1 transfer counter (TCR1)	0219 <sub>16</sub>	
003A <sub>16</sub>		021A <sub>16</sub>	Humming 8/4 register
003B <sub>16</sub>		021B <sub>16</sub>	
003C <sub>16</sub>	DMA1 control register (DM1CON)	021C <sub>16</sub>	Humming 24/18 register 0
003D <sub>16</sub>		021D <sub>16</sub>	
003E <sub>16</sub>		021E <sub>16</sub>	Humming 24/18 register 1
003F <sub>16</sub>		021F <sub>16</sub>	
0040 <sub>16</sub>		0220 <sub>16</sub>	
0041 <sub>16</sub>			
0042 <sub>16</sub>			
0043 <sub>16</sub>			
		033F <sub>16</sub>	

Note: Location in the SFR area where nothing is allocated are reserved. Do not access these areas for read or write.

Figure 2.1.2 Location of peripheral unit control registers (1)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

0340 <sub>16</sub>	Timer B3, 4, 5 count start flag (TBSR)	0380 <sub>16</sub>	Count start flag (TABSR)
0341 <sub>16</sub>		0381 <sub>16</sub>	Clock prescaler reset flag (CPSRF)
0342 <sub>16</sub>		0382 <sub>16</sub>	One-shot start flag (ONSF)
0343 <sub>16</sub>		0383 <sub>16</sub>	Trigger select register (TRGSR)
0344 <sub>16</sub>		0384 <sub>16</sub>	Up-down flag (UDF)
0345 <sub>16</sub>		0385 <sub>16</sub>	
0346 <sub>16</sub>		0386 <sub>16</sub>	Timer A0 (TA0)
0347 <sub>16</sub>		0387 <sub>16</sub>	
0348 <sub>16</sub>		0388 <sub>16</sub>	Timer A1 (TA1)
0349 <sub>16</sub>		0389 <sub>16</sub>	
034A <sub>16</sub>		038A <sub>16</sub>	Timer A2 (TA2)
034B <sub>16</sub>		038B <sub>16</sub>	
034C <sub>16</sub>		038C <sub>16</sub>	Timer A3 (TA3)
034D <sub>16</sub>		038D <sub>16</sub>	
034E <sub>16</sub>		038E <sub>16</sub>	Timer A4 (TA4)
034F <sub>16</sub>		038F <sub>16</sub>	
0350 <sub>16</sub>	Timer B3 register (TB3)	0390 <sub>16</sub>	Timer B0 (TB0)
0351 <sub>16</sub>		0391 <sub>16</sub>	
0352 <sub>16</sub>	Timer B4 register (TB4)	0392 <sub>16</sub>	Timer B1 (TB1)
0353 <sub>16</sub>		0393 <sub>16</sub>	
0354 <sub>16</sub>	Timer B5 register (TB5)	0394 <sub>16</sub>	Timer B2 (TB2)
0355 <sub>16</sub>		0395 <sub>16</sub>	
0356 <sub>16</sub>		0396 <sub>16</sub>	Timer A0 mode register (TA0MR)
0357 <sub>16</sub>		0397 <sub>16</sub>	Timer A1 mode register (TA1MR)
0358 <sub>16</sub>		0398 <sub>16</sub>	Timer A2 mode register (TA2MR)
0359 <sub>16</sub>		0399 <sub>16</sub>	Timer A3 mode register (TA3MR)
035A <sub>16</sub>		039A <sub>16</sub>	Timer A4 mode register (TA4MR)
035B <sub>16</sub>	Timer B3 mode register (TB3MR)	039B <sub>16</sub>	Timer B0 mode register (TB0MR)
035C <sub>16</sub>	Timer B4 mode register (TB4MR)	039C <sub>16</sub>	Timer B1 mode register (TB1MR)
035D <sub>16</sub>	Timer B5 mode register (TB5MR)	039D <sub>16</sub>	Timer B2 mode register (TB2MR)
035E <sub>16</sub>		039E <sub>16</sub>	
035F <sub>16</sub>	Interrupt cause select register (IFSR)	039F <sub>16</sub>	
0360 <sub>16</sub>	SI/O3 transmit/receive register (S3TRR)	03A0 <sub>16</sub>	UART0 transmit/receive mode register (U0MR)
0361 <sub>16</sub>		03A1 <sub>16</sub>	UART0 bit rate generator (U0BRG)
0362 <sub>16</sub>	SI/O3 control register (S3C)	03A2 <sub>16</sub>	
0363 <sub>16</sub>	SI/O3 bit rate generator (S3BRG)	03A3 <sub>16</sub>	UART0 transmit buffer register (U0TB)
0364 <sub>16</sub>	SI/O4 transmit/receive register (S4TRR)	03A4 <sub>16</sub>	UART0 transmit/receive control register 0 (U0C0)
0365 <sub>16</sub>		03A5 <sub>16</sub>	UART0 transmit/receive control register 1 (U0C1)
0366 <sub>16</sub>	SI/O4 control register (S4C)	03A6 <sub>16</sub>	
0367 <sub>16</sub>	SI/O4 bit rate generator (S4BRG)	03A7 <sub>16</sub>	UART0 receive buffer register (U0RB)
0368 <sub>16</sub>		03A8 <sub>16</sub>	UART1 transmit/receive mode register (U1MR)
0369 <sub>16</sub>		03A9 <sub>16</sub>	UART1 bit rate generator (U1BRG)
036A <sub>16</sub>		03AA <sub>16</sub>	
036B <sub>16</sub>		03AB <sub>16</sub>	UART1 transmit buffer register (U1TB)
036C <sub>16</sub>		03AC <sub>16</sub>	UART1 transmit/receive control register 0 (U1C0)
036D <sub>16</sub>		03AD <sub>16</sub>	UART1 transmit/receive control register 1 (U1C1)
036E <sub>16</sub>		03AE <sub>16</sub>	
036F <sub>16</sub>		03AF <sub>16</sub>	UART1 receive buffer register (U1RB)
0370 <sub>16</sub>		03B0 <sub>16</sub>	UART transmit/receive control register 2 (UCON)
0371 <sub>16</sub>		03B1 <sub>16</sub>	
0372 <sub>16</sub>		03B2 <sub>16</sub>	
0373 <sub>16</sub>		03B3 <sub>16</sub>	
0374 <sub>16</sub>		03B4 <sub>16</sub>	Flash memory control register (FER)
0375 <sub>16</sub>	UART2 special mode register 3(U2SMR3)	03B5 <sub>16</sub>	
0376 <sub>16</sub>	UART2 special mode register 2(U2SMR2)	03B6 <sub>16</sub>	
0377 <sub>16</sub>	UART2 special mode register (U2SMR)	03B7 <sub>16</sub>	
0378 <sub>16</sub>	UART2 transmit/receive mode register (U2MR)	03B8 <sub>16</sub>	DMA0 request cause select register (DM0SL)
0379 <sub>16</sub>	UART2 bit rate generator (U2BRG)	03B9 <sub>16</sub>	
037A <sub>16</sub>		03BA <sub>16</sub>	DMA1 request cause select register (DM1SL)
037B <sub>16</sub>	UART2 transmit buffer register (U2TB)	03BB <sub>16</sub>	
037C <sub>16</sub>		03BC <sub>16</sub>	
037D <sub>16</sub>	UART2 transmit/receive control register 0 (U2C0)	03BD <sub>16</sub>	CRC data register (CRCD)
037E <sub>16</sub>	UART2 transmit/receive control register 1 (U2C1)	03BE <sub>16</sub>	CRC input register (CRCIN)
037F <sub>16</sub>	UART2 receive buffer register (U2RB)	03BF <sub>16</sub>	

Note : Locations in the SFR area where nothing is allocated are reserved areas. Do not access these areas for read or write.

Figure 2.1.3 Location of peripheral unit control registers (2)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

03C0 <sub>16</sub>	A-D register 0 (AD0)
03C1 <sub>16</sub>	
03C2 <sub>16</sub>	A-D register 1 (AD1)
03C3 <sub>16</sub>	
03C4 <sub>16</sub>	A-D register 2 (AD2)
03C5 <sub>16</sub>	
03C6 <sub>16</sub>	A-D register 3 (AD3)
03C7 <sub>16</sub>	
03C8 <sub>16</sub>	A-D register 4 (AD4)
03C9 <sub>16</sub>	
03CA <sub>16</sub>	A-D register 5 (AD5)
03CB <sub>16</sub>	
03CC <sub>16</sub>	A-D register 6 (AD6)
03CD <sub>16</sub>	
03CE <sub>16</sub>	A-D register 7 (AD7)
03CF <sub>16</sub>	
03D0 <sub>16</sub>	
03D1 <sub>16</sub>	
03D2 <sub>16</sub>	
03D3 <sub>16</sub>	
03D4 <sub>16</sub>	A-D control register 2 (ADCON2)
03D5 <sub>16</sub>	
03D6 <sub>16</sub>	A-D control register 0 (ADCON0)
03D7 <sub>16</sub>	A-D control register 1 (ADCON1)
03D8 <sub>16</sub>	D-A register 0 (DA0)
03D9 <sub>16</sub>	
03DA <sub>16</sub>	D-A register 1 (DA1)
03DB <sub>16</sub>	
03DC <sub>16</sub>	D-A control register (DACON)
03DD <sub>16</sub>	
03DE <sub>16</sub>	
03DF <sub>16</sub>	
03E0 <sub>16</sub>	Port P0 (P0)
03E1 <sub>16</sub>	Port P1 (P1)
03E2 <sub>16</sub>	Port P0 direction register (PD0)
03E3 <sub>16</sub>	Port P1 direction register (PD1)
03E4 <sub>16</sub>	Port P2 (P2)
03E5 <sub>16</sub>	Port P3 (P3)
03E6 <sub>16</sub>	Port P2 direction register (PD2)
03E7 <sub>16</sub>	Port P3 direction register (PD3)
03E8 <sub>16</sub>	Port P4 (P4)
03E9 <sub>16</sub>	Port P5 (P5)
03EA <sub>16</sub>	Port P4 direction register (PD4)
03EB <sub>16</sub>	Port P5 direction register (PD5)
03EC <sub>16</sub>	Port P6 (P6)
03ED <sub>16</sub>	Port P7 (P7)
03EE <sub>16</sub>	Port P6 direction register (PD6)
03EF <sub>16</sub>	Port P7 direction register (PD7)
03F0 <sub>16</sub>	Port P8 (P8)
03F1 <sub>16</sub>	Port P9 (P9)
03F2 <sub>16</sub>	Port P8 direction register (PD8)
03F3 <sub>16</sub>	Port P9 direction register (PD9)
03F4 <sub>16</sub>	Port P10 (P10)
03F5 <sub>16</sub>	
03F6 <sub>16</sub>	Port P10 direction register (PD10)
03F7 <sub>16</sub>	
03F8 <sub>16</sub>	
03F9 <sub>16</sub>	
03FA <sub>16</sub>	
03FB <sub>16</sub>	
03FC <sub>16</sub>	Pull-up control register 0 (PUR0)
03FD <sub>16</sub>	Pull-up control register 1 (PUR1)
03FE <sub>16</sub>	Pull-up control register 2 (PUR2)
03FF <sub>16</sub>	Port control register (PCR)

Note: Locations in the SFR area where nothing is allocated are reserved areas. Do not access these areas for read or write.

Figure 2.1.4 Location of peripheral unit control registers (3)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.2 Central Processing Unit (CPU)

The CPU has 13 registers shown in Figure 2.2.1. Seven of these registers (R0, R1, R2, R3, A0, A1, and FB) come in two sets; therefore, these have two register banks.

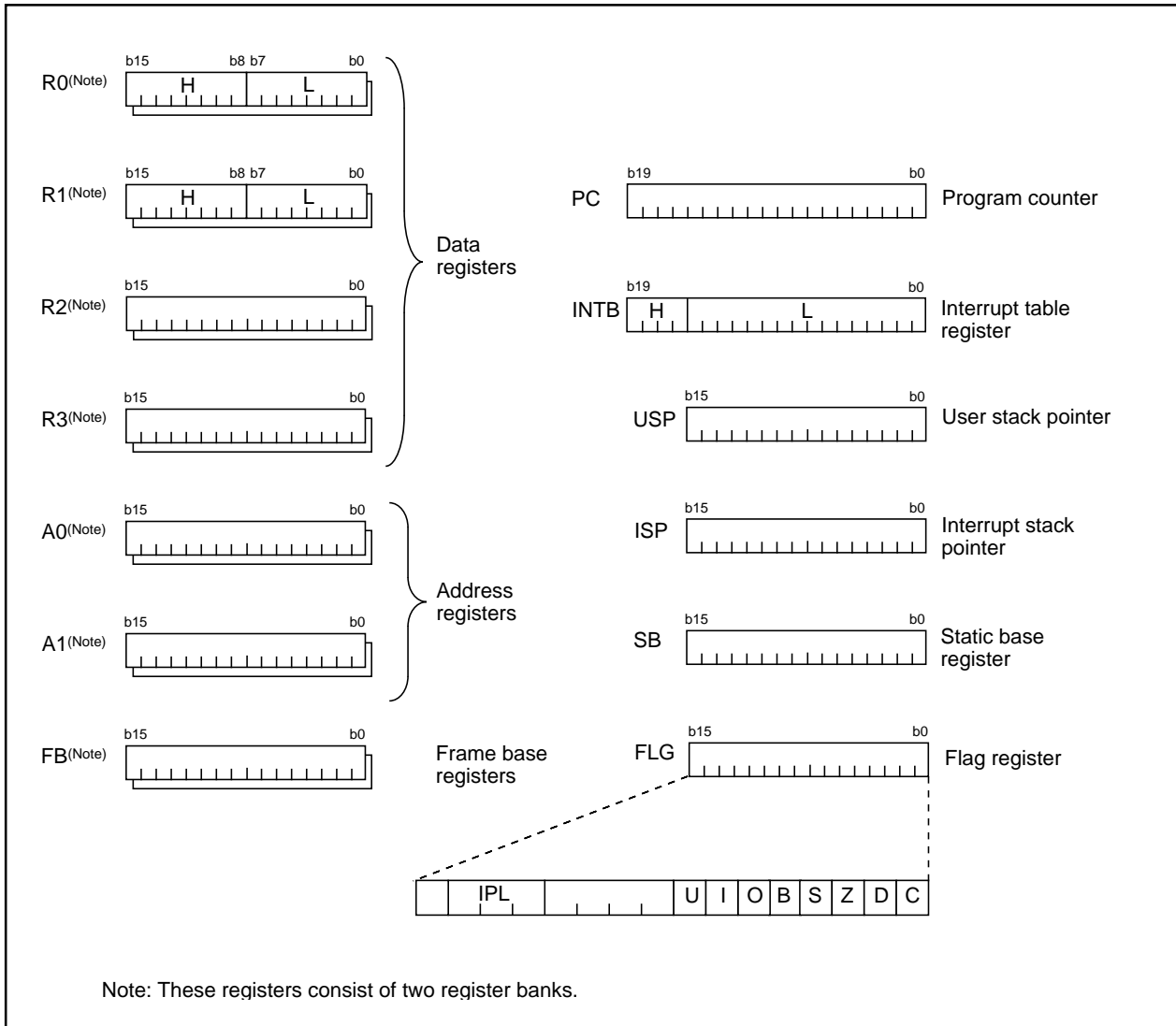


Figure 2.2.1 Central processing unit register

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

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

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

### (2) Address registers (A0 and A1)

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

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

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (3) Frame base register (FB)

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

## (4) Program counter (PC)

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

## (5) Interrupt table register (INTB)

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

## (6) Stack pointer (USP/ISP)

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

Your desired type of stack pointer (USP or ISP) can be selected by a stack pointer select flag (U flag).

This flag is located at the position of bit 7 in the flag register (FLG).

## (7) Static base register (SB)

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

## (8) Flag register (FLG)

Flag register (FLG) is configured with 11 bits, each bit is used as a flag. Figure 2.2.2 shows the flag register (FLG). The following explains the function of each flag:

- **Bit 0: Carry flag (C flag)**

This flag retains a carry, borrow, or shift-out bit that has occurred in the arithmetic/logic unit.

- **Bit 1: Debug flag (D flag)**

This flag enables a single-step interrupt.

When this flag is "1", a single-step interrupt is generated after instruction execution. This flag is cleared to "0" when the interrupt is acknowledged.

- **Bit 2: Zero flag (Z flag)**

This flag is set to "1" when an arithmetic operation resulted in 0; otherwise, cleared to "0".

- **Bit 3: Sign flag (S flag)**

This flag is set to "1" when an arithmetic operation resulted in a negative value; otherwise, cleared to "0".

- **Bit 4: Register bank select flag (B flag)**

This flag chooses a register bank. Register bank 0 is selected when this flag is "0"; register bank 1 is selected when this flag is "1".

- **Bit 5: Overflow flag (O flag)**

This flag is set to "1" when an arithmetic operation resulted in overflow; otherwise, cleared to "0".

- **Bit 6: Interrupt enable flag (I flag)**

This flag enables a maskable interrupt.

An interrupt is disabled when this flag is "0", and is enabled when this flag is "1". This flag is cleared to "0" when the interrupt is acknowledged.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

- Bit 7: Stack pointer select flag (U flag)**

Interrupt stack pointer (ISP) is selected when this flag is "0" ; user stack pointer (USP) is selected when this flag is "1".

This flag is cleared to "0" when a hardware interrupt is acknowledged or an INT instruction of software interrupt Nos. 0 to 31 is executed.

- Bits 8 to 11: Reserved area**

- Bits 12 to 14: Processor interrupt priority level (IPL)**

Processor interrupt priority level (IPL) is configured with three bits, for specification of up to eight processor interrupt priority levels from level 0 to level 7.

If a requested interrupt has priority greater than the processor interrupt priority level (IPL), the interrupt is enabled.

- Bit 15: Reserved area**

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

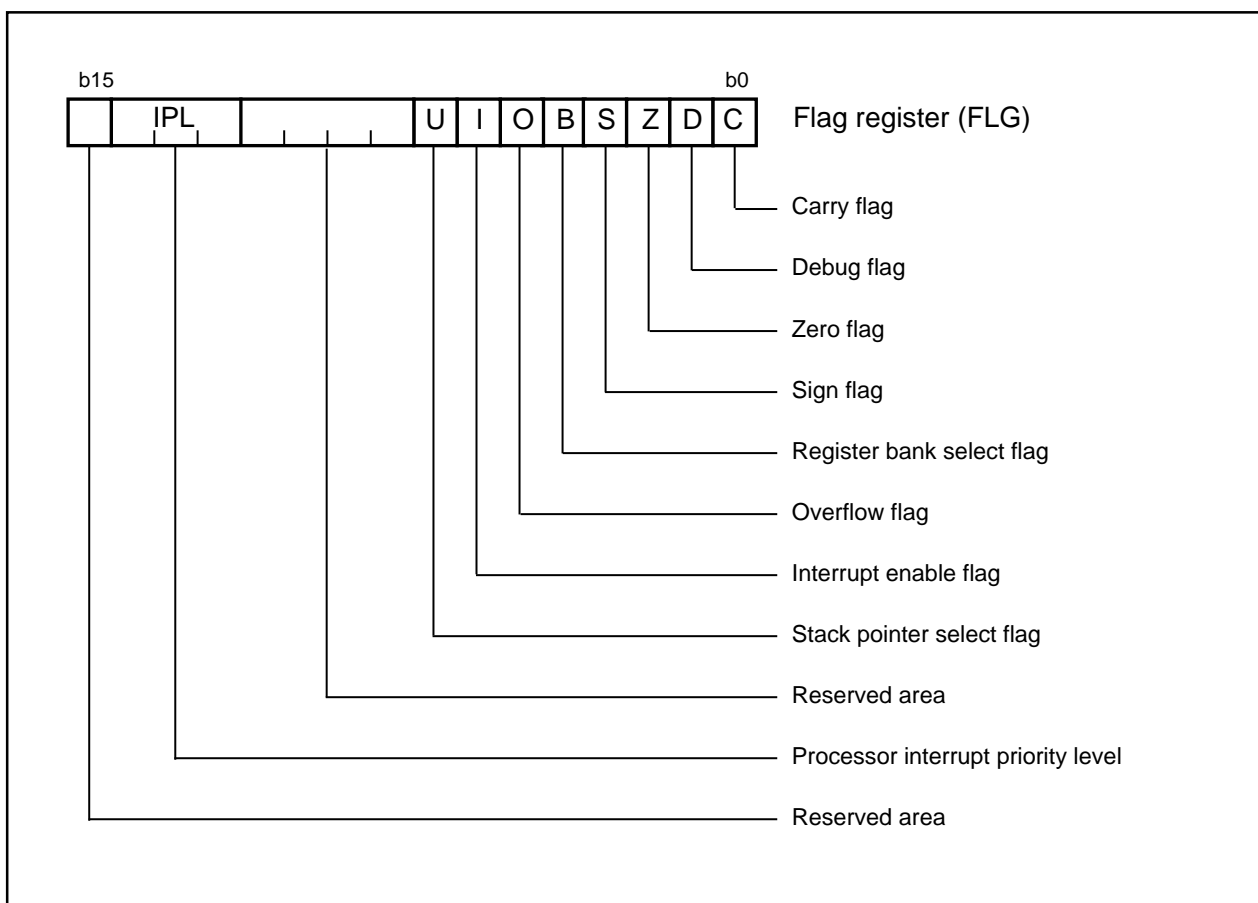


Figure 2.2.2 Flag register (FLG)



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## 2.3 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" (0.2V<sub>CC</sub> max.) for at least 20 cycles. When the reset pin level is then returned to the "H" level while main clock is stable, the reset status is cancelled and program execution resumes from the address in the reset vector table.

Figure 2.3.1 shows the example reset circuit. Figure 2.3.2 shows the reset sequence.

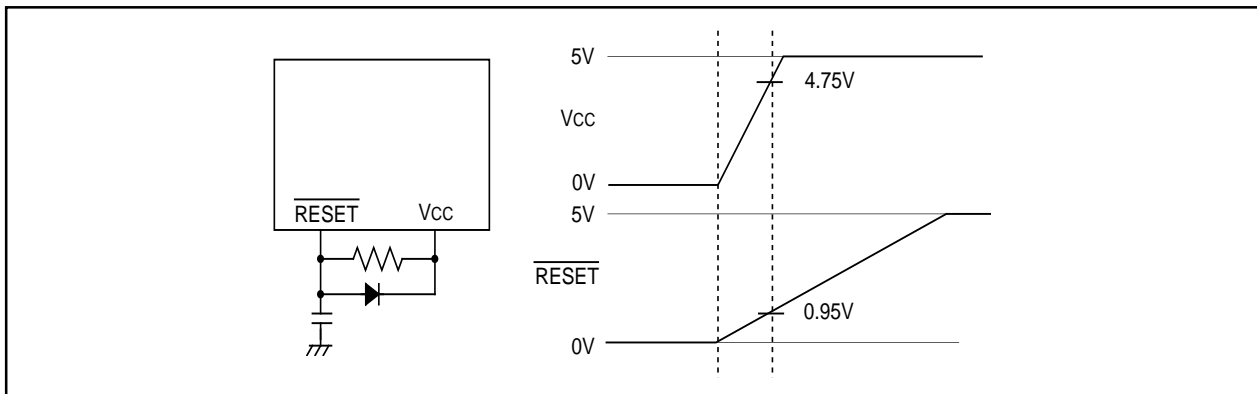


Figure 2.3.1 Example reset circuit

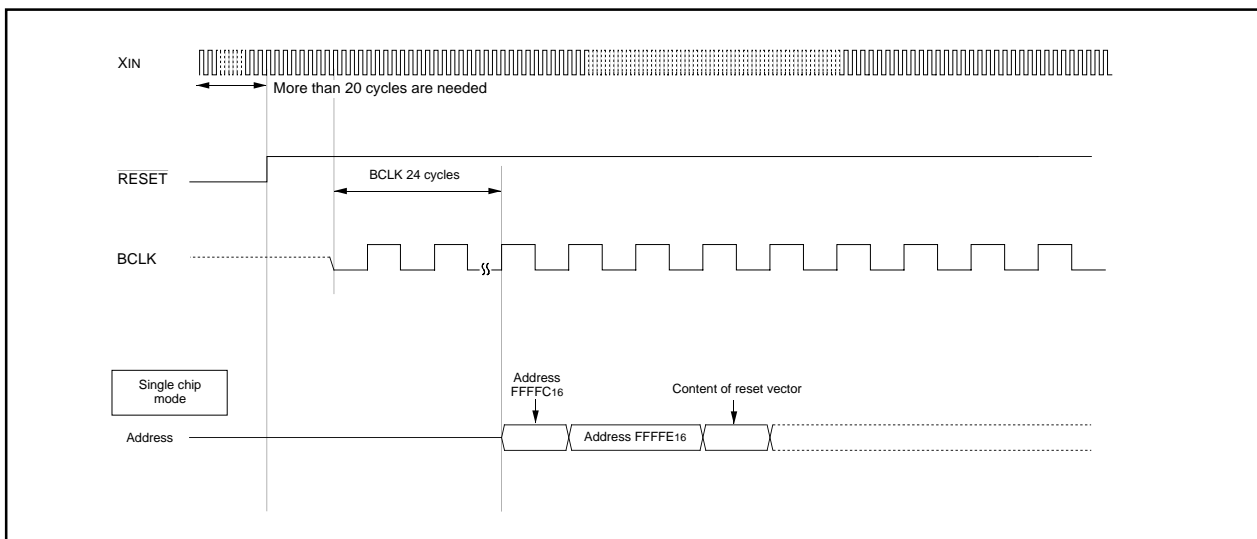


Figure 2.3.2 Reset sequence

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

Table 2.3.1 shows the statuses of the other pins while the  $\overline{\text{RESET}}$  pin level is "L". Figures 2.3.3 and 2.3.4 show the internal status of the microcomputer immediately after the reset is cancelled.

**Table 2.3.1 Pin status when  $\overline{\text{RESET}}$  pin level is "L"**

Pin name	Status
P0 to P10	Input port (floating)
P11	Output port
CVIN1,SVREF,SYNCIN,M1,FSCIN	Input port
LP2,LP3,LP4	Output port

## 2.3.1 Software Reset

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

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

Processor mode register 0 (Note)	(0004 <sub>16</sub> )...	00 <sub>16</sub>	Timer B0 interrupt control register	(005A <sub>16</sub> )...	XXXX?000
Processor mode register 1	(0005 <sub>16</sub> )...	00000XX0	Timer B1 interrupt control register	(005B <sub>16</sub> )...	XXXX?000
System clock control register 0	(0006 <sub>16</sub> )...	010001000	Timer B2 interrupt control register	(005C <sub>16</sub> )...	XXXX?000
System clock control register 1	(0007 <sub>16</sub> )...	001000000	INT0 interrupt control register	(005D <sub>16</sub> )...	XX00?000
Chip select control register	(0008 <sub>16</sub> )...	000000001	INT1 interrupt control register	(005E <sub>16</sub> )...	XX00?000
Address match interrupt enable register	(0009 <sub>16</sub> )...	XXXXXX00	INT2 interrupt control register	(005F <sub>16</sub> )...	XX00?000
Protect register	(000A <sub>16</sub> )...	XXXXXX00	Slice RAM address control register	(020E <sub>16</sub> )...	00 <sub>16</sub>
Watchdog timer control register	(000F <sub>16</sub> )...	000???		(020F <sub>16</sub> )...	00 <sub>16</sub>
Address match interrupt register 0	(0010 <sub>16</sub> )...	00 <sub>16</sub>	Slice RAM data control register	(0210 <sub>16</sub> )...	00 <sub>16</sub>
	(0011 <sub>16</sub> )...	00 <sub>16</sub>		(0211 <sub>16</sub> )...	00 <sub>16</sub>
	(0012 <sub>16</sub> )...	XXXX0000	Address control register for expansion register	(0216 <sub>16</sub> )...	00 <sub>16</sub>
Address match interrupt register 1	(0014 <sub>16</sub> )...	00 <sub>16</sub>		(0217 <sub>16</sub> )...	00 <sub>16</sub>
	(0015 <sub>16</sub> )...	00 <sub>16</sub>	Data control register for expansion register	(0218 <sub>16</sub> )...	00 <sub>16</sub>
	(0016 <sub>16</sub> )...	XXXX0000		(0219 <sub>16</sub> )...	00 <sub>16</sub>
DMA0 control register	(002C <sub>16</sub> )...	00000?00	Humming 8/4 register	(021A <sub>16</sub> )...	00 <sub>16</sub>
DMA1 control register	(003C <sub>16</sub> )...	00000?00		(021B <sub>16</sub> )...	00 <sub>16</sub>
INT3 interrupt control register	(0044 <sub>16</sub> )...	XX00?000	Humming 24/18 register0	(021C <sub>16</sub> )...	00 <sub>16</sub>
Timer B5 interrupt control register	(0045 <sub>16</sub> )...	XXXX?000		(021D <sub>16</sub> )...	00 <sub>16</sub>
Timer B4 interrupt control register	(0046 <sub>16</sub> )...	XXXX?000	Humming 24/18 register1	(021E <sub>16</sub> )...	00 <sub>16</sub>
Timer B3 interrupt control register	(0047 <sub>16</sub> )...	XXXX?000		(021F <sub>16</sub> )...	00 <sub>16</sub>
SI/O4 interrupt control register	(0048 <sub>16</sub> )...	XX00?000	Timer B3,4,5 count start flag	(0340 <sub>16</sub> )...	000XXXXX
SI/O3 interrupt control register	(0049 <sub>16</sub> )...	XX00?000	Timer B3 mode register	(035B <sub>16</sub> )...	00?XX000
Bus collision detection interrupt control register	(004A <sub>16</sub> )...	XXXX?000	Timer B4 mode register	(035C <sub>16</sub> )...	00?XX000
DMA0 interrupt control register	(004B <sub>16</sub> )...	XXXX?000	Timer B5 mode register	(035D <sub>16</sub> )...	00?XX000
DMA1 interrupt control register	(004C <sub>16</sub> )...	XXXX?000	Interrupt cause select register	(035F <sub>16</sub> )...	00 <sub>16</sub>
Key input interrupt control register	(004D <sub>16</sub> )...	XXXX?000	SI/O3 control register	(0362 <sub>16</sub> )...	40 <sub>16</sub>
A-D conversion interrupt control register	(004E <sub>16</sub> )...	XXXX?000	SI/O4 control register	(0366 <sub>16</sub> )...	40 <sub>16</sub>
UART2 transmit interrupt control register	(004F <sub>16</sub> )...	XXXX?000	UART2 special mode register 2	(0376 <sub>16</sub> )...	00 <sub>16</sub>
UART2 receive interrupt control register	(0050 <sub>16</sub> )...	XXXX?000	UART2 special mode register	(0377 <sub>16</sub> )...	00 <sub>16</sub>
UART0 transmit interrupt control register	(0051 <sub>16</sub> )...	XXXX?000	UART2 transmit/receive mode register	(0378 <sub>16</sub> )...	00 <sub>16</sub>
UART0 receive interrupt control register	(0052 <sub>16</sub> )...	XXXX?000	UART2 transmit/receive control register 0	(037C <sub>16</sub> )...	00001000
UART1 transmit interrupt control register	(0053 <sub>16</sub> )...	XXXX?000	UART2 transmit/receive control register 1	(037D <sub>16</sub> )...	00000010
UART1 receive interrupt control register	(0054 <sub>16</sub> )...	XXXX?000			
Timer A0 interrupt control register	(0055 <sub>16</sub> )...	XXXX?000			
Timer A1 interrupt control register	(0056 <sub>16</sub> )...	XXXX?000			
Timer A2 interrupt control register	(0057 <sub>16</sub> )...	XXXX?000			
Timer A3 interrupt control register	(0058 <sub>16</sub> )...	XXXX?000			
Timer A4 interrupt control register	(0059 <sub>16</sub> )...	XXXX?000			

x : Nothing is mapped to this bit  
? : Undefined  
The content of other registers and RAM is undefined when the microcomputer is reset. The initial values must therefore be set.

Figure 2.3.3 Device's internal status after a reset is cleared

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

Count start flag	(0380 <sub>16</sub> )...	00 <sub>16</sub>	D-A control register	(03DC <sub>16</sub> )...	00 <sub>16</sub>
Clock prescaler reset flag	(0381 <sub>16</sub> )...	0 x x x x x x x	Port P0 direction register	(03E2 <sub>16</sub> )...	00 <sub>16</sub>
One-shot start flag	(0382 <sub>16</sub> )...	0 0 x 0 0 0 0 0	Port P1 direction register	(03E3 <sub>16</sub> )...	00 <sub>16</sub>
Trigger select flag	(0383 <sub>16</sub> )...	00 <sub>16</sub>	Port P2 direction register	(03E6 <sub>16</sub> )...	00 <sub>16</sub>
Up-down flag	(0384 <sub>16</sub> )...	00 <sub>16</sub>	Port P3 direction register	(03E7 <sub>16</sub> )...	00 <sub>16</sub>
Timer A0 mode register	(0396 <sub>16</sub> )...	00 <sub>16</sub>	Port P4 direction register	(03EA <sub>16</sub> )...	00 <sub>16</sub>
Timer A1 mode register	(0397 <sub>16</sub> )...	00 <sub>16</sub>	Port P5 direction register	(03EB <sub>16</sub> )...	00 <sub>16</sub>
Timer A2 mode register	(0398 <sub>16</sub> )...	00 <sub>16</sub>	Port P6 direction register	(03EE <sub>16</sub> )...	00 <sub>16</sub>
Timer A3 mode register	(0399 <sub>16</sub> )...	00 <sub>16</sub>	Port P7 direction register	(03EF <sub>16</sub> )...	00 <sub>16</sub>
Timer A4 mode register	(039A <sub>16</sub> )...	00 <sub>16</sub>	Port P8 direction register	(03F2 <sub>16</sub> )...	0 0 x 0 0 0 0 0
Timer B0 mode register	(039B <sub>16</sub> )...	0 0 ? x 0 0 0 0	Port P9 direction register	(03F3 <sub>16</sub> )...	00 <sub>16</sub>
Timer B1 mode register	(039C <sub>16</sub> )...	0 0 ? x 0 0 0 0	Port P10 direction register	(03F6 <sub>16</sub> )...	00 <sub>16</sub>
Timer B2 mode register	(039D <sub>16</sub> )...	0 0 ? x 0 0 0 0	Pull-up control register 0	(03FC <sub>16</sub> )...	00 <sub>16</sub>
UART0 transmit/receive mode register	(03A0 <sub>16</sub> )...	00 <sub>16</sub>	Pull-up control register 1(Note)	(03FD <sub>16</sub> )...	00 <sub>16</sub>
UART0 transmit/receive control register 0	(03A4 <sub>16</sub> )...	0 0 0 0 1 0 0 0	Pull-up control register 2	(03FE <sub>16</sub> )...	00 <sub>16</sub>
UART0 transmit/receive control register 1	(03A5 <sub>16</sub> )...	0 0 0 0 0 0 1 0	Port control register	(03FF <sub>16</sub> )...	00 <sub>16</sub>
UART1 transmit/receive mode register	(03A8 <sub>16</sub> )...	00 <sub>16</sub>	Data registers (R0/R1/R2/R3)		0000 <sub>16</sub>
UART1 transmit/receive control register 0	(03AC <sub>16</sub> )...	0 0 0 0 1 0 0 0	Address registers (A0/A1)		0000 <sub>16</sub>
UART1 transmit/receive control register 1	(03AD <sub>16</sub> )...	0 0 0 0 0 0 1 0	Frame base register (FB)		0000 <sub>16</sub>
UART transmit/receive control register 2	(03B0 <sub>16</sub> )...	x 0 0 0 0 0 0 0	Interrupt table register (INTB)		00000 <sub>16</sub>
DMA0 cause select register	(03B8 <sub>16</sub> )...	00 <sub>16</sub>	User stack pointer (USP)		0000 <sub>16</sub>
DMA1 cause select register	(03BA <sub>16</sub> )...	00 <sub>16</sub>	Interrupt stack pointer (ISP)		0000 <sub>16</sub>
A-D control register 2	(03D4 <sub>16</sub> )...	0 0 0 0 x x x 0	Static base register (SB)		0000 <sub>16</sub>
A-D control register 0	(03D6 <sub>16</sub> )...	0 0 0 0 0 ? ? ?	Flag register (FLG)		0000 <sub>16</sub>
A-D control register 1	(03D7 <sub>16</sub> )...	00 <sub>16</sub>			

x : Nothing is mapped to this bit  
? : Undefined

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

Note: When the VCC level is applied to the CNVSS pin, it is 02<sub>16</sub> at a reset.

Figure 2.3.4 Device's internal status after a reset is cleared

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.4 Processor Mode

### (1) Types of Processor Mode

Processor mode can be used at microprocessor mode.

One of three processor modes can be selected: single-chip mode and memory expansion 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 and register settings. (See "Bus Settings" for details.)

### (2) Setting Processor Modes

The processor mode is set using the processor mode bits (bits 1 and 0 at address 000416). Do not set the processor mode bits to "102" and "112".

Changing the processor mode bits selects the mode. Therefore, never change the processor mode bits when changing the contents of other bits.

- **Applying VSS to CNVSS pin**

The microcomputer begins operation in single-chip mode after being reset. Memory expansion mode is selected by writing "012" to the processor mode is selected bits..

Figure 2.4.1 shows the processor mode register 0 and 1.

Figure 2.4.2 shows the memory maps applicable for each of the modes when memory area does not be expanded (normal mode).

Note : Microprocessor mode cannot be used.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## Processor mode register 0 (Note 1)

								Symbol	Address	When reset								
<table border="1"><tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr></table>								b7	b6	b5	b4	b3	b2	b1	b0	PM0	0004 <sub>16</sub>	00 <sub>16</sub>
b7	b6	b5	b4	b3	b2	b1	b0											
								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: Inhibited	○	○						
								PM01			○	○						
								PM02	R/W mode select bit	0 : RD,BHE,WR 1 : RD,WRH,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 (Note 3)	○	○						
								PM05			○	○						
								PM06	Port P4 <sub>0</sub> to P4 <sub>3</sub> function select bit (Note 2)	0 : Address output 1 : Port function (Address is not output)	○	○						
								PM07	BCLK output disable bit	0 : BCLK is output 1 : BCLK is not output (Pin is left floating)	○	○						

Notes 1: Set bit 1 of the protect register (address 000A<sub>16</sub>) to "1" when writing new values to this register.

2: Valid in memory expansion mode.

3: If the entire space is of multiplexed bus in memory expansion mode, chose an 8-bit width.

The higher-order address becomes a port if the entire space multiplexed bus is chosen, so only 256 bytes can be in used in each chip select.

## Processor mode register 1 (Note)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	When reset
	0	0	0	0	×	×	0	PM1	0005 <sub>16</sub>	00000XX0 <sub>2</sub>

Bit symbol	Bit name	Function	R	W
	Reserved bit	Must always be set to "0"	○	○
	Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.		—	—
	Reserved bit	Must always be set to "0"	—	○
	Reserved bit	Must always be set to "0"	○	○
PM17	Wait bit	0 : No wait state 1 : Wait state inserted	○	○

Note: Set bit 1 of the protect register (address 000A<sub>16</sub>) to "1" when writing new values to this register.

Figure 2.4.1 Processor mode registers

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

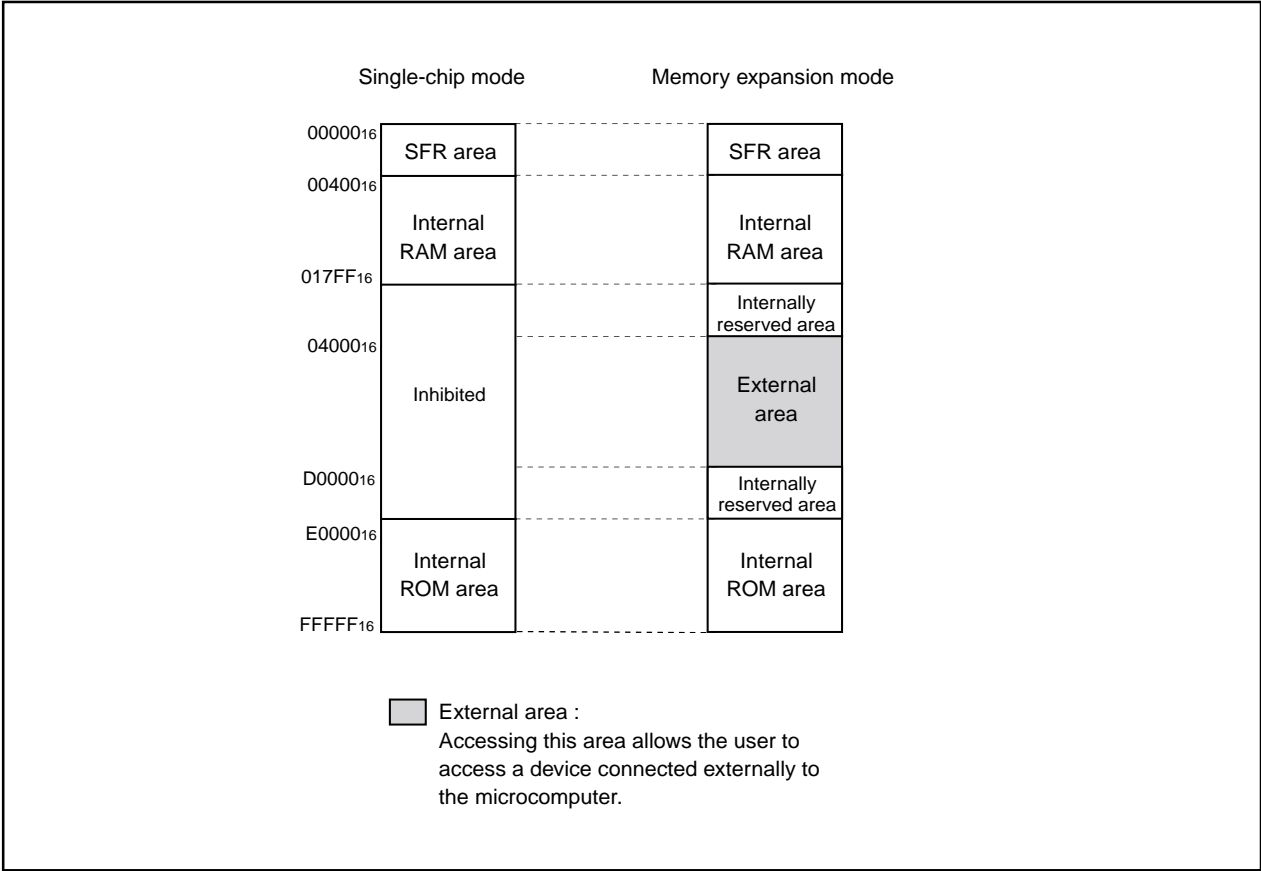


Figure 2.4.2 Memory maps in each processor mode

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.4.1 Bus settings

The BYTE pin and bits 4 to 6 of the processor mode register 0 (address 000416) are used to change the bus settings. Table 2.4.1 shows the factors used to change the bus settings.

**Table 2.4.1 Factors for switching bus settings**

Bus setting	Switching factor
Switching external address bus width	Bit 6 of processor mode register 0
Switching external data bus width	BYTE pin
Switching between separate and multiplex bus	Bits 4 and 5 of processor mode register 0

### (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. P40 to P43 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 P40 to P43 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 D0 to D7 are multiplexed with A0 to A7.

With a 16-bit data bus selected (BYTE pin = "L"), the 8 bits from D0 to D7 are multiplexed with A1 to A8. D8 to D15 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 P56.

Before using the multiplex bus for access, be sure to insert a software wait.

If the entire space is of multiplexed bus in memory expansion mode, choose an 8-bit width.

The higher-order address become a port of the entire space multiplexed bus is chosen, so only 256 bytes can be used in each chip select.



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

**Table 2.4.2 Pin functions for processor mode**

Processor mode	Single-chip mode	Memory expansion mode				Memory expansion mode
Multiplexed bus space select bit		"01" [ Either CS1 or CS2 is for multiplexed bus and others are for separate bus ]		"00" (separate bus)		"11"(Note 1) [ 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"
P00 to P07	I/O port	Data bus	Data bus	Data bus	Data bus	I/O port
P10 to P17	I/O port	I/O port	Data bus	I/O port	Data bus	I/O port
P20	I/O port	Address bus /data bus (Note)	Address bus	Address bus	Address bus	Address bus /data bus
P21 to P27	I/O port /data bus (Note)	Address bus /data bus (Note)	Address bus	Address bus	Address bus /data bus	Address bus
P30	I/O port /data bus (Note)	Address bus	Address bus	Address bus	Address bus	A8/D7
P31 to P37	I/O port	Address bus	Address bus	Address bus	Address bus	I/O port
P40 to P43 Port P40 to P43 function select bit = 1	I/O port	I/O port	I/O port	I/O port	I/O port	I/O port
P40 to P43 Port P40 to P43 function select bit = 0	I/O port	Address bus	Address bus	Address bus	Address bus	I/O port
P44 to P47	I/O port	CS (chip select) or programmable I/O port (For details, refer to "Bus control")				
P50 to P53	I/O port	Outputs RD, WRL, WRH, and BCLK or RD, BHE, WR, and BCLK (For details, refer to "Bus control")				
P54	I/O port	HLDA	HLDA	HLDA	HLDA	HLDA
P55	I/O port	HOLD	HOLD	HOLD	HOLD	HOLD
P56	I/O port	ALE	ALE	ALE	ALE	ALE
P57	I/O port	RDY	RDY	RDY	RDY	RDY

Note 1: If the entire space is of multiplexed bus in memory expansion mode, chose an 8-bit width.

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.

2: Address bus when in separate bus mode.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.4.2 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.

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

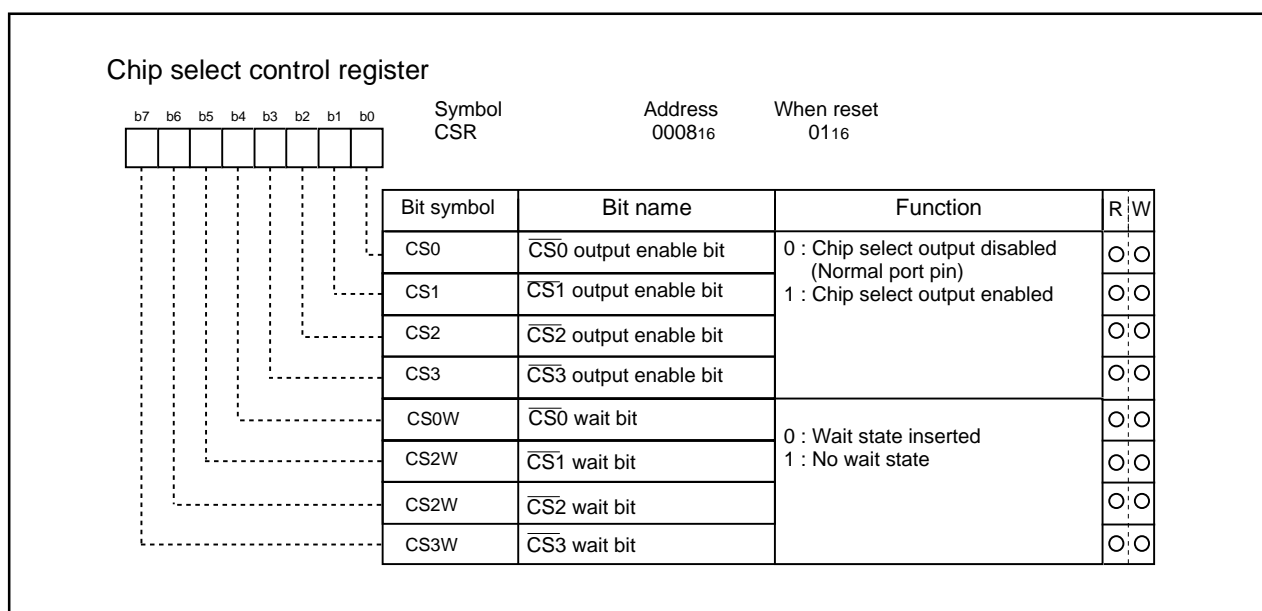
The chip select signal is output using the same pins as P44 to P47. Bits 0 to 3 of the chip select control register (address 0008<sub>16</sub>) set each pin to function as a port or to output the chip select signal. The chip select control register is valid in memory expansion mode. IN single-chip mode, P44 to P47 function as programmable I/O ports regardless of the value in the chip select control register.

Figure 2.4.3 shows the chip select control register.

The chip select signal can be used to split the external area. Tables 2.4.3 show the external memory areas specified using the chip select signal.

**Table 2.4.3 External areas specified by the chip select signals**

Processor mode	Chip select signal			
	$\overline{CS0}$	$\overline{CS1}$	$\overline{CS2}$	$\overline{CS3}$
Memory expansion mode	30000 <sub>16</sub> to CFFFF <sub>16</sub> (640K bytes)	28000 <sub>16</sub> to 2FFFF <sub>16</sub> (32K bytes)	08000 <sub>16</sub> to 27FFF <sub>16</sub> (128K bytes)	04000 <sub>16</sub> to 07FFF <sub>16</sub> (16K bytes)



**Figure 2.4.3 Chip select control register**

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

### (3) Read/write signals

With a 16-bit data bus (BYTE pin = "L"), bit 2 of the processor mode register 0 (address 0004<sub>16</sub>) select 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<sub>16</sub>) to "0".) Tables 2.4.4 and 2.4.5 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<sub>16</sub>) 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<sub>16</sub>) to "1".

**Table 2.4.4 Operation of  $\overline{RD}$ ,  $\overline{WRL}$ , and  $\overline{WRH}$  signals**

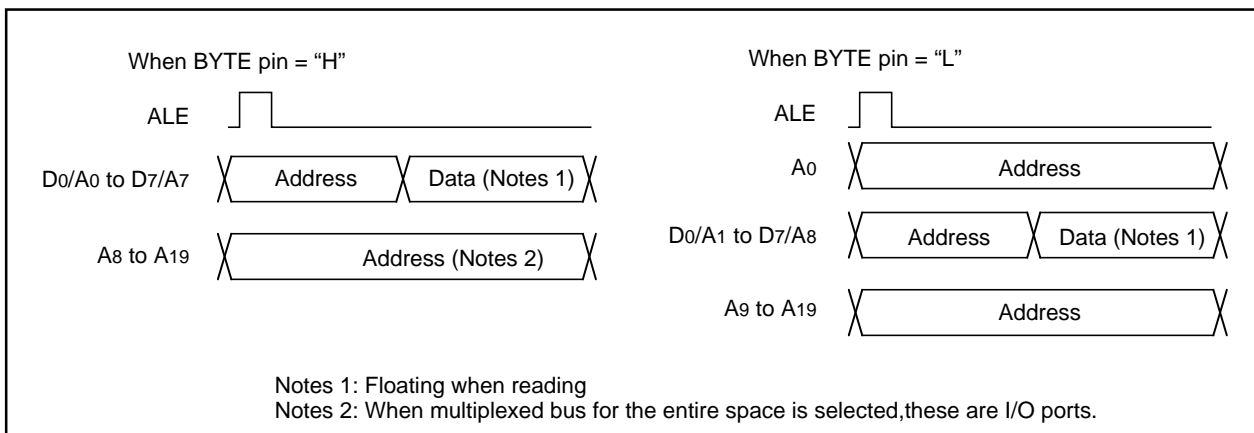
Data bus width	$\overline{RD}$	$\overline{WRL}$	$\overline{WRH}$	Status of external data bus
16-bit (BYTE = "L")	L	H	H	Read data
	H	L	H	Write 1 byte of data to even address
	H	H	L	Write 1 byte of data to odd address
	H	L	L	Write data to both even and odd addresses

**Table 2.4.5 Operation of  $\overline{RD}$ ,  $\overline{WR}$ , 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 2.4.4 ALE signal and address/data bus**

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (5) The $\overline{\text{RDY}}$ signal

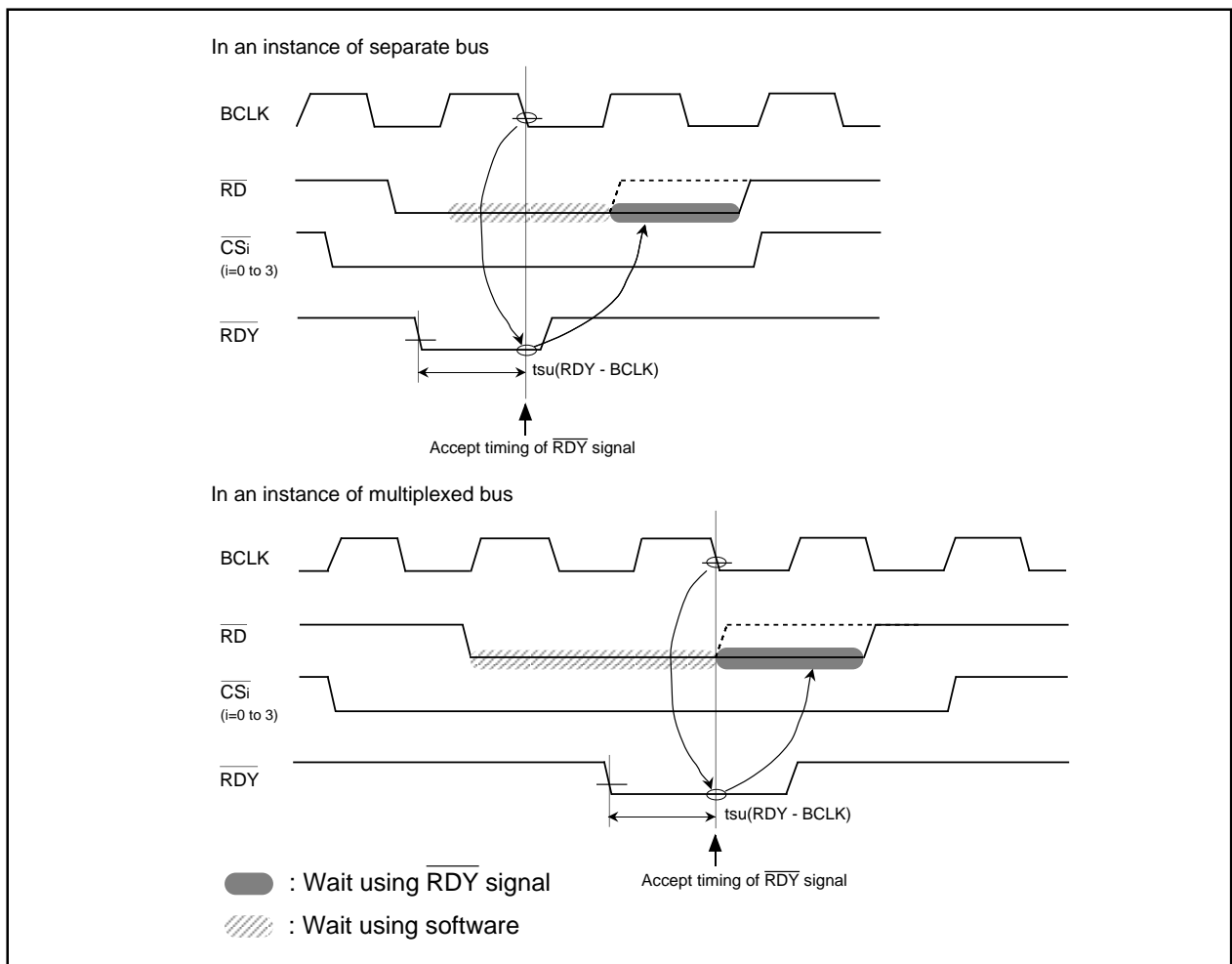
$\overline{\text{RDY}}$  is a signal that facilitates access to an external device that requires long access time. As shown in Figure 2.4.5, if an “L” is being input to the  $\overline{\text{RDY}}$  at the BCLK falling edge, the bus turns to the wait state. If an “H” is being input to the  $\overline{\text{RDY}}$  pin at the BCLK falling edge, the bus cancels the wait state. Table 2.4.6 shows the state of the microcomputer with the bus in the wait state, and Figure 2.4.5 shows an example in which the  $\overline{\text{RD}}$  signal is prolonged by the  $\overline{\text{RDY}}$  signal.

The  $\overline{\text{RDY}}$  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<sub>16</sub>) are set to “0”. The  $\overline{\text{RDY}}$  signal is invalid when setting “1” to all bits 4 to 7 of the chip select control register (address 0008<sub>16</sub>), but the  $\overline{\text{RDY}}$  pin should be treated as properly as in non-using.

**Table 2.4.6 Microcomputer status in ready state (Note)**

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

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



**Figure 2.4.5 Example of  $\overline{\text{RD}}$  signal extended by  $\overline{\text{RDY}}$  signal**

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## (6) Hold signal

The hold signal is used to transfer the bus privileges from the CPU to the external circuits. Inputting "L" to the  $\overline{\text{HOLD}}$  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  $\overline{\text{HLDA}}$  pin as long as "L" is input to the  $\overline{\text{HOLD}}$  pin. Table 2.4.7 shows the microcomputer status in the hold state.

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

$\overline{\text{HOLD}} > \text{DMAC} > \text{CPU}$

**Figure 2.4.6 Bus-using priorities**

**Table 2.4.7 Microcomputer status in hold state**

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

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

Table 2.4.8 shows the external bus status when the internal area is accessed.

**Table 2.4.8 External bus status when the internal area is accessed**

Item		SFR accessed	Internal 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
$\overline{\text{RD}}$ , $\overline{\text{WR}}$ , $\overline{\text{WRL}}$ , $\overline{\text{WRH}}$		$\overline{\text{RD}}$ , $\overline{\text{WR}}$ , $\overline{\text{WRL}}$ , $\overline{\text{WRH}}$ output	Output "H"
$\overline{\text{BHE}}$		$\overline{\text{BHE}}$ output	Maintain status before accessed status of external area
$\overline{\text{CS}}$		Output "H"	Output "H"
ALE		Output "L"	Output "L"

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (8) BCLK output

The user can choose the BCLK output by use of bit 7 of processor mode register 0 (0004<sub>16</sub>) (Note). When set to "1", the output floating.

Note: Before attempting to change the contents of the processor mode register 0, set bit 1 of the protectregister (address 000A<sub>16</sub>) 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<sub>16</sub>) (Note) and bits 4 to 7 of the chip select control register (address 0008<sub>16</sub>).

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", a wait is applied to all memory areas (two or three BCLK cycles), regardless of the contents of bits 4 to 7 of the chip select control register. Set this bit after referring to the recommended operating conditions (main clock input oscillation frequency) of the electric characteristics. However, when the user is using the  $\overline{\text{RDY}}$  signal, the relevant bit in the chip select control register's bits 4 to 7 must be set to "0".

When the wait bit of the processor mode register 1 is "0", software waits can be set independently for each areas selected using the chip select signal. Bits 4 to 7 of the chip select control register correspond to chip selects  $\overline{\text{CS0}}$  to  $\overline{\text{CS3}}$ . 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, insert a software wait if using the multiplex bus to access the external memory area.

Table 2.4.9 shows the software wait and bus cycles. Figure 2.4.7 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<sub>16</sub>) to "1".

**Table 2.4.9 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	3 BCLK cycles
	Multiplex bus	1	0 (Note)	3 BCLK cycles

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

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

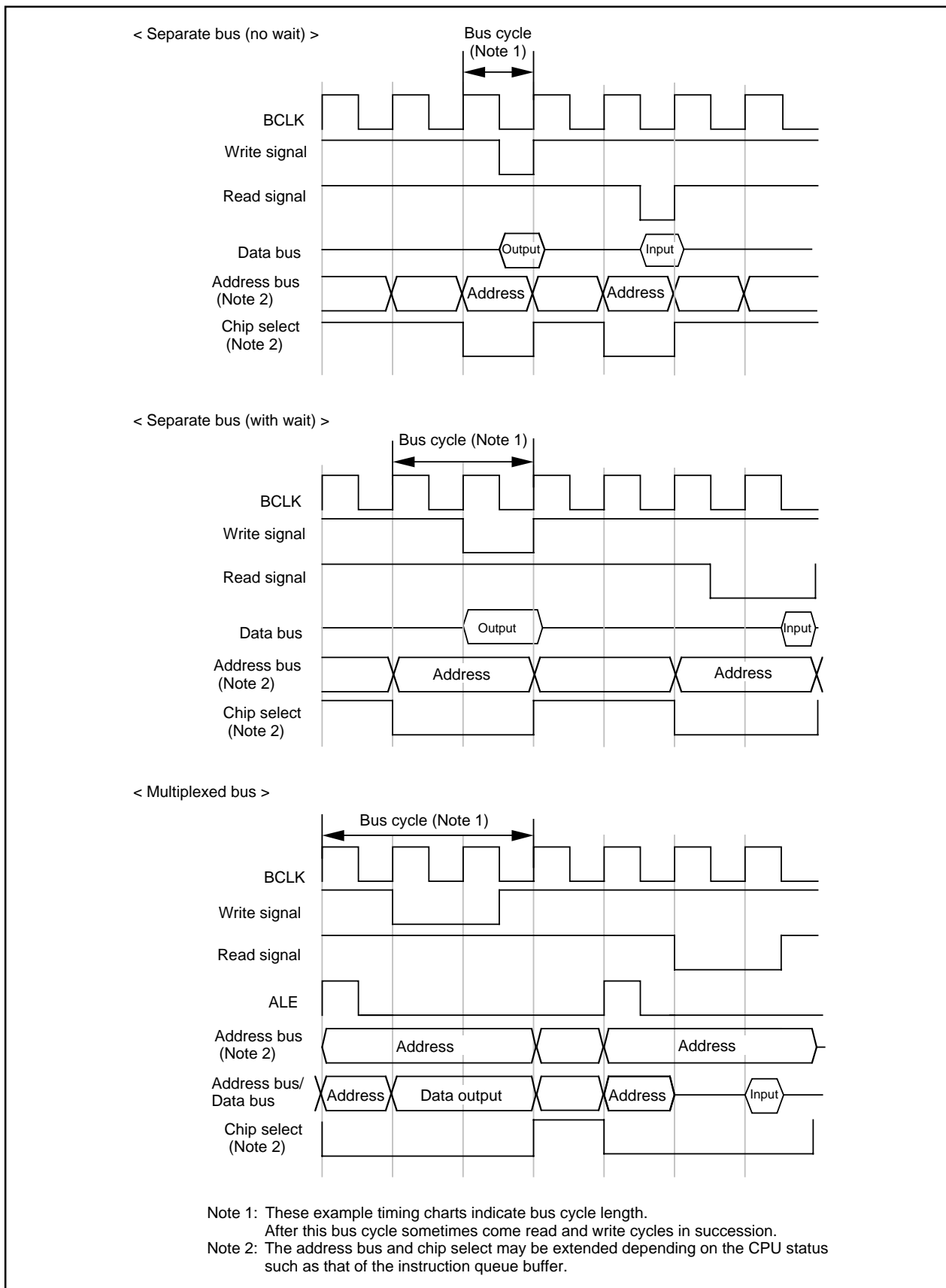


Figure 2.4.7 Typical bus timings using software wait

# M306H2FCFP

 SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.5 Clock Generating Circuit

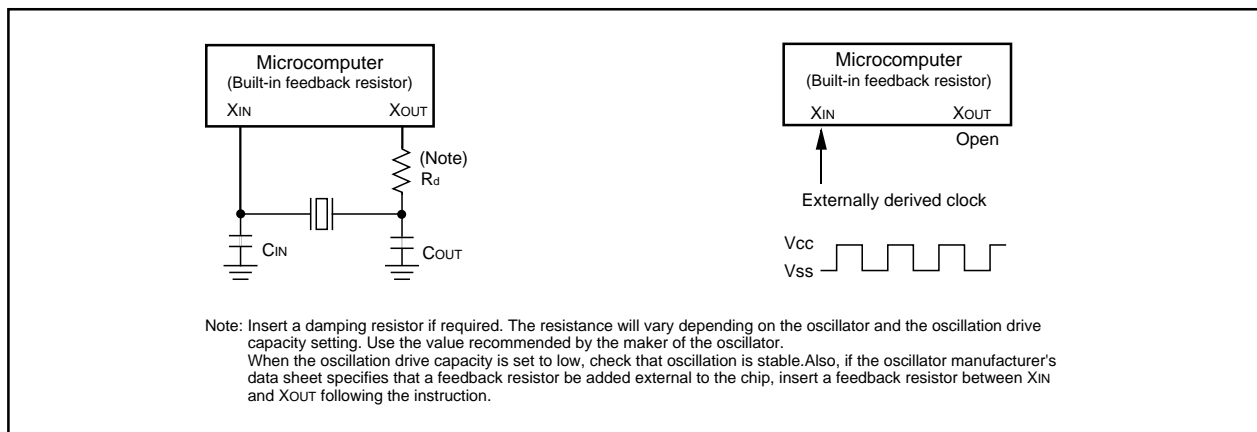
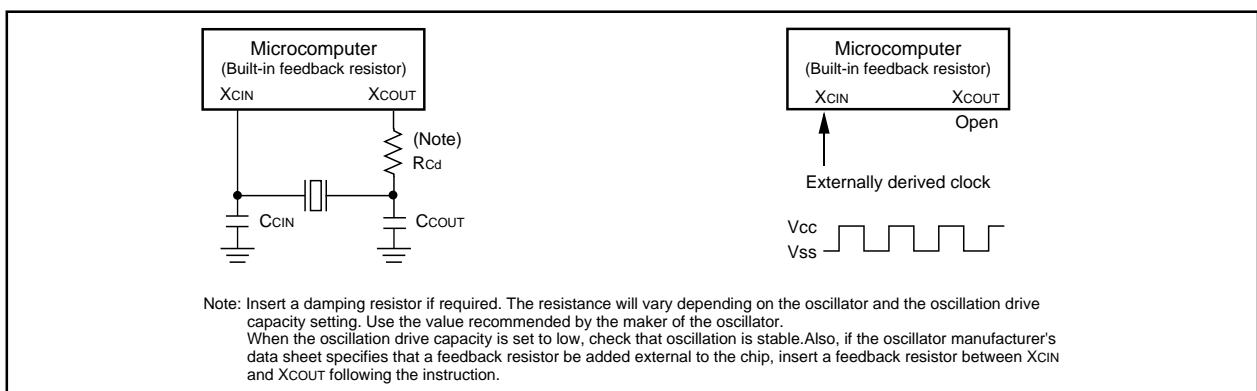
The clock generating circuit contains two oscillator circuits that supply the operating clock sources to the CPU and internal peripheral units.

**Table 2.5.1 Main clock and sub clock generating circuits**

	Main clock generating circuit	Sub clock generating circuit
Use of clock	<ul style="list-style-type: none"> <li>CPU's operating clock source</li> <li>Internal peripheral units' operating clock source</li> </ul>	<ul style="list-style-type: none"> <li>CPU's operating clock source</li> <li>Timer A/B's count clock source</li> </ul>
Usable oscillator	Ceramic or crystal oscillator	Crystal oscillator
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT
Oscillation stop/restart function	Available	Available
Oscillator status immediately after reset	Oscillating	Stopped
Other	Externally derived clock can be input	

### 2.5.1 Example of oscillator circuit

Figure 2.5.1 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 2.5.2 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 2.5.1 and 2.5.2 vary with each oscillator used. Use the values recommended by the manufacturer of your oscillator.


**Figure 2.5.1 Examples of main clock**

**Figure 2.5.2 Examples of sub clock**



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.5.2 Clock Control

Figure 2.5.3 shows the block diagram of the clock generating circuit.

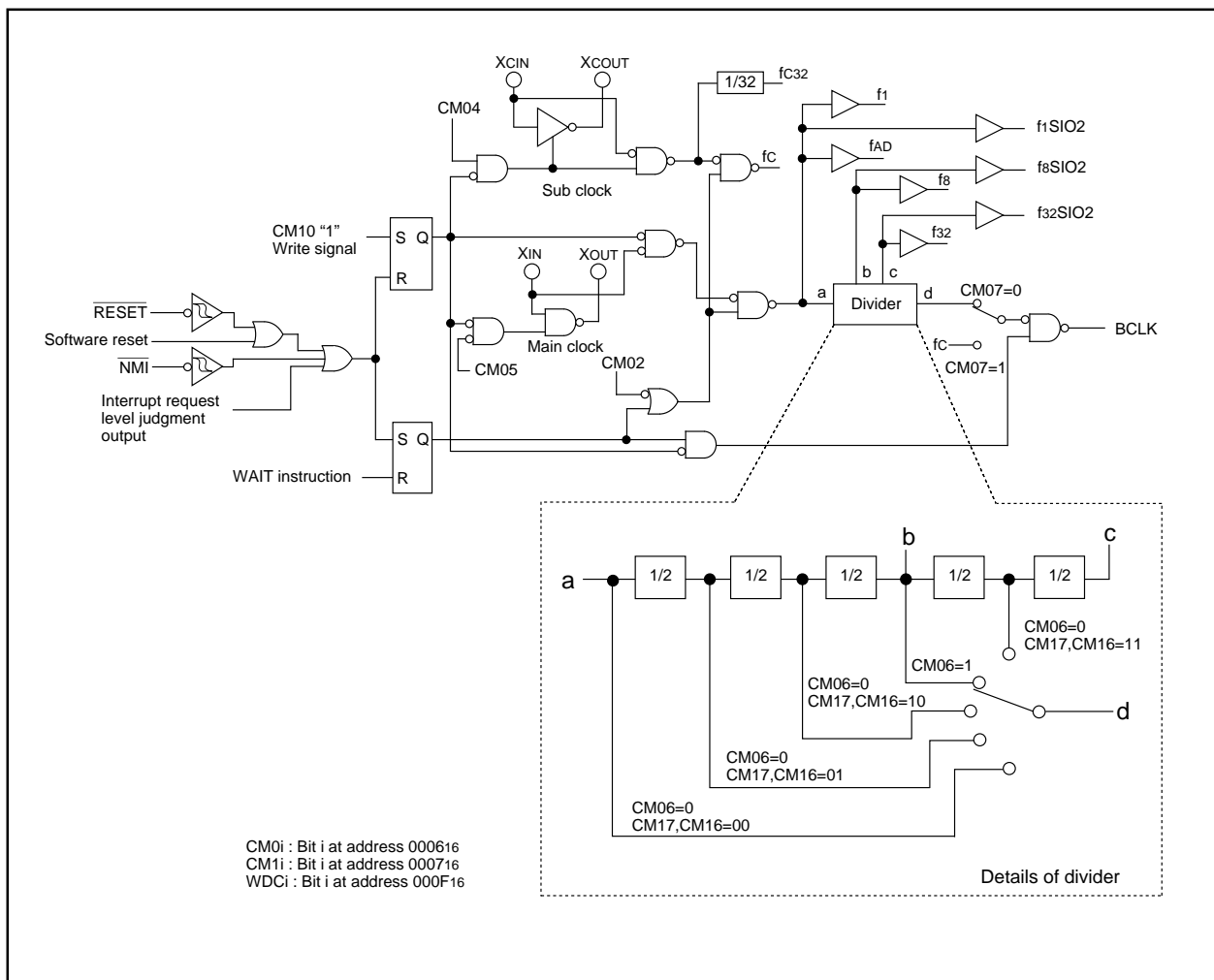


Figure 2.5.3 Clock generating circuit

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

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

## (1) Main clock

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

## (2) Sub-clock

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

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

## (3) BCLK

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

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

## (4) Peripheral function clock(f<sub>1</sub>, f<sub>8</sub>, f<sub>32</sub>, f<sub>1SIO2</sub>, f<sub>8SIO2</sub>, f<sub>32SIO2</sub>, f<sub>AD</sub>)

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

## (5) fc<sub>32</sub>

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

## (6) fc

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

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

Figure 2.5.4 shows the system clock control registers 0 and 1.

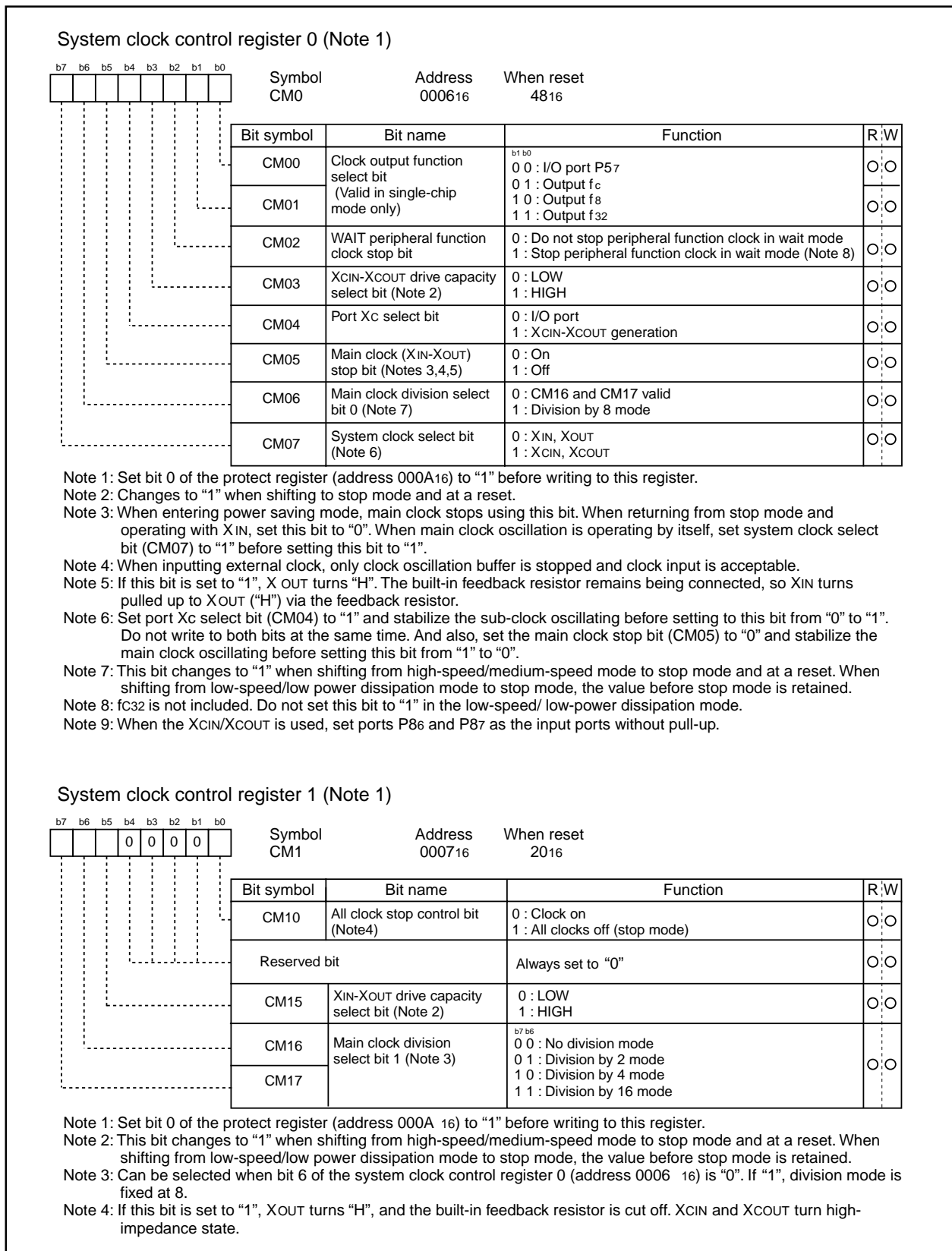


Figure 2.5.4 Clock control registers 0 and 1

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.5.3 Clock output

In single-chip mode, the clock output function select bits (bits 0 and 1 at address 0006<sub>16</sub>) enable f<sub>8</sub>, f<sub>32</sub>, or f<sub>c</sub> to be output from the P57/CLKOUT pin. When the WAIT peripheral function clock stop bit (bit 2 at address 0006<sub>16</sub>) is set to "1", the output of f<sub>8</sub> and f<sub>32</sub> stops when a WAIT instruction is executed.

## 2.5.4 Stop Mode

Writing "1" to the main clock and sub-clock stop control bit (bit 0 at address 0007<sub>16</sub>) stops oscillation and the microcomputer enters stop mode. In stop mode, the content of the internal RAM is retained provided that V<sub>CC</sub> remains above 2V.

The internal oscillator circuit of expansion function (Data acquisition / humming function) stops oscillation when expansion register XTAL\_VCO, PDC\_VCO\_ON, VPS\_VCO\_ON = "L".

Because the oscillation, BCLK, f<sub>1</sub> to f<sub>32</sub>, f<sub>1</sub>SI02 to f<sub>32</sub>SI02, f<sub>c</sub>, f<sub>c</sub>32, and f<sub>AD</sub> stops 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 UART<sub>i</sub>(i = 0 to 2) SI/O3,4 functions provided an external clock is selected. Table 2.5.2 shows the status of the ports in stop mode. Stop mode is cancelled by a hardware reset or interrupt. If an interrupt is to be used to cancel stop mode, that interrupt must first have been enabled. If returning by an interrupt, that interrupt routine is executed. When shifting from high-speed/medium-speed mode to stop mode and at a reset, the main clock division select bit 0 (bit 6 at address 0006<sub>16</sub>) is set to "1". When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

**Table 2.5.2 Port status during stop mode**

Pin		Memory expansion mode	Single-chip mode
Address bus, data bus, CS <sub>0</sub> to CS <sub>3</sub> , BHE		Retains status before wait mode	
RD, WR, WRL, WRH		"H "	
HLDA, BCLK		"H "	
ALE		"H "	
Port		Retains status before wait mode	Retains status before wait mode
CLKOUT	When f <sub>c</sub> selected	Valid only in single-chip mode	"H "
	When f <sub>8</sub> , f <sub>32</sub> selected	Valid only in single-chip mode	Retains status before wait mode

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## 2.5.5 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 watchdog timer stop. Writing "1" to the WAIT peripheral function clock stop bit and executing a WAIT instruction stops the clock being supplied to the internal peripheral functions, allowing power dissipation to be reduced. However, peripheral function clock fC32 does not stop so that the peripherals using fC32 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 2.5.3 shows the status of the ports in wait mode.

Wait mode is cancelled by a hardware reset or an interrupt. When using an interrupt to exit wait mode, make sure the interrupt used for that purpose is enabled and those not used for that purpose have their priority levels set to "0" before entering wait mode. When restored from wait mode by an interrupt, the microcomputer restarts operation from the interrupt routine using as BCLK the clock with which it was operating when the WAIT instruction was executed. When using a hardware reset or  $\overline{\text{NMI}}$  interrupt only, be sure to set the priority levels of all other interrupts to 0 before entering wait mode.

**Table 2.5.3 Port status during wait mode**

Pin		Memory expansion mode	Single-chip mode
Address bus, data bus, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$		Retains status before stop mode	
$\overline{\text{BHE}}$			
RD, WR, WRL, WRH		"H"	
HLDA, BCLK		"H"	
ALE		"H"	
Port		Retains status before wait mode	Retains status before wait mode
CLKOUT	When fc selected	Valid only in single-chip mode	Does not stop
	When f8, f32 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.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## 2.5.6 Status Transition Of BCLK

Power dissipation can be reduced and low-voltage operation achieved by changing the count source for BCLK. Table 2.5.4 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<sub>16</sub>) changes to “1” when shifting from high-speed/medium-speed to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained. The following shows the operational modes of BCLK.

### (1) Division by 2 mode

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

### (2) Division by 4 mode

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

### (3) Division by 8 mode

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

### (4) Division by 16 mode

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

### (5) No-division mode

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

### (6) Low-speed mode

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

### (7) Low power dissipation mode

fc is the BCLK and the main clock is stopped.

Note :

Before the count source for BCLK can be changed from X<sub>IN</sub> to X<sub>CIN</sub> or vice versa, the clock to which the count source is going to be switched must be oscillating stably. Allow a wait time in software for the oscillation to stabilize before switching over the clock.

**Table 2.5.4 Operating modes dictated by settings of system clock control registers 0 and 1**

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

CM1i: bit i of address 0007<sub>16</sub>

CM0i: bit i of address 0006<sub>16</sub>

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.5.7 Power control

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

### Modes

Power control is available in three modes.

#### (a) Normal operation mode

##### • High-speed mode

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

##### • Medium-speed mode

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

##### • Low-speed mode

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

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

When in single-chip mode, the device can be operated with a low supply voltage ( $V_{CC} = 3.0\text{ V}$ ) only during low power dissipation mode. Before entering or exiting low power dissipation mode, always make sure the supply voltage  $V_{CC}$  is 5 V.

Note: When operating with a low supply voltage, be aware that only the CPU, ROM, RAM, input/output ports, timers (timers A and B), and the interrupt control circuit can be used. All other internal resources (e.g., data slicer, DMAC, A/D, and D/A) cannot be used.

#### (b) Wait mode

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

#### (c) Stop mode

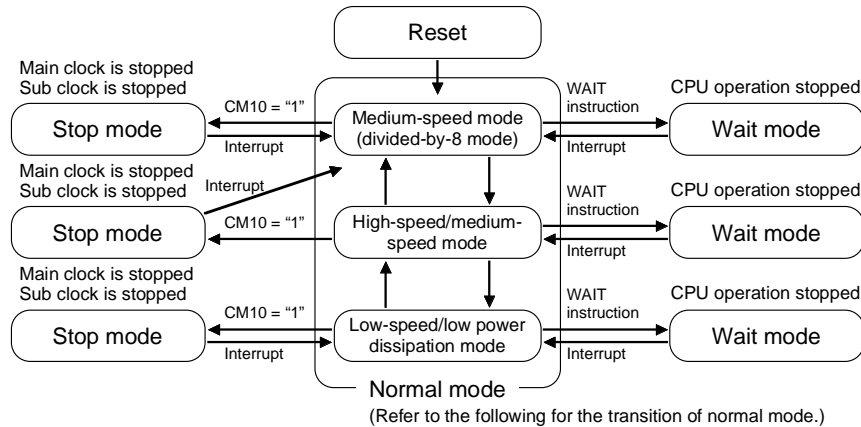
The main clock and the sub-clock oscillators stop. The CPU and all built-in peripheral functions stop. This mode, among the three modes listed here, is the most effective in decreasing power consumption.

Figure 2.5.5 is the state transition diagram of the above modes.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## Transition of stop mode, wait mode



## Transition of normal mode

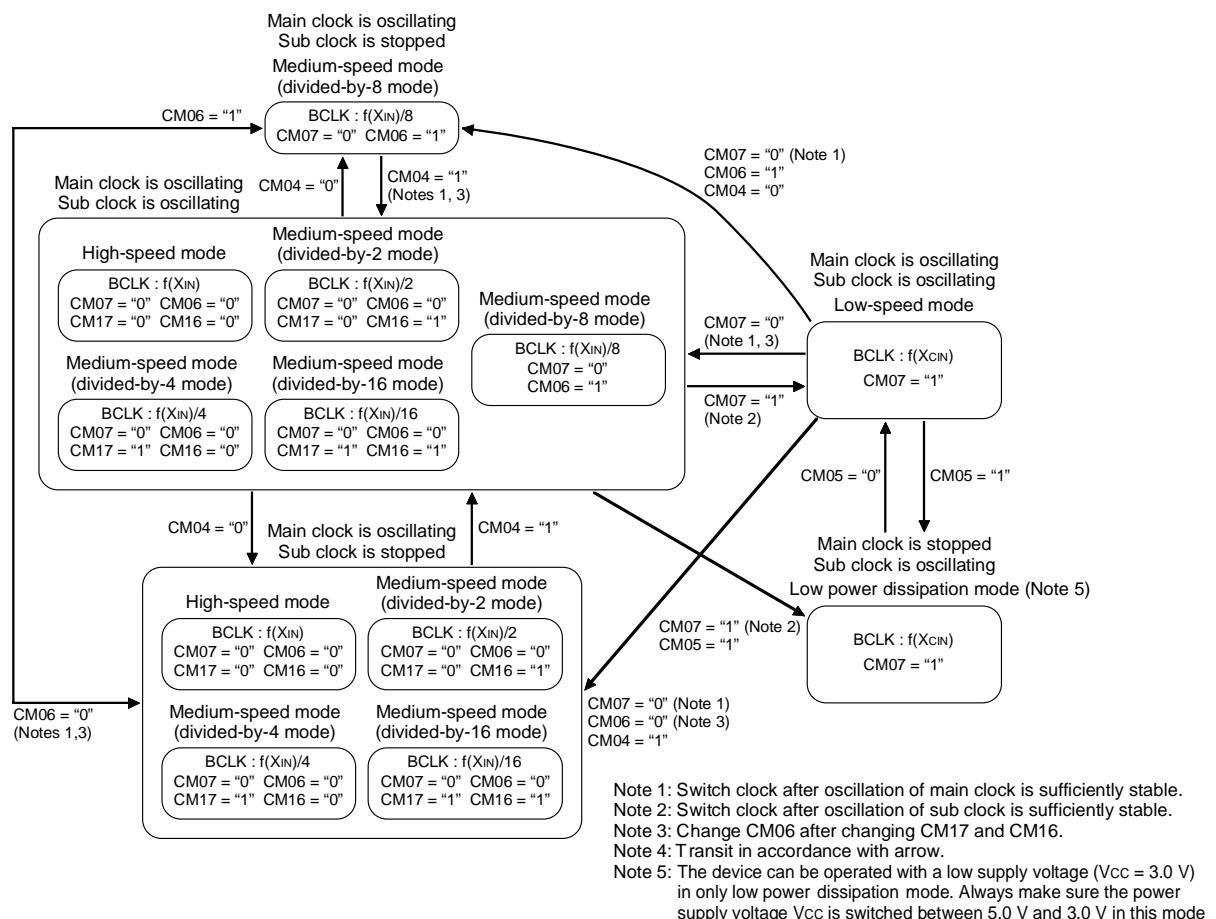


Figure 2.5.5 State transition diagram of Power control mode



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.6 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 2.6.1 shows the protect register. The values in the processor mode register 0 (address 0004<sub>16</sub>), processor mode register 1 (address 0005<sub>16</sub>), system clock control register 0 (address 0006<sub>16</sub>), system clock control register 1 (address 0007<sub>16</sub>), port P9 direction register (address 03F3<sub>16</sub>), SI/O3 control register (address 0362<sub>16</sub>) and SI/O4 control register (address 0366<sub>16</sub>) can only be changed when the respective bit in the protect register is set to "1". Therefore, important outputs can be allocated to port P9.

If, after "1" (write-enabled) has been written to the port P9 direction register and SI/Oi control register (i=3,4) write-enable bit (bit 2 at address 000A<sub>16</sub>), 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 000A<sub>16</sub>) and processor mode register 0 and 1 write-enable bit (bit 1 at 000A<sub>16</sub>) 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".

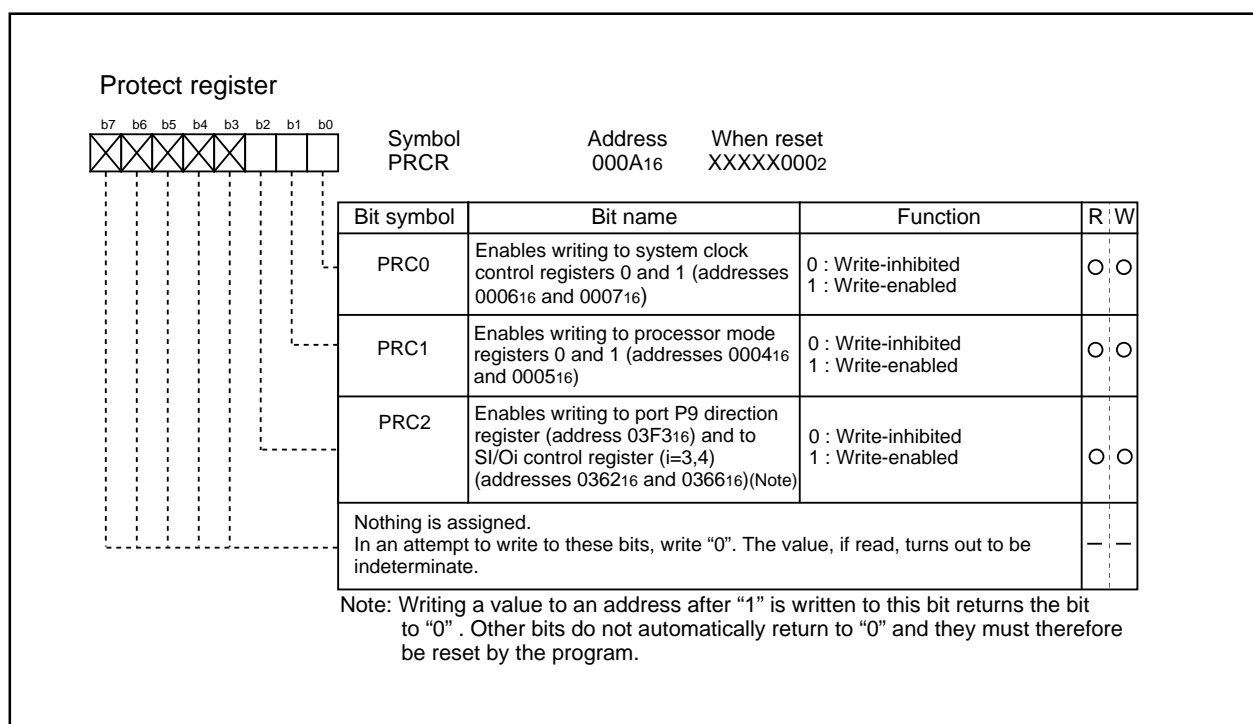


Figure 2.6.1 Protect register

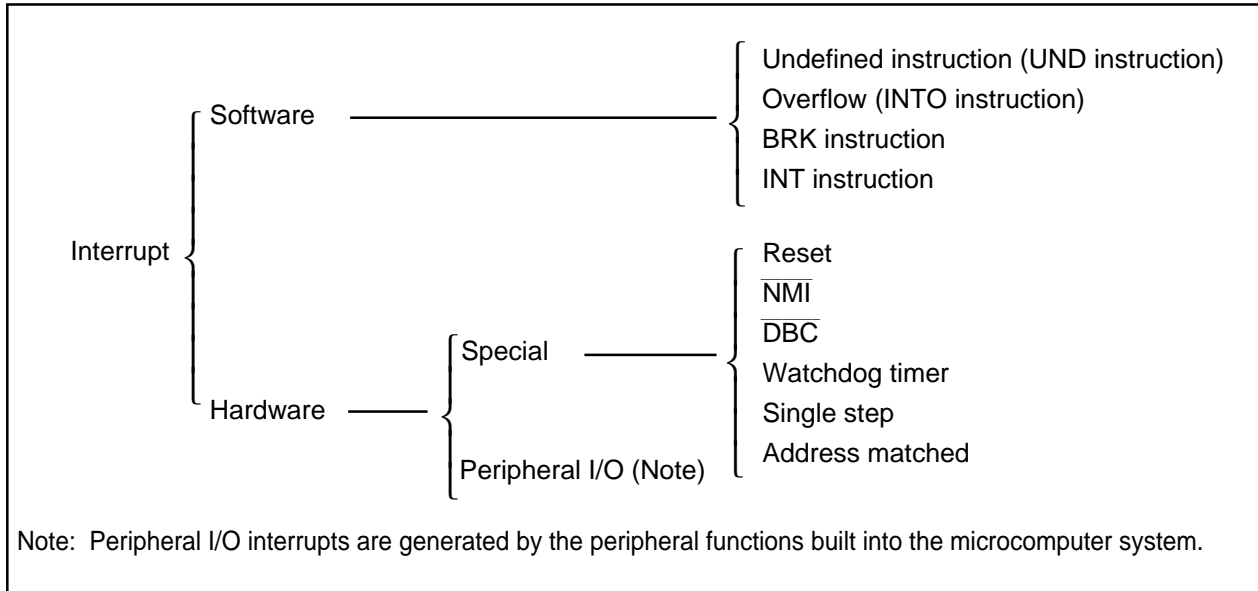
# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.7 Interrupt

### 2.7.1 Interrupt

Figure 2.7.1 lists the types of interrupts.



**Figure 2.7.1 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.

## 2.7.2 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 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.

## 2.7.3 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.

- **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.

- **Watchdog timer interrupt**

Generated by the watchdog timer.

- **Single-step interrupt**

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

- **Address match interrupt**

An address match interrupt occurs immediately before the instruction held in the address indicated by the address match interrupt register is executed with the address match interrupt enable bit set to “1”.

If an address other than the first address of the instruction in the address match interrupt register is set, no address match interrupt occurs. For address match interrupt, see 2.7.10 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, SI/O3 and SI/O4 transmission interrupt**

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

- **UART0, UART1, UART2/ACK, SI/O3 and SI/O4 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  $\overline{\text{INT5}}$  interrupt**

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

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.7.4 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 2.7.2 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.

	MSB	LSB
Vector address + 0	Low address	
Vector address + 1	Mid address	
Vector address + 2	0 0 0 0	High address
Vector address + 3	0 0 0 0	0 0 0 0

**Figure 2.7.2 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 FFFDC<sub>16</sub> to FFFFF<sub>16</sub>. One vector table comprises four bytes. Set the first address of interrupt routine in each vector table. Table 2.7.1 shows the interrupts assigned to the fixed vector tables and addresses of vector tables.

**Table 2.7.1 Interrupts assigned to the fixed vector tables and addresses of vector tables**

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks
Undefined instruction	FFFD <sub>C16</sub> to FFFD <sub>F16</sub>	Interrupt on UND instruction
Overflow	FFFE <sub>016</sub> to FFFE <sub>316</sub>	Interrupt on INTO instruction
BRK instruction	FFFE <sub>416</sub> to FFFE <sub>716</sub>	If the vector contains FF <sub>16</sub> , program execution starts from the address shown by the vector in the variable vector table
Address match	FFFE <sub>816</sub> to FFFE <sub>B16</sub>	There is an address-matching interrupt enable bit
Single step (Note)	FFFE <sub>C16</sub> to FFFE <sub>F16</sub>	Do not use
Watchdog timer	FFFF <sub>016</sub> to FFFF <sub>316</sub>	
DBC (Note)	FFFF <sub>416</sub> to FFFF <sub>716</sub>	Do not use
NMI	FFFF <sub>816</sub> to FFFF <sub>B16</sub>	External interrupt by input to NMI pin
Reset	FFFF <sub>C16</sub> to FFFF <sub>F16</sub>	

Note: Interrupts used for debugging purposes only.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## • Variable vector tables

The addresses in the variable vector table can be modified, according to the user's settings. Indicate the first address using the interrupt table register (INTB). The 256-byte area subsequent to the address the INTB indicates becomes the area for the variable vector tables. One vector table comprises four bytes. Set the first address of the interrupt routine in each vector table. Table 2.7.2 shows the interrupts assigned to the variable vector tables and addresses of vector tables.

**Table 2.7.2 Interrupts assigned to the variable vector tables and addresses of vector tables**

Software interrupt number	Vector table address Address (L) to address (H)	Interrupt source	Remarks
Software interrupt number 0	+0 to +3 (Note 1)	BRK instruction	Cannot be masked I flag
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)	SI/O4/ $\overline{\text{INT5}}$ (Note 2)	
Software interrupt number 9	+36 to +39 (Note 1)	SI/O3/ $\overline{\text{INT4}}$ (Note 2)	
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)	Key input interrupt	
Software interrupt number 14	+56 to +59 (Note 1)	A-D	
Software interrupt number 15	+60 to +63 (Note 1)	UART2 transmit/NACK (Note 3)	
Software interrupt number 16	+64 to +67 (Note 1)	UART2 receive/ACK (Note 3)	
Software interrupt number 17	+68 to +71 (Note 1)	UART0 transmit	
Software interrupt number 18	+72 to +75 (Note 1)	UART0 receive	
Software interrupt number 19	+76 to +79 (Note 1)	UART1 transmit	
Software interrupt number 20	+80 to +83 (Note 1)	UART1 receive	
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 bit (bit 6, 7 in address 035F16 ).

Note 3: When IIC mode is selected, NACK and ACK interrupts are selected.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

---

## 2.7.5 Interrupt Control

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

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

Figure 2.7.3 shows the memory map of the interrupt control registers.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

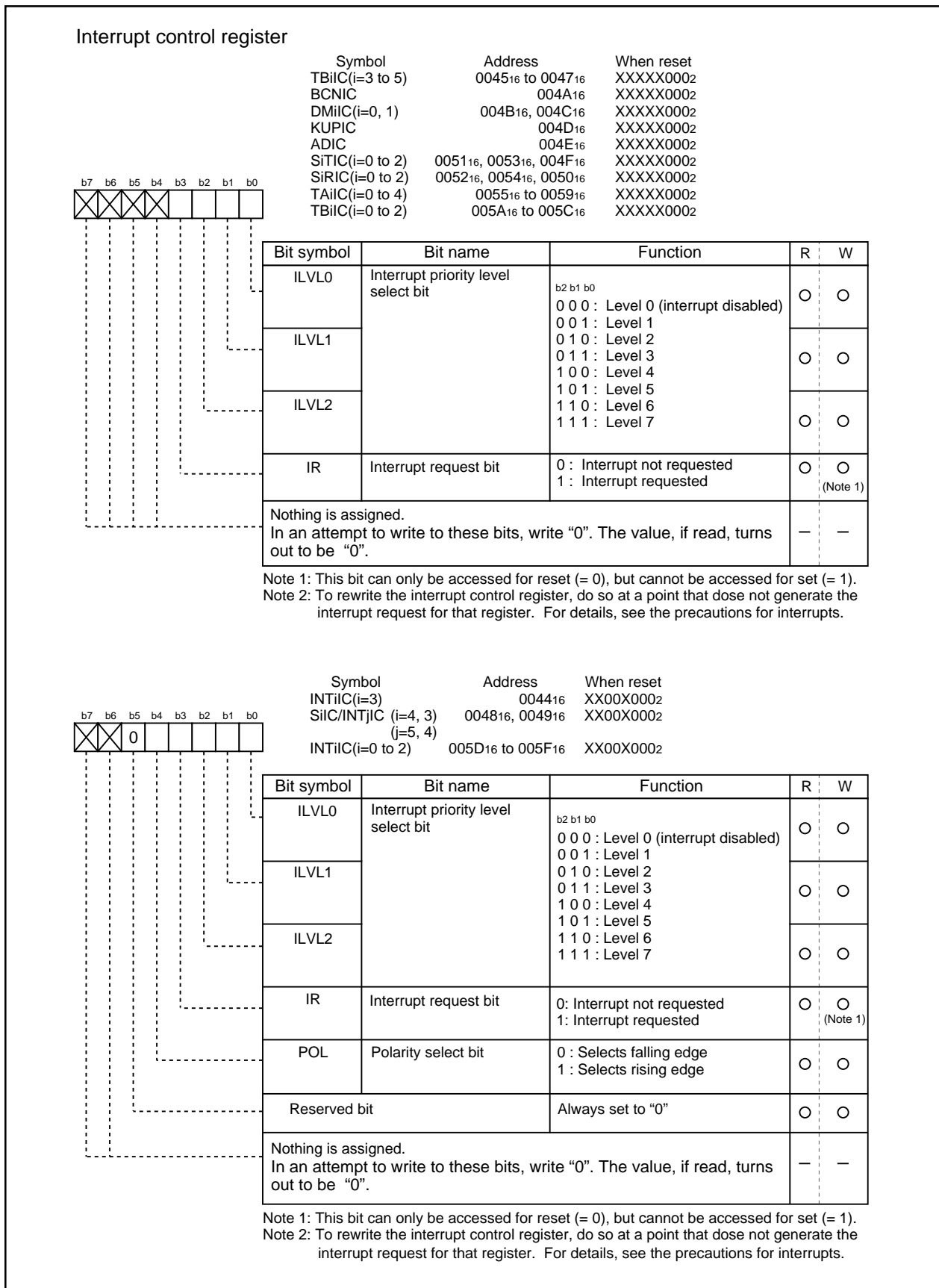


Figure 2.7.3 Interrupt control registers



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (1) Interrupt Enable Flag (I flag)

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

## (2) 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").

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

Set the interrupt priority level using the interrupt priority level select bit, which is one of the component bits of the interrupt control register. When an interrupt request occurs, the interrupt priority level is compared with the IPL. The interrupt is enabled only when the priority level of the interrupt is higher than the IPL. Therefore, setting the interrupt priority level to "0" disables the interrupt.

Table 2.7.3 shows the settings of interrupt priority levels and Table 2.7.4 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 2.7.3 Settings of interrupt priority levels**

Interrupt priority level select bit	Interrupt priority level	Priority order
b2 b1 b0 0 0 0	Level 0 (interrupt disabled)	———
0 0 1	Level 1	<div>Low</div> <div>↓</div> <div>High</div>
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	

**Table 2.7.4 Interrupt levels enabled according to the contents of the IPL**

IPL	Enabled interrupt priority levels
IPL <sub>2</sub> IPL <sub>1</sub> IPL <sub>0</sub> 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

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (4) Rewrite the interrupt control register

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

### Example 1:

```
INT_SWITCH1:
  FCLR  I           ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  NOP                     ; 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 are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

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

Instructions : AND, OR, BCLR, BSET

## 2.7.6 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:

- (a) CPU gets the interrupt information (the interrupt number and interrupt request level) by reading address 00000<sub>16</sub>.
- (b) 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.
- (c) 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)
- (d) Saves the content of the temporary register (Note) within the CPU in the stack area.
- (e) Saves the content of the program counter (PC) in the stack area.
- (f) 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.

### (1) Interrupt Response Time

'Interrupt response time' is the period between the instant an interrupt occurs and the instant the first instruction within the interrupt routine has been executed. This time comprises the period from the occurrence of an interrupt to the completion of the instruction under execution at that moment (a) and the time required for executing the interrupt sequence (b). Figure 2.7.4 shows the interrupt response time.

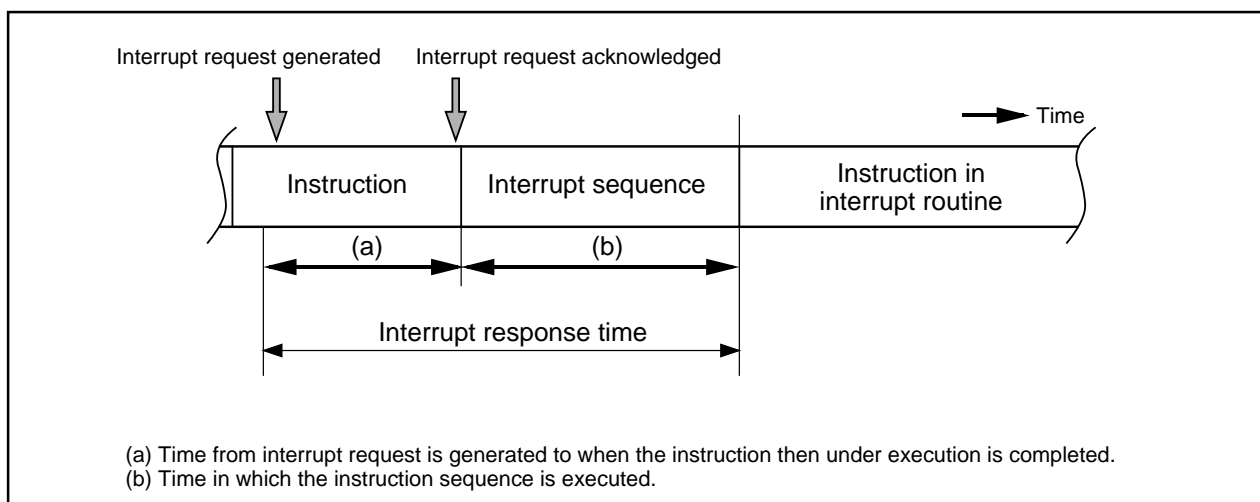


Figure 2.7.4 Interrupt response time

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

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

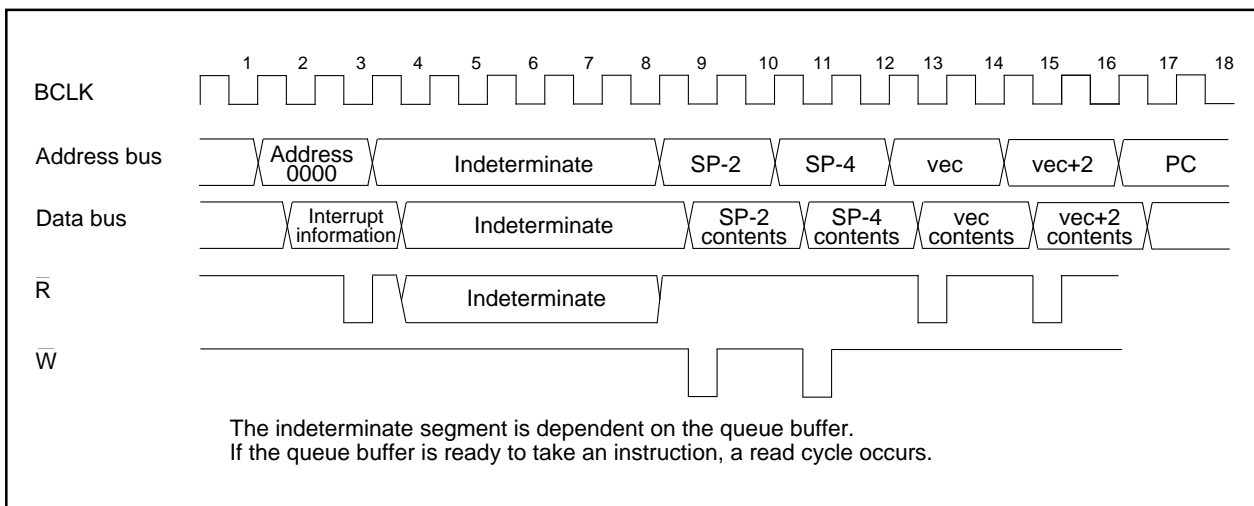
Time (b) is as shown in Table 2.7.5

**Table 2.7.5 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)

Notes 1: Add 2 cycles in the case of a DBC interrupt; add 1 cycle in the case either of an address coincidence interrupt or of a single-step interrupt.

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



**Figure 2.7.5 Time required for executing the interrupt sequence**

## (2) 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 2.7.6 is set in the IPL.

**Table 2.7.6 Relationship between interrupts without interrupt priority levels and IPL**

Interrupt sources without priority levels	Value set in the IPL
Watchdog timer, $\overline{\text{NMI}}$	7
Reset	0
Other	Not changed

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (3) 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 2.7.6 shows the state of the stack as it was before the acceptance of the interrupt request, and the state the stack after the acceptance of the interrupt request.

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

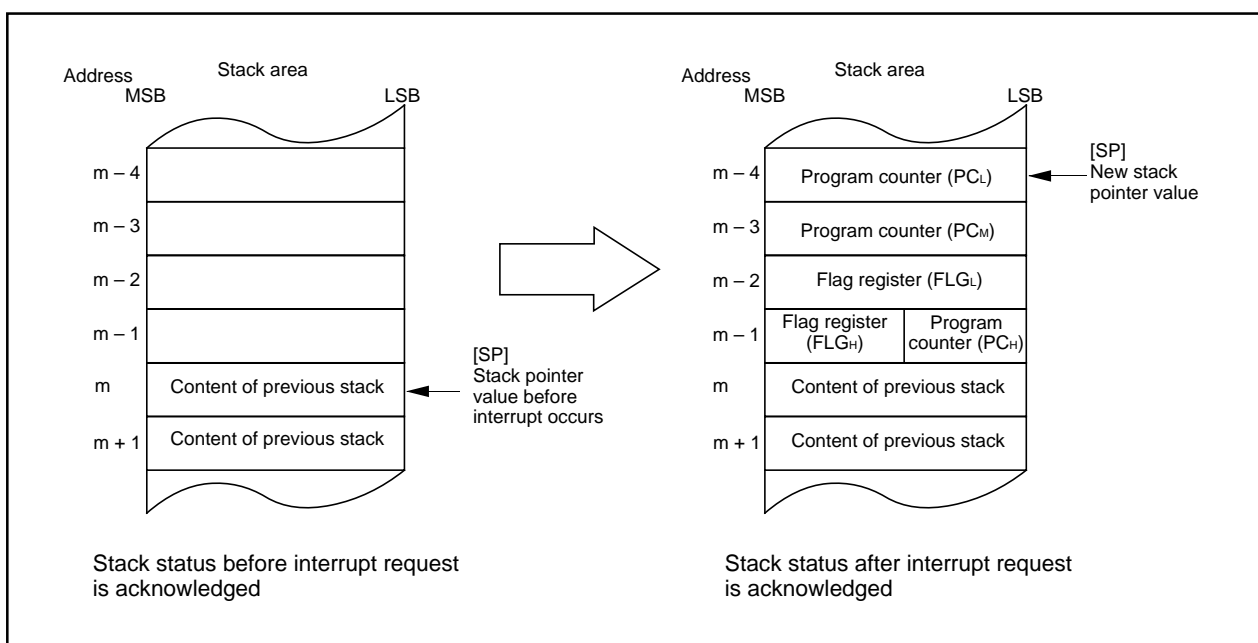


Figure 2.7.6 State of stack before and after acceptance of interrupt request

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

The operation of saving registers carried out in the interrupt sequence is dependent on whether the content of the stack pointer, at the time of acceptance of an interrupt request, is even or odd. If the content of the stack pointer (Note) is even, the content of the flag register (FLG) and the content of the program counter (PC) are saved, 16 bits at a time. If odd, their contents are saved in two steps, 8 bits at a time. Figure 2.7.7 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).

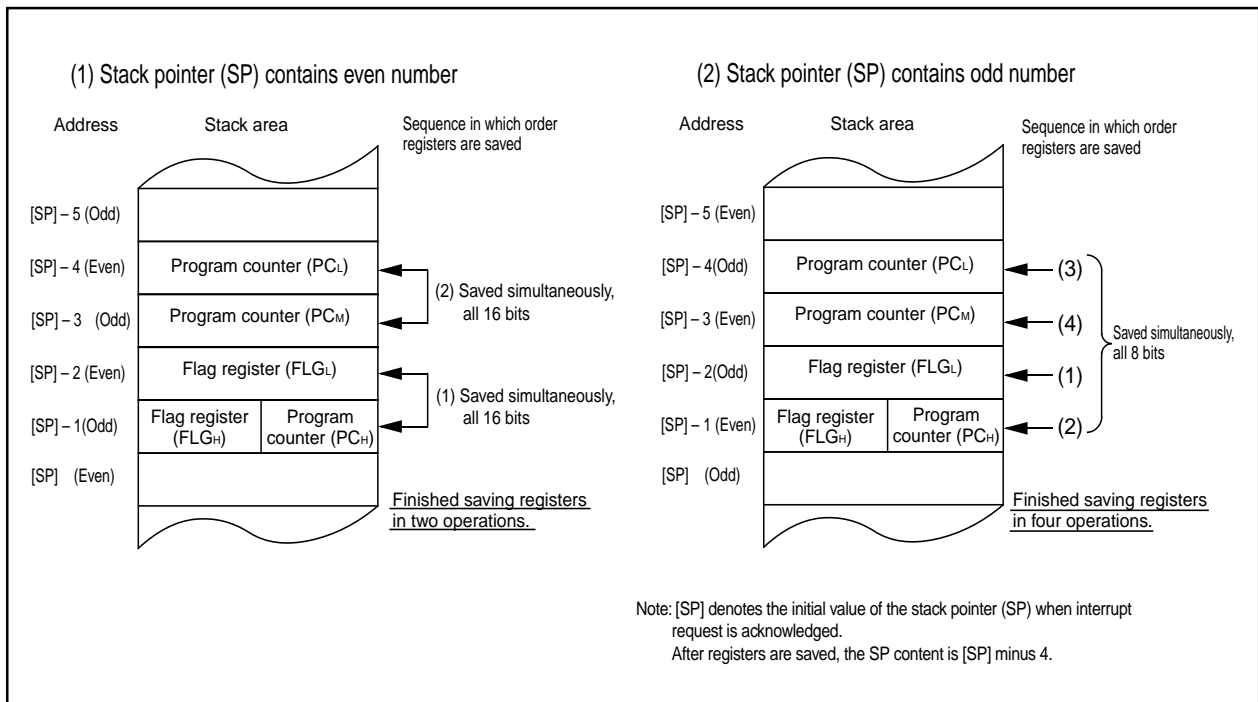


Figure 2.7.7 Operation of saving registers

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (4) Returning from an Interrupt Routine

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

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

## (5) Interrupt Priority

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

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

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

Figure 2.7.8 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 >  $\overline{\text{NMI}}$  >  $\overline{\text{DBC}}$  > Watchdog timer > Peripheral I/O > Single step > Address match

**Figure 2.7.8 Hardware interrupts priorities**

## (6) Interrupt resolution circuit

When two or more interrupts are generated simultaneously, this circuit selects the interrupt with the highest priority level. Figure 2.7.9 shows the circuit that judges the interrupt priority level.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

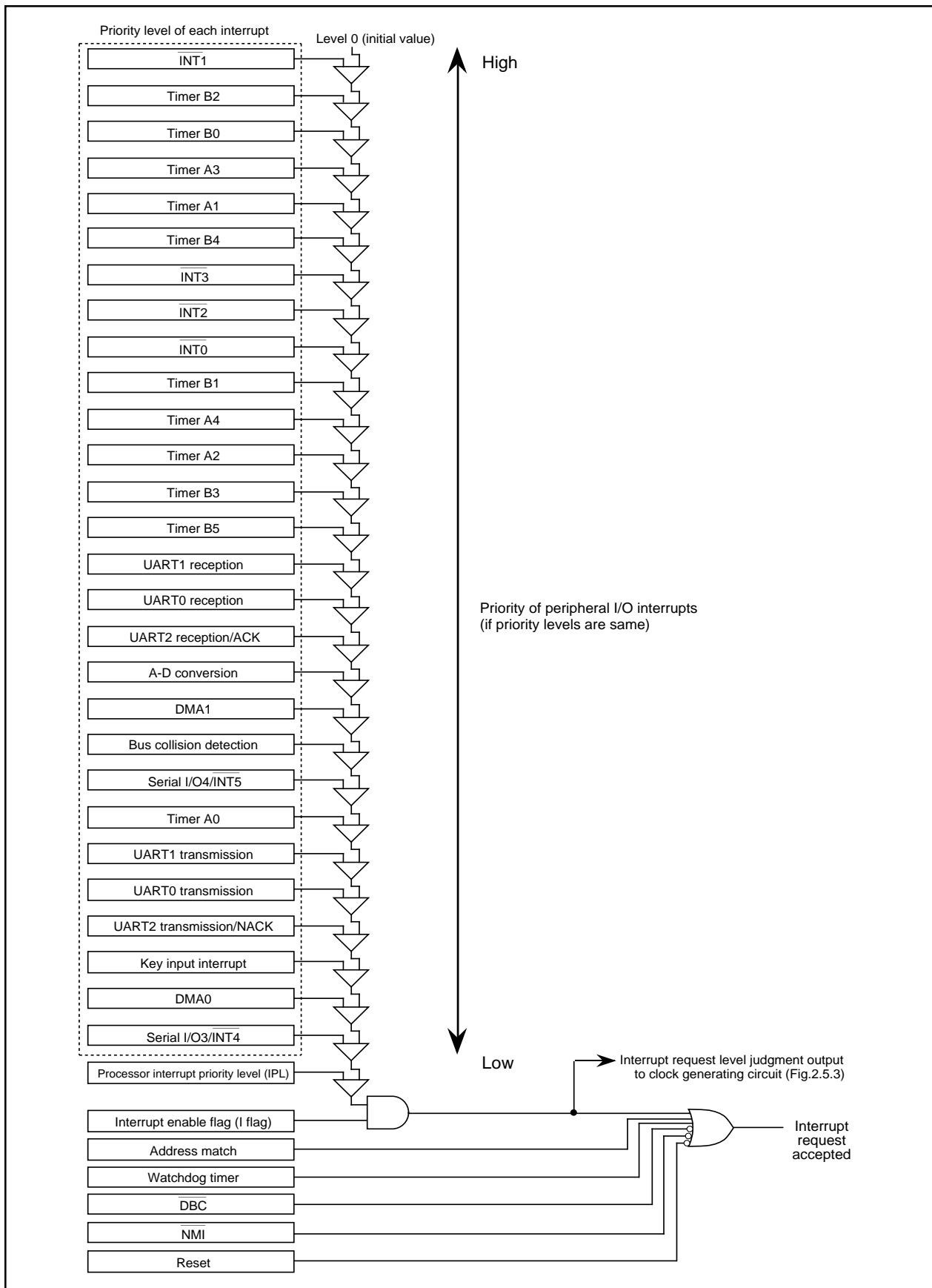


Figure 2.7.9 Maskable interrupts priorities (peripheral I/O interrupts)



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.7.7 $\overline{\text{INT}}$ Interrupt

$\overline{\text{INT0}}$  to  $\overline{\text{INT5}}$  are triggered by the edges of external inputs. The edge polarity is selected using the polarity select bit.

Of interrupt control registers, 0048<sub>16</sub> is used both as serial I/O4 and external interrupt  $\overline{\text{INT5}}$  input control register, and 0049<sub>16</sub> is used both as serial I/O3 and as external interrupt  $\overline{\text{INT4}}$  input control register. Use the interrupt request cause select bits - bits 6 and 7 of the interrupt request cause select register (035F<sub>16</sub>) - 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.

Either of the interrupt control registers - 0048<sub>16</sub>, 0049<sub>16</sub> - has the polarity-switching bit. Be sure to set this bit to "0" to select an serial I/O as the interrupt request cause.

As for external interrupt input, an interrupt can be generated both at the rising edge and at the falling edge by setting "1" in the INT<sub>i</sub> interrupt polarity switching bit of the interrupt request cause select register (035F<sub>16</sub>). To select both edges, set the polarity switching bit of the corresponding interrupt control register to 'falling edge' ("0").

Figure 2.7.10 shows the Interrupt request cause select register.

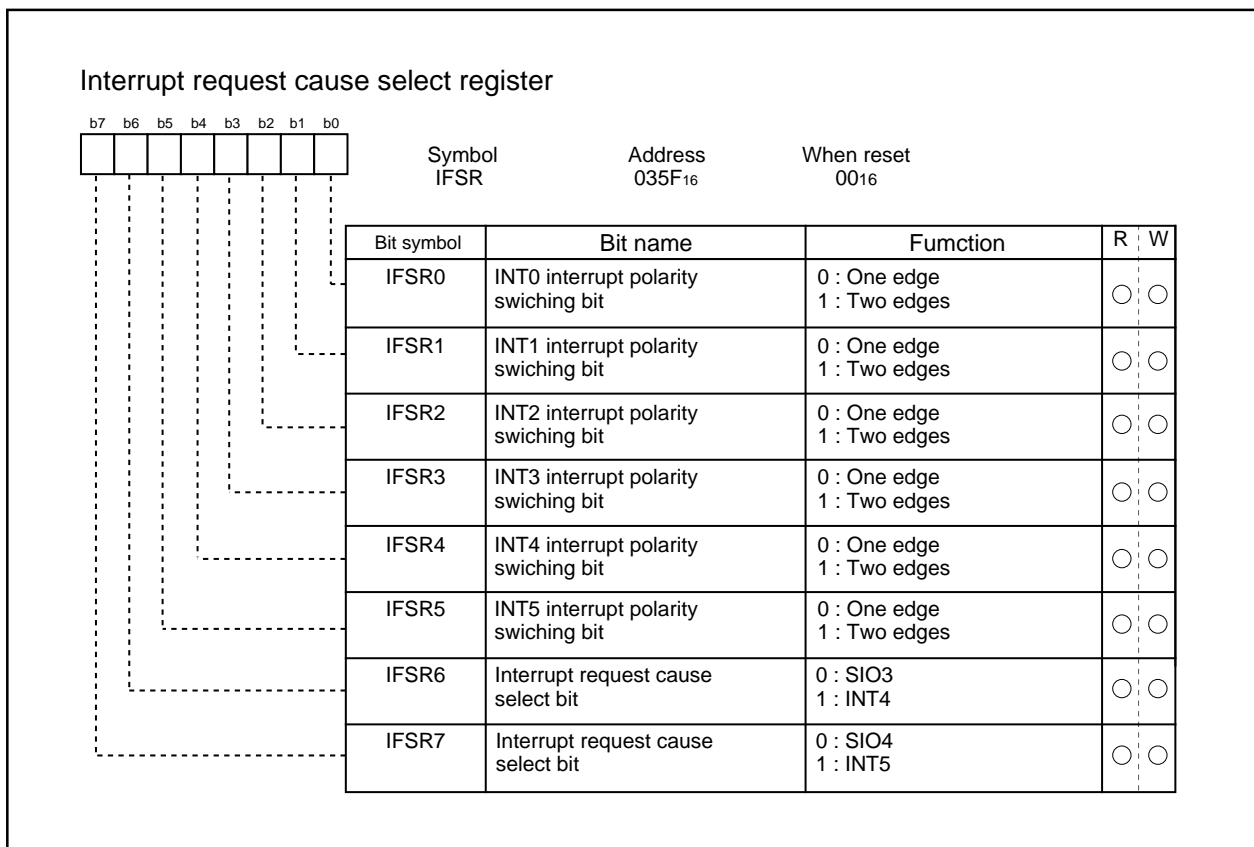


Figure 2.7.10 Interrupt request cause select register

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## 2.7.8 $\overline{\text{NMI}}$ Interrupt

An  $\overline{\text{NMI}}$  interrupt is generated when the input to the P85/ $\overline{\text{NMI}}$  pin changes from "H" to "L". The  $\overline{\text{NMI}}$  interrupt is a non-maskable external interrupt. The pin level can be checked in the port P85 register (bit 5 at address 03F0<sub>16</sub>).

This pin cannot be used as a normal port input.

## 2.7.9 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 2.7.11 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.

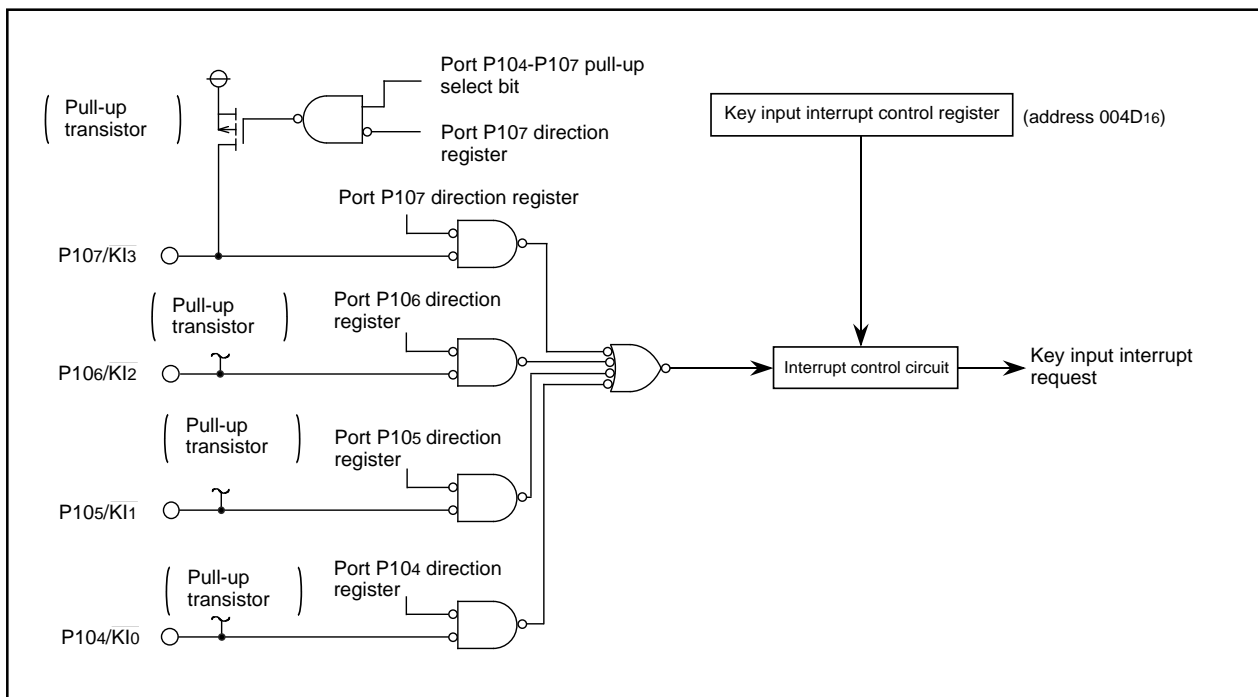


Figure 2.7.11 Block diagram of key input interrupt

M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

2.7.10 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. Note that when using the external data bus in width of 8 bits, the address match interrupt cannot be used for external area.

Figure 2.7.12 shows the address match interrupt-related registers.

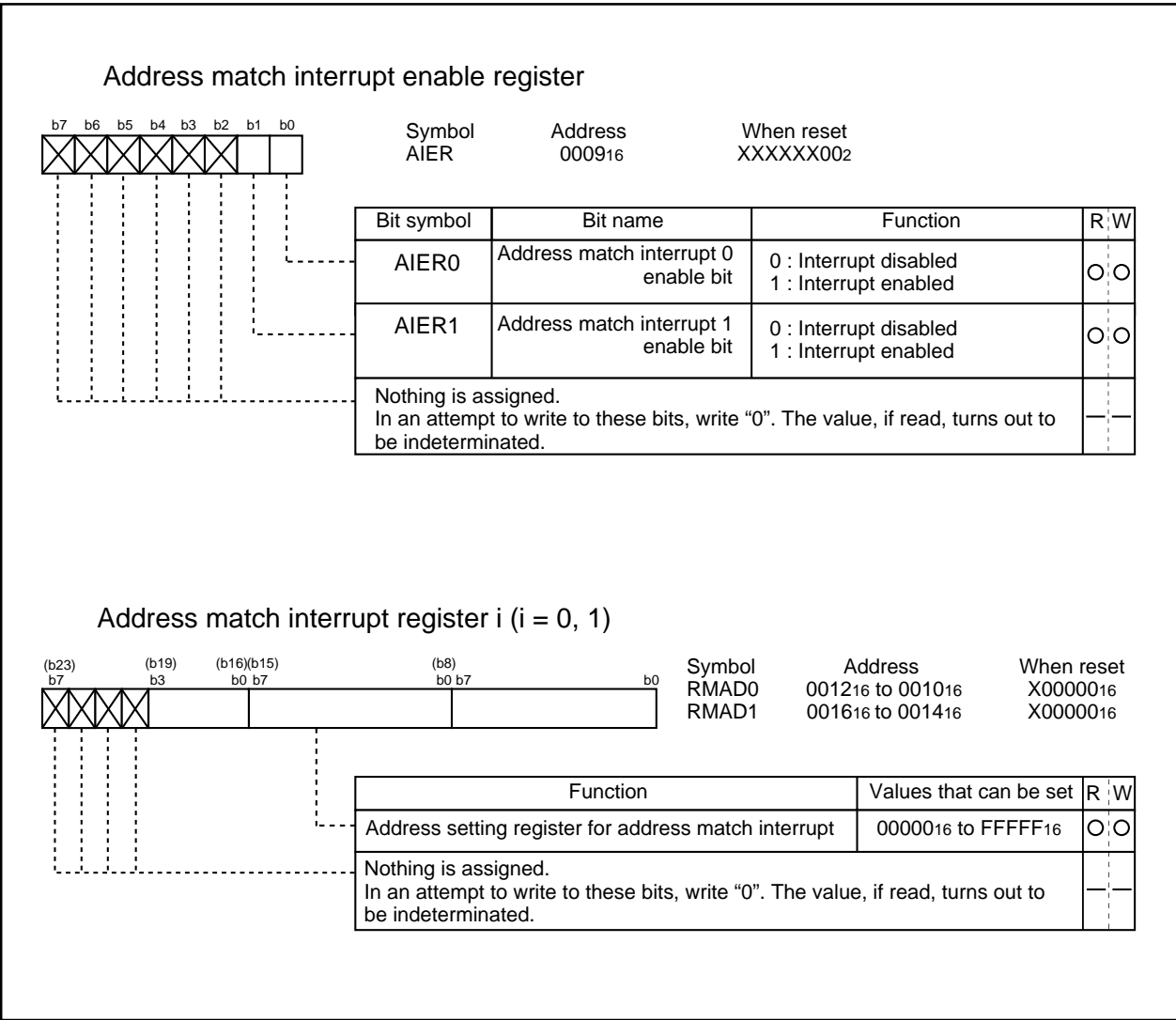


Figure 2.7.12 Address match interrupt-related registers

## 2.7.11 Precautions for Interrupts

### (1) Reading address 00000<sub>16</sub>

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

### (2) Setting the stack pointer

- The value of the stack pointer immediately after reset is initialized to 0000<sub>16</sub>. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt. 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 interrupts including the  $\overline{\text{NMI}}$  interrupt is prohibited.

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

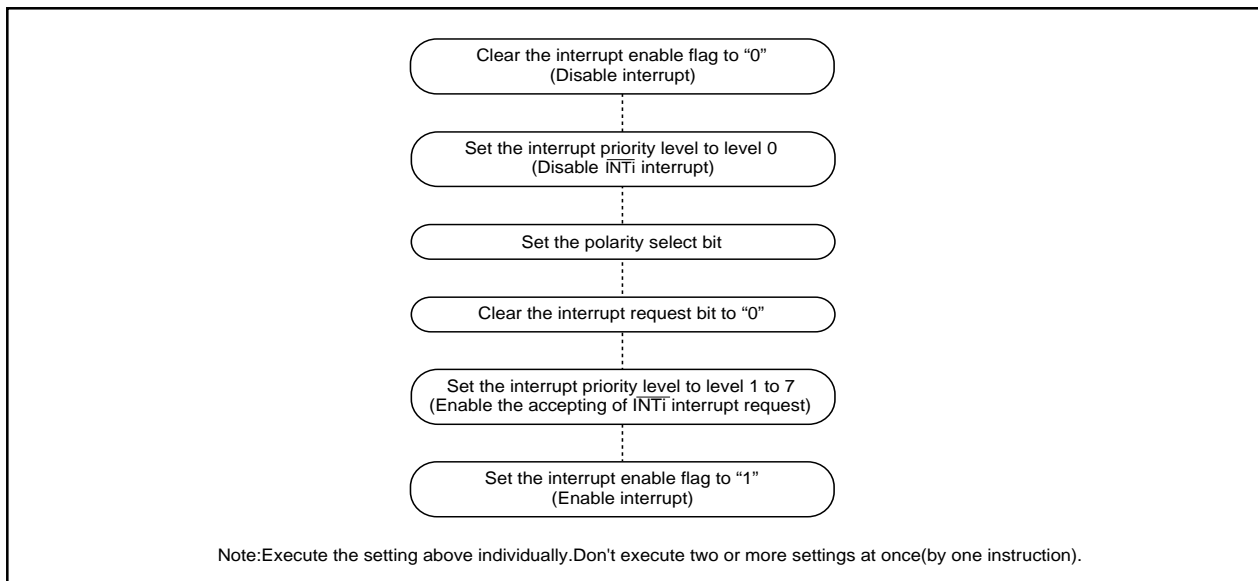
- As for the  $\overline{\text{NMI}}$  interrupt pin, an interrupt cannot be disabled. Connect it to the Vcc pin via a resistor (pull-up) if unused. Be sure to work on it.
- The  $\overline{\text{NMI}}$  pin also serves as P85, which is exclusively input. Reading the contents of the P8 register allows 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  $\overline{\text{NMI}}$  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 2.7.13 shows the procedure for changing the  $\overline{\text{INT}}$  interrupt generate factor.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER



**Figure 2.7.13 Switching condition of INT interrupt request**

## (5) Rewrite the interrupt control register

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

### Example 1:

```

INT_SWITCH1:
    FCLR    I            ; Disable interrupts.
    AND.B   #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
    NOP                     ; 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 are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

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

Instructions : AND, OR, BCLR, BSET

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.8 Watchdog Timer

The watchdog timer has the function of detecting when the program is out of control. The watchdog timer is a 15-bit counter which down-counts the clock derived by dividing the BCLK using the prescaler. A watchdog timer interrupt is generated when an underflow occurs in the watchdog timer. When XIN is selected for the 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 pre-scaler.

With XIN chosen for BCLK

$$\text{Watchdog timer period} = \frac{\text{pre-scaler dividing ratio (16 or 128)} \times \text{watchdog timer count (32768)}}{\text{BCLK}}$$

With XCIN chosen for BCLK

$$\text{Watchdog timer period} = \frac{\text{pre-scaler dividing ratio (2)} \times \text{watchdog timer count (32768)}}{\text{BCLK}}$$

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

The watchdog timer is initialized by writing to the watchdog timer start register (address 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).

Figure 2.8.1 shows the block diagram of the watchdog timer. Figure 2.8.2 shows the watchdog timer-related registers.

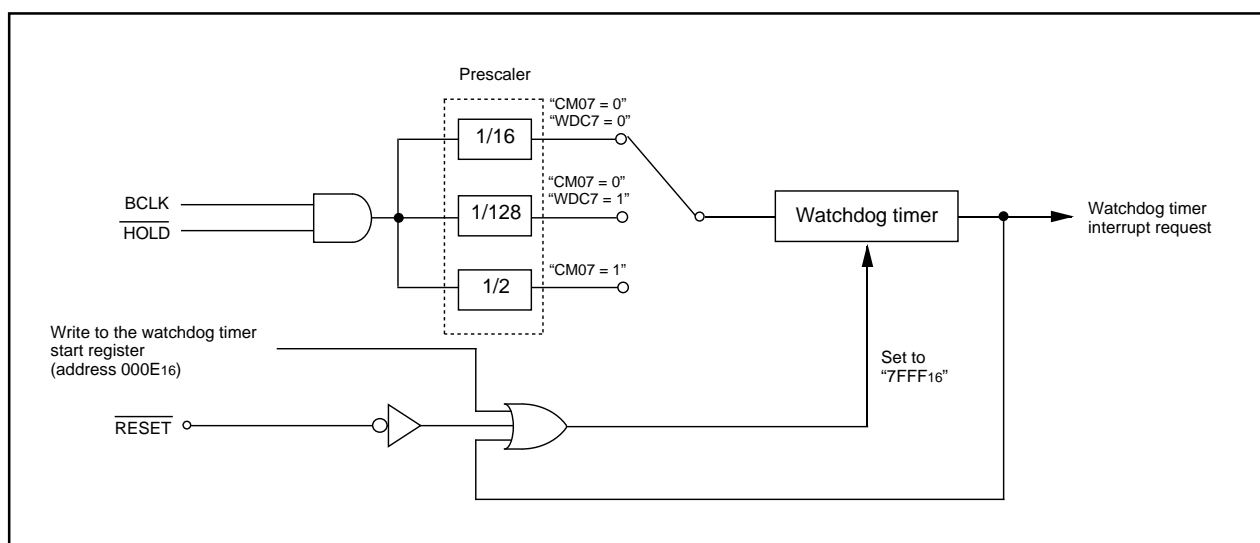


Figure 2.8.1 Block diagram of watchdog timer

M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

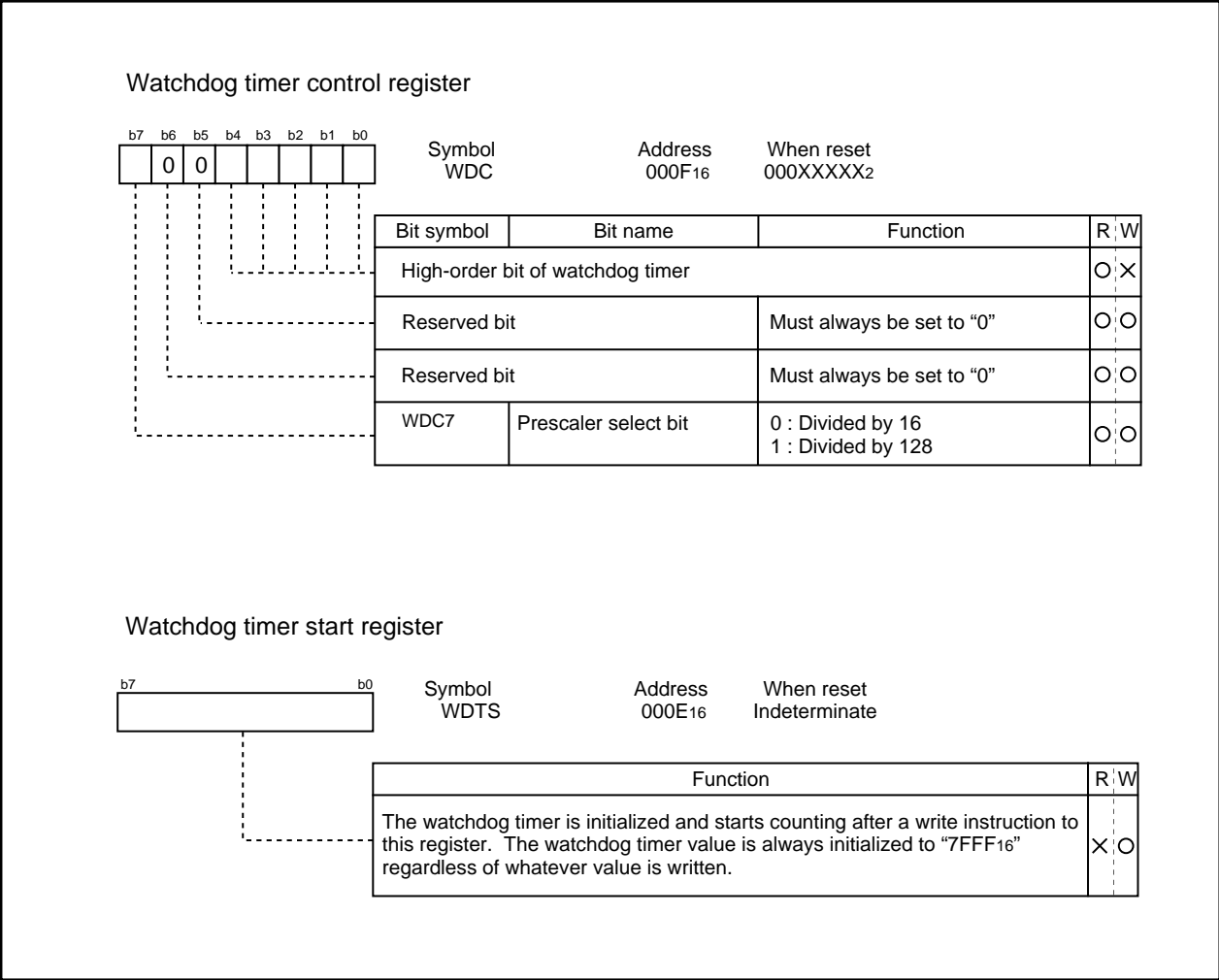


Figure 2.8.2 Watchdog timer control and start registers

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.9 DMAC

This microcomputer has two DMAC (direct memory access controller) channels that allow data to be sent to memory without using the CPU. DMAC shares the same data bus with the CPU. The DMAC is given a higher right of using the bus than the CPU, which leads to working the cycle stealing method. On this account, the operation from the occurrence of DMA transfer request signal to the completion of 1-word (16-bit) or 1-byte (8-bit) data transfer can be performed at high speed. Figure 2.9.1 shows the block diagram of the DMAC. Table 2.9.1 shows the DMAC specifications. Figures 2.9.2 to 2.9.4 show the registers used by the DMAC.

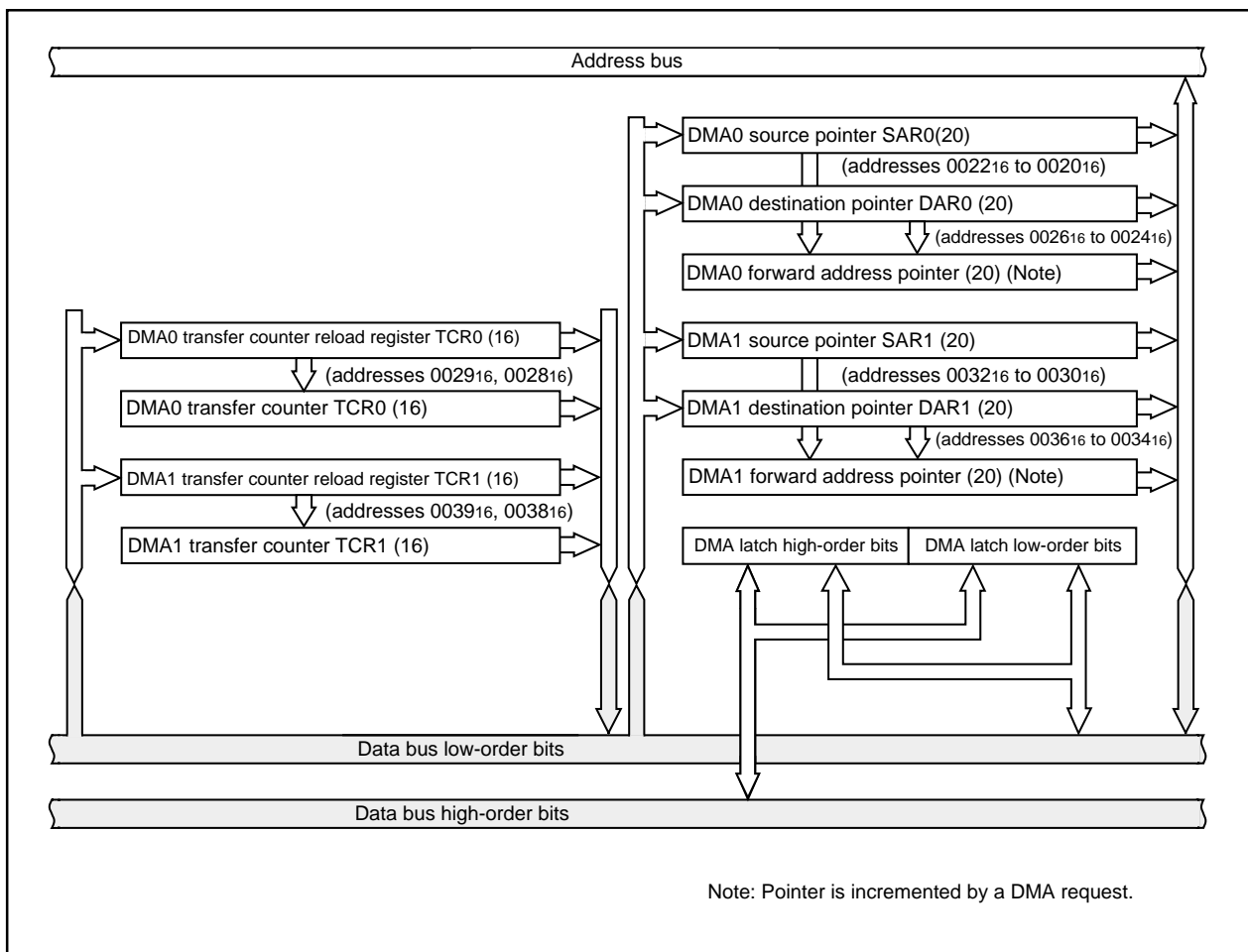


Figure 2.9.1 Block diagram of DMAC

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.



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

**Table 2.9.1 DMAC specifications**

Item	Specification
No. of channels	2 (cycle steal method)
Transfer memory space	<ul style="list-style-type: none"> <li>• From any address in the 1M bytes space to a fixed address</li> <li>• From a fixed address to any address in the 1M bytes space</li> <li>• From a fixed address to a fixed address</li> </ul> (Note that DMA-related registers [0020 <sub>16</sub> to 003F <sub>16</sub> ] 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 INT <sub>0</sub> or INT <sub>1</sub> (INT <sub>0</sub> can be selected by DMA0, INT <sub>1</sub> by DMA1) or both edge Timer A0 to timer A4 interrupt requests Timer B0 to timer B5 interrupt requests UART0 transfer and reception interrupt requests UART1 transfer and reception interrupt requests UART2 transfer and reception interrupt requests Serial I/O3, 4 interrupt requests A-D conversion interrupt requests Software triggers
Channel priority	DMA0 takes precedence if DMA0 and DMA1 requests are generated simultaneously
Transfer unit	8 bits or 16 bits
Transfer address direction	forward/fixed (forward direction cannot be specified for both source and destination simultaneously)
Transfer mode	<ul style="list-style-type: none"> <li>• Single transfer mode After the transfer counter underflows, the DMA enable bit turns to "0", and the DMAC turns inactive</li> <li>• 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.</li> </ul>
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	<ul style="list-style-type: none"> <li>• When the DMA enable bit is set to "0", the DMAC is inactive.</li> <li>• After the transfer counter underflows in single transfer mode</li> </ul>
Forward address pointer and reload timing for 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.

M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

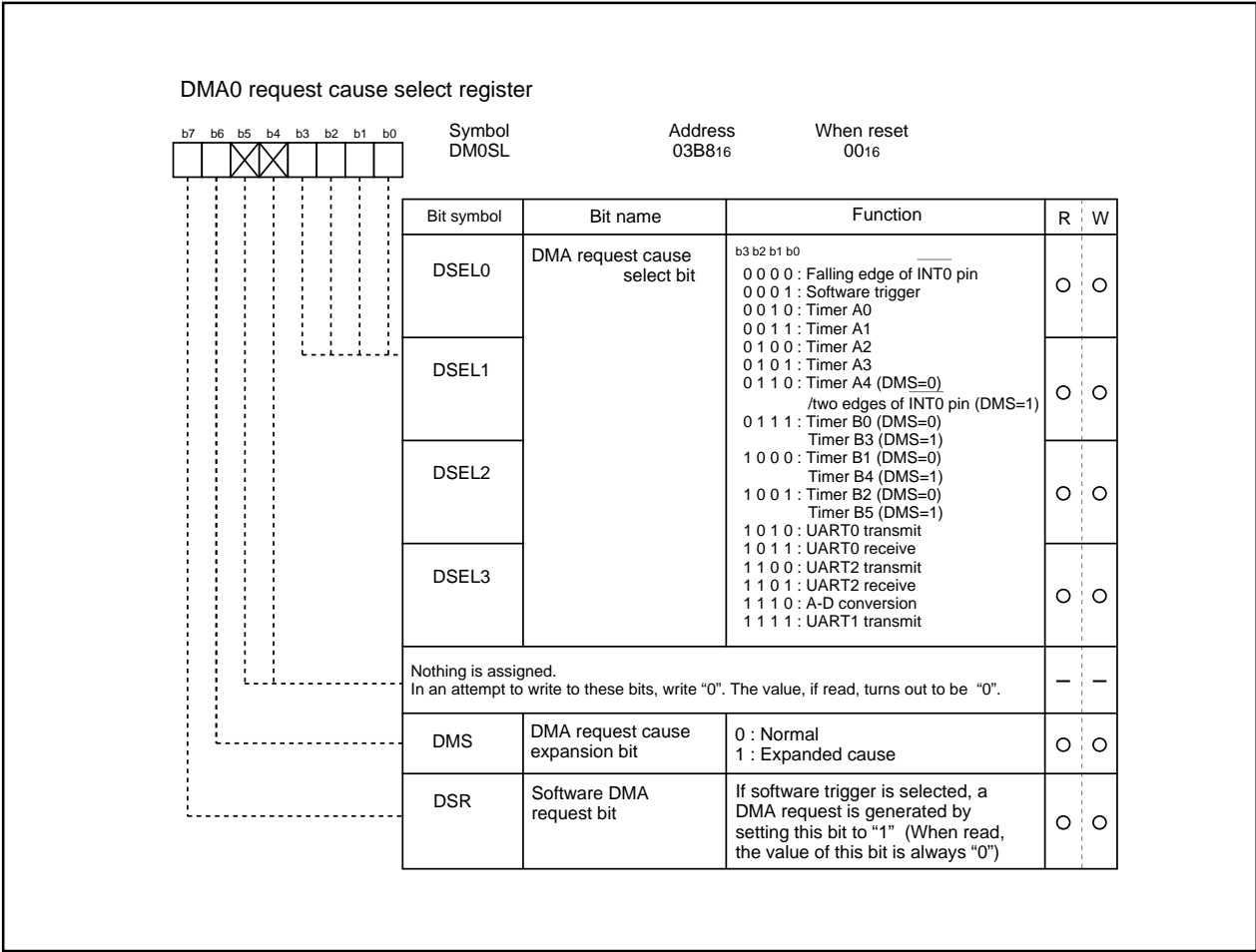


Figure 2.9.2 DMAC register (1)

# M306H2FCFP

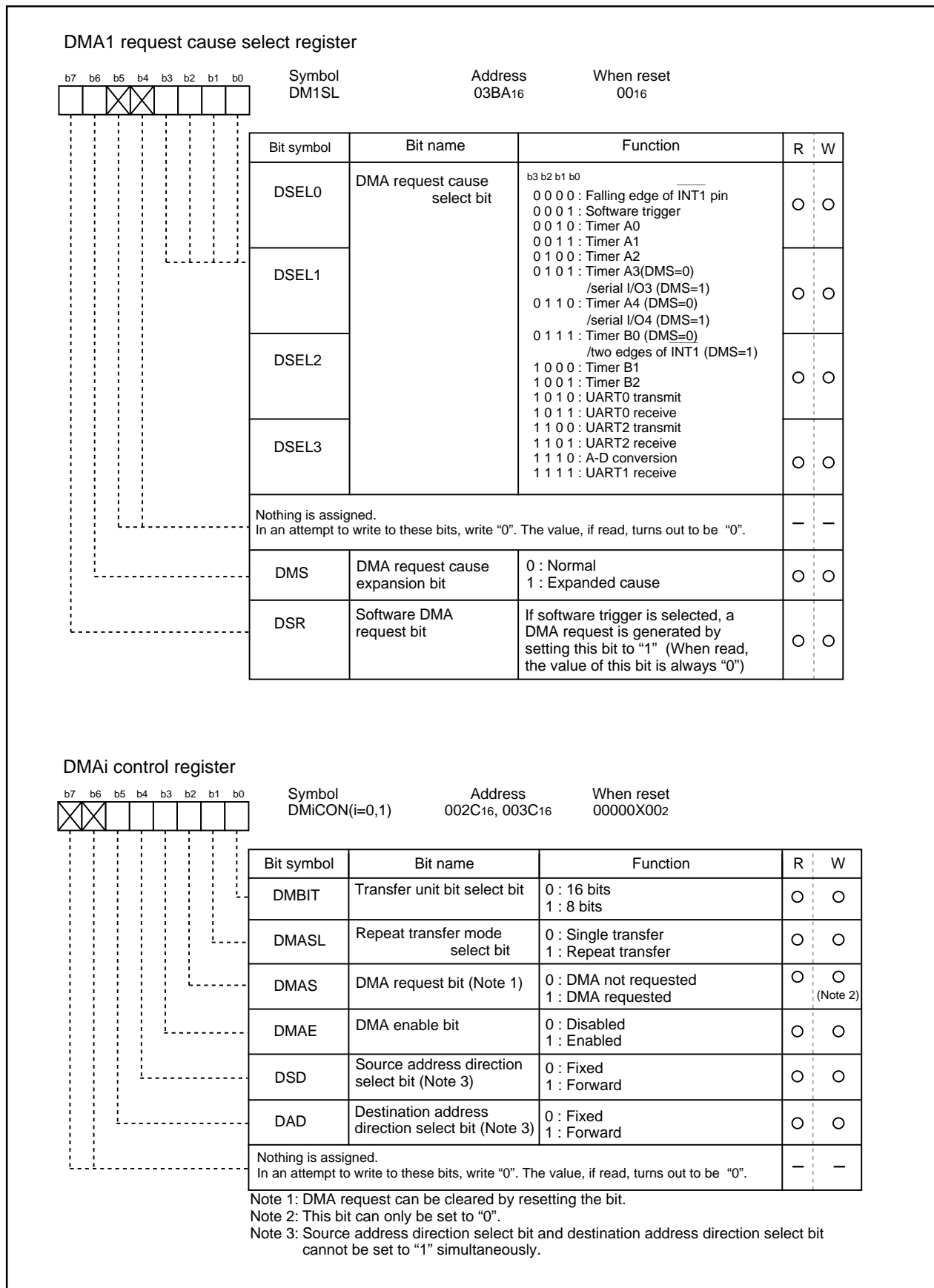
 SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER


Figure 2.9.3 DMAC register (2)

M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

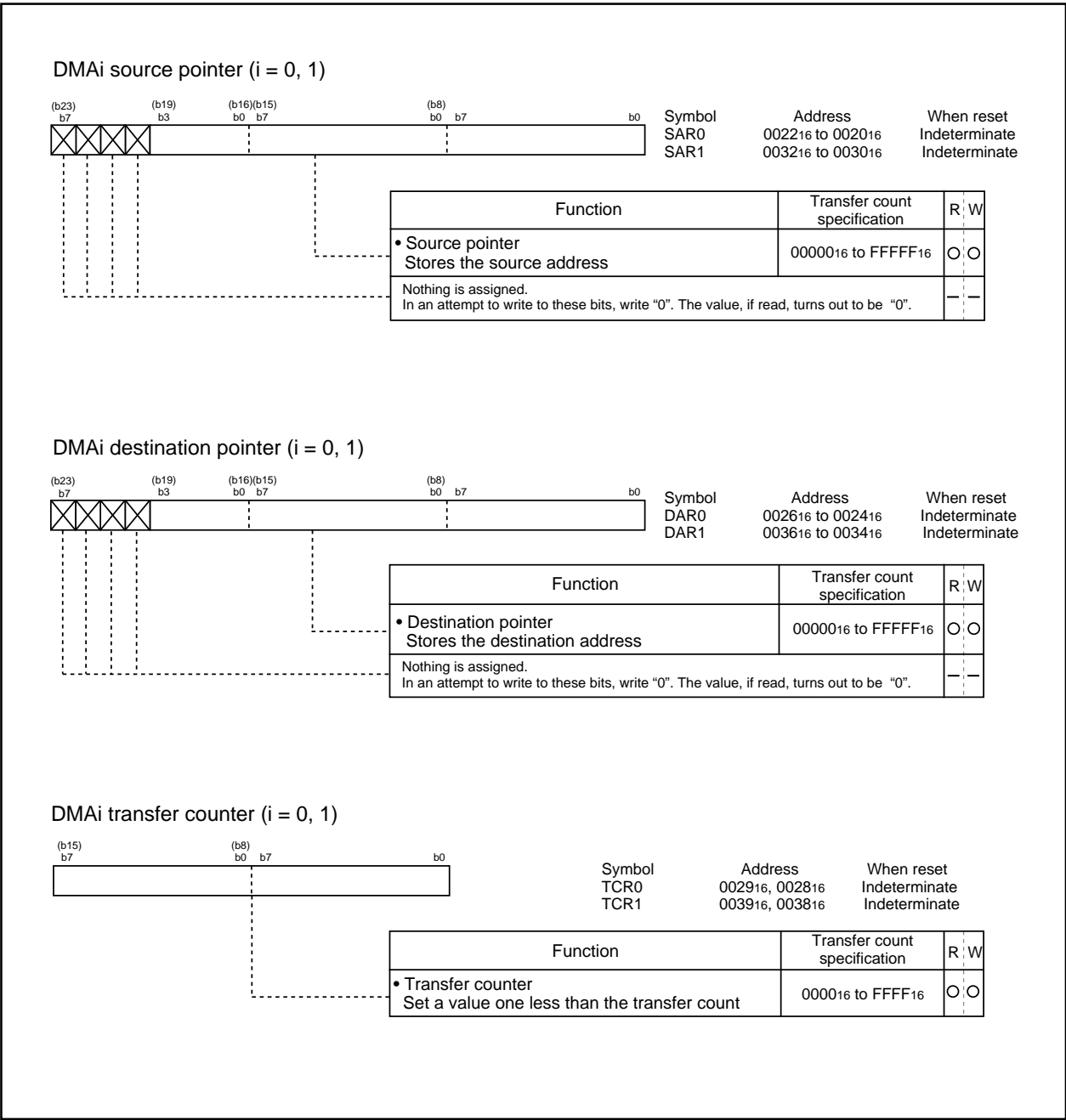


Figure 2.9.4 DMAC register (3)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (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 and, the level of the BYTE pin. In memory expansion 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

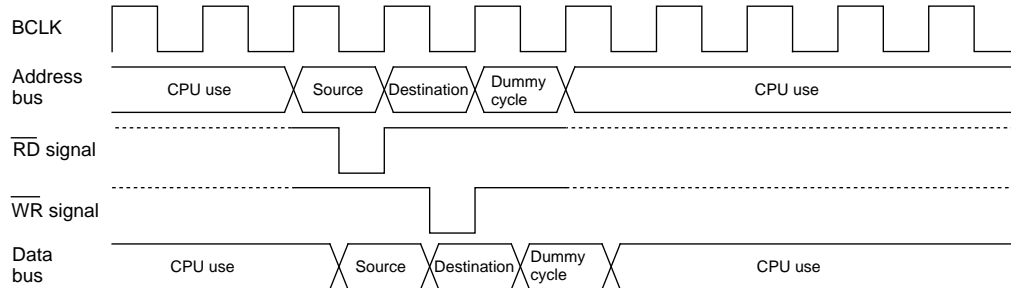
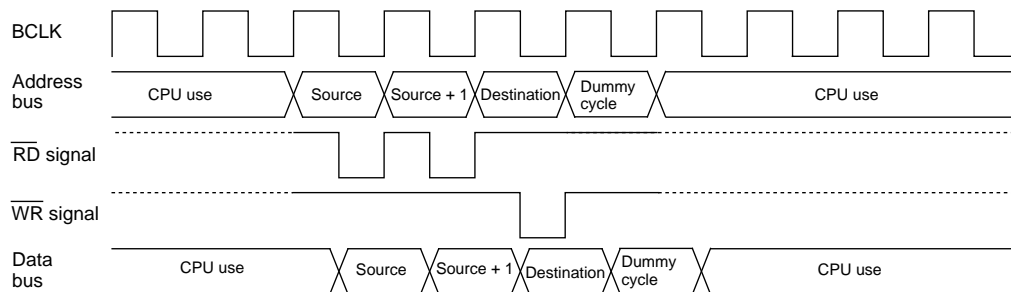
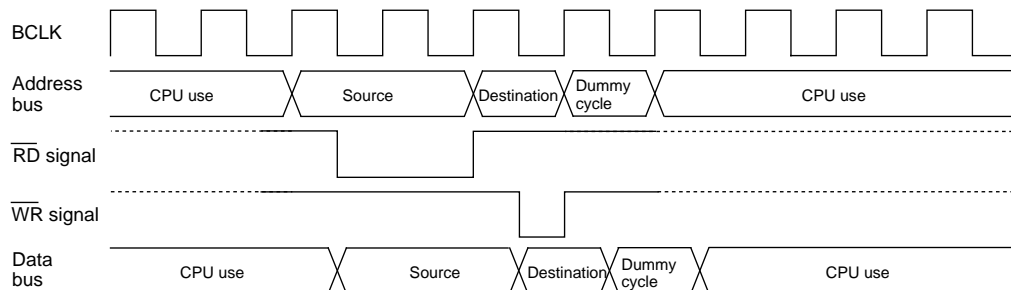
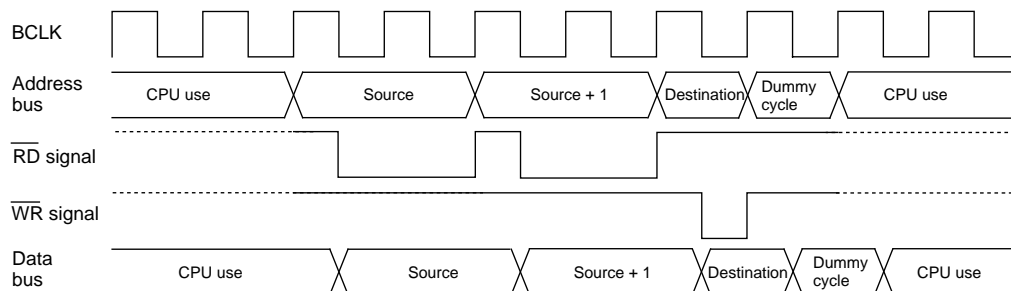
When transferring 16-bit data over an 8-bit data bus (BYTE pin = "H") in memory expansion mode, the 16 bits of data are sent in two 8-bit blocks. Therefore, two bus cycles are required for reading the data and two are required for writing the data. Also, in contrast to when the CPU accesses internal memory, when the DMAC accesses internal memory (internal 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 2.9.5 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 2.9.5, 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.

## M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER**(1) 8-bit transfers****16-bit transfers from even address 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 write cycles).**

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

**Figure 2.9.5 Example of the transfer cycles for a source read**

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (2) DMAC transfer cycles

Any combination of even or odd transfer read and write addresses is possible. Table 2.9.2 shows the number of DMAC transfer cycles.

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

$$\text{No. of transfer cycles per transfer unit} = \text{No. of read cycles} \times j + \text{No. of write cycles} \times k$$

**Table 2.9.2 No. of DMAC transfer cycles**

Transfer unit	Bus width	Access address	Single-chip mode		Memory expansion 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		
Internal ROM/RAM No wait	Internal ROM/RAM With wait	SFR area	Separate bus No wait	Separate bus With wait	Multiplex bus
1	2	2	1	2	3

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## 2.9.1 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 one of the source pointer and 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 given above, so the DMAC operates again from the initial state at the instant "1" is overwritten to the DMA enable bit.

## 2.9.2 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.

For the selection of DMA request factors, see the descriptions of the DMAi factor selection register. The DMA request bit turns to "1" if the DMA transfer request signal occurs regardless of the DMAC's state (regardless of whether the DMA enable bit is set "1" or to "0"). It turns to "0" immediately before data transfer starts.

In addition, it 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 factor selection bit causes the DMA request bit to turn to "1". So be sure to set the DMA request bit to "0" after the DMA request factor selection bit is 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 "1" due to an internal factor is timed to be effected immediately before the transfer starts.

### (2) External factors

An external factor is a factor caused to occur by the leading edge of input from the INTi pin (i depends on which DMAC channel is used).

Selecting the INTi pins as external factors using the DMA request factor selection bit causes 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 factor selection bit (synchronizes with the trailing edge of the input signal to each 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.



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

### (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.

An example in which DMA transfer is carried out in minimum cycles at the time when DMA transfer request signals due to external factors concurrently occur.

Figure 2.9.6 An example of DMA transfer effected by external factors.

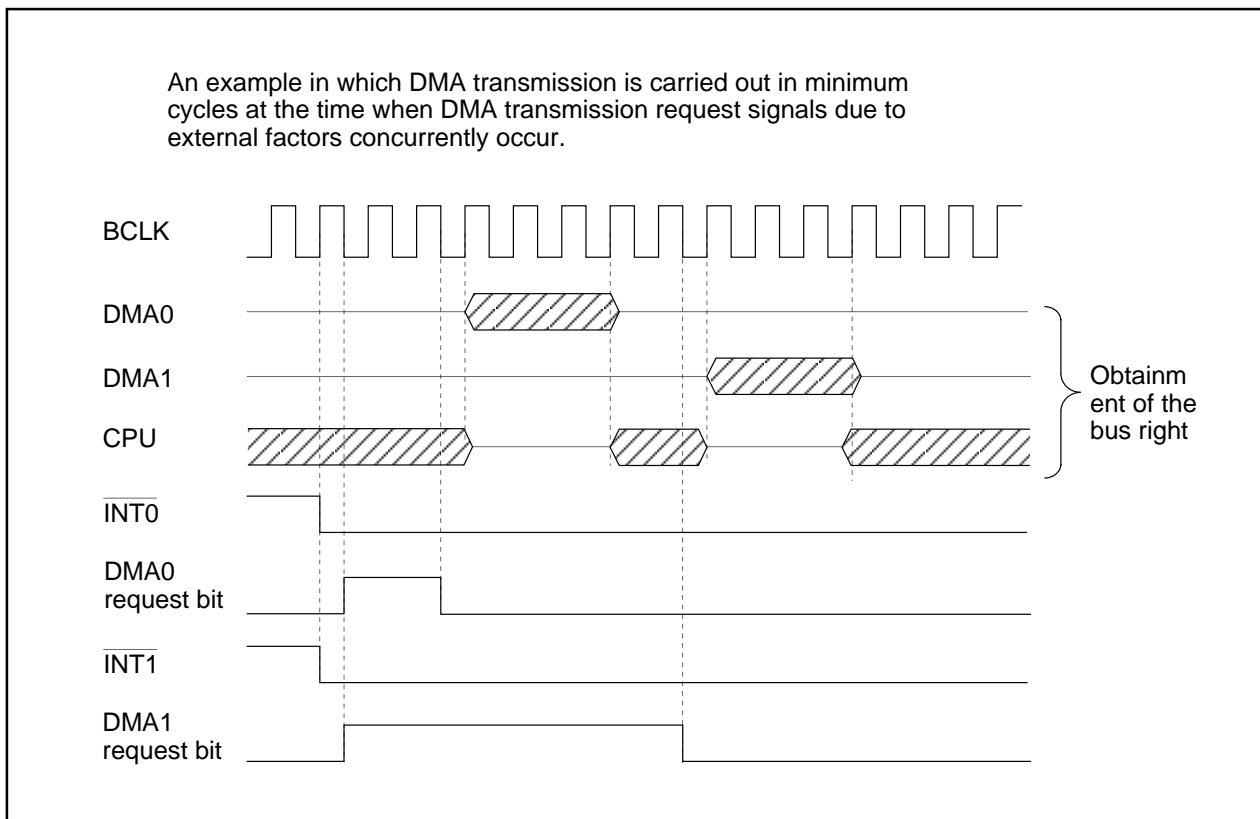


Figure 2.9.6 An example of DMA transfer effected by external factors

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.10 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 2.10.1 and 2.10.2 show the block diagram of timers.

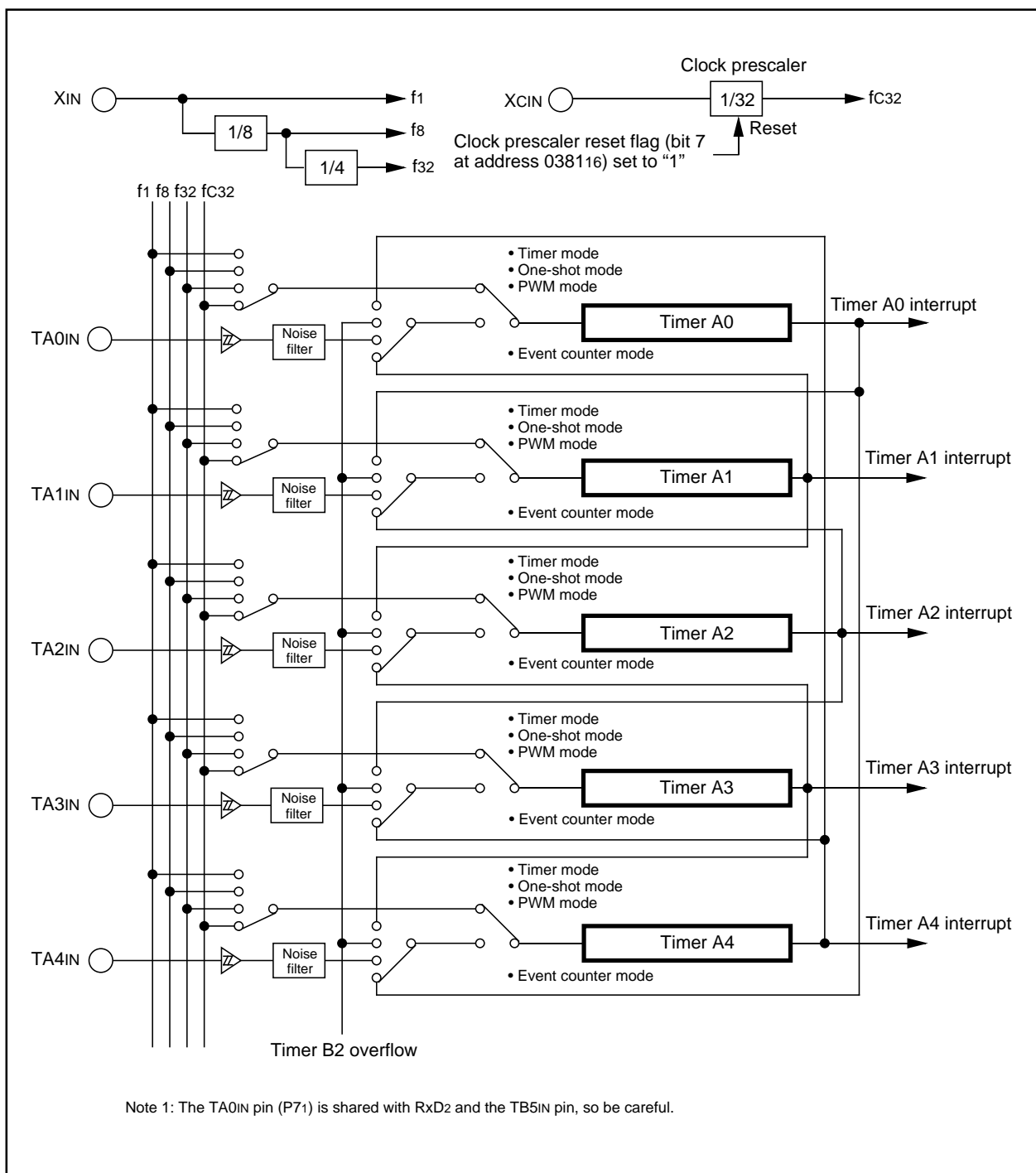


Figure 2.10.1 Timer A block diagram

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

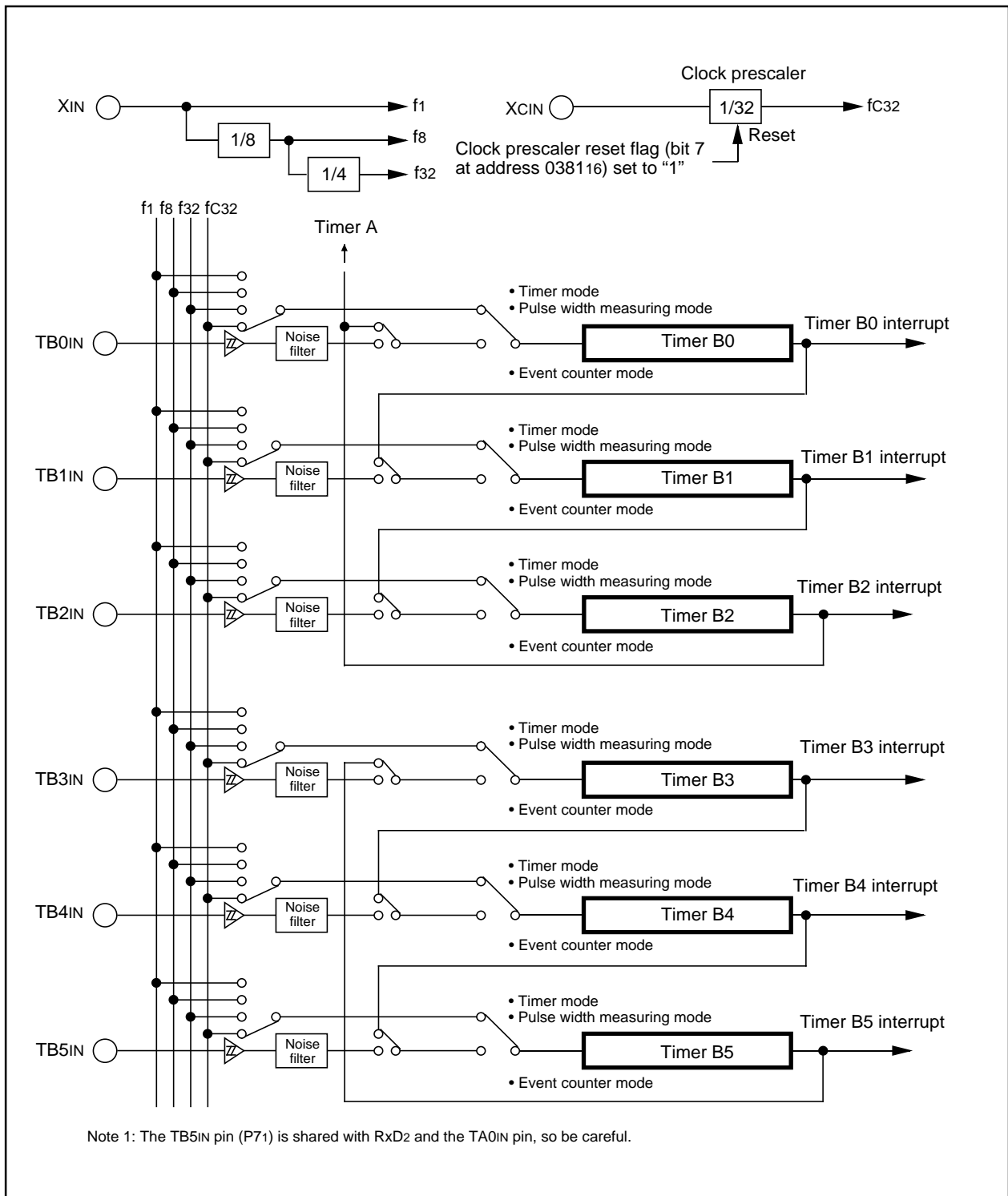


Figure 2.10.2 Timer B block diagram

M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

2.10.1 Timer A

Figure 2.10.3 shows the block diagram of timer A. Figures 2.10.4 to 2.10.6 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<sub>16</sub>”.
- Pulse width modulation (PWM) mode: The timer outputs pulses of a given width.

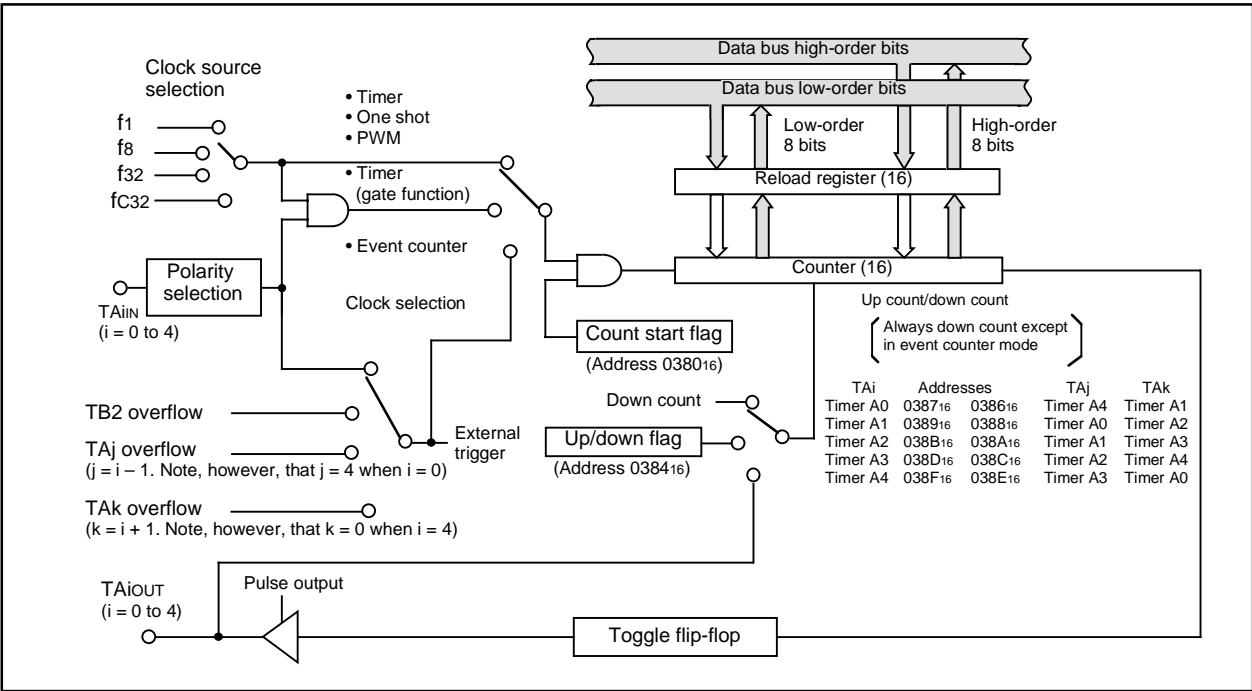


Figure 2.10.3 Block diagram of timer A

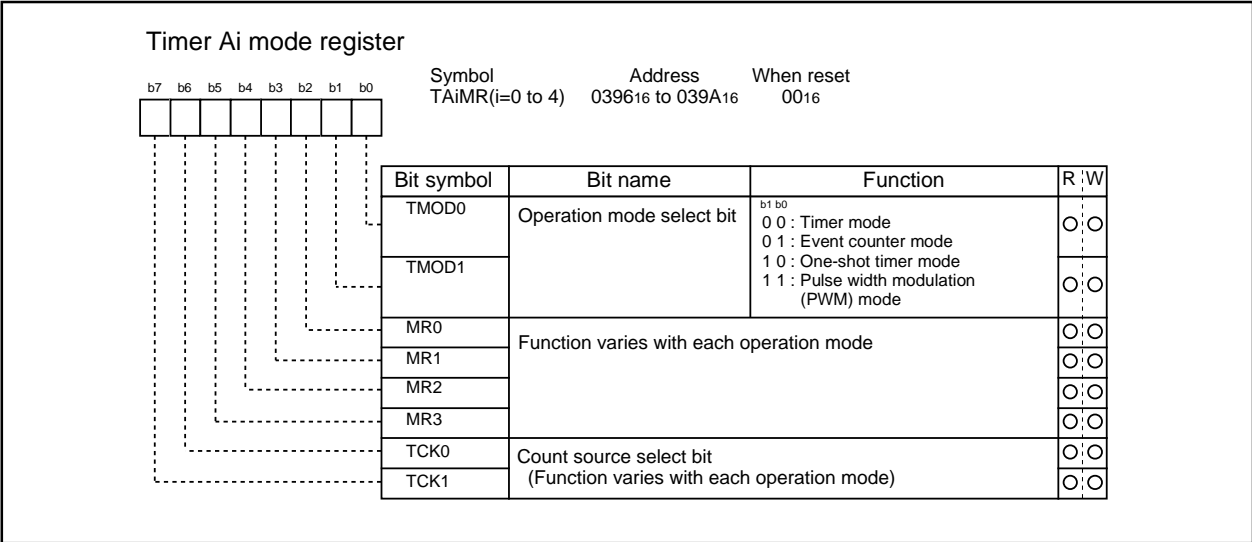
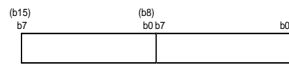


Figure 2.10.4 Timer A-related registers (1)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

Timer Ai register (Note 1)



Symbol	Address	When reset
TA0	0387 <sub>16</sub> , 0386 <sub>16</sub>	Indeterminate
TA1	0389 <sub>16</sub> , 0388 <sub>16</sub>	Indeterminate
TA2	038B <sub>16</sub> , 038A <sub>16</sub>	Indeterminate
TA3	038D <sub>16</sub> , 038C <sub>16</sub>	Indeterminate
TA4	038F <sub>16</sub> , 038E <sub>16</sub>	Indeterminate

Function	Values that can be set	R/W
• Timer mode Counts an internal count source	0000 <sub>16</sub> to FFFF <sub>16</sub>	O/O
• Event counter mode Counts pulses from an external source or timer overflow	0000 <sub>16</sub> to FFFF <sub>16</sub>	O/O
• One-shot timer mode Counts a one shot width	0000 <sub>16</sub> to FFFF <sub>16</sub> (Note 2,4)	X/O
• Pulse width modulation mode (16-bit PWM) Functions as a 16-bit pulse width modulator	0000 <sub>16</sub> to FFFE <sub>16</sub> (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 <sub>16</sub> ~ FE <sub>16</sub> (high-order address) 00 <sub>16</sub> ~ FF <sub>16</sub> (low-order address) (Note 3,4)	X/O

Note 1: Read and write data in 16-bit units.

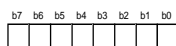
Note 2: When the timer Ai register is set to "0000<sub>16</sub>", the counter does not operate and the timer Ai interrupt request is not generated. When the pulse is set to output, the pulse does not output from the TAIOUT pin.

Note 3: When the timer Ai register is set to "0000<sub>16</sub>", 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<sub>16</sub>".

Note 4: Use MOV instruction to write to this register.

Note 5: In the case of using "Event counter mode" as "Free-Run type", the timer register contents may be unknown when counting begins. (Refer 3. Usage Precaution.)

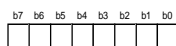
Count start flag



Symbol	Address	When reset
TABSR	0380 <sub>16</sub>	00 <sub>16</sub>

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 <sub>16</sub>	00 <sub>16</sub>

Bit symbol	Bit name	Function	R/W
TA0UD	Timer A0 up/down flag	0 : Down count 1 : Up count	O/O
TA1UD	Timer A1 up/down flag		O/O
TA2UD	Timer A2 up/down flag	This specification becomes valid when the up/down flag content is selected for up/down switching cause	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		X/O
TA3P	Timer A3 two-phase pulse signal processing select bit	0 : two-phase pulse signal processing disabled 1 : two-phase pulse signal processing enabled	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 2.10.5 Timer A-related registers (2)

# M306H2FCFP

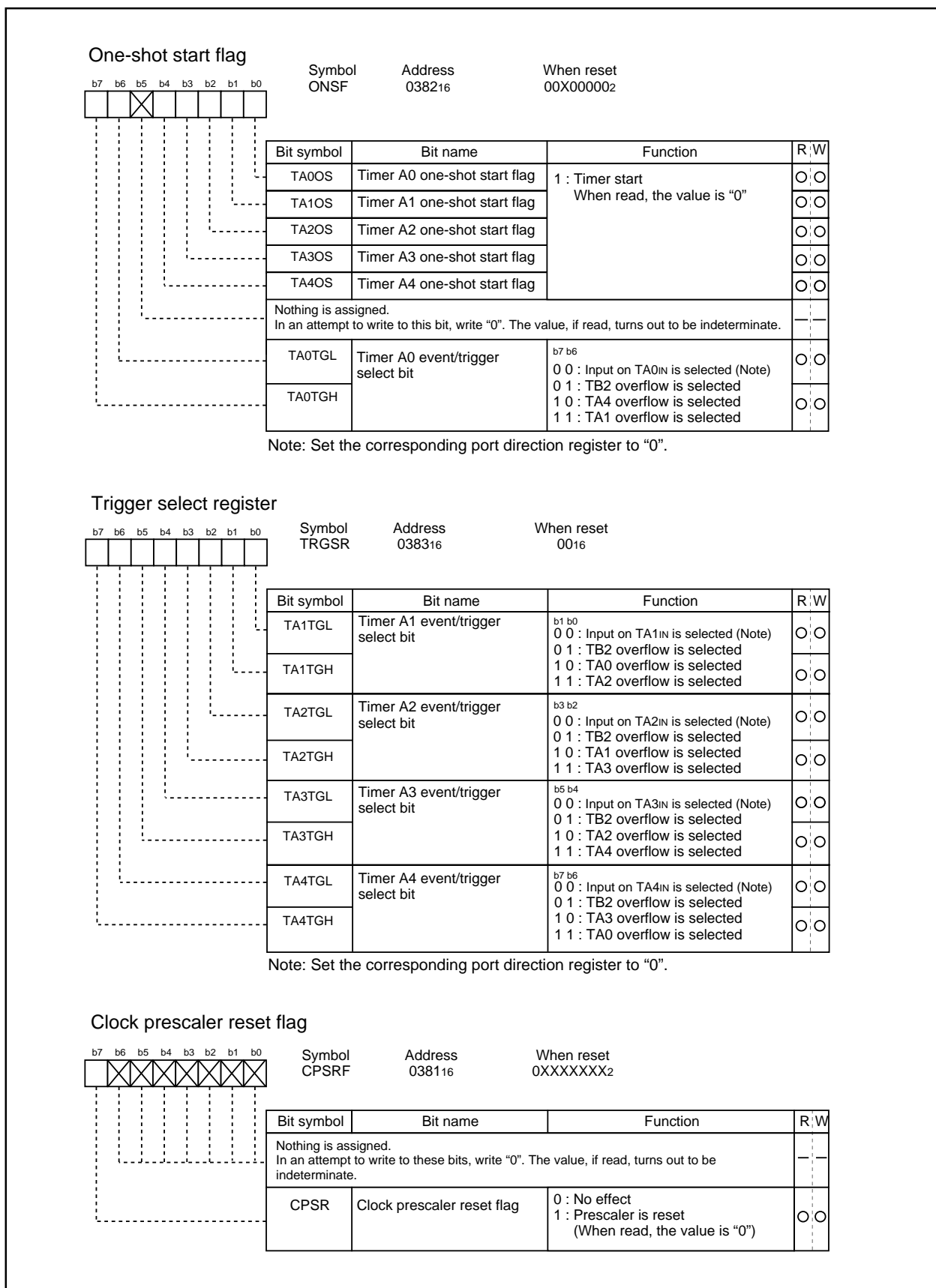
 SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER


Figure 2.10.6 Timer A-related registers (3)

# M306H2FCFP

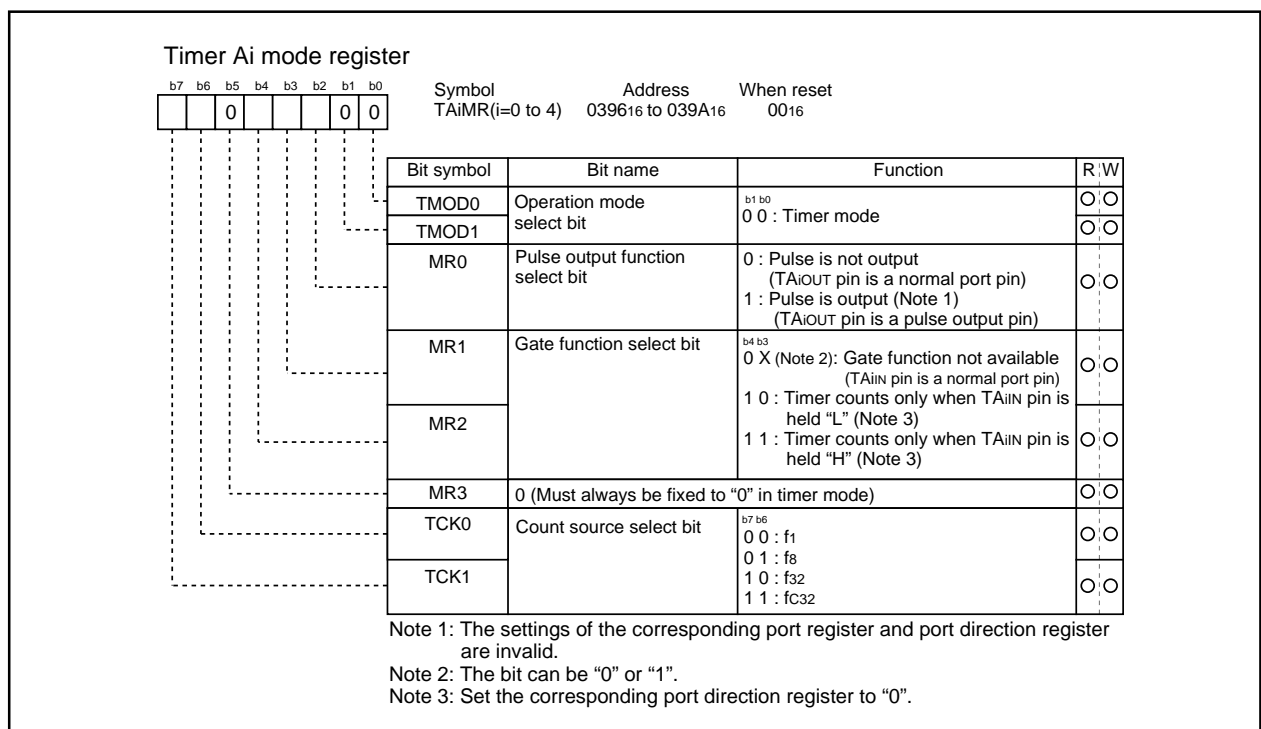
 SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 2.10.1) Figure 2.10.7 shows the timer Ai mode register in timer mode.

**Table 2.10.1 Specifications of timer mode**

Item	Specification
Count source	f <sub>1</sub> , f <sub>8</sub> , f <sub>32</sub> , f <sub>C32</sub>
Count operation	<ul style="list-style-type: none"> <li>Down count</li> <li>When the timer underflows, it reloads the reload register contents before continuing counting</li> </ul>
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	<ul style="list-style-type: none"> <li>When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter</li> <li>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)</li> </ul>
Select function	<ul style="list-style-type: none"> <li>Gate function Counting can be started and stopped by the TAiIN pin's input signal</li> <li>Pulse output function Each time the timer underflows, the TAiOUT pin's polarity is reversed</li> </ul>


**Figure 2.10.7 Timer Ai mode register in timer mode**

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (2) Event counter mode

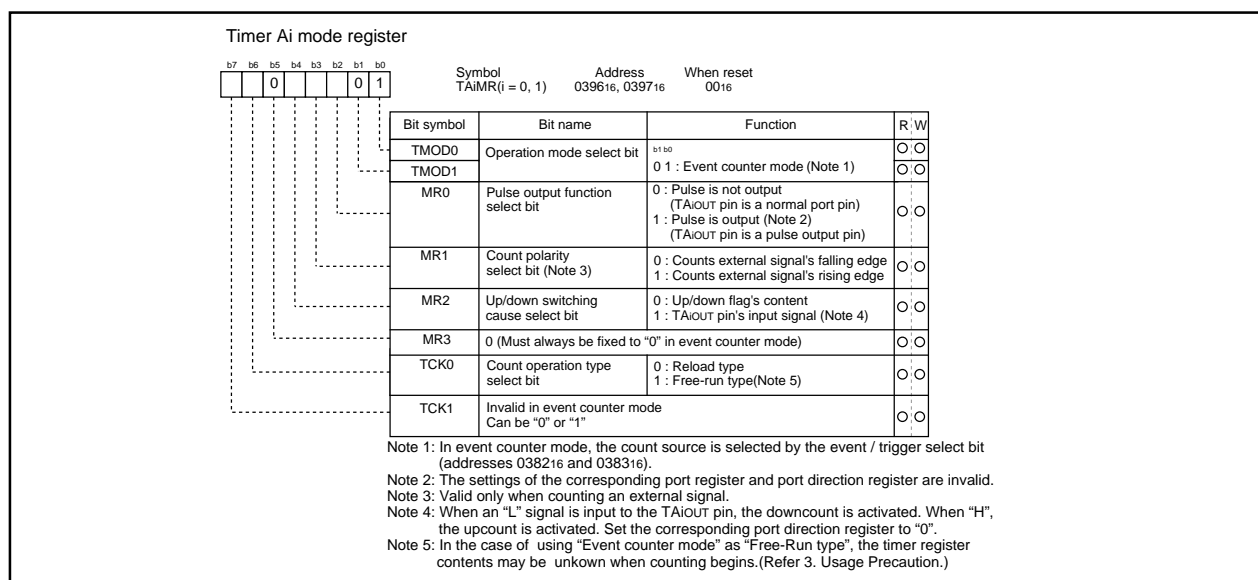
In this mode, the timer counts an external signal or an internal timer's overflow. Timers A0 and A1 can count a single-phase external signal. Timers A2, A3, and A4 can count a single-phase and a two-phase external signal. Table 2.10.2 lists timer specifications when counting a single-phase external signal. Figure 2.10.8 shows the timer Ai mode register in event counter mode.

Table 2.10.3 lists timer specifications when counting a two-phase external signal. Figure 2.10.9 shows the timer Ai mode register in event counter mode.

**Table 2.10.2 Timer specifications in event counter mode (when not processing two-phase pulse signal)**

Item	Specification
Count source	<ul style="list-style-type: none"> <li>External signals input to TAIin pin (effective edge can be selected by software)</li> <li>TB2 overflow, TAJ overflow</li> </ul>
Count operation	<ul style="list-style-type: none"> <li>Up count or down count can be selected by external signal or software</li> <li>When the timer overflows or underflows, it reloads the reload register contents before continuing counting (Note)</li> </ul>
Divide ratio	$1 / (FFFF_{16} - n + 1)$ for up count $1 / (n + 1)$ for down count      n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer overflows or underflows
TAiin pin function	Programmable I/O port or count source input
TAiOUT pin function	Programmable I/O port, pulse output, or up/down count select input
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	<ul style="list-style-type: none"> <li>When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter</li> <li>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)</li> </ul>
Select function	<ul style="list-style-type: none"> <li>Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it</li> <li>Pulse output function Each time the timer overflows or underflows, the TAIOUT pin's polarity is reversed</li> </ul>

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



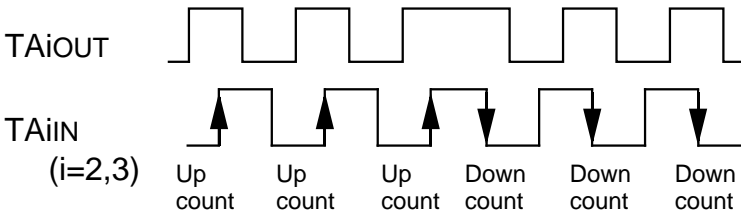
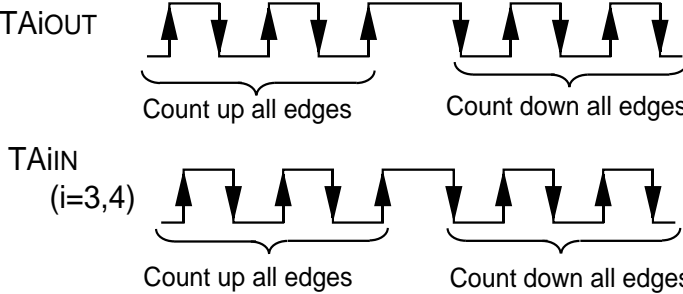
**Figure 2.10.8 Timer Ai mode register in event counter mode**



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

Table 2.10.3 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 (Note1)
Divide ratio	$1/(FFFF_{16} - n + 1)$ for up count $1/(n + 1)$ for down count $n$ : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	Timer overflows or underflows
TAiIN pin function	Two-phase pulse input
TAiOUT pin function	Two-phase pulse input
Read from timer	Count value can be read out by reading timer A2, A3, or A4 register
Write to timer	• When counting stopped When a value is written to timer A2, A3, or A4 register, it is written to both reload register and counter • When counting in progress When a value is written to timer A2, A3, or A4 register, it is written to only reload register. (Transferred to counter at next reload time.)
Select function(Note 2)	<ul style="list-style-type: none"> <li>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"</li> </ul>  <ul style="list-style-type: none"> <li>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.</li> </ul> 

Note 1: This does not apply when the free-run function is selected.

Note 2: Timer A3 alone can be selected. Timer A2 is fixed to normal processing operation, and timer A4 is fixed to multiply-by-4 processing operation.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

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

b7b6b5b4b3b2b1b0								Symbol	Address	When reset		
		0				0	1	TAiMR(i = 2 to 4)	0398 <sub>16</sub> to 039A <sub>16</sub>	00 <sub>16</sub>		
								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(Note 5)	○	○
								TCK1	Invalid in event counter mode Can be "0" or "1"		○	○

Note 1: In event counter mode, the count source is selected by the event / trigger select bit (addresses 0382<sub>16</sub> and 0383<sub>16</sub>).

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 TA<sub>i</sub>OUT pin, the downcount is activated. When "H", the upcount is activated. Set the corresponding port direction register to "0".

Note 5: In the case of using "Event counter mode" as "Free-Run type", the timer register contents may be unknown when counting begins.(Refer 3. Usage Precaution.)

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

b7b6b5b4b3b2b1b0								Symbol	Address	When reset								
<table border="1"><tr><td></td><td></td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>										0	1	0	0	0	1	TAiMR(i = 2 to 4)	0398 <sub>16</sub> to 039A <sub>16</sub>	00 <sub>16</sub>
		0	1	0	0	0	1											
								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(Note 3)	○	○						
								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<sub>16</sub>) is set to "1". Also, always be sure to set the event/trigger select bit (address 0383<sub>16</sub>) to "00".

Note 3: In the case of using "Event counter mode" as "Free-Run type", the timer register contents may be unknown when counting begins.(Refer 3. Usage Precaution.)

Figure 2.10.9 Timer Ai mode register in event counter mode

# M306H2FCFP

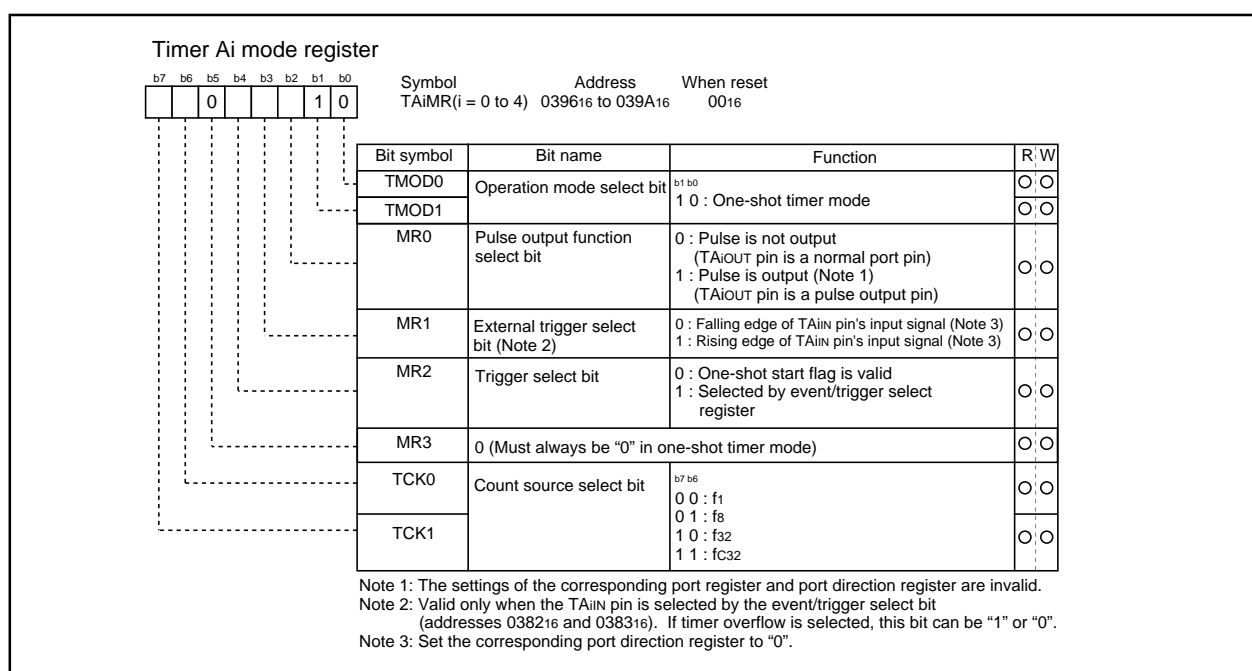
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (3) One-shot timer mode

In this mode, the timer operates only once. (See Table 2.10.4) When a trigger occurs, the timer starts up and continues operating for a given period. Figure 2.10.10 shows the timer Ai mode register in one-shot timer mode.

**Table 2.10.4 Timer specifications in one-shot timer mode**

Item	Specification
Count source	f1, f8, f32, fC32
Count operation	<ul style="list-style-type: none"> <li>The timer counts down</li> <li>When the count reaches 0000<sub>16</sub>, the timer stops counting after reloading a new count</li> <li>If a trigger occurs when counting, the timer reloads a new count and restarts counting</li> </ul>
Divide ratio	1/n    n : Set value
Count start condition	<ul style="list-style-type: none"> <li>An external trigger is input</li> <li>The timer overflows</li> <li>The one-shot start flag is set (= 1)</li> </ul>
Count stop condition	<ul style="list-style-type: none"> <li>A new count is reloaded after the count has reached 0000<sub>16</sub></li> <li>The count start flag is reset (= 0)</li> </ul>
Interrupt request generation timing	The count reaches 0000 <sub>16</sub>
TAiN pin function	Programmable I/O port or trigger input
TAiOUT pin function	Programmable I/O port or pulse output
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	<ul style="list-style-type: none"> <li>When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter</li> <li>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)</li> </ul>



**Figure 2.10.10 Timer Ai mode register in one-shot timer mode**

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

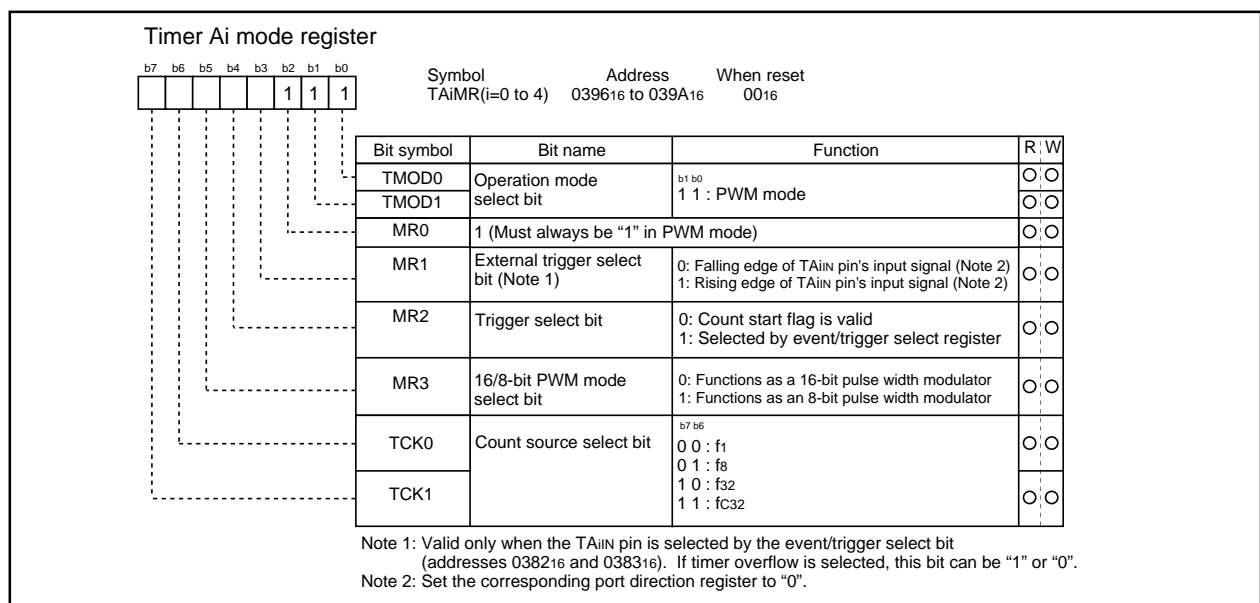
## (4) Pulse width modulation (PWM) mode

In this mode, the timer outputs pulses of a given width in succession. (See Table 2.10.5) In this mode, the counter functions as either a 16-bit pulse width modulator or an 8-bit pulse width modulator.

Figure 2.10.11 shows the timer Ai mode register in pulse width modulation mode. Figure 2.10.12 shows the example of how a 16-bit pulse width modulator operates. Figure 2.10.13 shows the example of how an 8-bit pulse width modulator operates.

**Table 2.10.5 Timer specifications in pulse width modulation mode**

Item	Specification
Count source	f1, f8, f32, fc32
Count operation	<ul style="list-style-type: none"> <li>The timer counts down (operating as an 8-bit or a 16-bit pulse width modulator)</li> <li>The timer reloads a new count at a rising edge of PWM pulse and continues counting</li> <li>The timer is not affected by a trigger that occurs when counting</li> </ul>
16-bit PWM	<ul style="list-style-type: none"> <li>High level width <math>n / f_i</math> <math>n</math>: Set value</li> <li>Cycle time <math>(2^{16}-1) / f_i</math> fixed</li> </ul>
8-bit PWM	<ul style="list-style-type: none"> <li>High level width <math>n \times (m+1) / f_i</math> <math>n</math>: values set to timer Ai register's high-order address</li> <li>Cycle time <math>(2^8-1) \times (m+1) / f_i</math> <math>m</math>: values set to timer Ai register's low-order address</li> </ul>
Count start condition	<ul style="list-style-type: none"> <li>External trigger is input</li> <li>The timer overflows</li> <li>The count start flag is set (= 1)</li> </ul>
Count stop condition	<ul style="list-style-type: none"> <li>The count start flag is reset (= 0)</li> </ul>
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	<ul style="list-style-type: none"> <li>When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter</li> <li>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)</li> </ul>



**Figure 2.10.11 Timer Ai mode register in pulse width modulation mode**

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

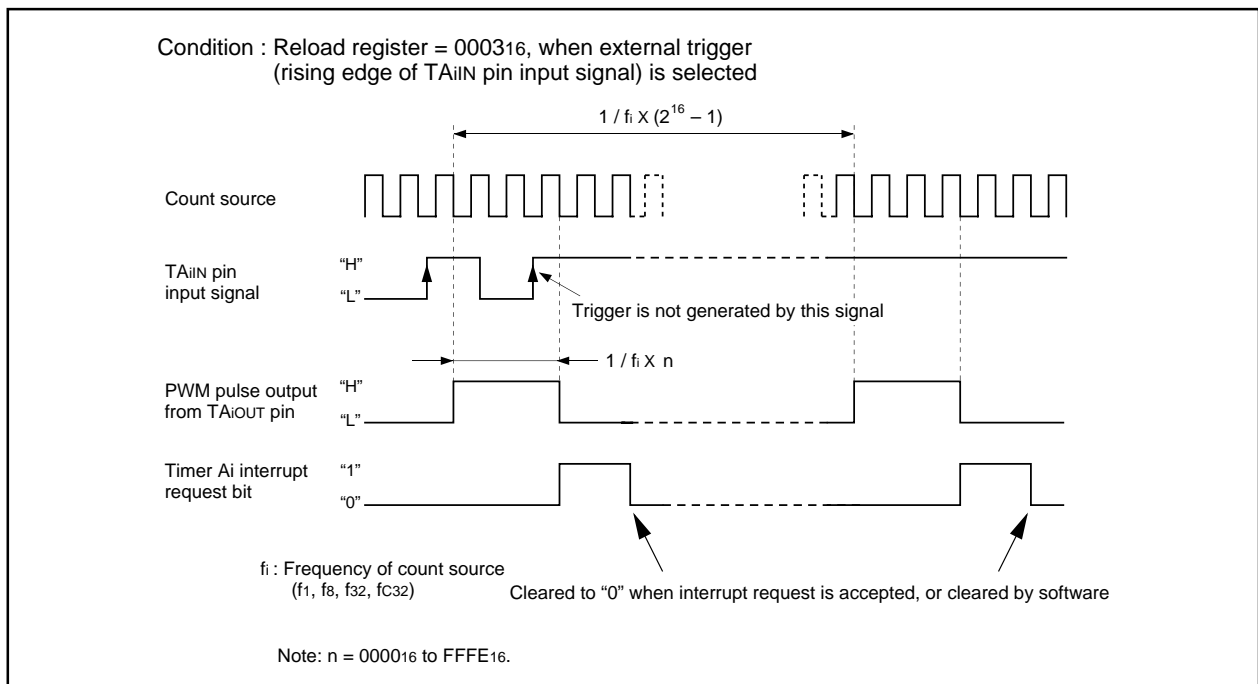


Figure 2.10.12 Example of how a 16-bit pulse width modulator operates

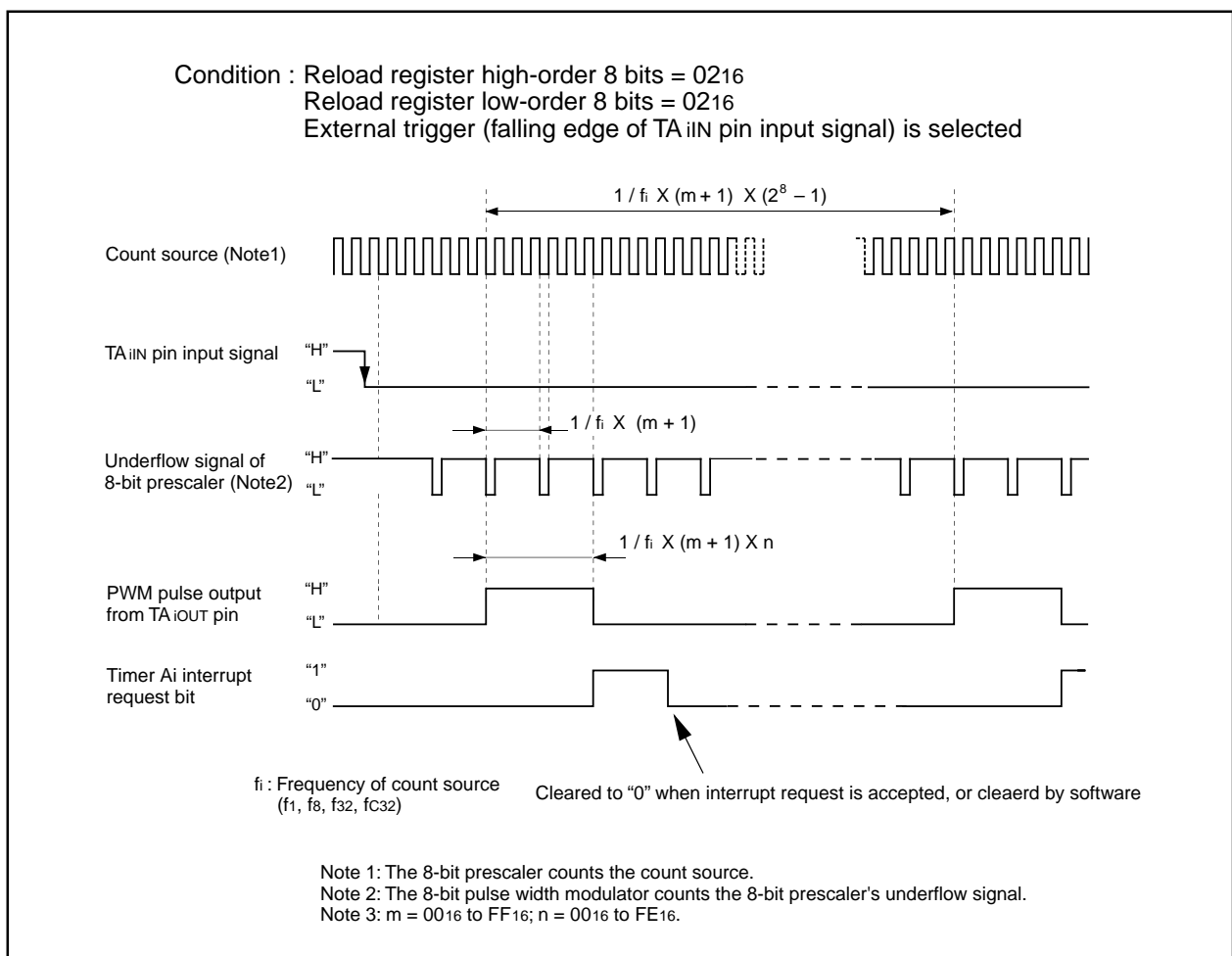


Figure 2.10.13 Example of how an 8-bit pulse width modulator operates

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.10.2 Timer B

Figure 2.10.14 shows the block diagram of timer B. Figures 2.10.15 and 2.10.16 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.

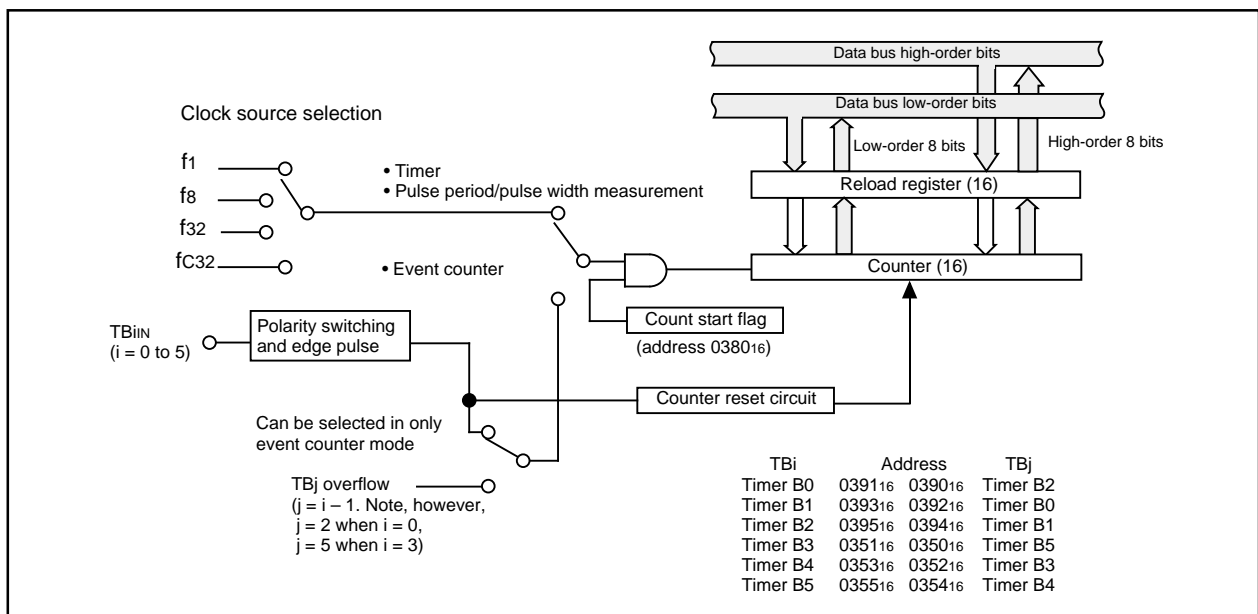


Figure 2.10.14 Block diagram of timer B

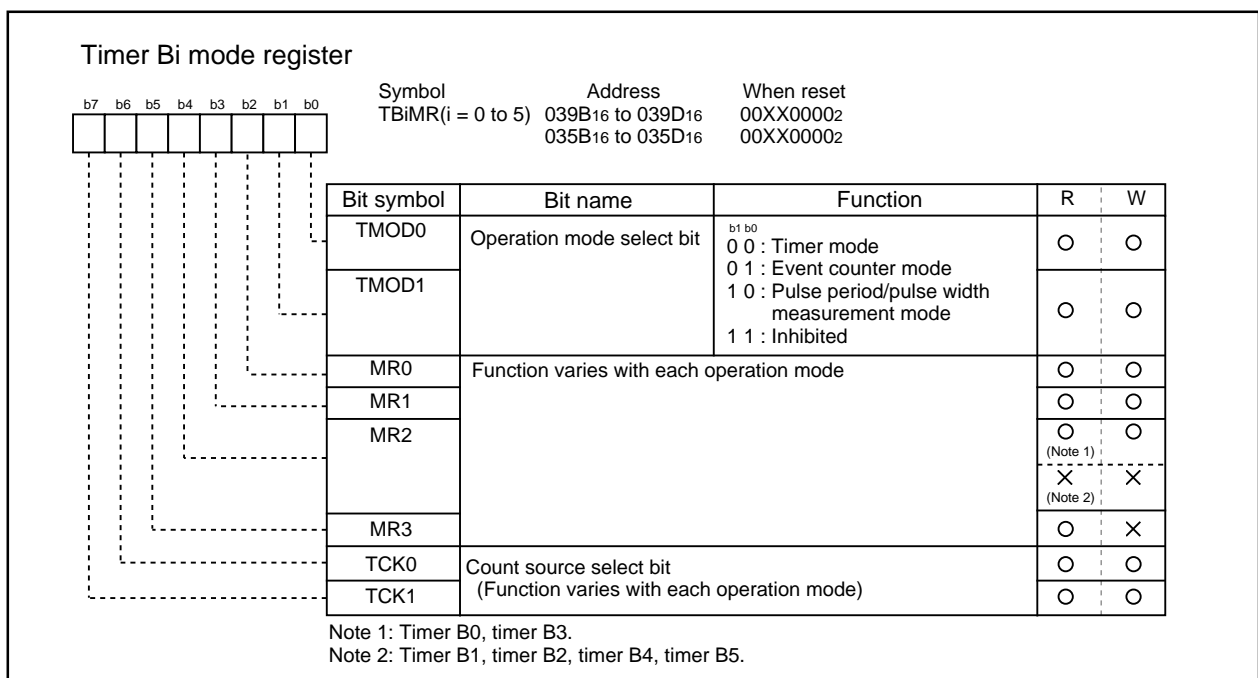
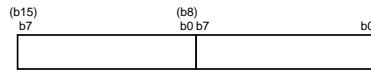


Figure 2.10.15 Timer B-related registers (1)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## Timer Bi register (Note)

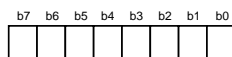


Symbol	Address	When reset
TB0	0391 <sub>16</sub> , 0390 <sub>16</sub>	Indeterminate
TB1	0393 <sub>16</sub> , 0392 <sub>16</sub>	Indeterminate
TB2	0395 <sub>16</sub> , 0394 <sub>16</sub>	Indeterminate
TB3	0351 <sub>16</sub> , 0350 <sub>16</sub>	Indeterminate
TB4	0353 <sub>16</sub> , 0352 <sub>16</sub>	Indeterminate
TB5	0355 <sub>16</sub> , 0354 <sub>16</sub>	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts the timer's period	0000 <sub>16</sub> to FFFF <sub>16</sub>	○	○
• Event counter mode Counts external pulses input or a timer overflow	0000 <sub>16</sub> to FFFF <sub>16</sub>	○	○
• 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 <sub>16</sub>	00 <sub>16</sub>

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		○	○

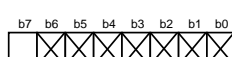
## Timer B3, 4, 5 count start flag



Symbol	Address	When reset
TBSR	0340 <sub>16</sub>	000XXXXX <sub>2</sub>

Bit symbol	Bit name	Function	R	W
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "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 <sub>16</sub>	0XXXXXXX <sub>2</sub>

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

Figure 2.10.16 Timer B-related registers (2)

# M306H2FCFP

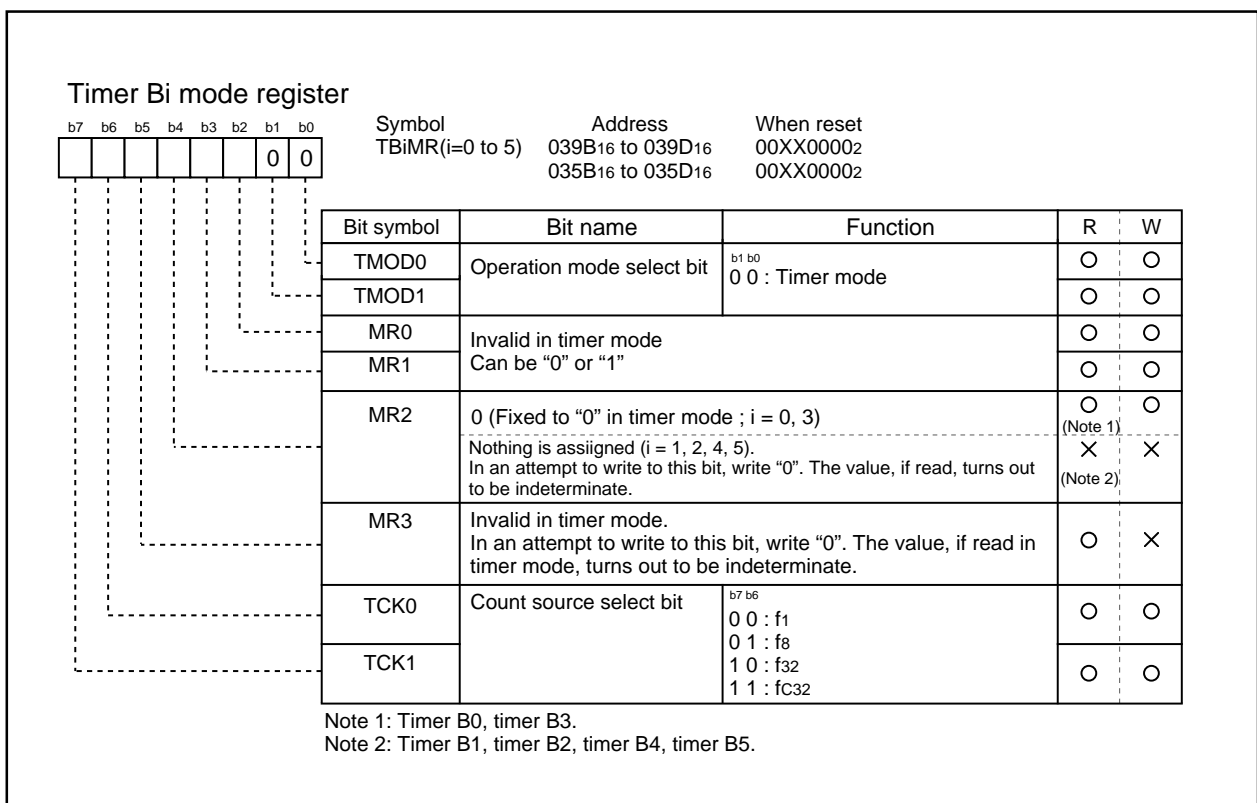
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## (1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 2.10.6) Figure 2.10.17 shows the timer Bi mode register in timer mode.

**Table 2.10.6 Timer specifications in timer mode**

Item	Specification
Count source	f1, f8, f32, fc32
Count operation	<ul style="list-style-type: none"> <li>Counts down</li> <li>When the timer underflows, it reloads the reload register contents before continuing counting</li> </ul>
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	Programmable I/O port
Read from timer	Count value is read out by reading timer Bi register
Write to timer	<ul style="list-style-type: none"> <li>When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter</li> <li>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)</li> </ul>



**Figure 2.10.17 Timer Bi mode register in timer mode**



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

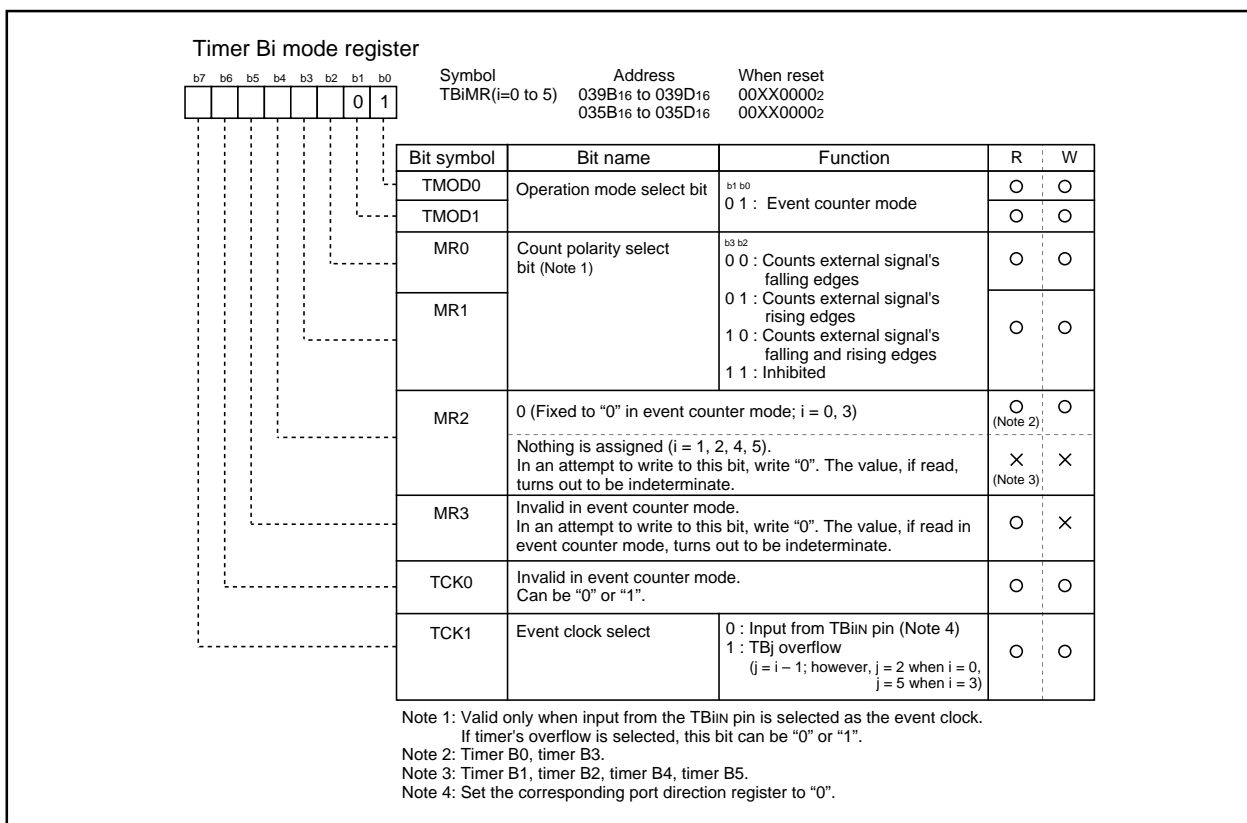
## (2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. (See Table 2.10.7)

Figure 2.10.18 shows the timer Bi mode register in event counter mode.

**Table 2.10.7 Timer specifications in event counter mode**

Item	Specification
Count source	<ul style="list-style-type: none"> <li>External signals input to TBiIN pin</li> <li>Effective edge of count source can be a rising edge, a falling edge, or falling and rising edges as selected by software</li> </ul>
Count operation	<ul style="list-style-type: none"> <li>Counts down</li> <li>When the timer underflows, it reloads the reload register contents before continuing counting</li> </ul>
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	<ul style="list-style-type: none"> <li>When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter</li> <li>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)</li> </ul>



**Figure 2.10.18 Timer Bi mode register in event counter mode**

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (3) Pulse period/pulse width measurement mode

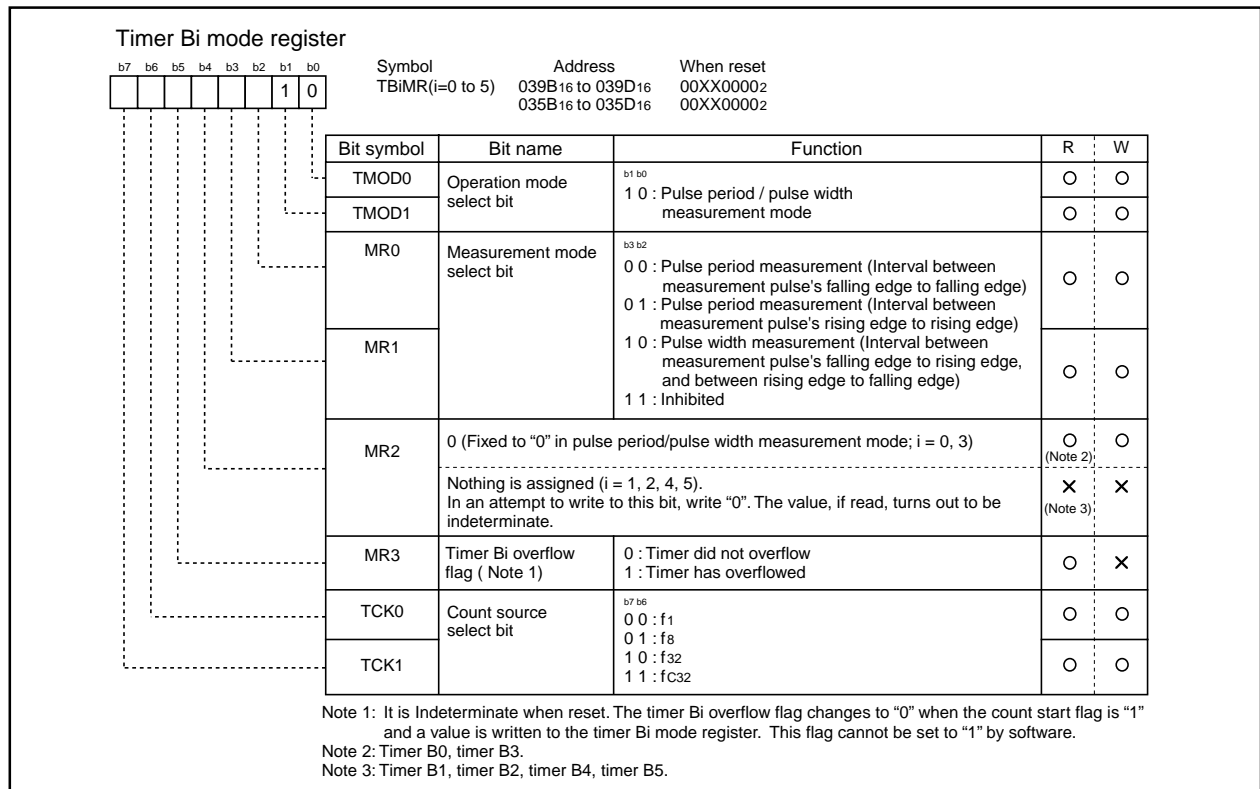
In this mode, the timer measures the pulse period or pulse width of an external signal. (See Table 2.10.8) Figure 2.10.19 shows the timer Bi mode register in pulse period/pulse width measurement mode. Figure 2.10.20 shows the operation timing when measuring a pulse period. Figure 2.10.21 shows the operation timing when measuring a pulse width.

**Table 2.10.8 Timer specifications in pulse period/pulse width measurement mode**

Item	Specification
Count source	f1, f8, f32, fC32
Count operation	<ul style="list-style-type: none"> <li>Up count</li> <li>Counter value "0000<sub>16</sub>" is transferred to reload register at measurement pulse's effective edge and the timer continues counting</li> </ul>
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	<ul style="list-style-type: none"> <li>When measurement pulse's effective edge is input (Note 1)</li> <li>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.)</li> </ul>
TBiIN 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.



**Figure 2.10.19 Timer Bi mode register in pulse period/pulse width measurement mode**

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

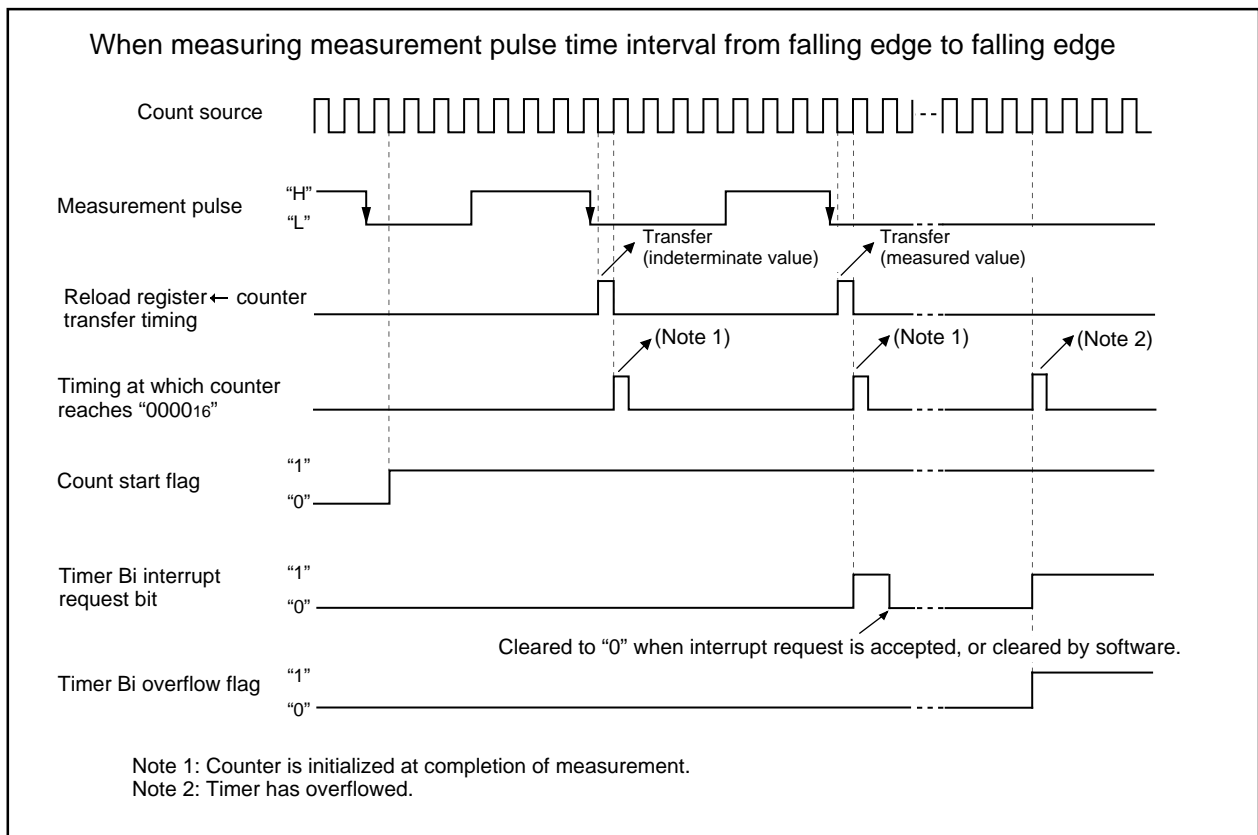


Figure 2.10.20 Operation timing when measuring a pulse period

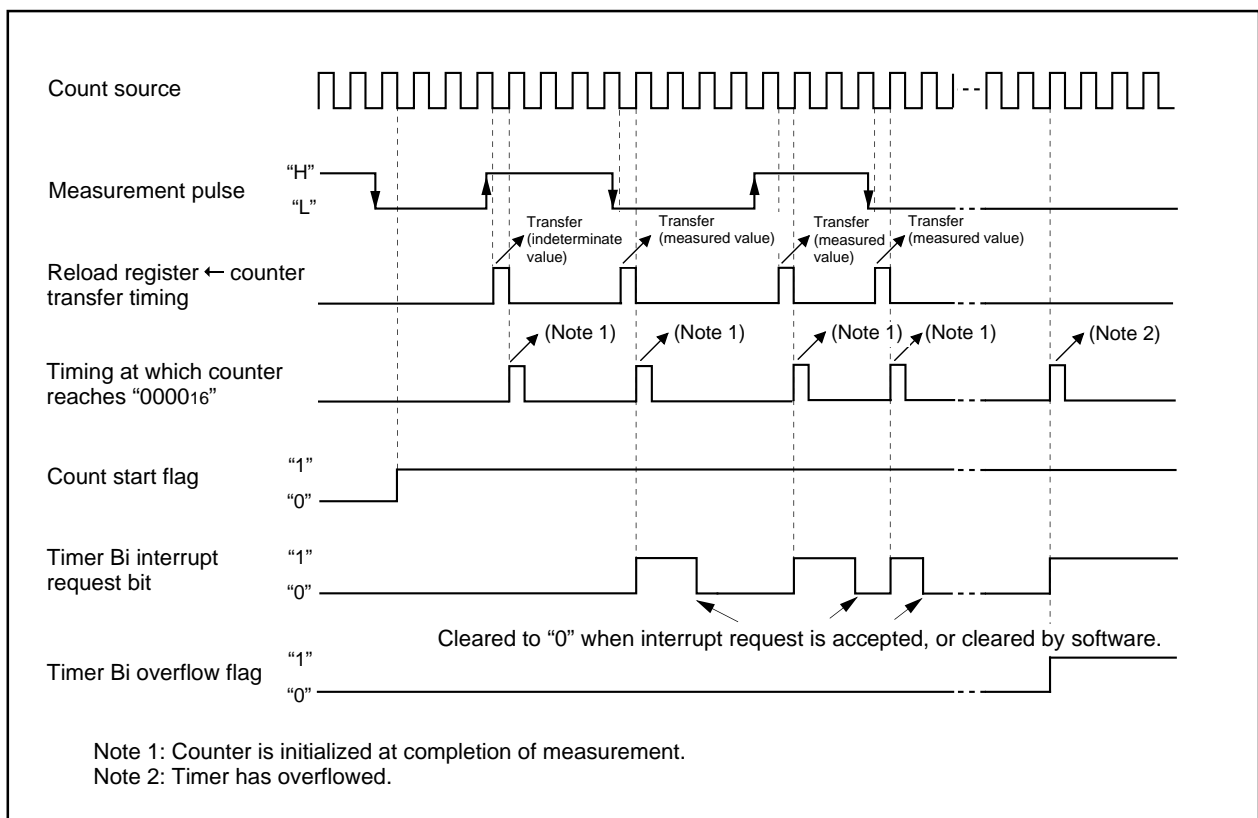


Figure 2.10.21 Operation timing when measuring a pulse width

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.11 Serial I/O

Serial I/O is configured as five channels: UART0, UART1, UART2, S I/O3 and S I/O4.

### 2.11.1 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 2.11.1 shows the block diagram of UART0, UART1 and UART2. Figures 2.11.2 and 2.11.3 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<sub>16</sub>, 03A8<sub>16</sub> and 0378<sub>16</sub>) 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.

UART0 through UART2 are almost equal in their functions with minor exceptions. UART2, in particular, is compliant with the SIM 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 2.11.1 shows the comparison of functions of UART0 through UART2, and Figures 2.11.4 to 2.11.8 show the registers related to UARTi.

Note: SIM : Subscriber Identity Module

**Table 2.11.1 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
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, RxD port output format	CMOS output	CMOS output	N-channel open-drain output
Parity error signal output	Impossible	Impossible	Possible (Note 4)
Bus collision detection	Impossible	Impossible	Possible

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.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

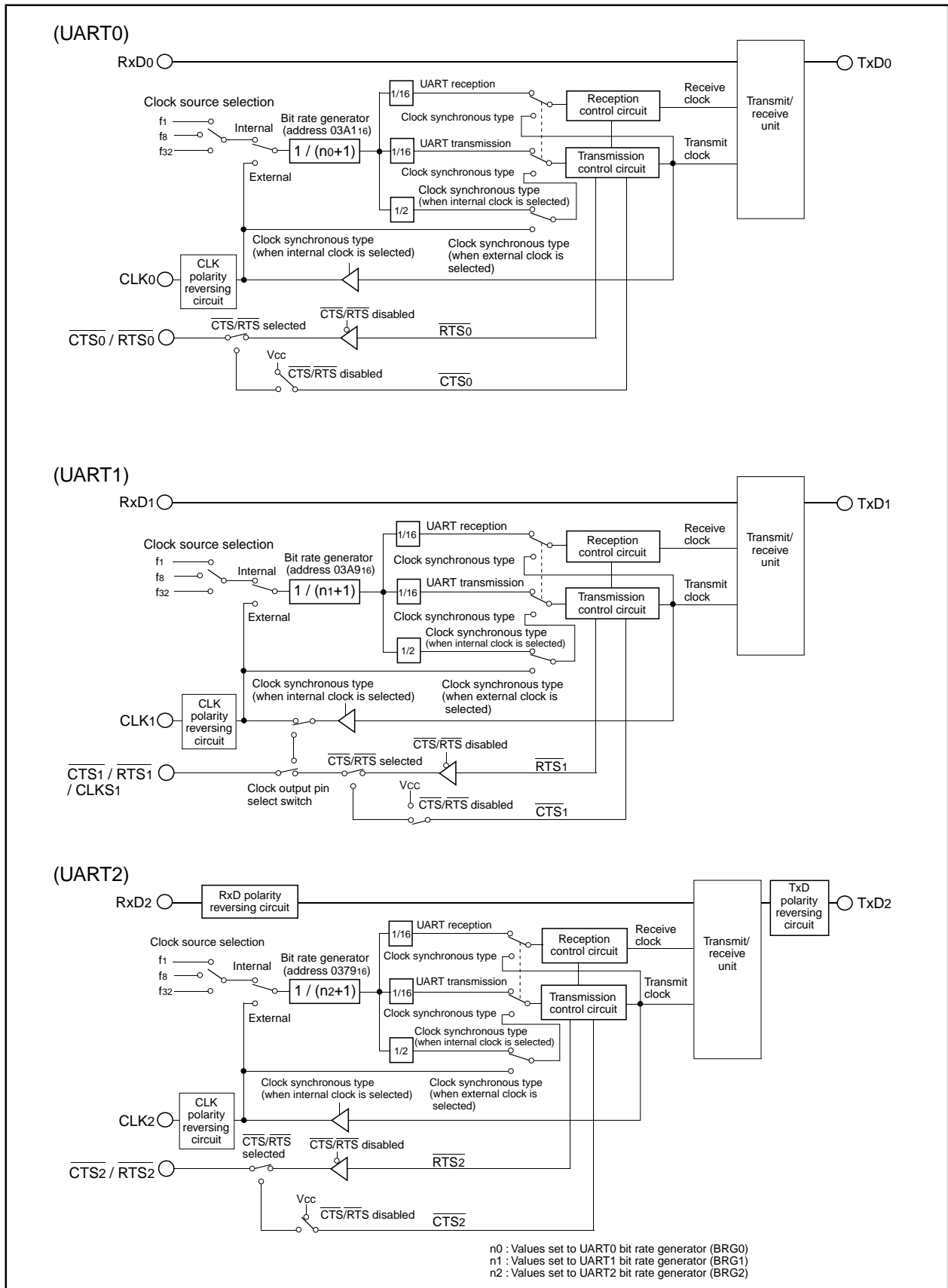


Figure 2.11.1 Block diagram of UARTi (i = 0 to 2)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

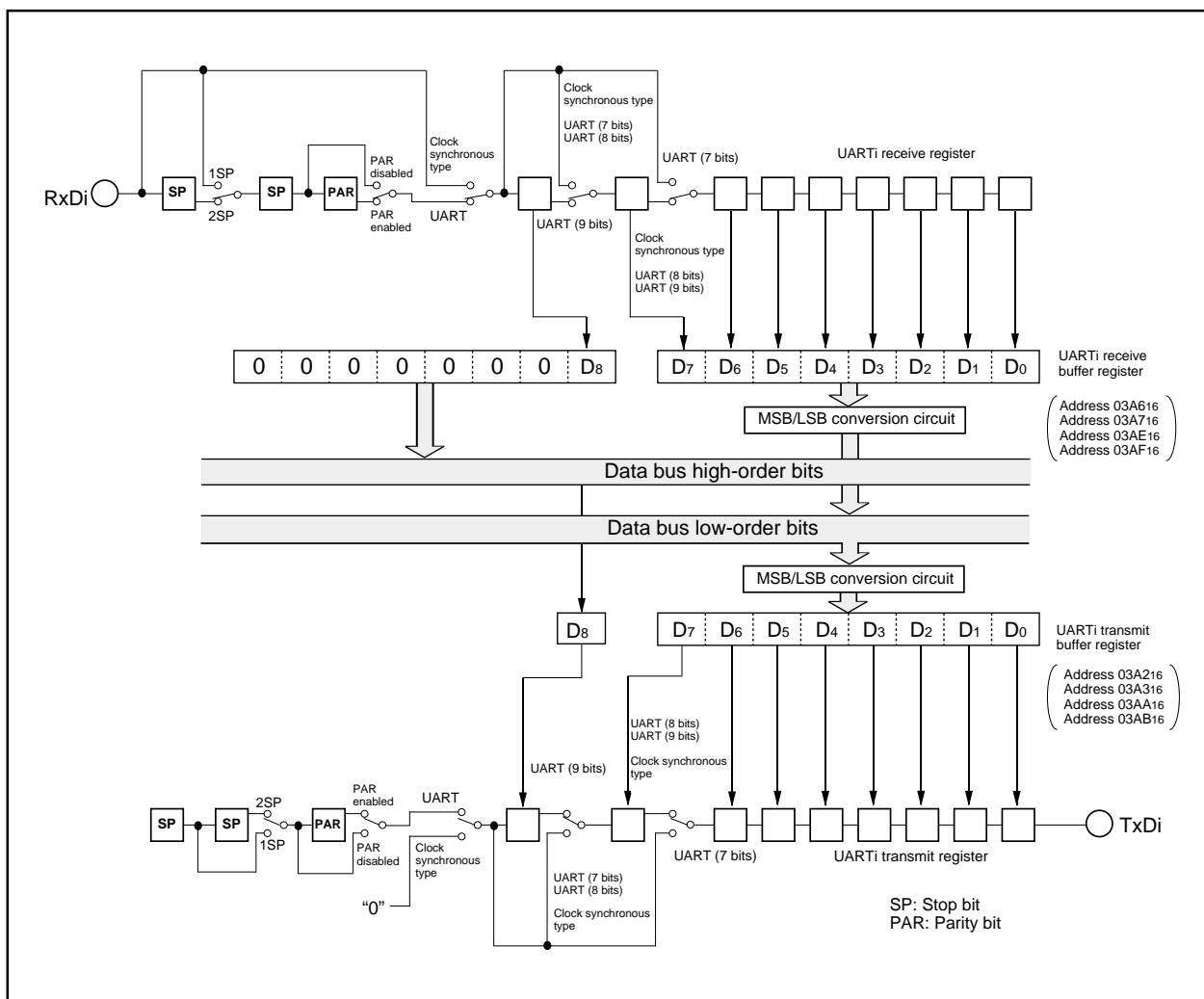


Figure 2.11.2 Block diagram of UARTi (i = 0, 1) transmit/receive unit

94

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

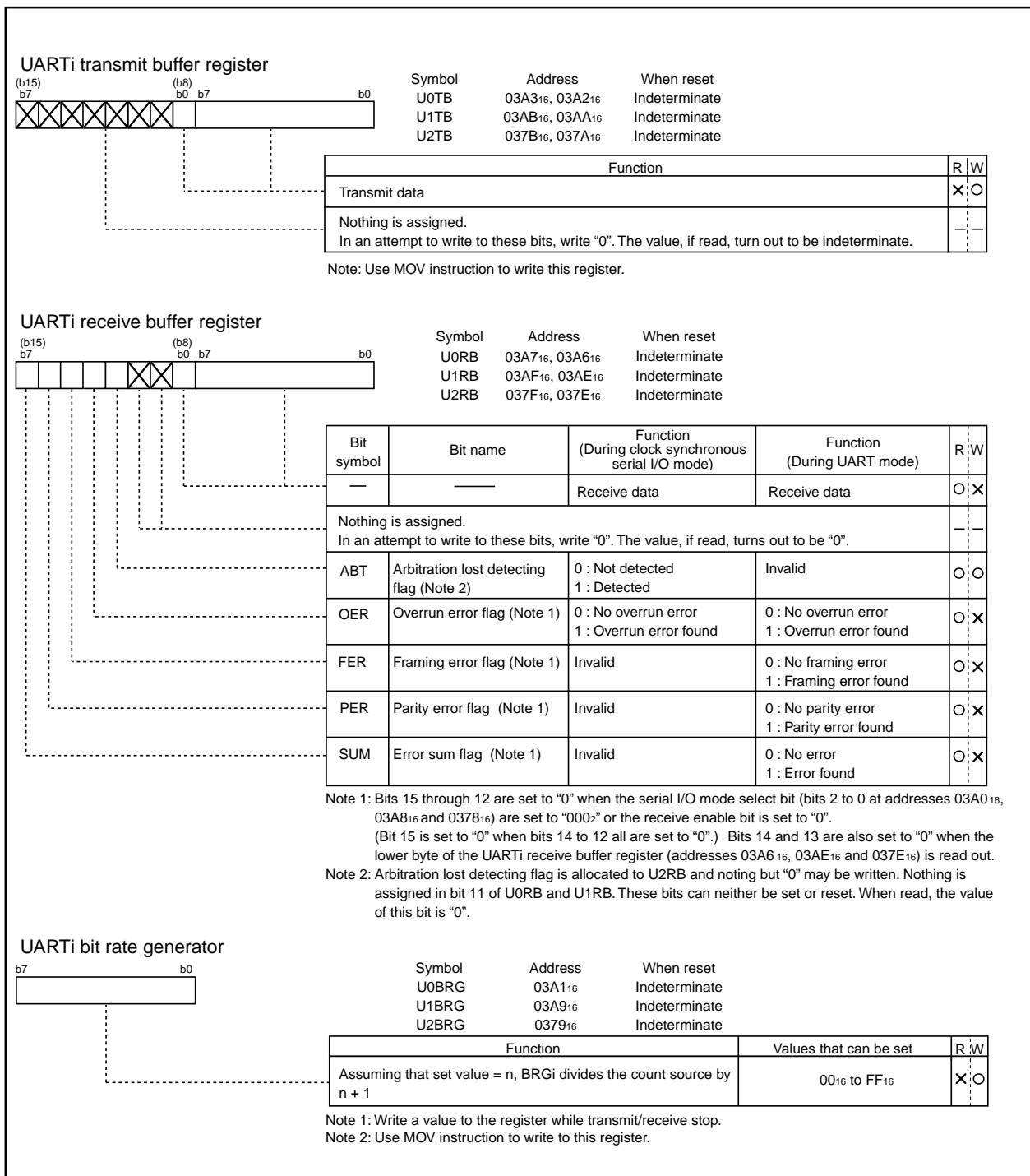


Figure 2.11.4 UARTi I/O-related registers (1)



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## UARTi transmit/receive mode register

Symbol								Address		When reset	
UiMR(i=0,1)								03A0 <sub>16</sub> , 03A8 <sub>16</sub>		00 <sub>16</sub>	
b7	b6	b5	b4	b3	b2	b1	b0				
	</										

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

## UART2 transmit/receive mode register

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	When reset
								U2MR	0378 <sub>16</sub>	00 <sub>16</sub>

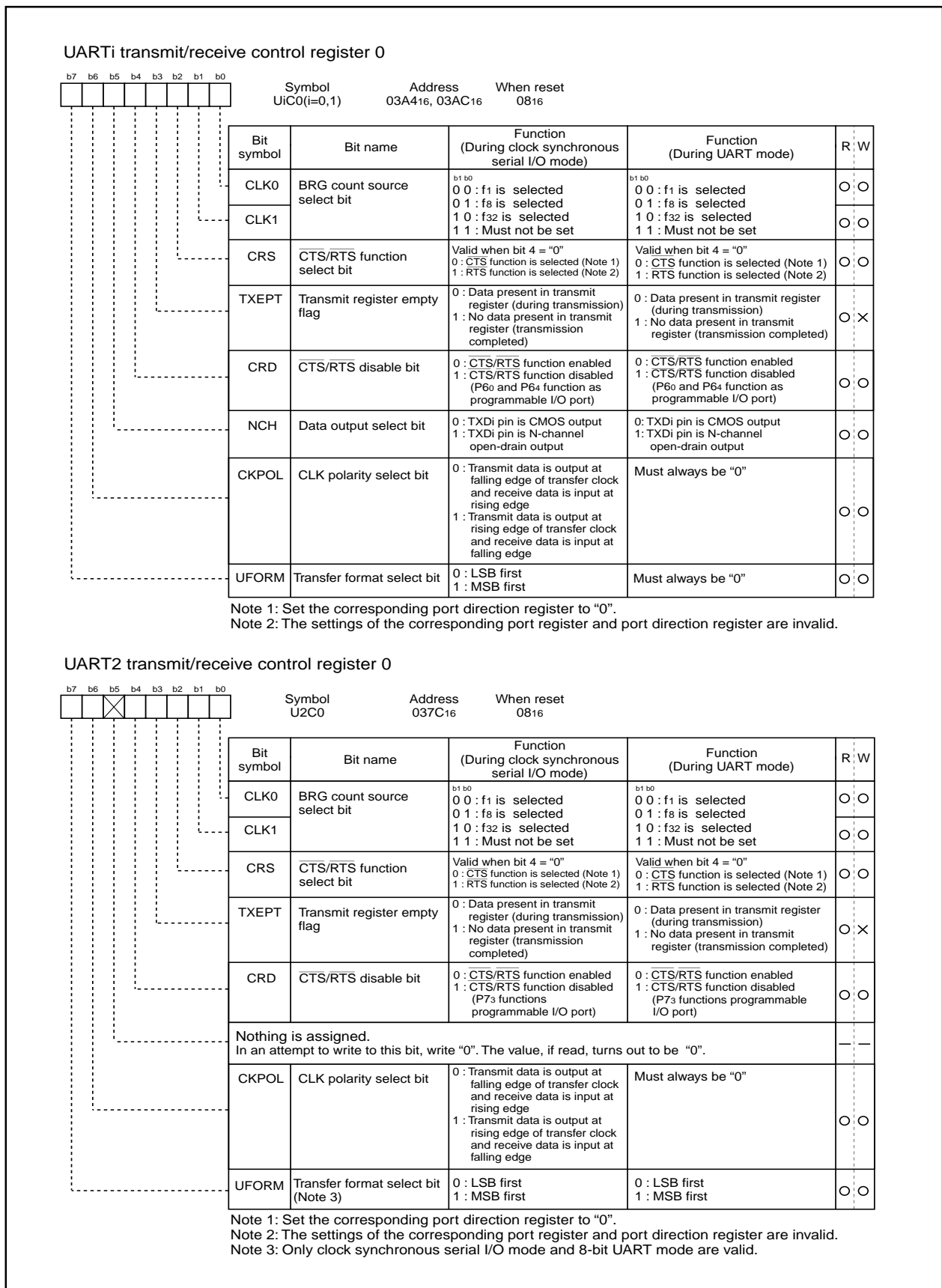
Notes1: Bit 2 to bit 0 are set to "010<sub>2</sub>" when I<sup>2</sup>C mode is used.

2: Set the corresponding direction register to "0".

Figure 2.11.5 UARTi I/O-related registers (2)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER



### Figure 2.11.6 UARTi I/O-related registers (3)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## UARTi transmit/receive control register 1

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

## UART2 transmit/receive control register 1

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	Invalid	○	○
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 2.11.7 UARTi I/O-related registers (4)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

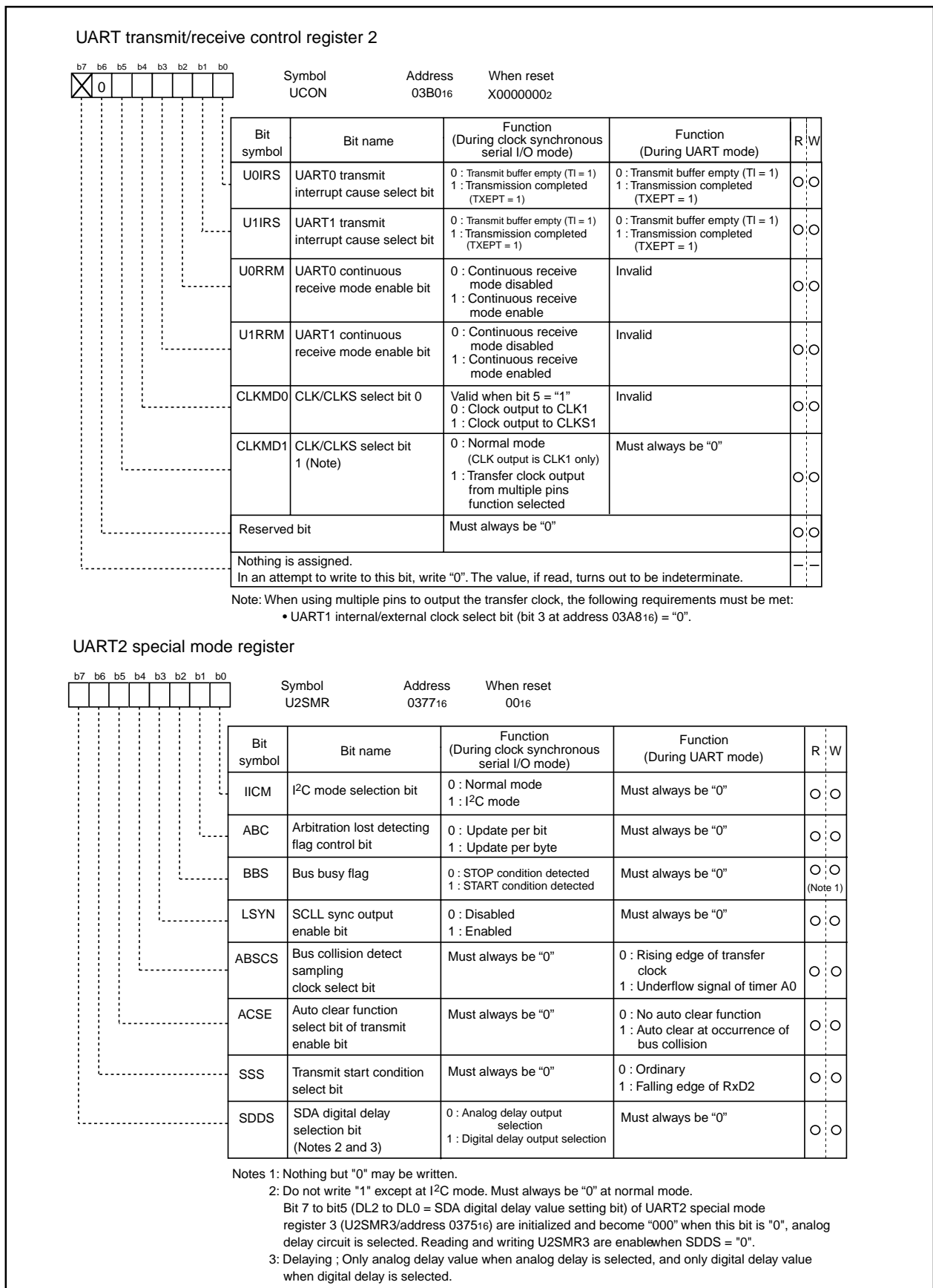


Figure 2.11.8 UARTi I/O-related registers (5)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## UART2 special mode register 2 (I<sup>2</sup>C bus exclusive register)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	When reset
								U2SMR2	0376 <sub>16</sub>	When reset 00 <sub>16</sub>
								</		

## UART2 special mode register 3 (I<sup>2</sup>C bus exclusive register)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	When reset
			X	X	X	X	X	U2SMR3	0375 <sub>16</sub>	Indeterminate (initializing value is "00 <sub>16</sub> " at SDDS = "1")

- Notes 1: Reading and writing is possible when bit7 (SDDS = SDA digital delay selection bit) of UART2 special mode register (U2SMR/address 0377<sub>16</sub>) is "1". When set SDDS = "1" and read out initialized value of UART2 special mode register 3(U2SMR3), this value is "00<sub>16</sub>".When set SDDS = "1" and write to UART2 special mode register 3(U2SMR3), set "0" to bit 0 to bit 4. When SDDS = "0", writing is enable. When read out, this value is indeterminate.
- 2: When SDDS = "0", this bit is initialized and become "000", selected analog delay circuit. This bit is become "000" after end reset released, and selected analog delay circuit. Reading out is possible when only SDDS = "1". when SDDS = "0", value which was read out is indeterminate.
- 3: Delaying ; Only analog delay value when analog delay is selected, and only digital delay value when digital delay is selected.
- 4: Delay level depends on SCL pin and SDA pin. And, when use external clock, delay is increase around 100ns. So test first, and use this.

Figure 2.11.9 UARTi -related registers (6)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.11.2 Clock synchronous serial I/O mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Tables 2.11.2 and 2.11.3 list the specifications of the clock synchronous serial I/O mode. Figur 2.11.10 shows the UARTi transmit/receive mode register.

**Table 2.11.2 Specifications of clock synchronous serial I/O mode (1)**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>Transfer data length: 8 bits</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>When internal clock is selected (bit 3 at addresses 03A0<sub>16</sub>, 03A8<sub>16</sub>, 0378<sub>16</sub> = "0") : <math>f_i / 2(n+1)</math> (Note 1) <math>f_i = f_1, f_8, f_{32}</math></li> <li>When external clock is selected (bit 3 at addresses 03A0<sub>16</sub>, 03A8<sub>16</sub>, 0378<sub>16</sub> = "1") : Input from CLKi pin</li> </ul>
Transmission/reception control	<ul style="list-style-type: none"> <li>CTS function/RTS function/CTS, RTS function chosen to be invalid</li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>To start transmission, the following requirements must be met: <ul style="list-style-type: none"> <li>Transmit enable bit (bit 0 at addresses 03A5<sub>16</sub>, 03AD<sub>16</sub>, 037D<sub>16</sub>) = "1"</li> <li>Transmit buffer empty flag (bit 1 at addresses 03A5<sub>16</sub>, 03AD<sub>16</sub>, 037D<sub>16</sub>) = "0"</li> <li>When CTS function selected, CTS input level = "L"</li> </ul> </li> <li>Furthermore, if external clock is selected, the following requirements must also be met: <ul style="list-style-type: none"> <li>CLKi polarity select bit (bit 6 at addresses 03A4<sub>16</sub>, 03AC<sub>16</sub>, 037C<sub>16</sub>) = "0": CLKi input level = "H"</li> <li>CLKi polarity select bit (bit 6 at addresses 03A4<sub>16</sub>, 03AC<sub>16</sub>, 037C<sub>16</sub>) = "1": CLKi input level = "L"</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>To start reception, the following requirements must be met: <ul style="list-style-type: none"> <li>Receive enable bit (bit 2 at addresses 03A5<sub>16</sub>, 03AD<sub>16</sub>, 037D<sub>16</sub>) = "1"</li> <li>Transmit enable bit (bit 0 at addresses 03A5<sub>16</sub>, 03AD<sub>16</sub>, 037D<sub>16</sub>) = "1"</li> <li>Transmit buffer empty flag (bit 1 at addresses 03A5<sub>16</sub>, 03AD<sub>16</sub>, 037D<sub>16</sub>) = "0"</li> </ul> </li> <li>Furthermore, if external clock is selected, the following requirements must also be met: <ul style="list-style-type: none"> <li>CLKi polarity select bit (bit 6 at addresses 03A4<sub>16</sub>, 03AC<sub>16</sub>, 037C<sub>16</sub>) = "0": CLKi input level = "H"</li> <li>CLKi polarity select bit (bit 6 at addresses 03A4<sub>16</sub>, 03AC<sub>16</sub>, 037C<sub>16</sub>) = "1": CLKi input level = "L"</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>When transmitting <ul style="list-style-type: none"> <li>Transmit interrupt cause select bit (bits 0, 1 at address 03B0<sub>16</sub>, bit 4 at address 037D<sub>16</sub>) = "0": Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed</li> <li>Transmit interrupt cause select bit (bits 0, 1 at address 03B0<sub>16</sub>, bit 4 at address 037D<sub>16</sub>) = "1": Interrupts requested when data transmission from UARTi transfer register is completed</li> </ul> </li> <li>When receiving <ul style="list-style-type: none"> <li>Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed</li> </ul> </li> </ul>
Error detection	<ul style="list-style-type: none"> <li>Overrun error (Note 2)</li> </ul> <p>This error occurs when the next data is ready before contents of UARTi receive buffer register are read out</p>

Note 1: "n" denotes the value 00<sub>16</sub> to FF<sub>16</sub> 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 does not change.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

**Table 2.11.3 Specifications of clock synchronous serial I/O mode (2)**

Item	Specification
Select function	<ul style="list-style-type: none"> <li>• CLK polarity selection Whether transmit data is output/input at the rising edge or falling edge of the transfer clock can be selected</li> <li>• LSB first/MSB first selection Whether transmission/reception begins with bit 0 or bit 7 can be selected</li> <li>• Continuous receive mode selection Reception is enabled simultaneously by a read from the receive buffer register</li> <li>• Transfer clock output from multiple pins selection (UART1) UART1 transfer clock can be chosen by software to be output from one of the two pins set</li> <li>• 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.</li> <li>• 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.</li> </ul>

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## UARTi transmit/receive mode registers

<div><div><div>b7</div><div>b6</div><div>b5</div><div>b4</div><div>b3</div><div>b2</div><div>b1</div><div>b0</div></div><div><div>0</div><div></div><div></div><div></div><div></div><div>0</div><div>0</div><div>1</div></div></div>								Symbol UiMR(i=0,1)	Address 03A0 <sub>16</sub> , 03A8 <sub>16</sub>	When reset 00 <sub>16</sub>		
<div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div></div>								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	<div></div> <div></div>	
								SMD1			<div></div> <div></div>	
								SMD2			<div></div> <div></div>	
								CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock (See note 1)	<div></div> <div></div>	
								STPS	Invalid in clock synchronous serial I/O mode		<div></div> <div></div>	
								PRY			<div></div> <div></div>	
								PRYE			<div></div> <div></div>	
								SLEP	0 (Must always be “0” in clock synchronous serial I/O mode)		<div></div> <div></div>	

Note 1: Set a corresponding direction register to "0."

## UART2 transmit/receive mode register

<div><div>b7b6b5b4b3b2b1b0</div><div><div>0000001</div></div></div>								Symbol U2MR	Address 0378 <sub>16</sub>	When reset 00 <sub>16</sub>				
<div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div>								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 (See note 2)				
								STPS	Invalid in clock synchronous serial I/O mode					
								PRY						
								PRYE						
								IOPOL	TxD, Rx D I/O polarity reverse bit (Note1)	0 : No reverse 1 : Reverse				

Notes 1: Usually set to "0".

2: Set a corresponding direction register to "0."

Figure 2.11.10 UARTi transmit/receive mode register in clock synchronous serial I/O mode



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

Table 2.11.4 lists the functions of the input/output pins during clock synchronous serial I/O mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs a "H". (If the N-channel open-drain is selected, this pin is in floating state.)

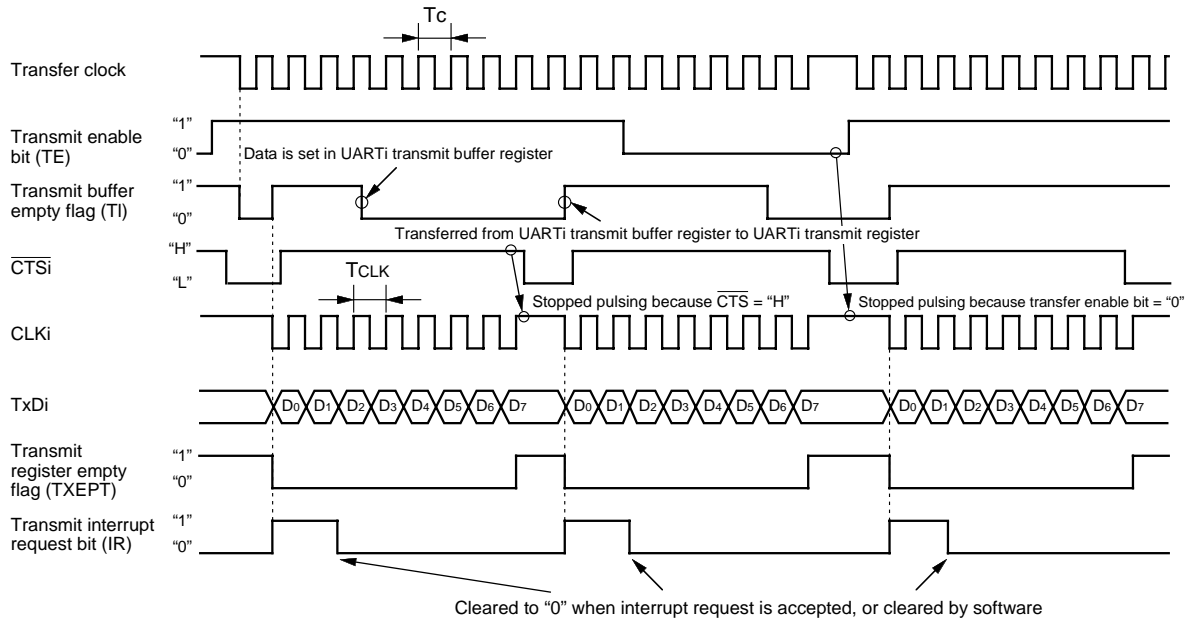
**Table 2.11.4 Input/output pin functions in clock synchronous serial I/O mode**

Pin name	Function	Method of selection
TxDi (P63, P67, P70)	Serial data output	(Outputs dummy data when performing reception only)
RxDi (P62, P66, P71)	Serial data input	Port P62, P66 and P71 direction register (bits 2 and 6 at address 03EE <sub>16</sub> , bit 1 at address 03EF <sub>16</sub> ) = "0" (Can be used as an input port when performing transmission only)
CLKi (P61, P65, P72)	Transfer clock output	Internal/external clock select bit (bit 3 at address 03A0 <sub>16</sub> , 03A8 <sub>16</sub> , 0378 <sub>16</sub> ) = "0"
	Transfer clock input	Internal/external clock select bit (bit 3 at address 03A0 <sub>16</sub> , 03A8 <sub>16</sub> , 0378 <sub>16</sub> ) = "1" Port P61, P65 and P72 direction register (bits 1 and 5 at address 03EE <sub>16</sub> , bit 2 at address 03EF <sub>16</sub> ) = "0"
$\overline{\text{CTS}}/\overline{\text{RTS}}$ (P60, P64, P73)	$\overline{\text{CTS}}$ input	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> , 037C <sub>16</sub> ) = "0" $\overline{\text{CTS}}/\overline{\text{RTS}}$ function select bit (bit 2 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> , 037C <sub>16</sub> ) = "0" Port P60, P64 and P73 direction register (bits 0 and 4 at address 03EE <sub>16</sub> , bit 3 at address 03EF <sub>16</sub> ) = "0"
	$\overline{\text{RTS}}$ output	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> , 037C <sub>16</sub> ) = "0" $\overline{\text{CTS}}/\overline{\text{RTS}}$ function select bit (bit 2 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> , 037C <sub>16</sub> ) = "1"
	Programmable I/O port	$\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at address 03A4 <sub>16</sub> , 03AC <sub>16</sub> , 037C <sub>16</sub> ) = "1"

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

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



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

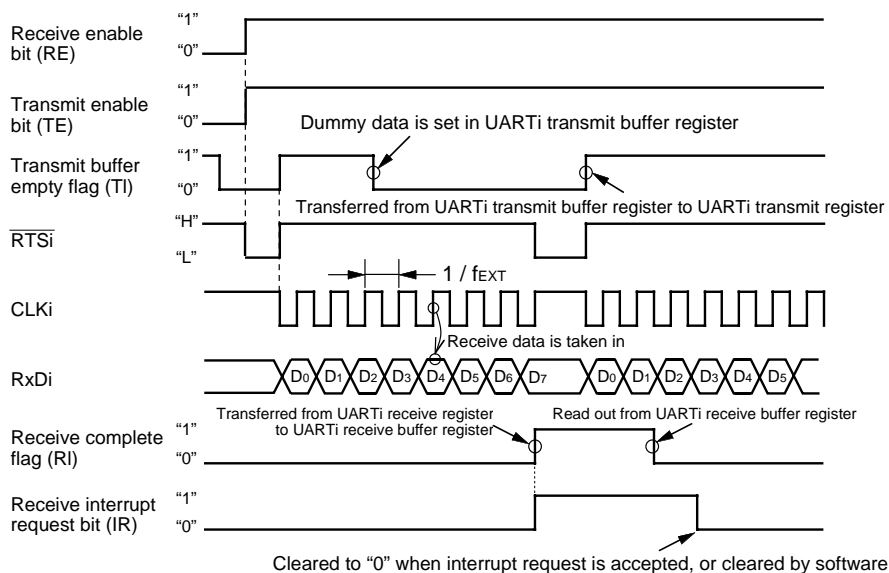


Figure 2.11.11 Typical transmit/receive timings in clock synchronous serial I/O mode

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (1) Polarity select function

As shown in Figure 2.11.12 the CLK polarity select bit (bit 6 at addresses 03A4<sub>16</sub>, 03AC<sub>16</sub>, 037C<sub>16</sub>) allows selection of the polarity of the transfer clock.

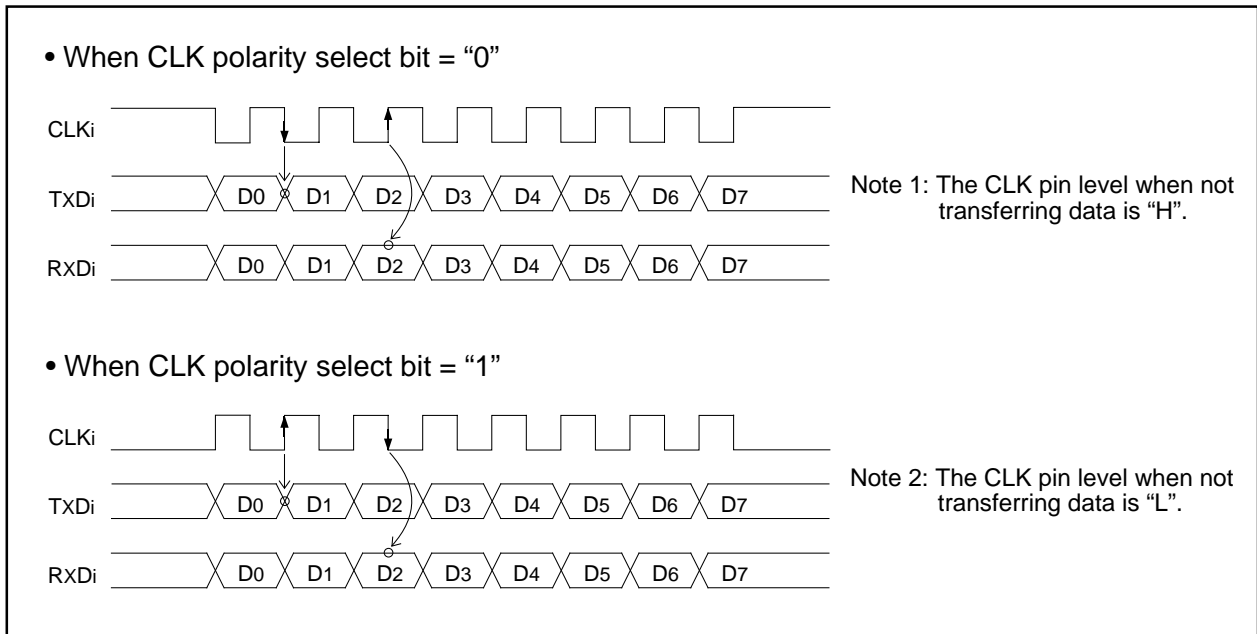


Figure 2.11.12 Polarity of transfer clock

## (2) LSB first/MSB first select function

As shown in Figure 2.11.13, when the transfer format select bit (bit 7 at addresses 03A4<sub>16</sub>, 03AC<sub>16</sub>, 037C<sub>16</sub>) = "0", the transfer format is "LSB first"; when the bit = "1", the transfer format is "MSB first".

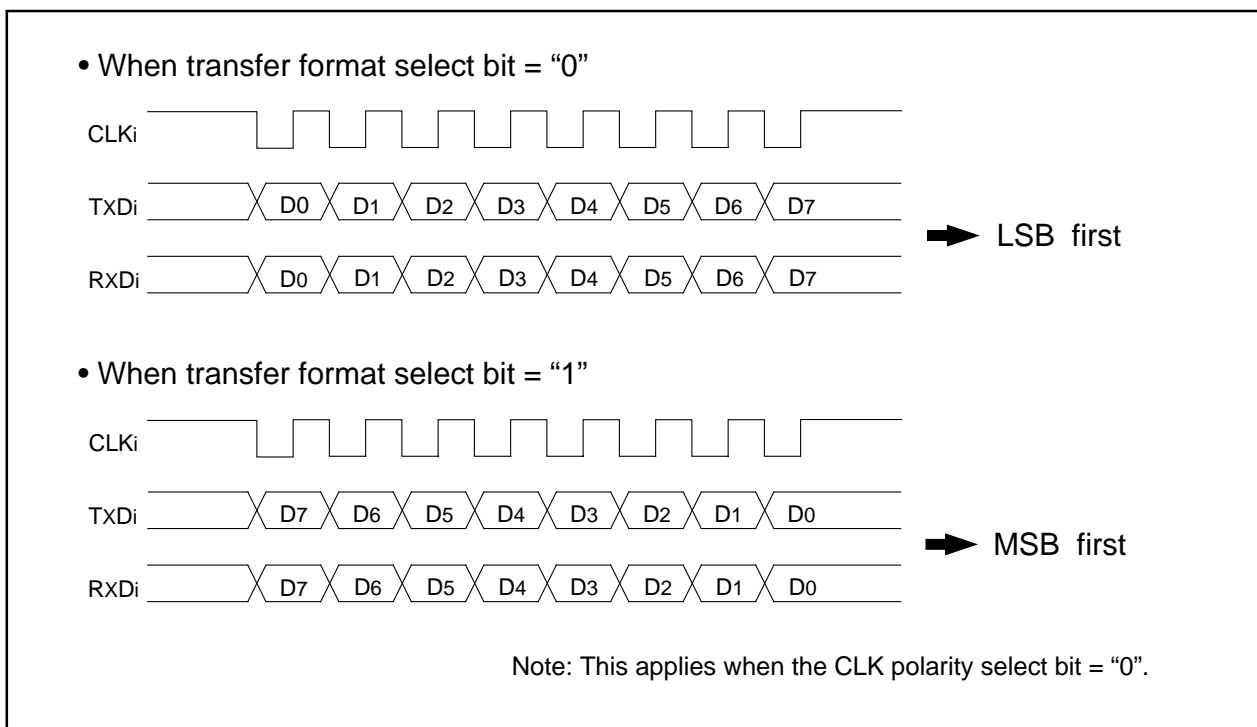


Figure 2.11.13 Transfer format

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

### (3) 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 03B016). (See Figure 2.11.14)  
The multiple pins function is valid only when the internal clock is selected for UART1.

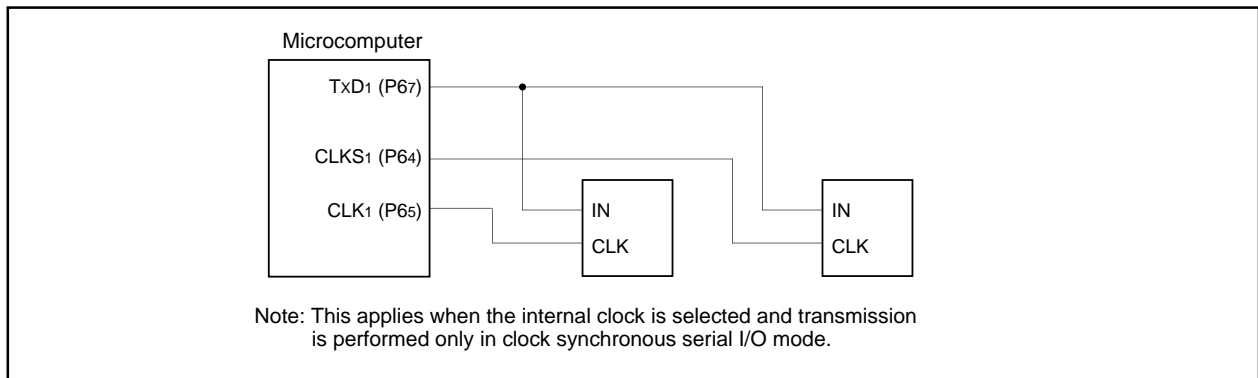


Figure 2.11.14 The transfer clock output from the multiple pins function usage

### (4) Continuous receive mode

If the continuous receive mode enable bit (bits 2 and 3 at address 03B016, bit 5 at address 037D16) 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.

### (5) Serial data logic switch function (UART2)

When the data logic select bit (bit6 at address 037D16) = "1", and writing to transmit buffer register or reading from receive buffer register, data is reversed. Figure 2.11.15 shows the example of serial data logic switch timing.

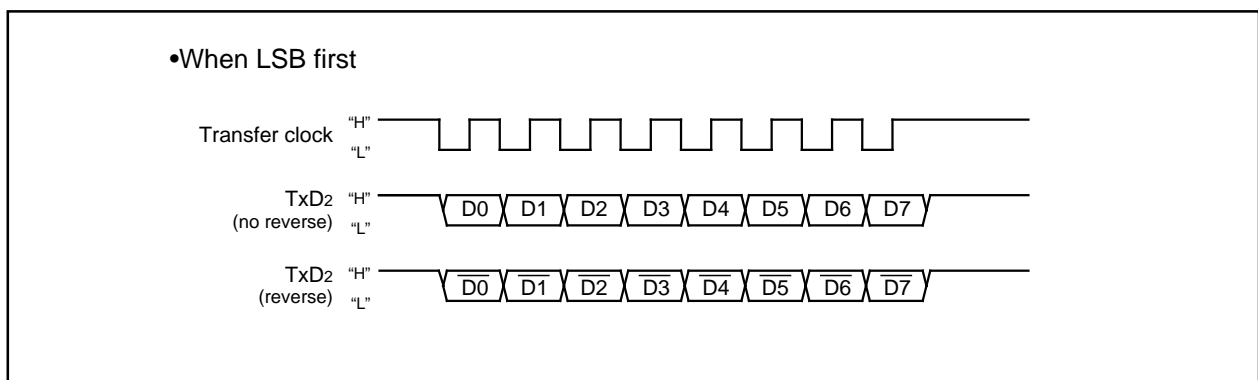


Figure 2.11.15 Serial data logic switch timing

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.11.3 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 2.11.5 and 2.11.6 list the specifications of the UART mode. Figure 2.11.16 shows the UARTi transmit/receive mode register.

**Table 2.11.5 Specifications of UART Mode (1)**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>• Character bit (transfer data): 7 bits, 8 bits, or 9 bits as selected</li> <li>• Start bit: 1 bit</li> <li>• Parity bit: Odd, even, or nothing as selected</li> <li>• Stop bit: 1 bit or 2 bits as selected</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>• When internal clock is selected (bit 3 at addresses 03A0<sub>16</sub>, 03A8<sub>16</sub>, 0378<sub>16</sub> = "0") : <math>f_i/16(n+1)</math> (Note 1) <math>f_i = f_1, f_8, f_{32}</math></li> <li>• When external clock is selected (bit 3 at addresses 03A0<sub>16</sub> and 03A8<sub>16</sub> = "1") : <math>f_{EXT}/16(n+1)</math> (Note 1) (Note 2) (Do not set external clock for UART2)</li> </ul>
Transmission/reception control	<ul style="list-style-type: none"> <li>• CTS function/RTS function/CTS, RTS function chosen to be invalid</li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>• To start transmission, the following requirements must be met: <ul style="list-style-type: none"> <li>- Transmit enable bit (bit 0 at addresses 03A5<sub>16</sub>, 03AD<sub>16</sub>, 037D<sub>16</sub>) = "1"</li> <li>- Transmit buffer empty flag (bit 1 at addresses 03A5<sub>16</sub>, 03AD<sub>16</sub>, 037D<sub>16</sub>) = "0"</li> <li>- When CTS function selected, CTS input level = "L"</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>• To start reception, the following requirements must be met: <ul style="list-style-type: none"> <li>- Receive enable bit (bit 2 at addresses 03A5<sub>16</sub>, 03AD<sub>16</sub>, 037D<sub>16</sub>) = "1"</li> <li>- Start bit detection</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• When transmitting <ul style="list-style-type: none"> <li>- Transmit interrupt cause select bits (bits 0,1 at address 03B0<sub>16</sub>, bit4 at address 037D<sub>16</sub>) = "0": Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed</li> <li>- Transmit interrupt cause select bits (bits 0, 1 at address 03B0<sub>16</sub>, bit4 at address 037D<sub>16</sub>) = "1": Interrupts requested when data transmission from UARTi transfer register is completed</li> </ul> </li> <li>• When receiving <ul style="list-style-type: none"> <li>- Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed</li> </ul> </li> </ul>
Error detection	<ul style="list-style-type: none"> <li>• Overrun error (Note 3) <ul style="list-style-type: none"> <li>This error occurs when the next data is ready before contents of UARTi receive buffer register are read out</li> </ul> </li> <li>• Framing error <ul style="list-style-type: none"> <li>This error occurs when the number of stop bits set is not detected</li> </ul> </li> <li>• Parity error <ul style="list-style-type: none"> <li>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</li> </ul> </li> <li>• Error sum flag <ul style="list-style-type: none"> <li>This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered</li> </ul> </li> </ul>

Note 1: 'n' denotes the value 00<sub>16</sub> to FF<sub>16</sub> that is set to the UARTi bit rate generator.

Note 2: f<sub>EXT</sub> 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 does not change.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

**Table 2.11.6 Specifications of UART Mode (2)**

Item	Specification
Select function	<ul style="list-style-type: none"> <li>• Sleep mode selection (UART0, UART1) This mode is used to transfer data to and from one of multiple slave micro-computers</li> <li>• Serial data logic switch (UART2) This function is reversing logic value of transferring data. Start bit, parity bit and stop bit are not reversed.</li> <li>• Tx/D, Rx/D I/O polarity switch (UART2) This function is reversing Tx/D port output and Rx/D port input. All I/O data level is reversed.</li> </ul>

# M306H2FCFP

 SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## UARTi transmit / receive mode registers

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	When reset	
								UiMR(i=0,1)	03A0 <sub>16</sub> , 03A8 <sub>16</sub>	00 <sub>16</sub>	
								Bit symbol	Bit name	Function	R/W
								SMD0	Serial I/O mode select bit	b2 b1 b0 1 0 0 : Transfer data 7 bits length 1 0 1 : Transfer data 8 bits length 1 1 0 : Transfer data 9 bits length	O
								SMD1			O
								SMD2			O
								CKDIR	Internal / external clock select bit	0 : Internal clock 1 : External clock (Note)	O
								STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	O
								PRY	Odd / even parity select bit	Valid when bit 6 = "1" 0 : Odd parity 1 : Even parity	O
								PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	O
								SLEP	Sleep select bit	0 : Sleep mode deselected 1 : Sleep mode selected	O

Note: Set a corresponding direction register to "0."

## UART2 transmit / receive mode register

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	When reset	
								U2MR	0378 <sub>16</sub>	00 <sub>16</sub>	
								Bit symbol	Bit name	Function	R/W
								SMD0	Serial I/O mode select bit	b2 b1 b0 1 0 0 : Transfer data 7 bits length 1 0 1 : Transfer data 8 bits length 1 1 0 : Transfer data 9 bits length	O
								SMD1			O
								SMD2			O
								CKDIR	Internal / external clock select bit	Must always be fixed to "0"	O
								STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	O
								PRY	Odd / even parity select bit	Valid when bit 6 = "1" 0 : Odd parity 1 : Even parity	O
								PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	O
								IOPOL	TxD, RxD I/O polarity reverse bit (Note)	0 : No reverse 1 : Reverse	O

Note: Usually set to "0".

Figure 2.11.16 UARTi transmit/receive mode register in UART mode

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

Table 2.11.7 lists the functions of the input/output pins during UART mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs a "H". (If the N-channel open-drain is selected, this pin is in floating state.)

**Table 2.11.7 Input/output pin functions in UART mode**

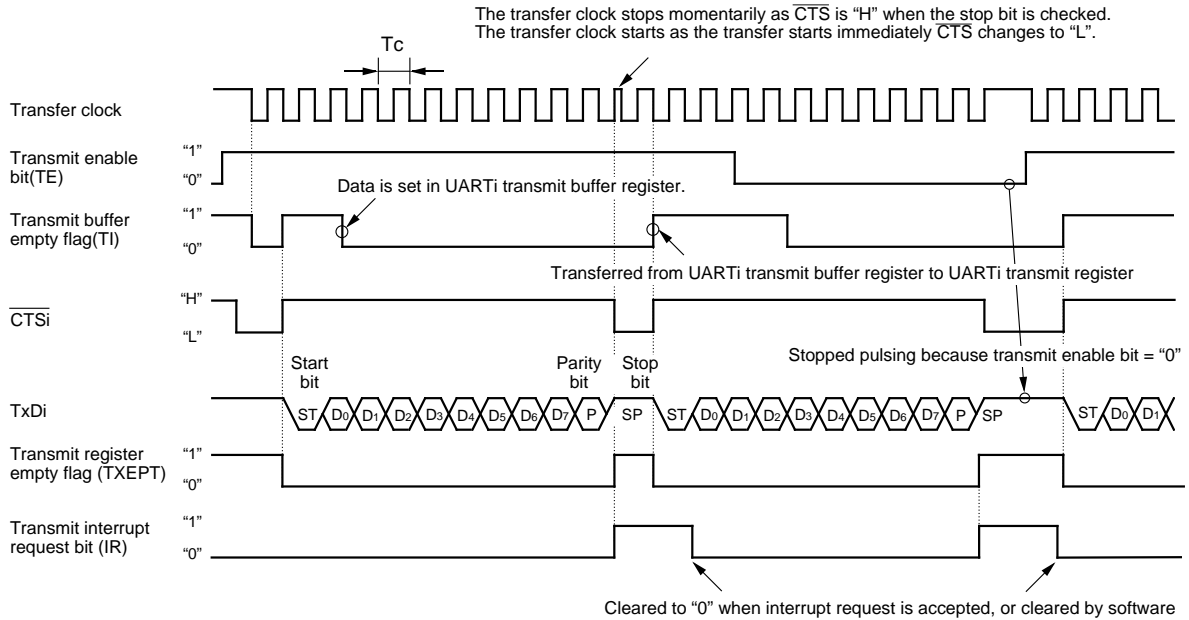
Pin name	Function	Method of selection
TxDi (P63, P67, P70)	Serial data output	
RxDi (P62, P66, P71)	Serial data input	Port P62, P66 and P71 direction register (bits 2 and 6 at address 03EE16, bit 1 at address 03EF16) = "0" (Can be used as an input port when performing transmission only)
CLKi (P61, P65, P72)	Programmable I/O port	Internal/external clock select bit (bit 3 at address 03A016, 03A816, 037816) = "0"
	Transfer clock input	Internal/external clock select bit (bit 3 at address 03A016, 03A816) = "1" Port P61, P65 direction register (bits 1 and 5 at address 03EE16) = "0" (Do not set external clock for UART2)
CTS <sub>i</sub> /RTS <sub>i</sub> (P60, P64, P73)	CTS input	CTS/RTS disable bit (bit 4 at address 03A416, 03AC16, 037C16) = "0" CTS/RTS function select bit (bit 2 at address 03A416, 03AC16, 037C16) = "0" Port P60, P64 and P73 direction register (bits 0 and 4 at address 03EE16, bit 3 at address 03EF16) = "0"
	RTS output	CTS/RTS disable bit (bit 4 at address 03A416, 03AC16, 037C16) = "0" CTS/RTS function select bit (bit 2 at address 03A416, 03AC16, 037C16) = "1"
	Programmable I/O port	CTS/RTS disable bit (bit 4 at address 03A416, 03AC16, 037C16) = "1"



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

• 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".

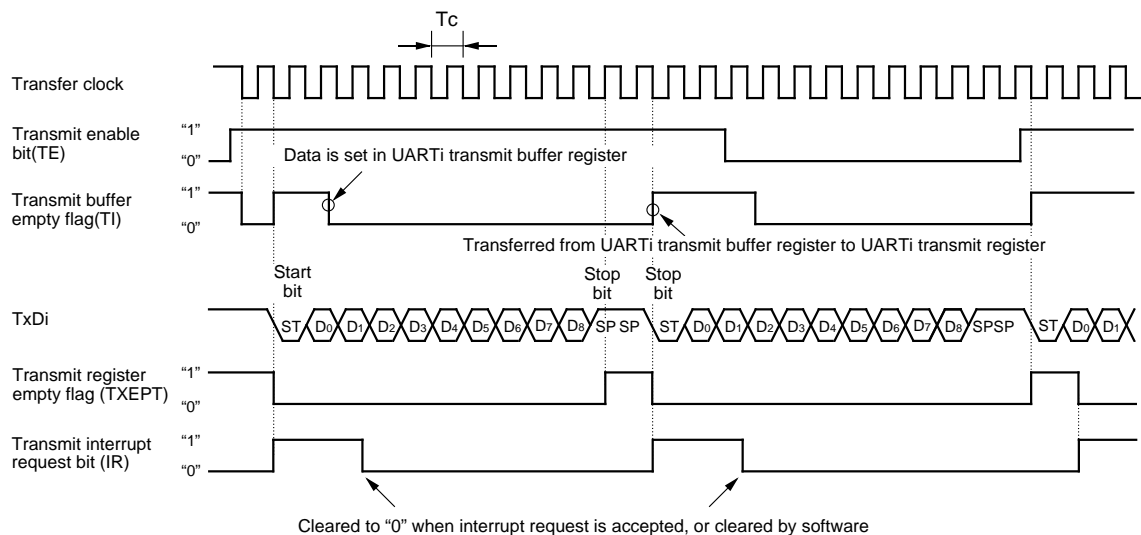
$$T_c = 16(n + 1) / f_i \text{ or } 16(n + 1) / f_{EXT}$$

$f_i$  : frequency of BRGi count source ( $f_1, f_8, f_{32}$ )

$f_{EXT}$  : 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".

$$T_c = 16(n + 1) / f_i \text{ or } 16(n + 1) / f_{EXT}$$

$f_i$  : frequency of BRGi count source ( $f_1, f_8, f_{32}$ )

$f_{EXT}$  : frequency of BRGi count source (external clock)

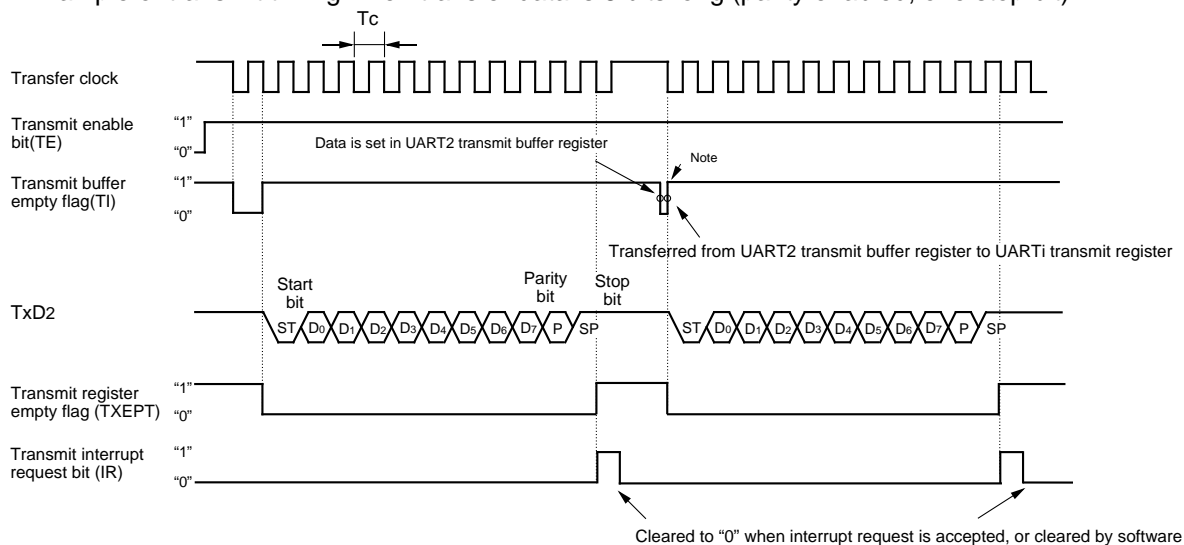
$n$  : value set to BRGi

Figure 2.11.17 Typical transmit timings in UART mode(UART0,UART1)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

- 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".

$T_c = 16 (n + 1) / f_i$   
 $f_i$  : frequency of BRG2 count source ( $f_1, f_8, f_{32}$ )  
 $n$  : value set to BRG2

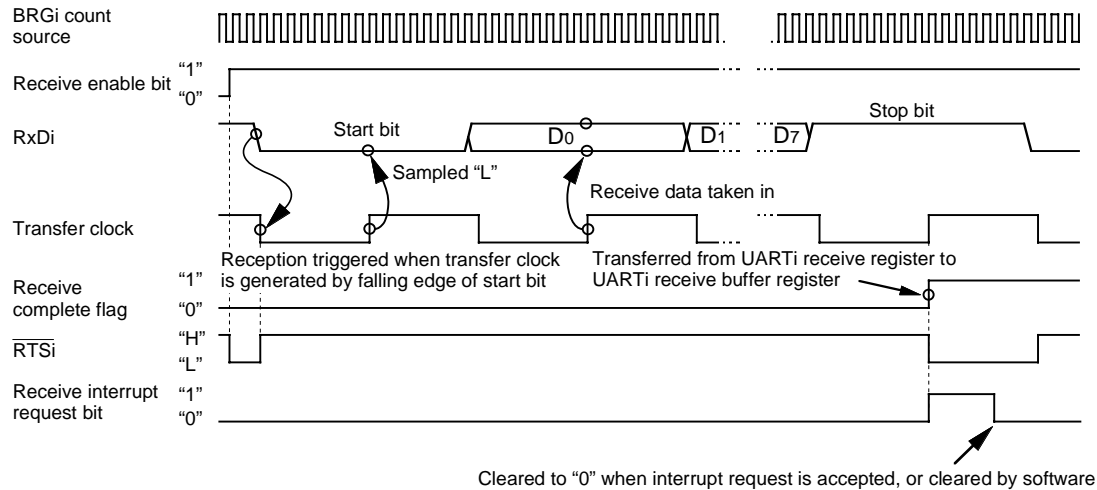
Note: The transmit is started with overflow timing of BRG after having written in a value at the transmit buffer in the above timing.

Figure 2.11.18 Typical transmit timings in UART mode(UART2)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

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



The above timing applies to the following settings :  
•Parity is disabled.  
•One stop bit.

Figure 2.11.19 Typical receive timing in UART mode

## (1) 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<sub>16</sub>, 03A8<sub>16</sub>) 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".

## (2) Function for switching serial data logic (UART2)

When the data logic select bit (bit 6 of address 037D<sub>16</sub>) is assigned 1, data is inverted in writing to the transmission buffer register or reading the reception buffer register. Figure 2.11.20 shows the example of timing for switching serial data logic.

- When LSB first, parity enabled, one stop bit

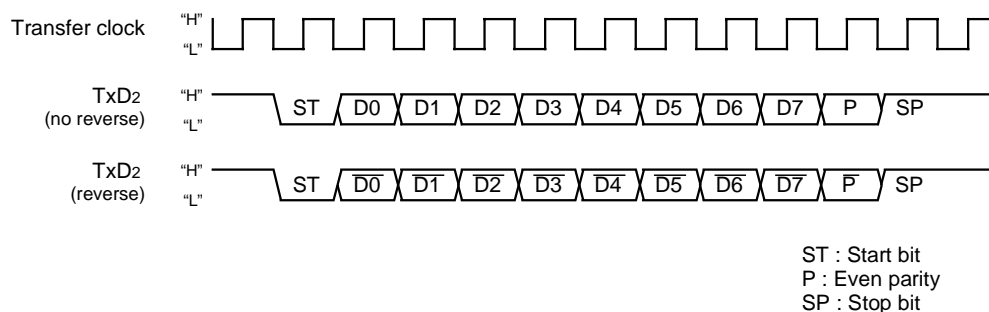


Figure 2.11.20 Timing for switching serial data logic

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

### (3) 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.

### (4) 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 2.11.21 shows the example of detection timing of a buss collision (in UART mode).

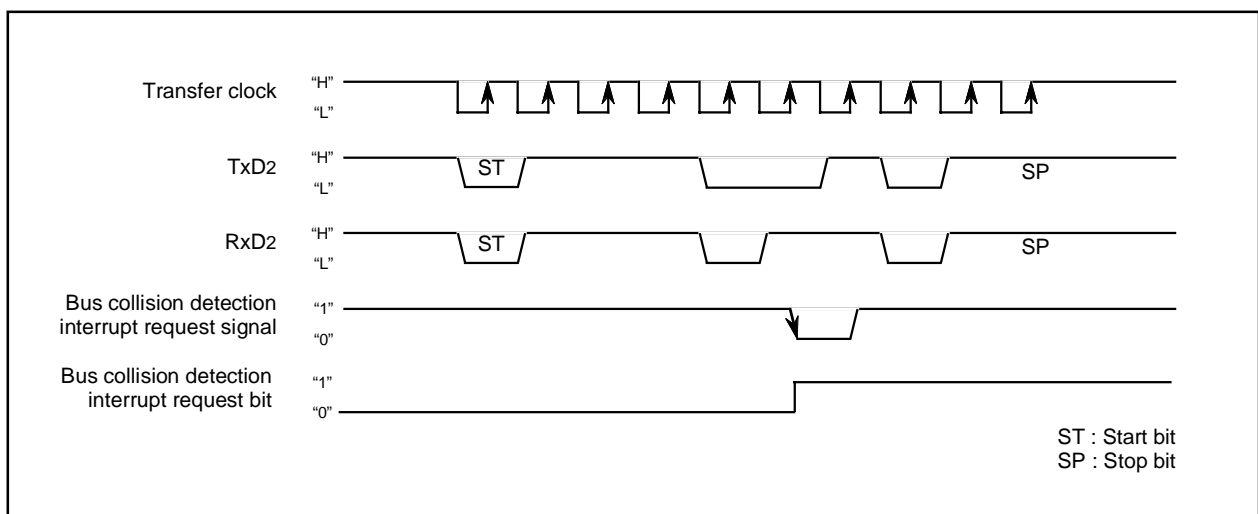


Figure 2.11.21 Detection timing of a bus collision (in UART mode)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

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

The SIM interface is used for connecting the microcomputer with a memory card or the like; adding some extra settings in UART2 clock-asynchronous serial I/O mode allows the user to effect this function. Table 2.11.8 shows the specifications of clock-asynchronous serial I/O mode (compliant with the SIM interface).

**Table 2.11.8 Specifications of clock-asynchronous serial I/O mode (compliant with the SIM interface)**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>Transfer data 8-bit UART mode (bit 2 through bit 0 of address 0378<sub>16</sub> = "1012")</li> <li>One stop bit (bit 4 of address 0378<sub>16</sub> = "0")</li> <li>With the direct format chosen <ul style="list-style-type: none"> <li>Set parity to "even" (bit 5 and bit 6 of address 0378<sub>16</sub> = "1" and "1" respectively)</li> <li>Set data logic to "direct" (bit 6 of address 037D<sub>16</sub> = "0").</li> <li>Set transfer format to LSB (bit 7 of address 037C<sub>16</sub> = "0").</li> </ul> </li> <li>With the inverse format chosen <ul style="list-style-type: none"> <li>Set parity to "odd" (bit 5 and bit 6 of address 0378<sub>16</sub> = "0" and "1" respectively)</li> <li>Set data logic to "inverse" (bit 6 of address 037D<sub>16</sub> = "1")</li> <li>Set transfer format to MSB (bit 7 of address 037C<sub>16</sub> = "1")</li> </ul> </li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>With the internal clock chosen (bit 3 of address 0378<sub>16</sub> = "0") : <math>f_i / 16 (n + 1)</math> (Note 1) : <math>f_i = f_1, f_8, f_{32}</math> (Do not set external clock)</li> </ul>
Transmission / reception control	<ul style="list-style-type: none"> <li>Disable the <math>\overline{\text{CTS}}</math> and <math>\overline{\text{RTS}}</math> function (bit 4 of address 037C<sub>16</sub> = "1")</li> </ul>
Other settings	<ul style="list-style-type: none"> <li>The sleep mode select function is not available for UART2</li> <li>Set transmission interrupt factor to "transmission completed" (bit 4 of address 037D<sub>16</sub> = "1")</li> </ul>
Transmission start condition	<ul style="list-style-type: none"> <li>To start transmission, the following requirements must be met: <ul style="list-style-type: none"> <li>Transmit enable bit (bit 0 of address 037D<sub>16</sub>) = "1"</li> <li>Transmit buffer empty flag (bit 1 of address 037D<sub>16</sub>) = "0"</li> </ul> </li> </ul>
Reception start condition	<ul style="list-style-type: none"> <li>To start reception, the following requirements must be met: <ul style="list-style-type: none"> <li>Reception enable bit (bit 2 of address 037D<sub>16</sub>) = "1"</li> <li>Detection of a start bit</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>When transmitting <ul style="list-style-type: none"> <li>When data transmission from the UART2 transfer register is completed (bit 4 of address 037D<sub>16</sub> = "1")</li> </ul> </li> <li>When receiving <ul style="list-style-type: none"> <li>When data transfer from the UART2 receive register to the UART2 receive buffer register is completed</li> </ul> </li> </ul>
Error detection	<ul style="list-style-type: none"> <li>Overrun error (see the specifications of clock-asynchronous serial I/O) (Note 2)</li> <li>Framing error (see the specifications of clock-asynchronous serial I/O)</li> <li>Parity error (see the specifications of clock-asynchronous serial I/O) <ul style="list-style-type: none"> <li>On the reception side, an "L" level is output from the TxD<sub>2</sub> pin by use of the parity error signal output function (bit 7 of address 037D<sub>16</sub> = "1") when a parity error is detected</li> <li>On the transmission side, a parity error is detected by the level of input to the RxD<sub>2</sub> pin when a transmission interrupt occurs</li> </ul> </li> <li>The error sum flag (see the specifications of clock-asynchronous serial I/O)</li> </ul>

Note 1: 'n' denotes the value 00<sub>16</sub> to FF<sub>16</sub> that is set to the UARTi 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 does not change.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

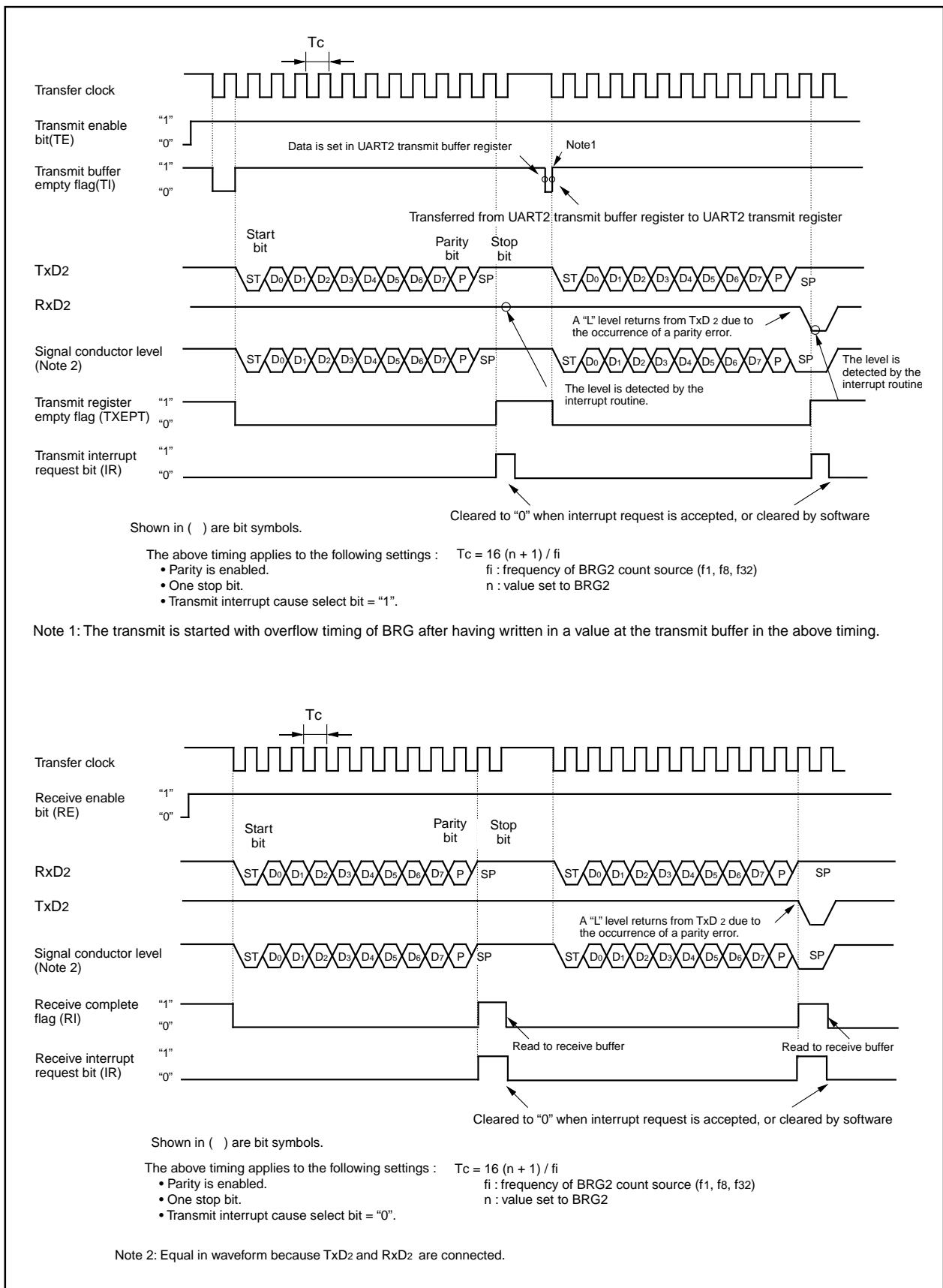


Figure 2.11.22 Typical transmit/receive timing in UART mode (compliant with the SIM interface)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

**(1) Parity error signal output function**

If a parity error is detected when the error signal output enable bit (address 037D16, bit 7) has been set to “1”, a low-level signal can be output from the TxD2 pin. Also, when operating in transmit mode, a transmit-complete interrupt is generated a half transfer clock cycle later than when the error signal output enable bit (address 037D16, bit 7) is set to “0”. Therefore, a parity error signal can be detected in the transmit-complete interrupt program. Figure 2.11.23 shows the timing at which a parity error signal is output.

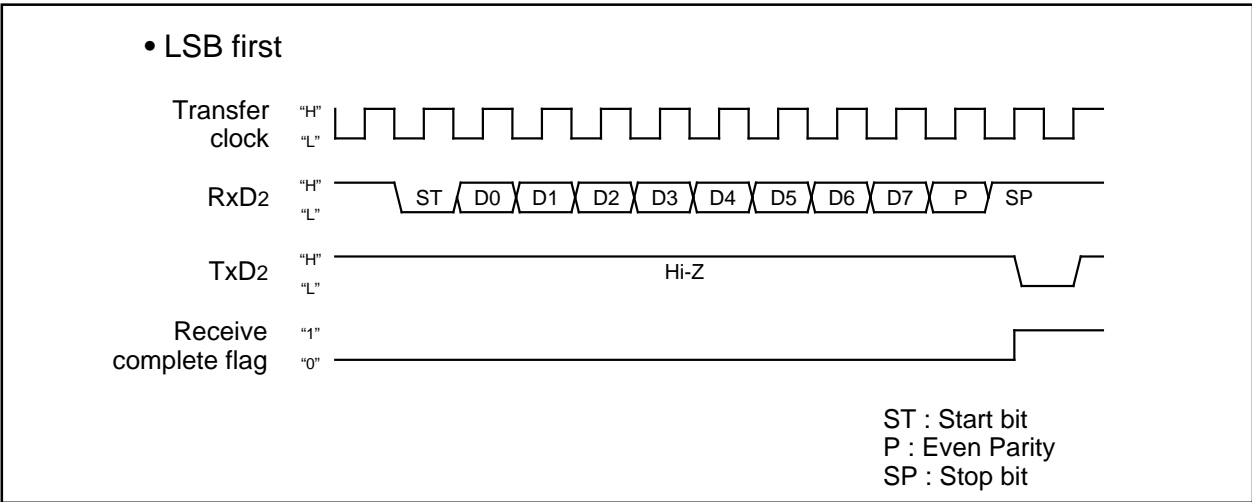


Figure 2.11.23 Output timing of the parity error signal

**(2) 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 2.11.24 shows the SIM interface format.

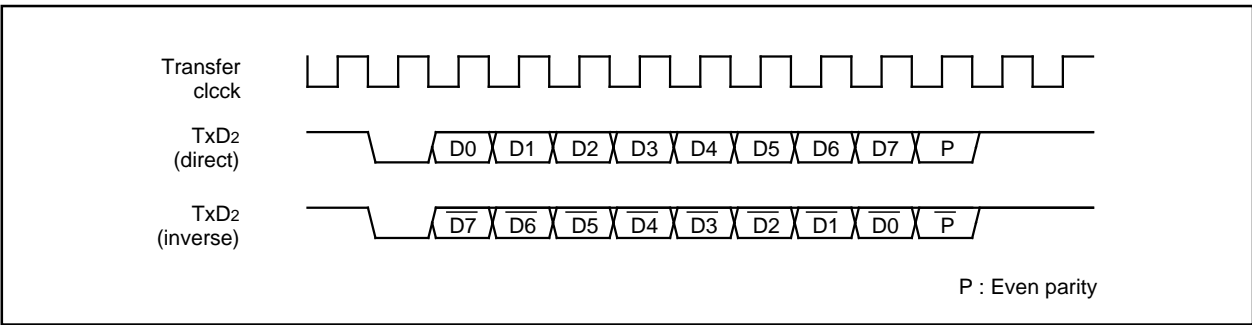


Figure 2.11.24 SIM interface format

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

Figure 2.11.25 shows the example of connecting the SIM interface. Connect TxD2 and RxD2 and apply pull-up.

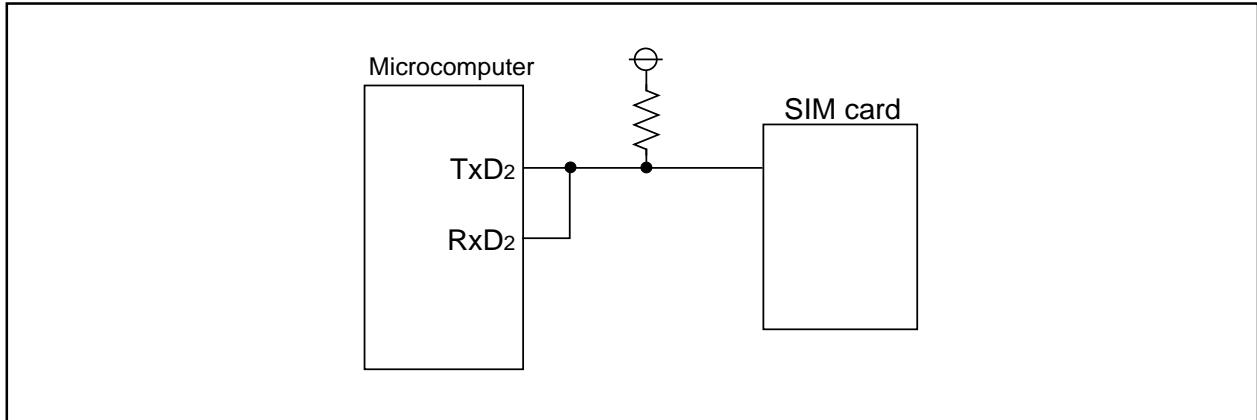


Figure 2.11.25 Connecting the SIM interface



### 2.11.5 UART2 Special Mode Register

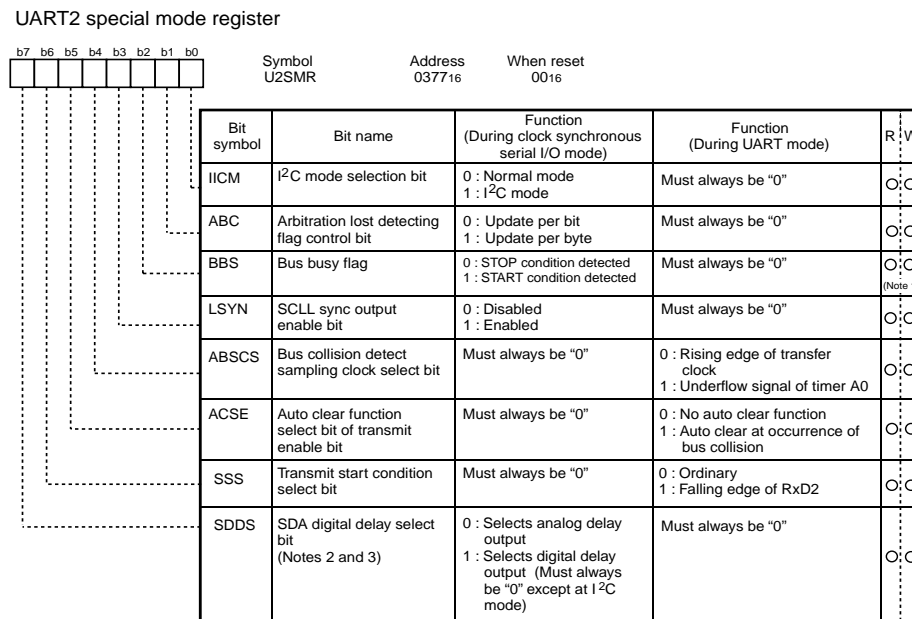
The UART2 special mode register (address 0377<sub>16</sub>) is used to control UART2 in various ways.

Figure 2.11.26 shows the UART2 special mode register.

Bit 0 of the UART special mode register (0377<sub>16</sub>) is used as the I<sup>2</sup>C mode selection bit.

Setting “1” in the I<sup>2</sup>C mode select bit (bit 0) goes the circuit to achieve the I<sup>2</sup>C bus interface effective.

Since this function uses clock-synchronous serial I/O mode, set this bit to “0” in UART mode.

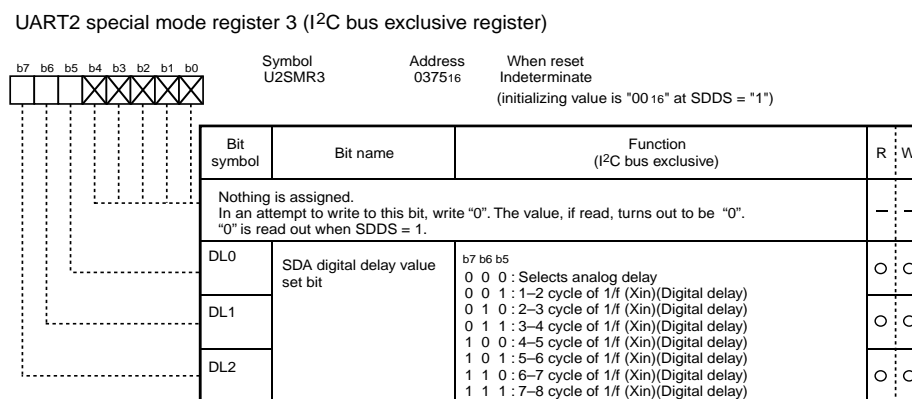


Notes 1: Nothing but "0" may be written.

2: Do not write "1" except at 1<sup>2</sup>C mode. Must always be "0" at normal mode.

Bit 7 to bit5 (DL2 to DL0 = SDA digital delay value setting bit) of UART2 special mode register 3 (U2SMR3/address 0375<sub>16</sub>) are initialized and become "000" when this bit is "0", analog delay circuit is selected. Reading and writing U2SMR are enable when SDDS = "0".

3: Delaying ; Only analog delay value when analog delay is selected, and only digital delay value when digital delay is selected.



**Notes 1:** Reading and writing is possible when bit7 (SDDS = SDA digital delay selection bit) of UART2 special mode register (U2SMR/address 0377 16) is "1". When set SDDS = "1" and read out initialized value of UART2 special mode register 3(U2SMR3), this value is "00 16".When set SDDS = "1" and write to UART2 special mode register 3(U2SMR3), set "0" to bit 0 to bit 4. When SDDS = "0", writing is enable. When read out, this value is indeterminate.

2: When SDDS = "0", this bit is initialized and become "000", selected analog delay circuit. This bit is become "000" after end reset released, and selected analog delay circuit. Reading out is possible when only SDDS = "1". when SDDS = "0", value which was read out is indeterminate.

3: Delaying ; Only analog delay value when analog delay is selected, and only digital delay value when digital delay is selected.

4: Delay level depends on SCL pin and SDA pin. And, when use external clock, delay is increase around 100ns. So test first, and use this.

Figure 2.11.26 UART2 special mode register

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

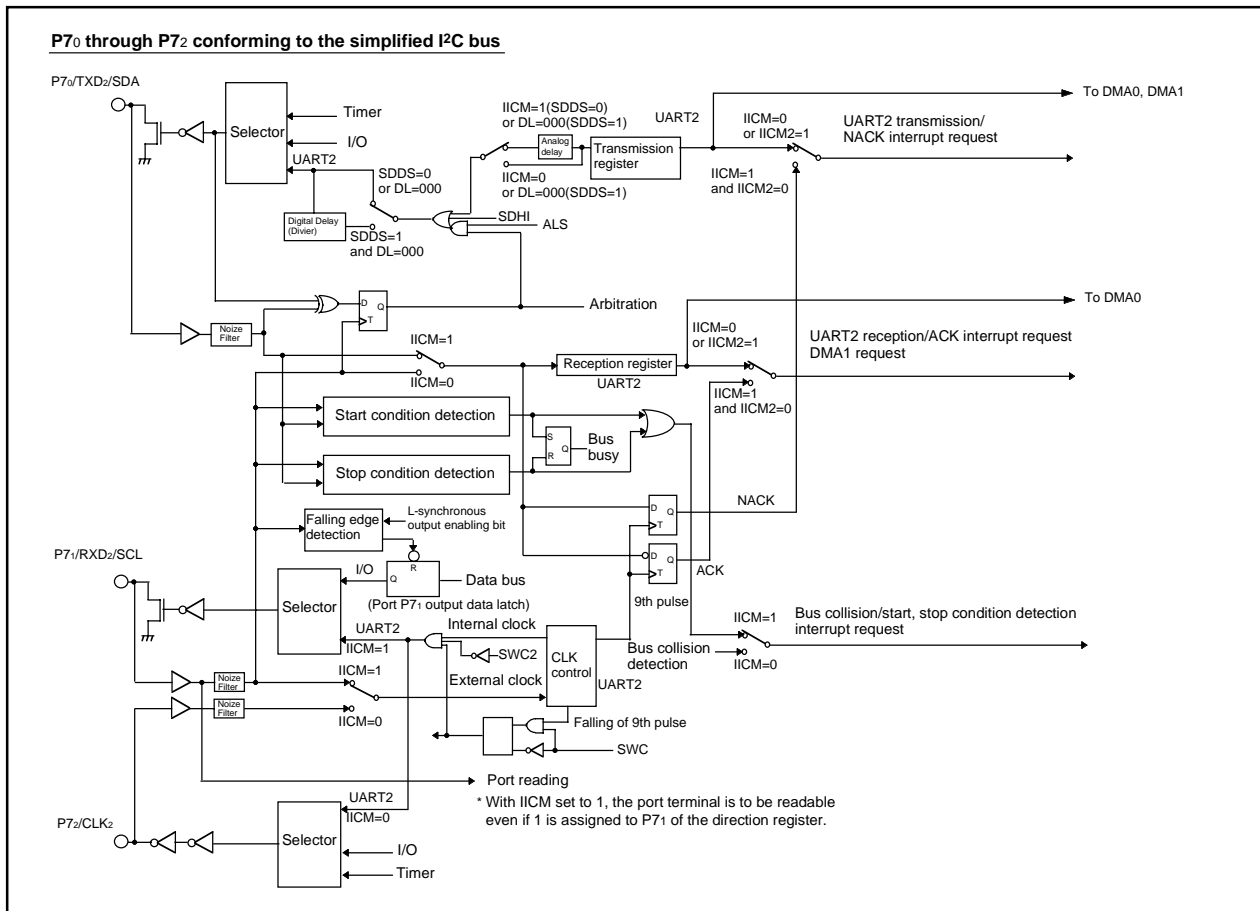


Figure 2.11.27 Functional block diagram for I<sup>2</sup>C mode

Table 2.11.9 Features in I<sup>2</sup>C mode

	Function	Normal mode	I <sup>2</sup> C 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(Digital / analog selection is possible)
5	P7 <sub>0</sub> at the time when UART2 is in use	TxD <sub>2</sub> (output)	SDA (input/output) (Note 3)
6	P7 <sub>1</sub> at the time when UART2 is in use	RxD <sub>2</sub> (input)	SCL (input/output)
7	P7 <sub>2</sub> at the time when UART2 is in use	CLK <sub>2</sub>	P7 <sub>2</sub>
8	DMA1 factor at the time when 1 1 0 1 is assigned to the DMA request factor selection bits	UART2 reception	Acknowledgment detection (ACK)
9	Noise filter width	15ns	50ns
10	Reading P7 <sub>1</sub>	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 <sub>0</sub> when the port is selected

Note 1: Make the settings given below when I<sup>2</sup>C mode is in use.

Set 0 1 0 in bits 2, 1, 0 of the UART2 transmission/reception mode register.

Disable the RTS/CTS function. 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.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

Figure 2.11.27 shows the functional block diagram for I<sup>2</sup>C mode. Setting "1" in the I<sup>2</sup>C mode selection bit (IICM) causes ports P70, P71, and P72 to work as data transmission-reception terminal SDA, clock input-output terminal SCL, and port P72 respectively. A delay circuit is added to the SDA transmission output, so the SDA output changes after SCL fully goes to "L". Can select analog delay or digital delay by SDA digital delay selection bit (7 bit of address 0377<sub>16</sub>). When select digital delay, can select delay to 2 cycle to 8 cycle of f<sub>1</sub> by UART2 special mode register 3 (address 0375<sub>16</sub>). Functions changed by I<sup>2</sup>C mode selection bit 2 is shown in below.

**Table 2.11.10 Delay circuit selection condition**

	Register value			Contents
	IICM	SDDS	DL	
Digital delay selection	1	1	001 to 111	When select digital delay, analog delay is not added. Only digital delay.
Analog delay selection	1	1	000	When select DL="000", analog delay is chosen regardless of the value of SDDS.
		0	(000)	When SDDS="0", DL is initialized and DL="000".
No delay	0	0	(000)	Delay circuit is not selected when IICM="0". But, must set SDDS="0" when IICM="0".

An attempt to read Port P71 (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 P70. 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 (P70) is detected with the SCL terminal (P71) staying "H". The stop condition detection interrupt refers to the interrupt that occurs when the rising edge of the SDA terminal (P70) is detected with the SCL terminal (P71) staying "H". The bus busy flag (bit 2 of the UART2 special 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 1101(UART2 reception) to the DMA1 request factor select bits provides the means to start up the DMA transfer by the effect of acknowledgment detection.

Bit 1 of the UART2 special mode register (0377<sub>16</sub>) is used as the arbitration loss 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 11 of the UART2 reception buffer register (037F<sub>16</sub>, 037E<sub>16</sub>), 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 update 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 enable bit. Setting this bit to "1" goes the P71 data register to "0" in synchronization with the SCL terminal level going to "L".

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

Some other functions added are explained here. Figure 2.11.28 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 transmit start condition select bit. Setting this bit to "1" starts the TxD transmission in synchronization with the falling edge of the RxD terminal.

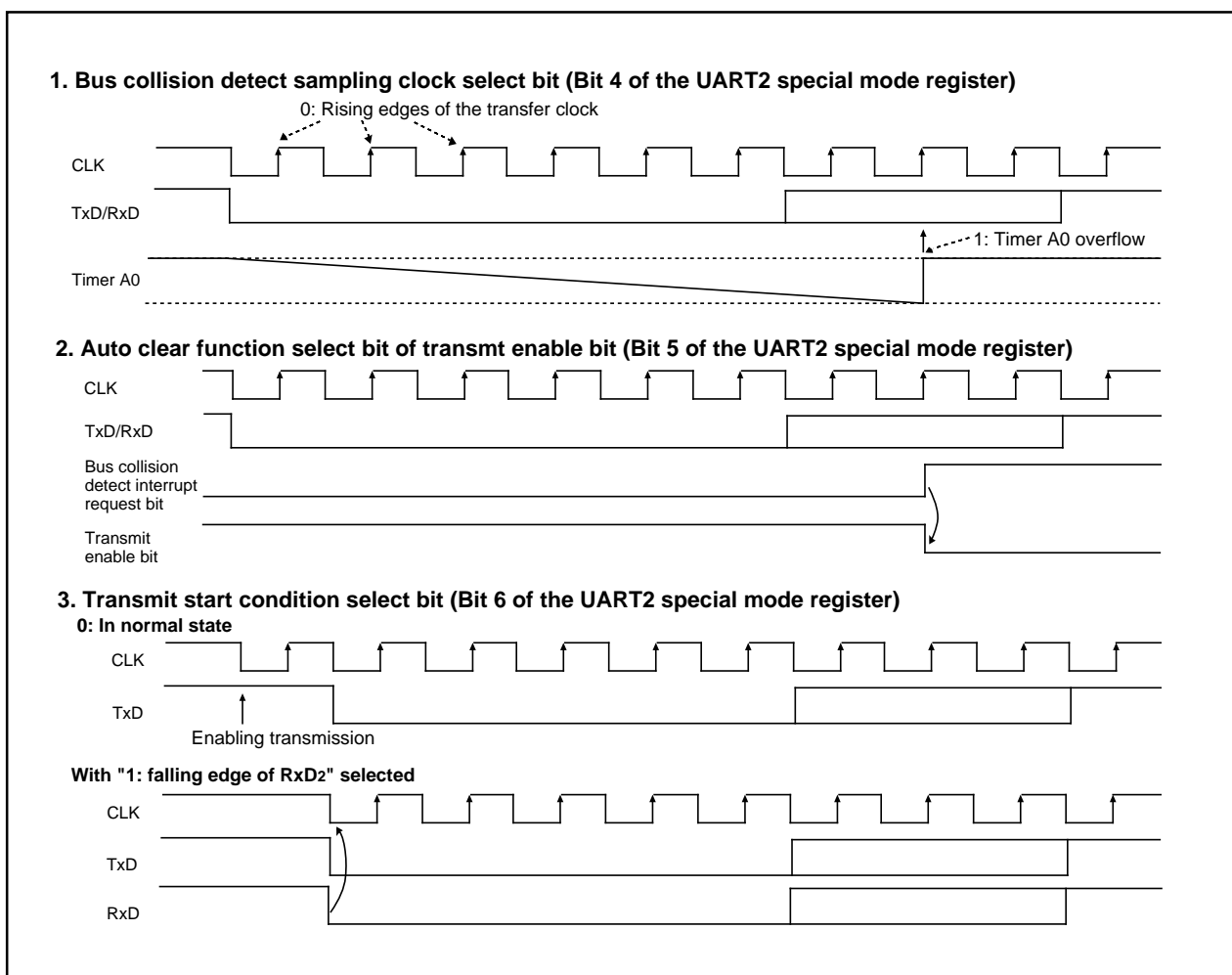


Figure 2.11.28 Some other functions added

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.11.6 UART2 Special Mode Register 2

UART2 special mode register 2 (address 0376<sub>16</sub>) is used to further control UART2 in I<sup>2</sup>C mode. Figure 2.11.29 shows the UART2 special mode register 2.

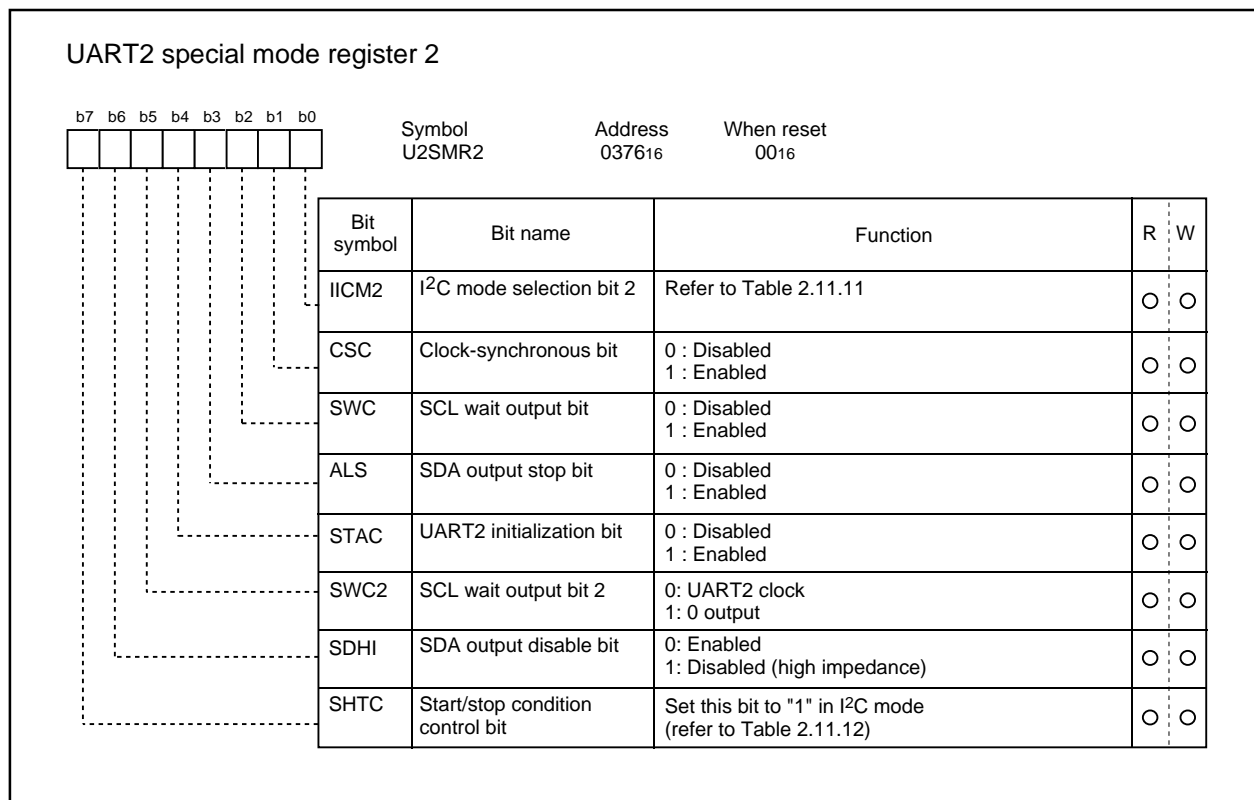


Figure 2.11.29 UART2 special mode register 2

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

Bit 0 of the UART2 special mode register 2 (address 0376<sub>16</sub>) is used as the I<sup>2</sup>C mode selection bit 2. Table 2.11.11 shows the types of control to be changed by I<sup>2</sup>C mode selection bit 2 when the I<sup>2</sup>C mode selection bit is set to "1". Table 2.11.12 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<sup>2</sup>C mode.

**Table 2.11.11 Functions changed by I<sup>2</sup>C mode selection bit 2**

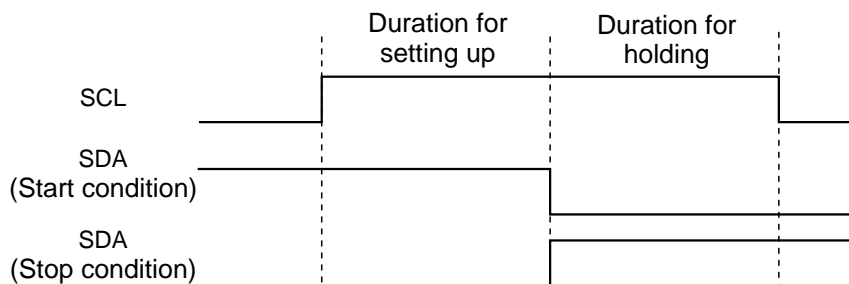
	Function	IICM2 = 0	IICM2 = 1
1	Factor of interrupt number 15	No acknowledgment detection (NACK)	UART2 transmission (the rising edge of the final bit of the clock)
2	Factor of interrupt number 16	Acknowledgment detection (ACK)	UART2 reception (the falling edge of the final bit of the clock)
3	DMA1 factor at the time when 1 1 0 1 is assigned to the DMA request factor selection 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 2.11.12 Timing characteristics of detecting the start condition and the stop condition(Note1)**

3 to 6 cycles < duration for setting-up (Note2)
3 to 6 cycles < duration for holding (Note2)

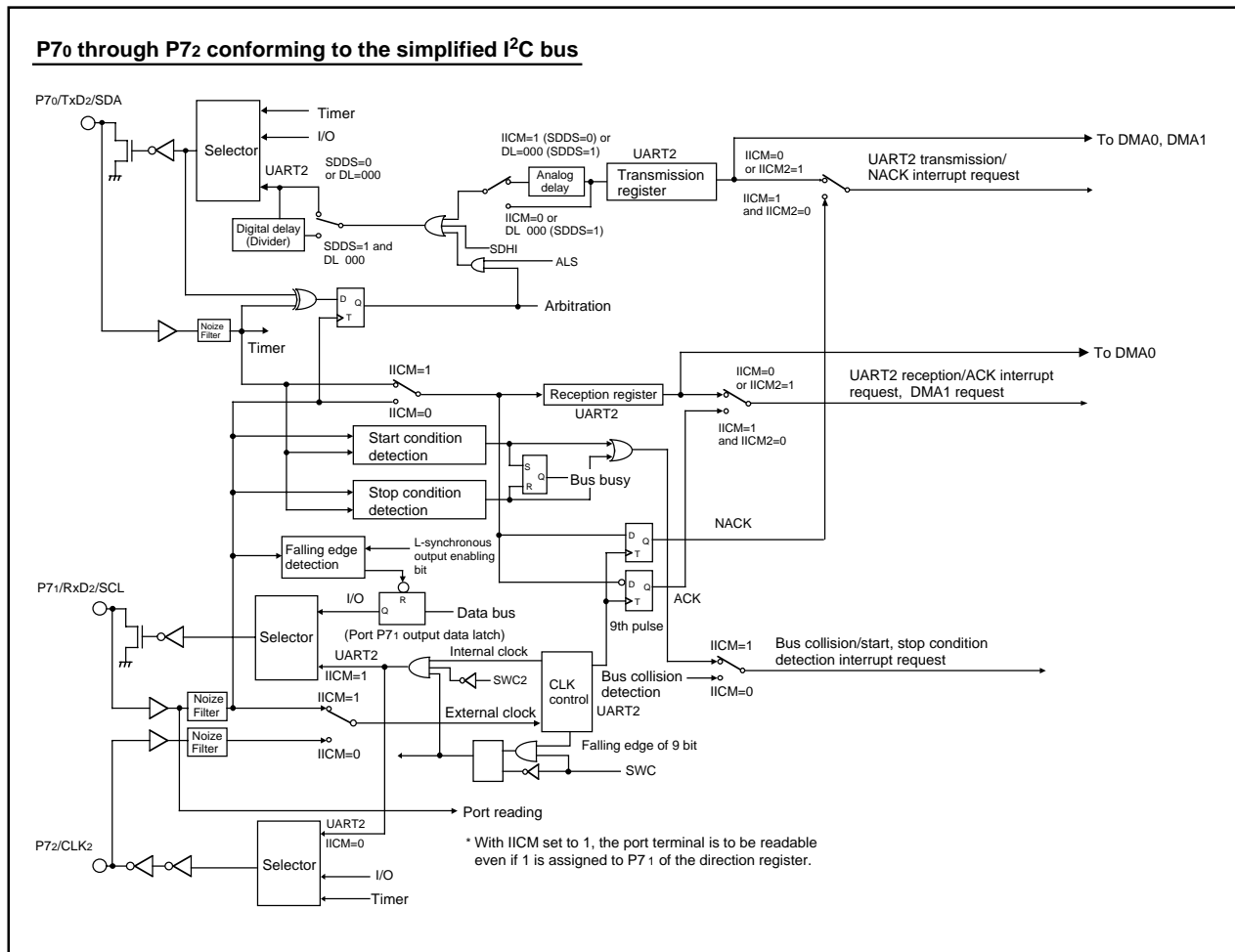
Note 1 : When the start/stop condition count bit is "1".

Note 2 : "cycles" is in terms of the input oscillation frequency f(XIN) of the main clock.



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER



**Figure 2.11.30 Functional block diagram for I<sup>2</sup>C mode**

Functions available in I<sup>2</sup>C mode are shown in Figure 2.11.30—a functional block diagram.

Bit 3 of the UART2 special mode register 2 (address 0376<sub>16</sub>) 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 loss detection flag is set to "1".

Bit 1 of the UART2 special mode register 2 (address 0376<sub>16</sub>) 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 (0376<sub>16</sub>) 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".

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

Bit 4 of the UART2 special mode register 2 (address 037616) 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, doesn't 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 doesn't change the value of the transmission buffer empty flag. To use this function, choose the external clock for the transfer clock. Bit 5 of the UART2 special mode register 2 (037616) 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 UART2 special mode register 2 (037616) 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 detection flag is turned on.



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.11.7 S I/O3, 4

S I/O3 and S I/O4 are exclusive clock-synchronous serial I/Os.

Figure 2.11.31 shows the S I/O3, 4 block diagram, and Figure 2.11.32 shows the S I/O3, 4 control register. Table 2.11.13 shows the specifications of S I/O3, 4.

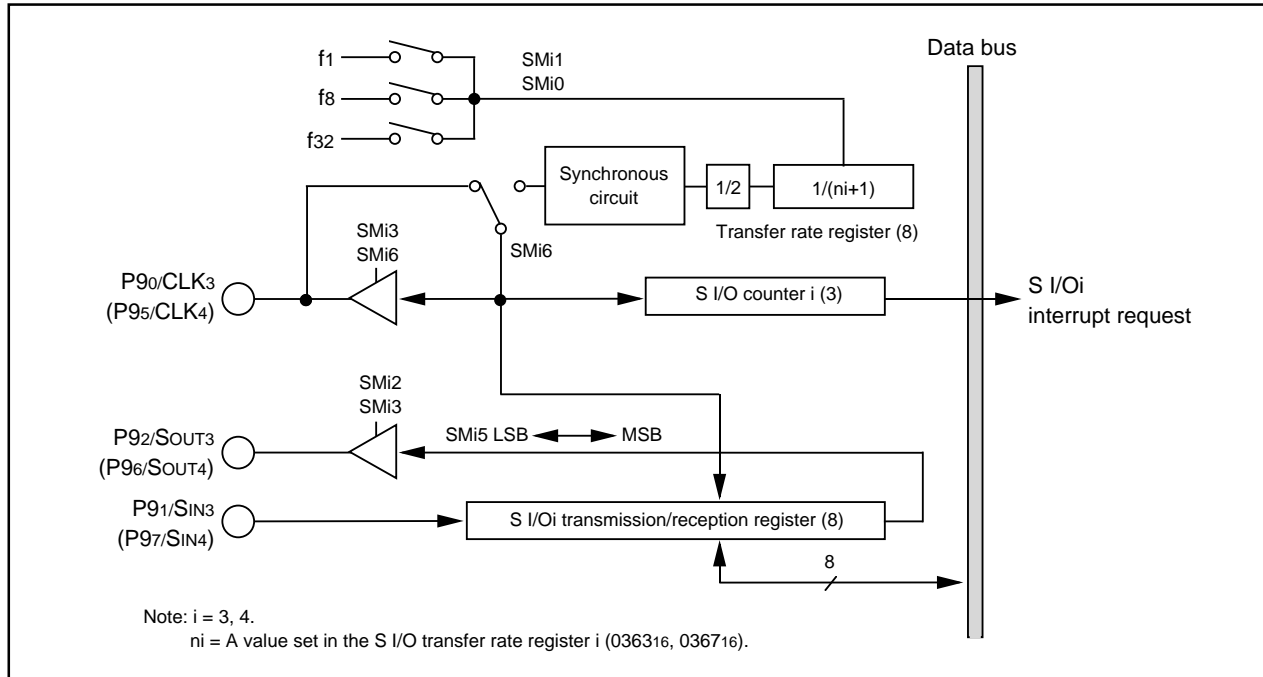
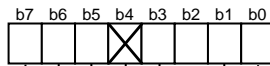


Figure 2.11.31 S I/O3, 4 block diagram

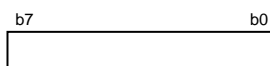
**M306H2FCFP**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER**S I/O<sub>i</sub> control register (i = 3, 4) (Note 1)**

Symbol	Address	When reset
SiC	0362 <sub>16</sub> , 0366 <sub>16</sub>	40 <sub>16</sub>

Bit symbol	Bit name	Description	R	W
SMi0	Internal synchronous clock select bit	b1 b0 0 0 : Selecting f <sub>1</sub>	<input type="radio"/>	<input type="radio"/>
SMi1		0 1 : Selecting f <sub>8</sub>	<input type="radio"/>	<input type="radio"/>
		1 0 : Selecting f <sub>32</sub>	<input type="radio"/>	<input type="radio"/>
		1 1 : Not to be used	<input type="radio"/>	<input type="radio"/>
SMi2	Sout <sub>i</sub> output disable bit	0 : Sout <sub>i</sub> output 1 : Sout <sub>i</sub> output disable (high impedance)	<input type="radio"/>	<input type="radio"/>
SMi3	S I/O <sub>i</sub> port select bit (Note 2)	0 : Input-output port 1 : Sout <sub>i</sub> output, CLK function	<input type="radio"/>	<input type="radio"/>
	Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be "0".		—	—
SMi5	Transfer direction select bit	0 : LSB first 1 : MSB first	<input type="radio"/>	<input type="radio"/>
SMi6	Synchronous clock select bit (Note 2)	0 : External clock 1 : Internal clock	<input type="radio"/>	<input type="radio"/>
SMi7	Sout <sub>i</sub> initial value set bit	Effective when SMi3 = 0 0 : L output 1 : H output	<input type="radio"/>	<input type="radio"/>

Note 1: Set "1" in bit 2 of the protection register (000A<sub>16</sub>) in advance to write to the S I/O<sub>i</sub> control register (i = 3, 4).

Note 2: When using the port as an input/output port by setting the S I/O<sub>i</sub> port select bit (i = 3, 4) to "1", be sure to set the sync clock select bit to "1".

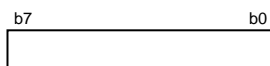
**S I/O<sub>i</sub> bit rate generator (Note 1, Note 2)**

Symbol	Address	When reset
S3BRG	0363 <sub>16</sub>	Indeterminate
S4BRG	0367 <sub>16</sub>	Indeterminate

Indeterminate	Values that can be set	R	W
Assuming that set value = n, BRG <sub>i</sub> divides the count source by n + 1	00 <sub>16</sub> to FF <sub>16</sub>	<input type="radio"/>	<input type="radio"/>

Note 1: Write a value to this register while transmit/receive halts.

Note 2: Use MOV instruction to write to this register.

**S I/O<sub>i</sub> transmit/receive register (Note)**

Symbol	Address	When reset
S3TRR	0360 <sub>16</sub>	Indeterminate
S4TRR	0364 <sub>16</sub>	Indeterminate

Indeterminate	R	W
Transmission/reception starts by writing data to this register. After transmission/reception finishes, reception data is input.	<input type="radio"/>	<input type="radio"/>

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

**Figure 2.11.32 S I/O<sub>3, 4</sub> related register**

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

**Table 2.11.13 Specifications of S I/O3, 4**

Item	Specifications
Transfer data format	<ul style="list-style-type: none"> <li>Transfer data length: 8 bits</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>With the internal clock selected (bit 6 of 0362<sub>16</sub>, 0366<sub>16</sub> = "1"): <math>f_{1/2(ni+1)}</math>, <math>f_{8/2(ni+1)}</math>, <math>f_{32/2(ni+1)}</math> (Note 1)</li> <li>With the external clock selected (bit 6 of 0362<sub>16</sub>, 0366<sub>16</sub> = 0): Input from the CLK<sub>i</sub> terminal (Note 2)</li> </ul>
Conditions for transmission/reception start	<ul style="list-style-type: none"> <li>To start transmit/reception, the following requirements must be met: <ul style="list-style-type: none"> <li>Select the synchronous clock (use bit 6 of 0362<sub>16</sub>, 0366<sub>16</sub>).</li> <li>Select a frequency dividing ratio if the internal clock has been selected (use bits 0 and 1 of 0362<sub>16</sub>, 0366<sub>16</sub>).</li> <li>SOUT<sub>i</sub> initial value set bit (use bit 7 of 0362<sub>16</sub>, 0366<sub>16</sub>) = 1.</li> <li>S I/O<sub>i</sub> port select bit (bit 3 of 0362<sub>16</sub>, 0366<sub>16</sub>) = 1.</li> <li>Select the transfer direction (use bit 5 of 0362<sub>16</sub>, 0366<sub>16</sub>)</li> <li>Write transfer data to SI/O<sub>i</sub> transmit/receive register (0360<sub>16</sub>, 0364<sub>16</sub>)</li> </ul> </li> <li>To use S I/O<sub>i</sub> interrupts, the following requirements must be met: <ul style="list-style-type: none"> <li>Clear the SI/O<sub>i</sub> interrupt request bit before writing transfer data to the SI/O<sub>i</sub> transmit/receive register (bit 3 of 0049<sub>16</sub>, 0048<sub>16</sub>) = 0.</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>Rising edge of the last transfer clock. (Note 3)</li> </ul>
Select function	<ul style="list-style-type: none"> <li>LSB first or MSB first selection Whether transmission/reception begins with bit 0 (LSB) or bit 7 (MSB) can be selected.</li> <li>Function for setting an SOUT<sub>i</sub> initial value selection When using an external clock for the transfer clock, the user can choose the SOUT<sub>i</sub> pin output level during a non-transfer time. For details on how to set, see Figure 2.11.33.</li> </ul>
Precaution	<ul style="list-style-type: none"> <li>Unlike UART0–2, SI/O<sub>i</sub> (i = 3, 4) is not divided for transfer register and buffer. Therefore, do not write the next transfer data to the SI/O<sub>i</sub> transmit/receive register (addresses 0360<sub>16</sub>, 0364<sub>16</sub>) during a transfer. When the internal clock is selected for the transfer clock, SOUT<sub>i</sub> 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 SI/O<sub>i</sub> transmit/receive register (addresses 0360<sub>16</sub>, 0364<sub>16</sub>) during this time, SOUT<sub>i</sub> is placed in the high-impedance state immediately upon writing and the data hold time is thereby reduced.</li> </ul>

Note 1: n is a value from 00<sub>16</sub> through FF<sub>16</sub> set in the S I/O<sub>i</sub> transfer rate register (i = 3, 4).

Note 2: With the external clock selected:

- Before data can be written to the SI/O<sub>i</sub> transmit/receive register (addresses 0360<sub>16</sub>, 0364<sub>16</sub>), the CLK<sub>i</sub> pin input must be in the low state. Also, before rewriting the SI/O<sub>i</sub> Control Register (addresses 0362<sub>16</sub>, 0366<sub>16</sub>)'s bit 7 (SOUT<sub>i</sub> initial value set bit), make sure the CLK<sub>i</sub> pin input is held low.
- The S I/O<sub>i</sub> 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.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (1) Functions for setting an SOUTi initial value

When using an external clock for the transfer clock, the SOUTi pin output level during a non-transfer time can be set to the high or the low state. Figure 2.11.33 shows the timing chart for setting an SOUTi initial value and how to set it.

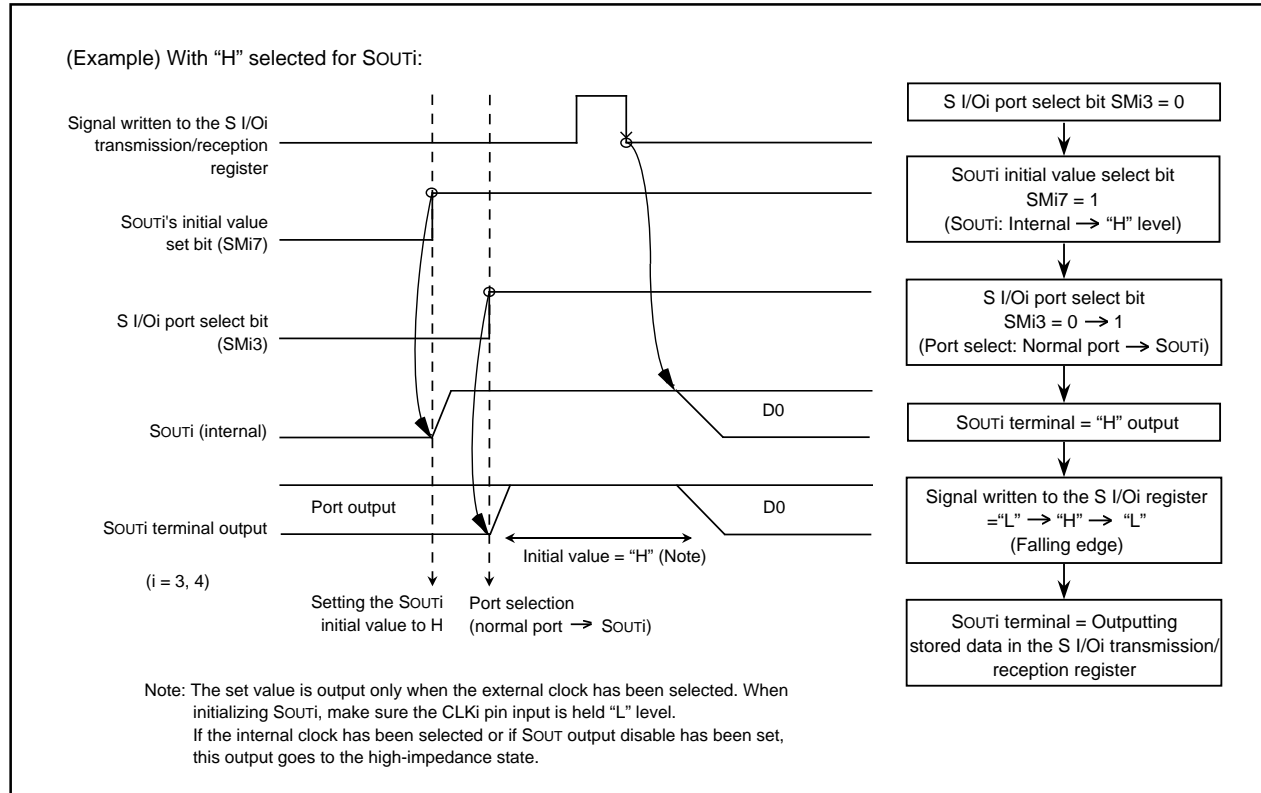


Figure 2.11.33 Timing chart for setting SOUTi's initial value and how to set it

## (2) S I/Oi operation timing

Figure 2.11.34 shows the S I/Oi operation timing

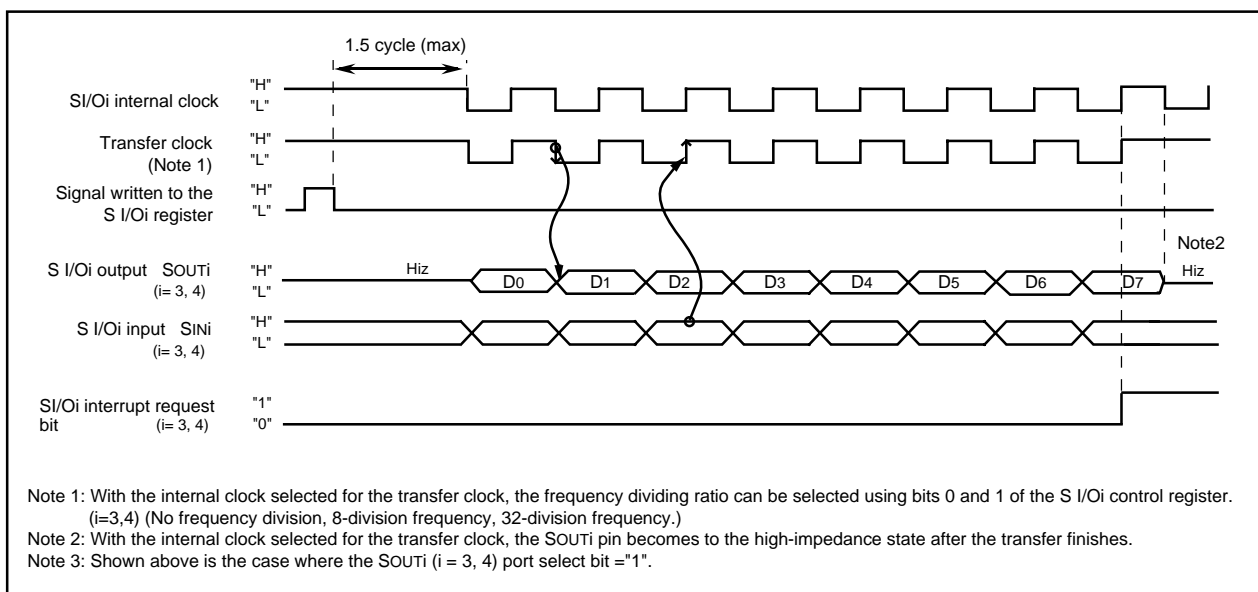


Figure 22.11.34 S I/Oi operation timing chart

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.12 A-D Converter

The A-D converter consists of one 8-bit successive approximation A-D converter circuit with a capacitive coupling amplifier. Pins P100 to P107, P95, and P96 also function as the analog signal input pins. The direction registers of these pins for A-D conversion must therefore be set to input. The Vref connect bit (bit 5 at address 03D716) can be used to isolate the resistance ladder of the A-D converter from the reference voltage input pin (VREF) when the A-D converter is not used. Doing so stops any current flowing into the resistance ladder from VREF, reducing the power dissipation. When using the A-D converter, start A-D conversion only after setting bit 5 of 03D716 to connect VREF.

The result of A-D conversion is stored in the A-D registers of the selected pins.

Table 2.12.1 shows the performance of the A-D converter. Figure 2.12.1 shows the block diagram of the A-D converter, and Figures 2.12.2 and 2.12.3 show the A-D converter-related registers.

**Table 2.12.1 Performance of A-D converter**

Item	Performance
Method of A-D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage (Note 1)	0V to AVCC (VCC)
Operating clock $\phi_{AD}$ (Note 2)	$f_{AD}/\text{divide-by-2}$ of $f_{AD}/\text{divide-by-4}$ of $f_{AD}$ , $f_{AD}=f(XIN)$
Resolution	8-bit
Absolute precision	<ul style="list-style-type: none"> <li>● Without sample and hold function <math>\pm 3\text{LSB}</math></li> <li>● With sample and hold function <math>\pm 2\text{LSB}</math></li> </ul>
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog input pins	8 pins (AN0 to AN7) + 2pins (ANEX0 and ANEX1)
A-D conversion start condition	<ul style="list-style-type: none"> <li>● Software trigger A-D conversion starts when the A-D conversion start flag changes to "1"</li> <li>● External trigger (can be retriggered) A-D conversion starts when the A-D conversion start flag is "1" and the <math>\overline{ADTRG}/P97</math> input changes from "H" to "L"</li> </ul>
Conversion speed per pin	<ul style="list-style-type: none"> <li>● Without sample and hold function 49 <math>\phi_{AD}</math> cycles</li> <li>● With sample and hold function 28 <math>\phi_{AD}</math> cycles</li> </ul>

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

Note 2: Without sample and hold function, set the  $\phi_{AD}$  frequency to 250kHz min.

With the sample and hold function, set the  $\phi_{AD}$  frequency to 1MHz min.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

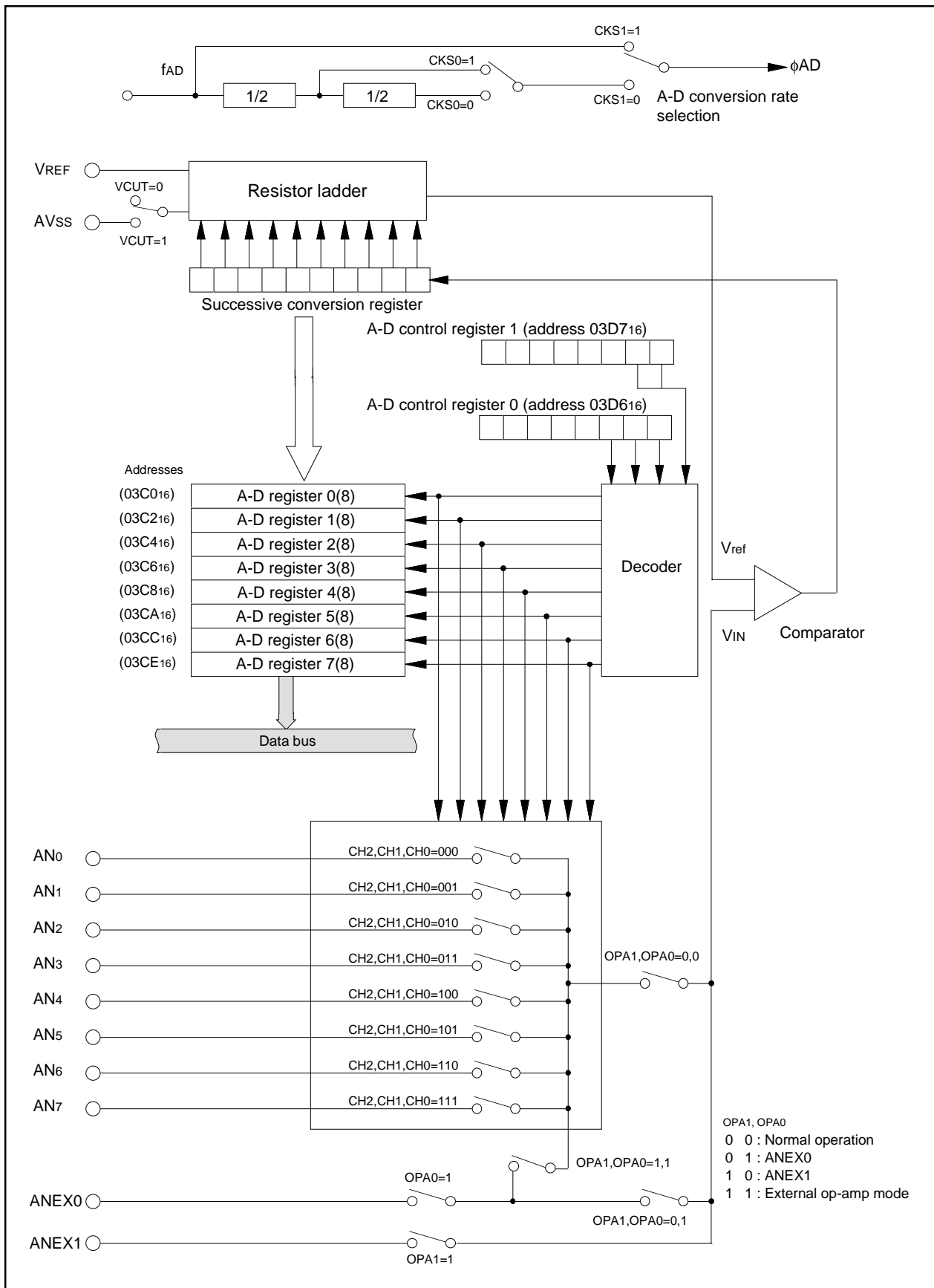


Figure 2.12.1 Block diagram of A-D converter

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

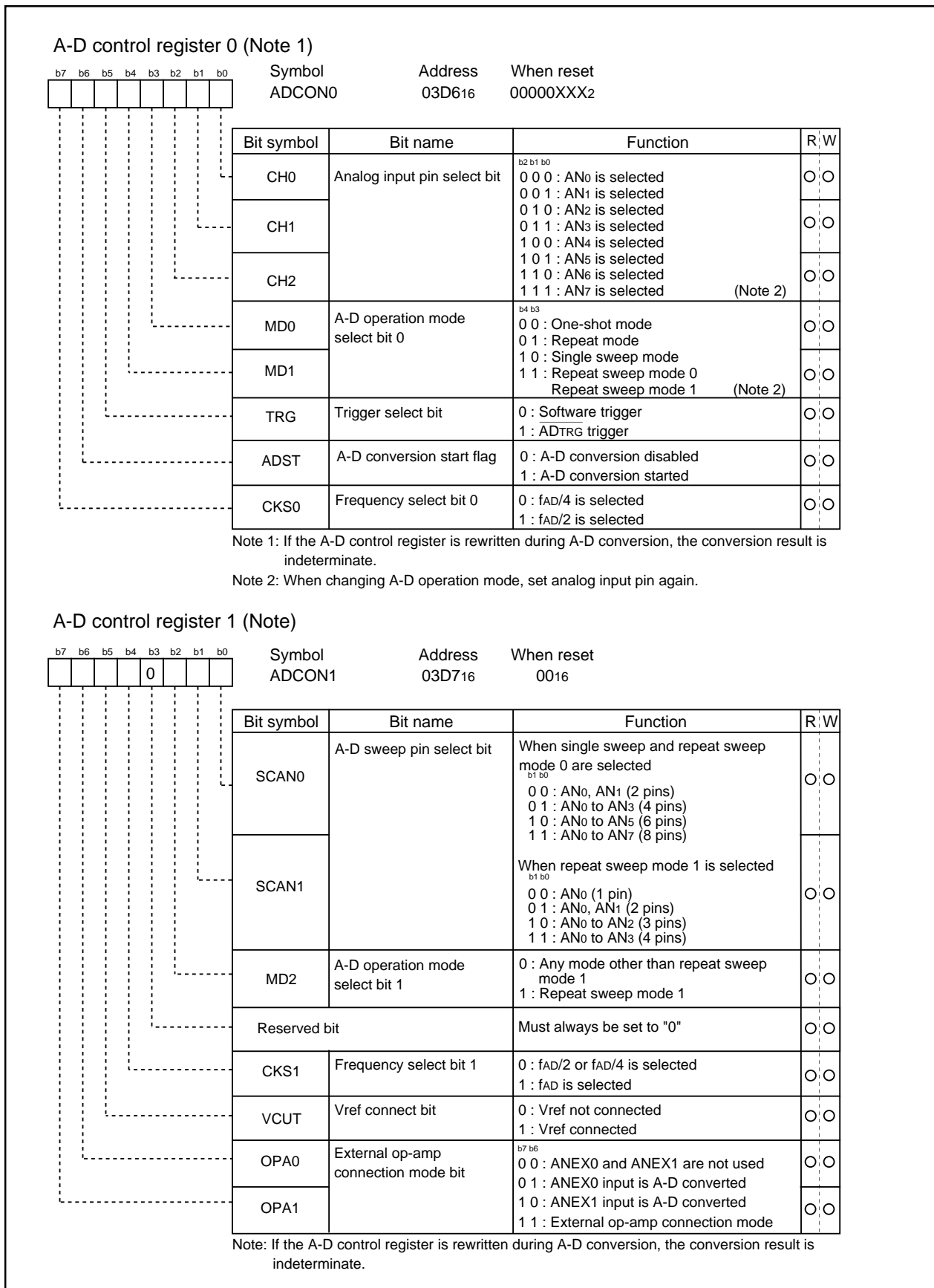


Figure 2.12.2 A-D converter-related registers (1)

M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

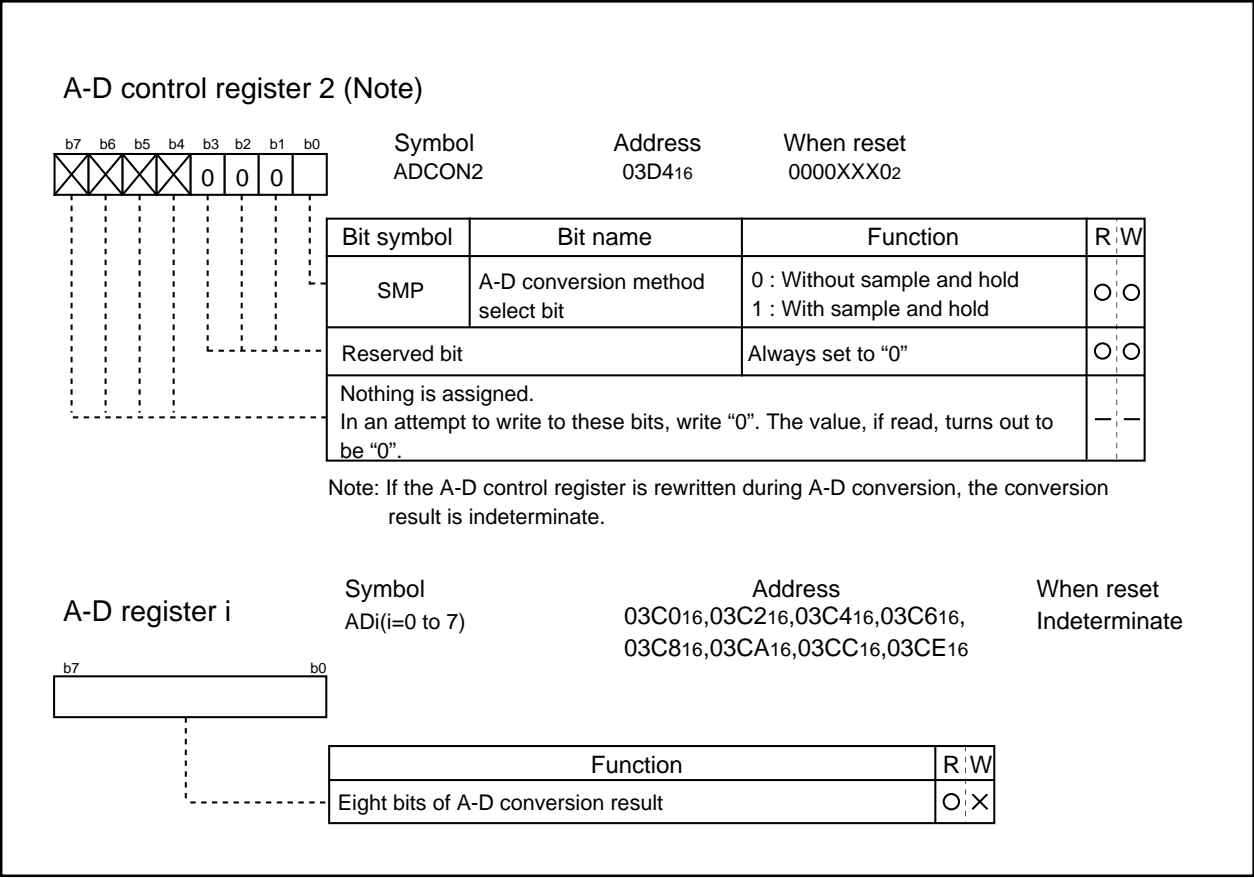


Figure 2.12.3 A-D converter-related registers (2)



# M306H2FCFP

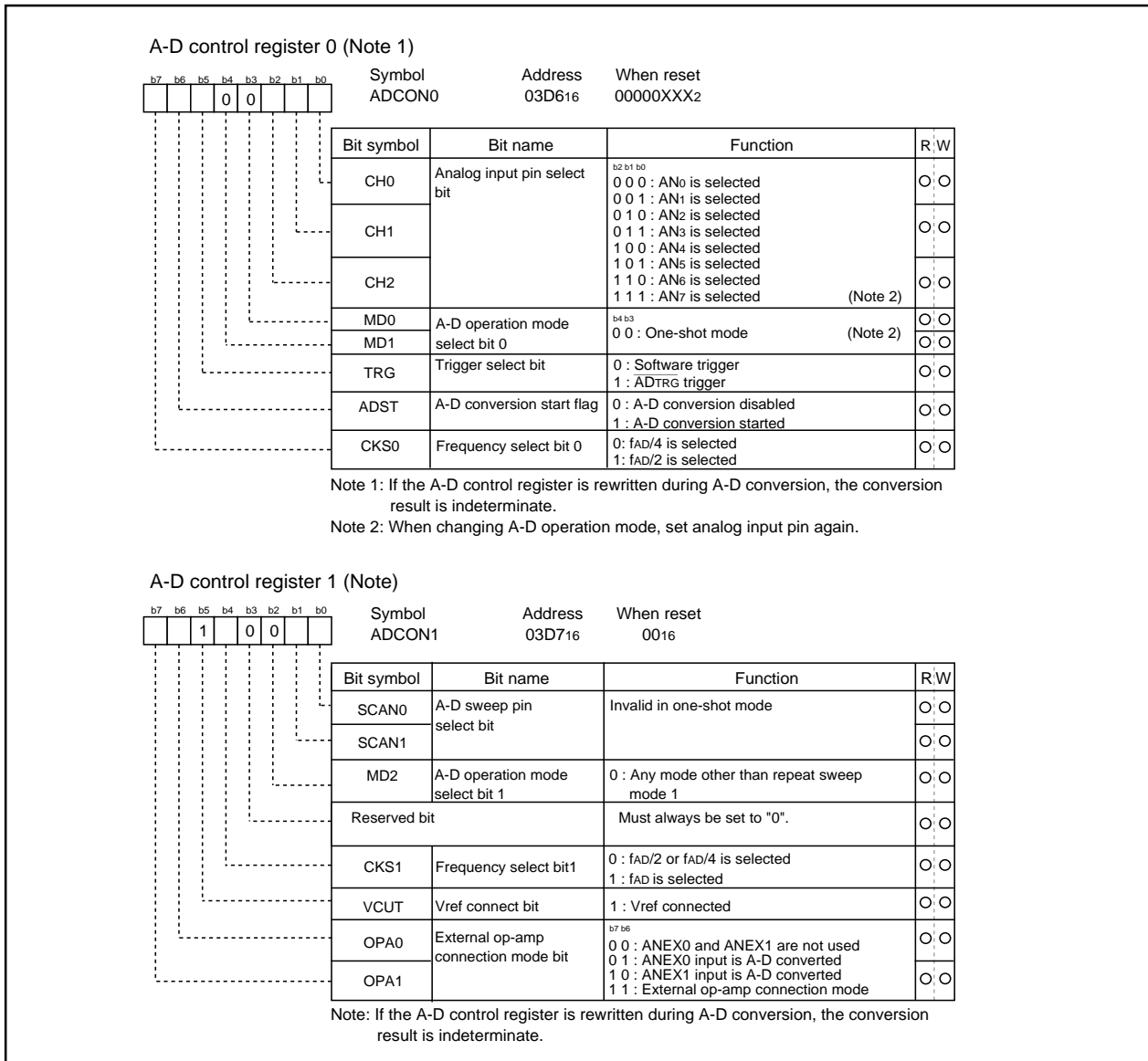
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (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 2.12.2 shows the specifications of one-shot mode. Figure 2.12.4 shows the A-D control register in one-shot mode.

**Table 2.12.2 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	<ul style="list-style-type: none"> <li>End of A-D conversion (A-D conversion start flag changes to "0", except when external trigger is selected)</li> <li>Writing "0" to A-D conversion start flag</li> </ul>
Interrupt request generation timing	End of A-D conversion
Input pin	One of AN0 to AN7, as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin



**Figure 2.12.4 A-D conversion register in one-shot mode**

# M306H2FCFP

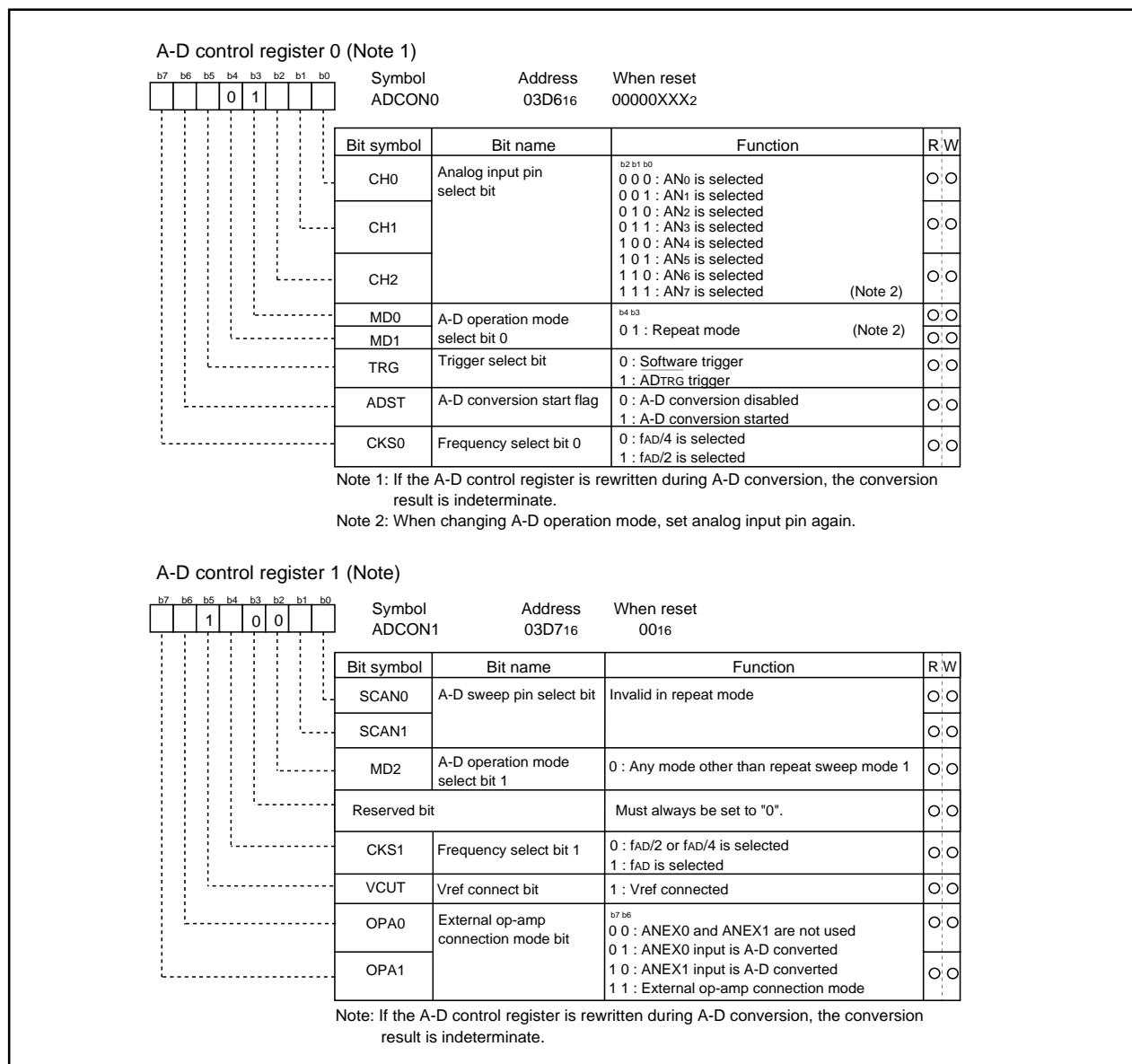
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (2) Repeat mode

In repeat mode, the pin selected using the analog input pin select bit is used for repeated A-D conversion. Table 2.12.3 shows the specifications of repeat mode. Figure 2.12.5 shows the A-D control register in repeat mode.

**Table 2.12.3 Repeat mode specifications**

Item	Specification
Function	The pin selected by the analog input pin select bit is used for repeated A-D conversion
Star condition	Writing "1" to A-D conversion start flag
Stop condition	Writing "0" to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	One of AN0 to AN7, as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin



**Figure 2.12.5 A-D conversion register in repeat mode**

# M306H2FCFP

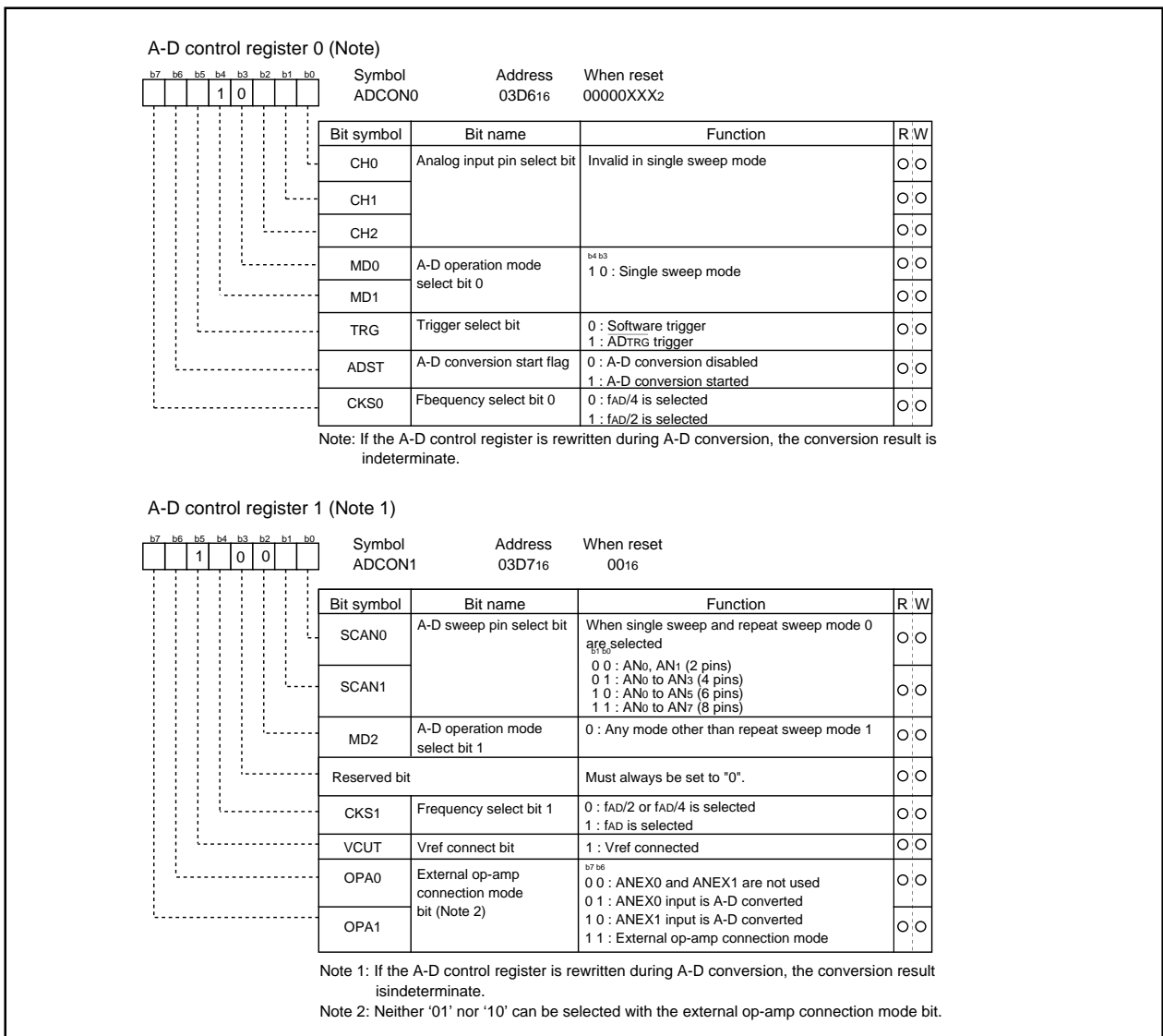
 SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

### (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 2.12.4 shows the specifications of single sweep mode. Figure 2.12.6 shows the A-D control register in single sweep mode.

**Table 2.12.4 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	<ul style="list-style-type: none"> <li>End of A-D conversion (A-D conversion start flag changes to "0", except when external trigger is selected)</li> <li>Writing "0" to A-D conversion start flag</li> </ul>
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


**Figure 2.12.6 A-D conversion register in single sweep mode**

# M306H2FCFP

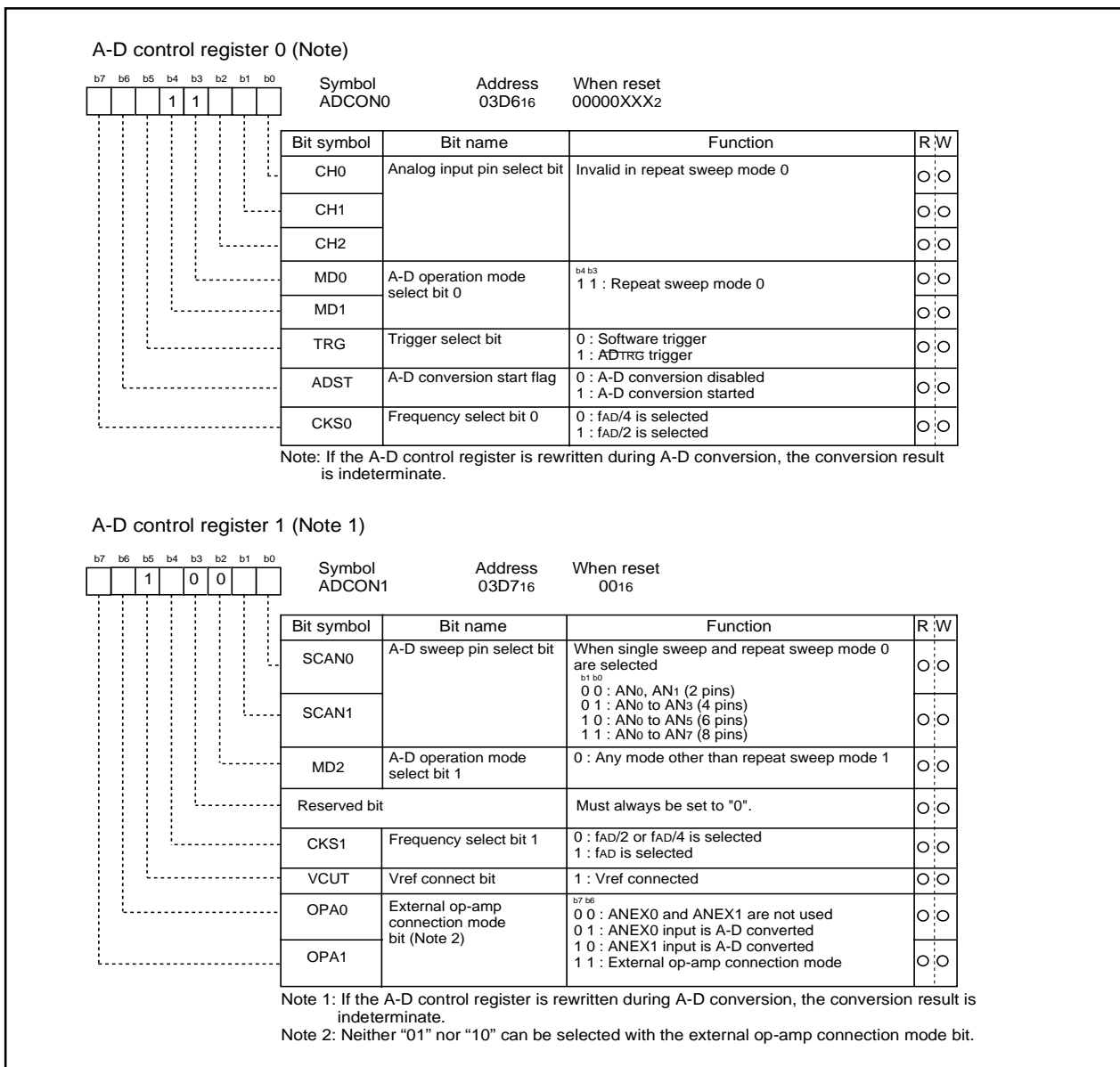
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (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 2.12.5 shows the specifications of repeat sweep mode 0. Figure 2.12.7 shows the A-D control register in repeat sweep mode 0.

**Table 2.12.5 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	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 (at any time)



**Figure 2.12.7 A-D conversion register in repeat sweep mode 0**

# M306H2FCFP

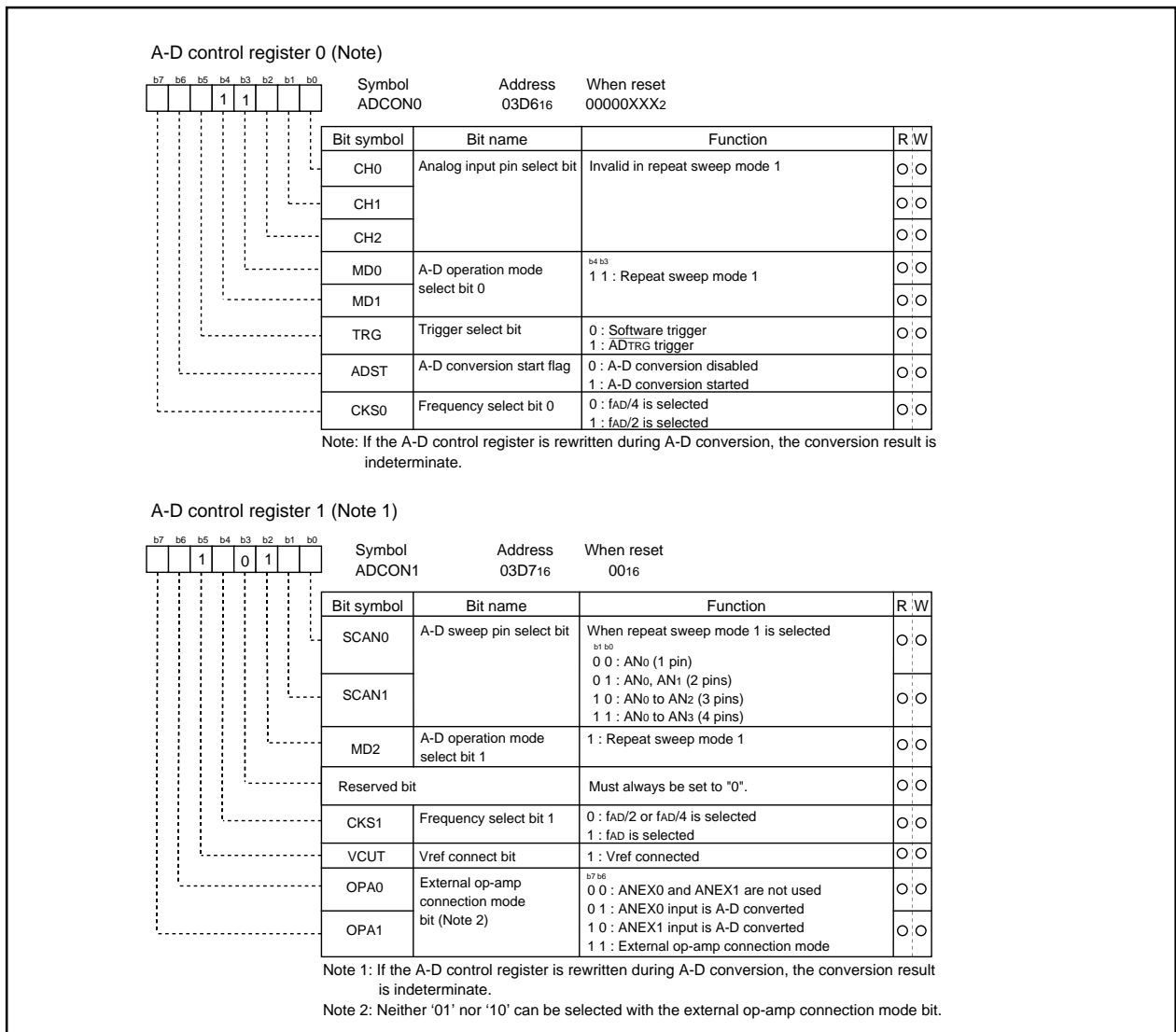
 SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (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 2.12.6 shows the specifications of repeat sweep mode 1. Figure 2.12.8 shows the A-D control register in repeat sweep mode 1.

**Table 2.12.6 Repeat sweep mode 1 specifications**

Item	Specification
Function	All pins perform repeat sweep A-D conversion, with emphasis on the pin or pins selected by the A-D sweep pin select bit Example : AN <sub>0</sub> selected AN <sub>0</sub> → AN <sub>1</sub> → AN <sub>0</sub> → AN <sub>2</sub> → AN <sub>0</sub> → AN <sub>3</sub> , 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	AN <sub>0</sub> (1 pin), AN <sub>0</sub> and AN <sub>1</sub> (2 pins), AN <sub>0</sub> to AN <sub>2</sub> (3 pins), AN <sub>0</sub> to AN <sub>3</sub> (4 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)


**Figure 2.12.8 A-D conversion register in repeat sweep mode 1**

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## (a) Sample and hold

Sample and hold is selected by setting bit 0 of the A-D control register 2 (address 03D4<sub>16</sub>) to "1". When sample and hold is selected, the rate of conversion of each pin increases. As a result, a 28 fAD cycle is achieved. 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<sub>16</sub>) 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<sub>16</sub>) 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<sub>16</sub>) is "1" and bit 7 is "1", input via AN<sub>0</sub> to AN<sub>7</sub> 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 2.12.9 is an example of how to connect the pins in external operation amp mode.

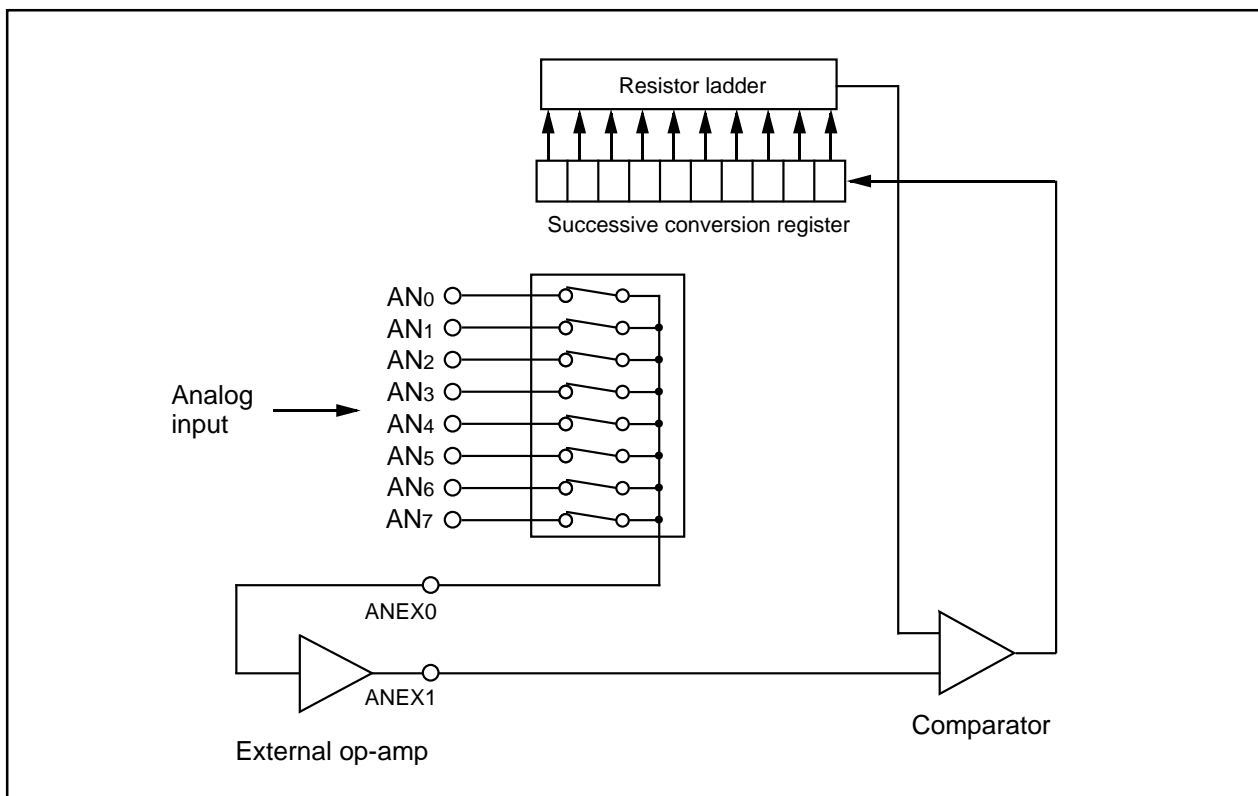


Figure 2.12.9 Example of external op-amp connection mode

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.13 D-A Converter

This is an 8-bit, R-2R type D-A converter. The microcomputer contains two independent D-A converters of this type.

D-A conversion is performed when a value is written to the corresponding D-A register. Bits 0 and 1 (D-A output enable bits) of the D-A control register decide if the result of conversion is to be output. Do not set the target port to output mode if D-A conversion is to be performed.

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

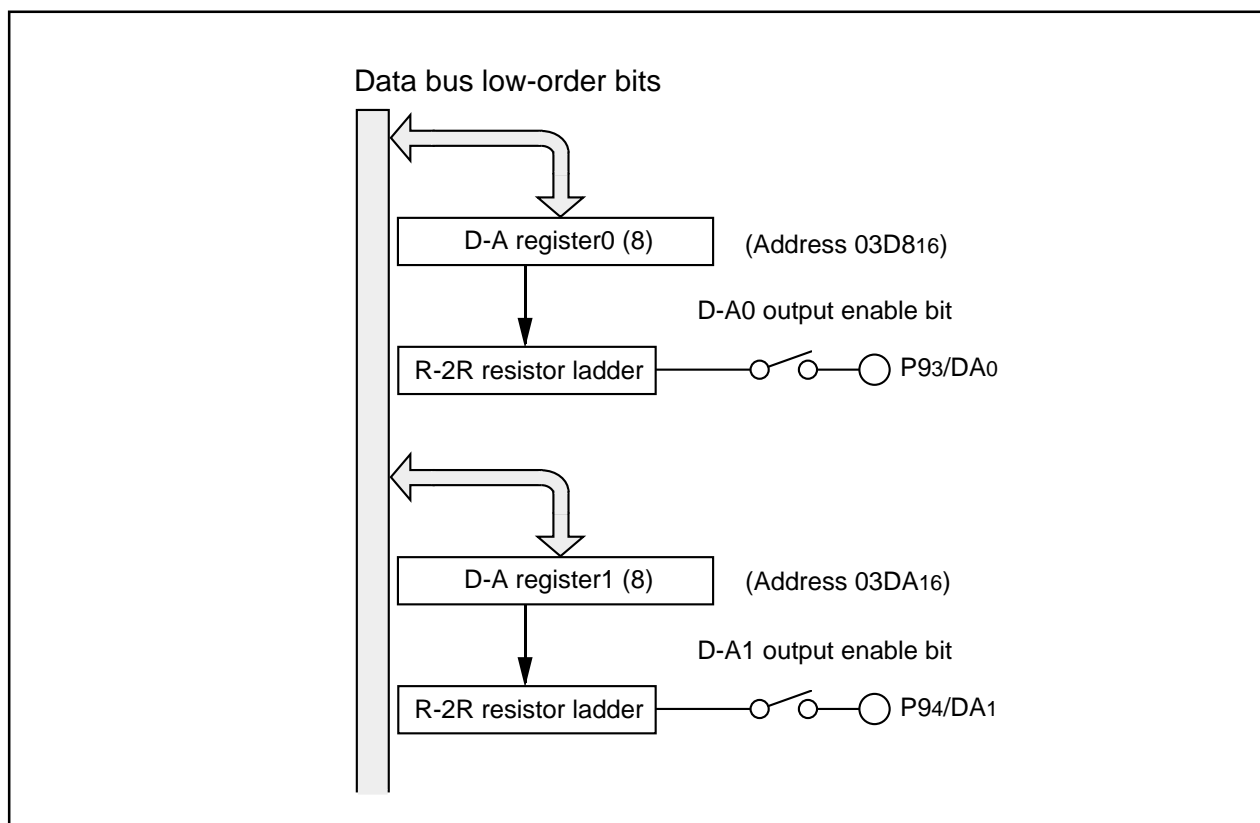
$$V = V_{REF} \times n / 256 \quad (n = 0 \text{ to } 255)$$

$V_{REF}$  : reference voltage

Table 2.13.1 lists the performance of the D-A converter. Figure 2.13.1 shows the block diagram of the D-A converter. Figure 2.13.2 shows the D-A control register. Figure 2.13.3 shows the D-A converter equivalent circuit.

**Table 2.13.1 Performance of D-A converter**

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



**Figure 2.13.1 Block diagram of D-A converter**

M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

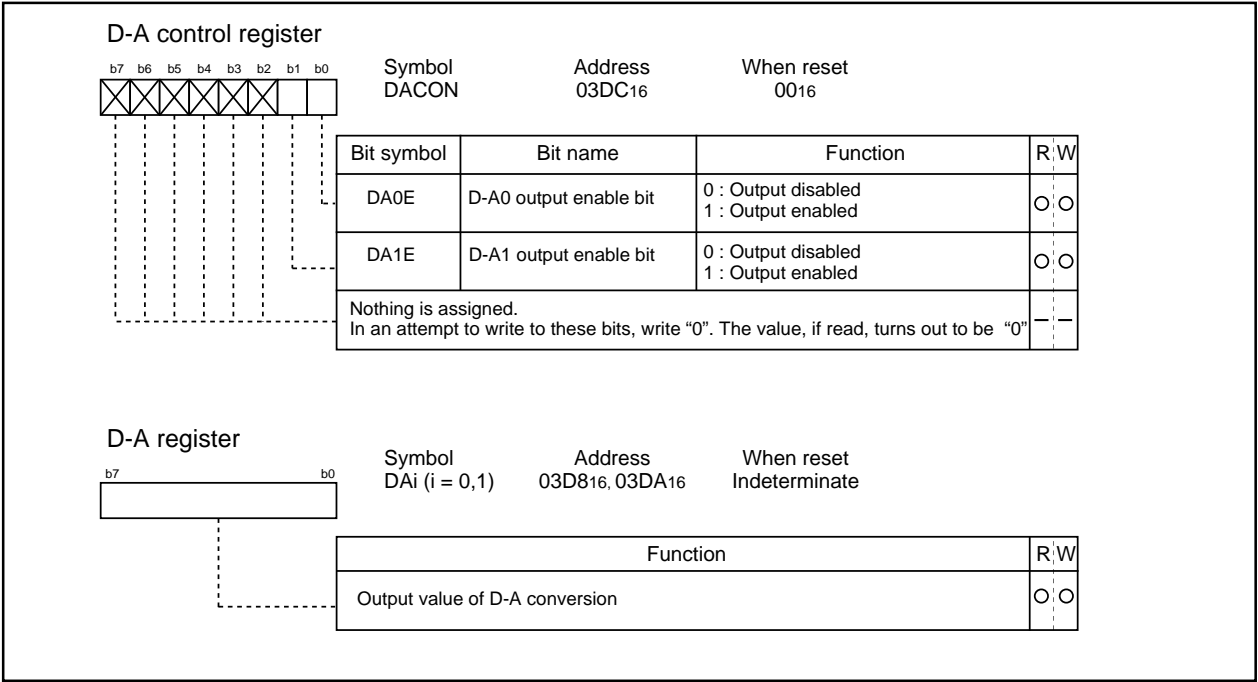


Figure 2.13.2 D-A control register

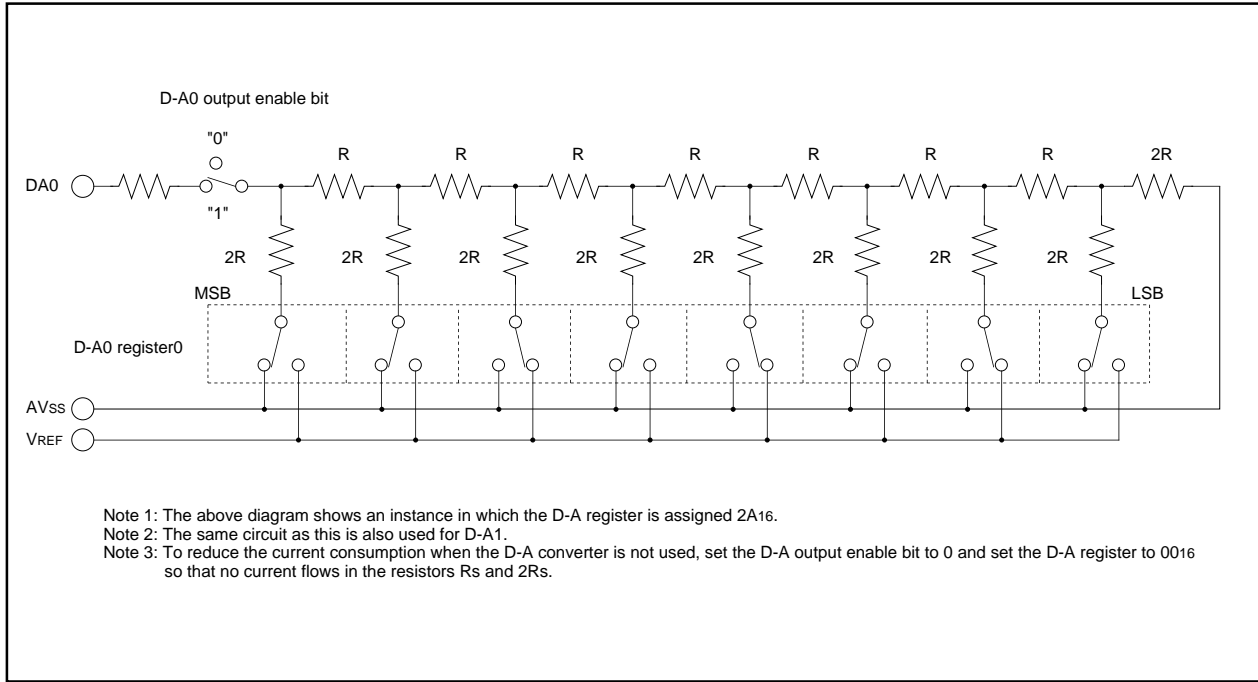


Figure 2.13.3 D-A converter equivalent circuit



M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

2.14 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 2.14.1 shows the block diagram of the CRC circuit. Figure 2.14.2 shows the CRC-related registers. Figure 2.14.3 shows the calculation example using the CRC calculation circuit

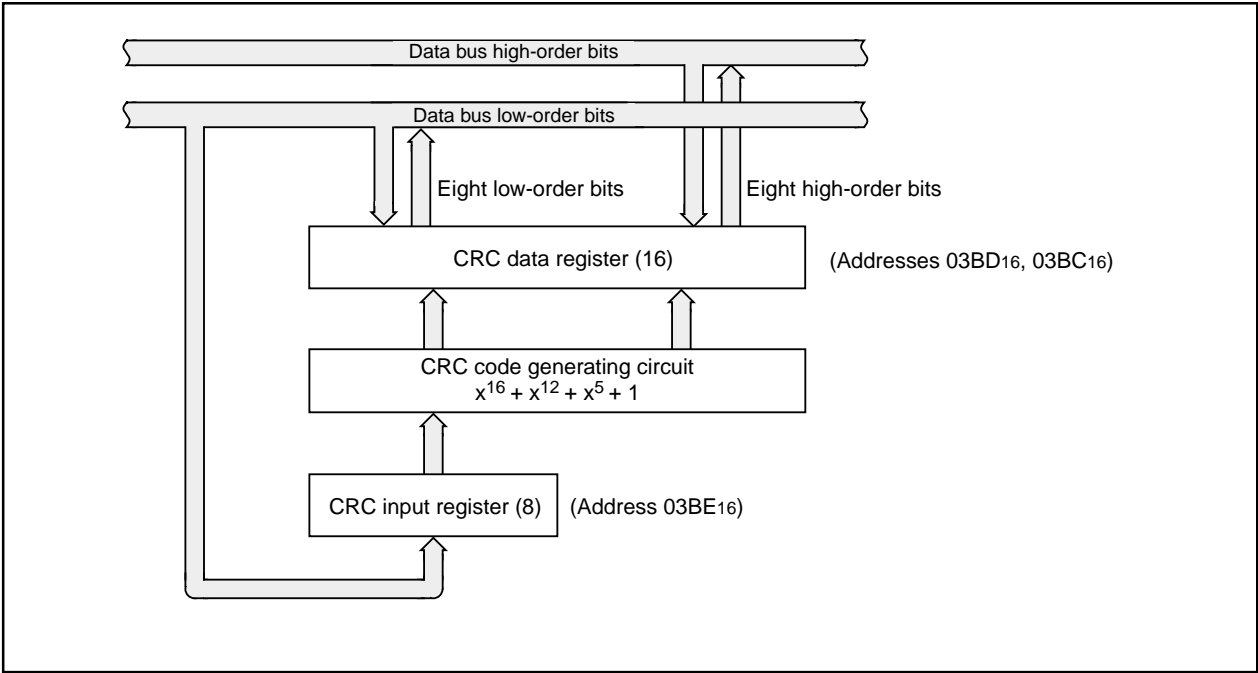


Figure 2.14.1 Block diagram of CRC circuit

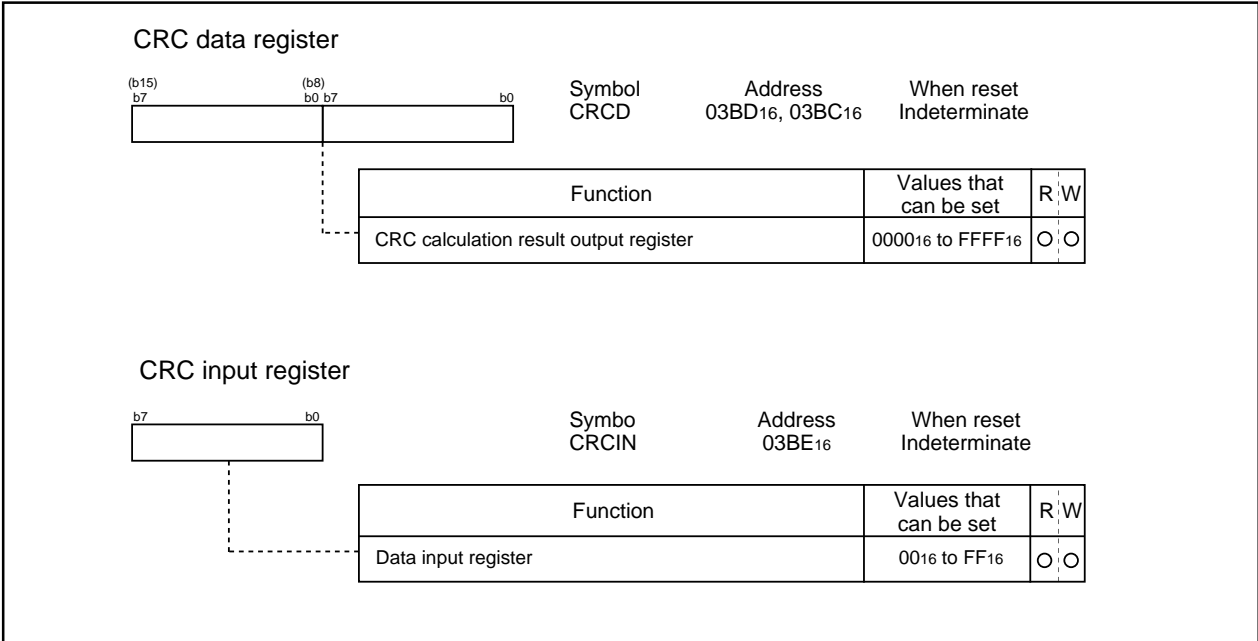


Figure 2.14.2 CRC-related registers

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

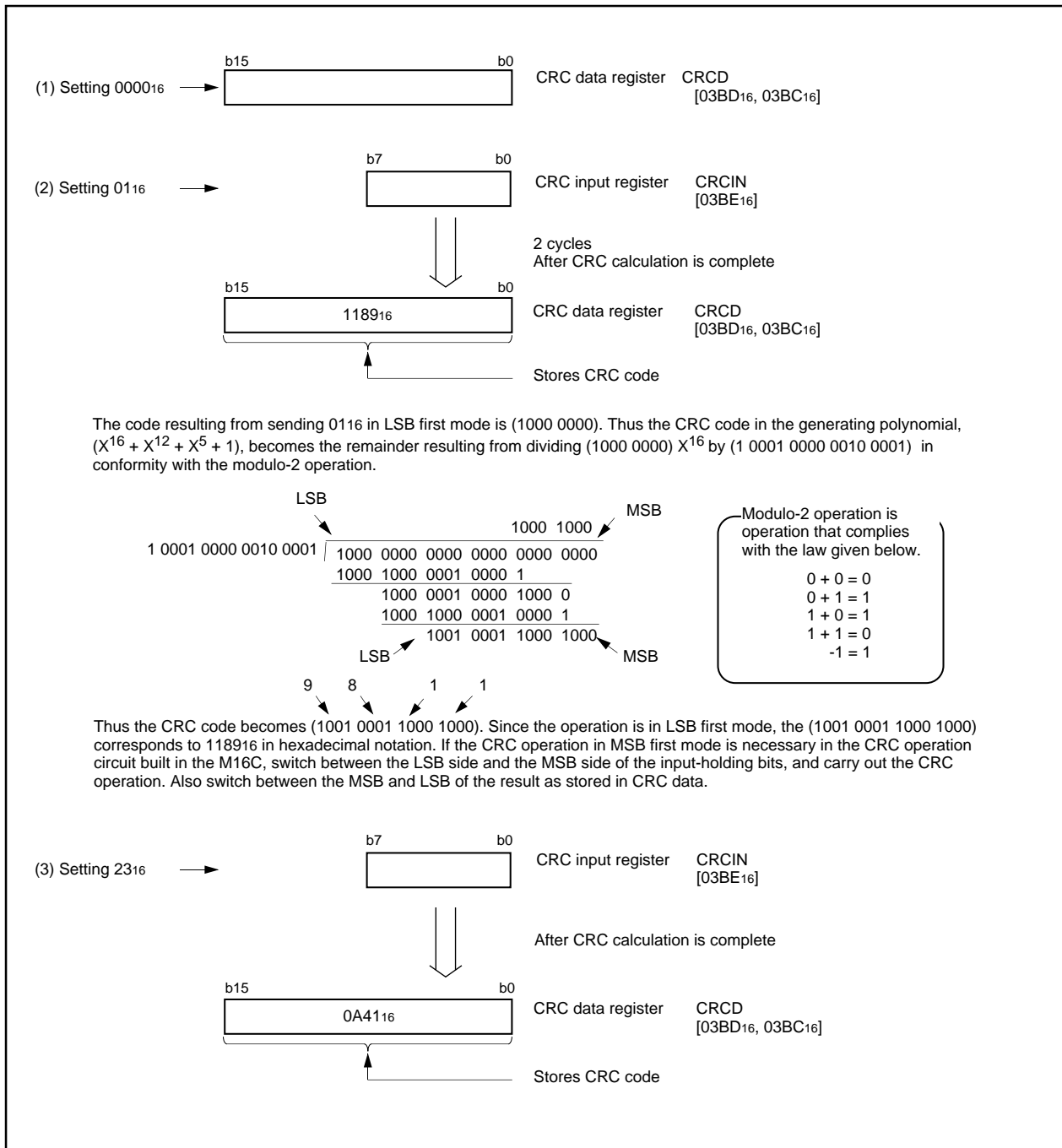


Figure 2.14.3 Calculation example using the CRC calculation circuit

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.15 Expansion Function

### 2.15.1 Expansion function description

Expansion function consists of data acquisition function and humming decoder function. Each function is controlled by expansion memories.

#### (1) Data acquisition function

Corresponds to

Hardware : TELETEXT, PDC, VPS, VBI and EPG-J

Software : XDS, WSS and VBI-ID

#### (2) Humming decoder function

8/4 humming and 24/18 humming

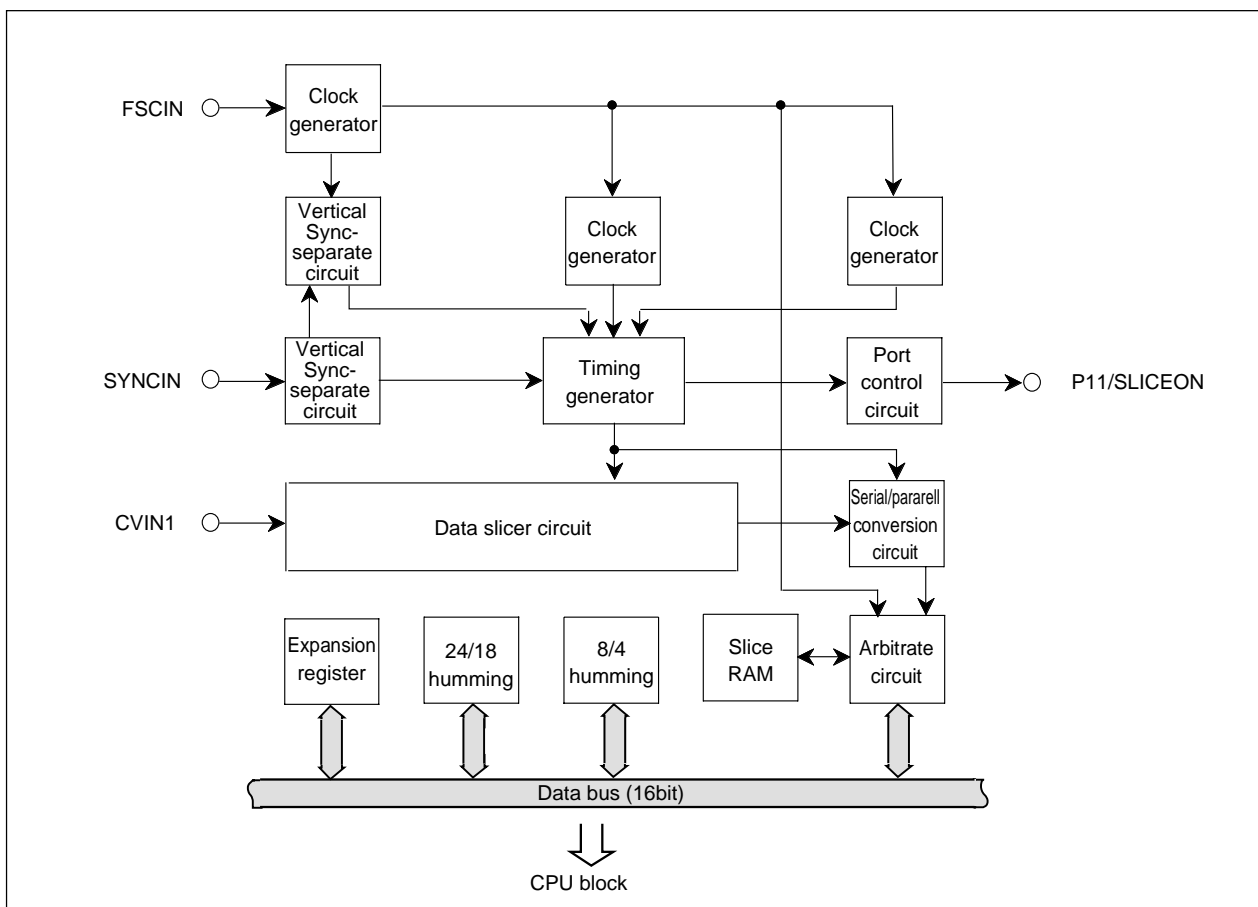


Figure 2.15.1 Block diagram of expansion function

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.15.2 Expansion memory

Expansion function memory is divided by 2 patterns ; Slice RAM and expansion register. (Humming decoder operates by the register placed on SFR). Data writing and read out to the Slice RAM and the expansion register are carried out 16 bit unit by the data setting register (addresses 020E<sub>16</sub>, 0210<sub>16</sub>, 0216<sub>16</sub> and 0218<sub>16</sub>) placed on SFR.

Contents of each memory and data setting register are shown in Table 2.15.1.

**Table 2.15.1 Expansion memory composition**

Expansion memory	Contents	Data setting register
Slice RAM	Store acquisition data.	Slice RAM address control register (020E <sub>16</sub> ) Slice RAM data control register (0210 <sub>16</sub> )
Expansion register	This register controls data acquisition	Expansion register address control register (0216 <sub>16</sub> ) Expansion register data control register (0218 <sub>16</sub> )

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## 2.15.3 Slice RAM

Store 18-line slice data. There are 3 types of Slice data : PDC, VPS and VBI. All data are stored to addresses which corresponds to acquisition line (ex. 22 line' data is stored to addresses 200<sub>16</sub> to 217<sub>16</sub> ). 24 addresses (SR00x to SR17x) are prepared for 1 line, acquisition data is stored in order from LSB side. Then, acquisition datas and field information are stored to the top address of each line. Slice RAM composite is shown in Table 2.15.2.

**Table 2.15.2 Slice RAM composition**

Slice RAM addresses (SA9 to SA0)	SD15	SD14	SD13	SD12	SD11	SD10	SD9	SD8	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0	Remarks
000 <sub>16</sub>	SR00F	SR00E	SR00D	SR00C	SR00B	SR00A	SR009	SR008	SR007	SR006	SR005	SR004	SR003	SR002	SR001	SR000	6th line or 318th line slice data
001 <sub>16</sub>	SR01F	SR01E	SR01D	SR01C	SR01B	SR01A	SR019	SR018	SR017	SR016	SR015	SR014	SR013	SR012	SR011	SR010	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
016 <sub>16</sub>	SR16F	SR16E	SR16D	SR16C	SR16B	SR16A	SR169	SR168	SR167	SR166	SR165	SR164	SR163	SR162	SR161	SR160	
017 <sub>16</sub>	SR17F	SR17E	SR17D	SR17C	SR17B	SR17A	SR179	SR178	SR177	SR176	SR175	SR174	SR173	SR172	SR171	SR170	
018 <sub>16</sub>	Unused area																
⋮																	
01F <sub>16</sub>																	
020 <sub>16</sub>	SR00F	SR00E	SR00D	SR00C	SR00B	SR00A	SR009	SR008	SR007	SR006	SR005	SR004	SR003	SR002	SR001	SR000	7th line or 319th line slice data
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
037 <sub>16</sub>	SR17F	SR17E	SR17D	SR17C	SR17B	SR17A	SR179	SR178	SR177	SR176	SR175	SR174	SR173	SR172	SR171	SR170	
040 <sub>16</sub>	⋮																8th line to 21th line or 320th line to 333 line slice data
⋮																	
1F7 <sub>16</sub>																	
200 <sub>16</sub>	SR00F	SR00E	SR00D	SR00C	SR00B	SR00A	SR009	SR008	SR007	SR006	SR005	SR004	SR003	SR002	SR001	SR000	22th line or 334th line slice data
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
217 <sub>16</sub>	SR17F	SR17E	SR17D	SR17C	SR17B	SR17A	SR179	SR178	SR177	SR176	SR175	SR174	SR173	SR172	SR171	SR170	
220 <sub>16</sub>	SR00F	SR00E	SR00D	SR00C	SR00B	SR00A	SR009	SR008	SR007	SR006	SR005	SR004	SR003	SR002	SR001	SR000	23th line or 335th line slice data
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
237 <sub>16</sub>	SR17F	SR17E	SR17D	SR17C	SR17B	SR17A	SR179	SR178	SR177	SR176	SR175	SR174	SR173	SR172	SR171	SR170	

For accessing to Slice RAM data, set accessing address (SA9 to SA0) (shown in Table 2.15.2) to Slice RAM address control register (address 020E<sub>16</sub> ). Then read out data from Slice RAM data control register (address 0210<sub>16</sub> ). When end the data reading, Slice RAM address control register increments address automatically. Then, next address data reading is possible. Do not access to unused area of each character codes. Must set address to each line because unused area has no address' automatically increment.

Slice RAM bit composition is shown in Figure 2.15.2, Slice RAM access registers are shown in Figure 2.15.3 and Slice RAM access block diagram is shown in Figure 2.15.4.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

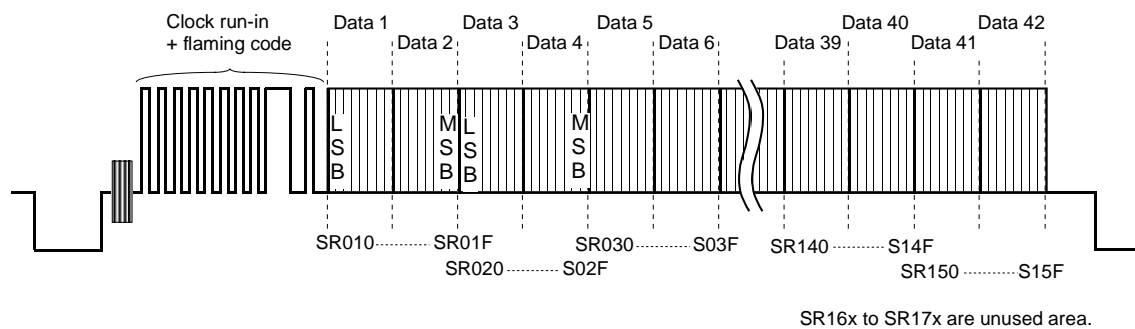
The each head address of the address is corresponded to acquisition line has stored next acquisition information.

	SR00F to SR004	SR003	SR002	SR001	SR000
PDC	0	field * (Note)	0	0	1
VPS	0	field * (Note)	0	1	0
VBI	0	field * (Note)	1	0	0
Other	0	0	0	0	0

Note : \* the first field : 1  
the second field : 0

## (1) PDC

In case of the PDC data, 16 bits (2 data) are stored for the 1 address from the LSB side.

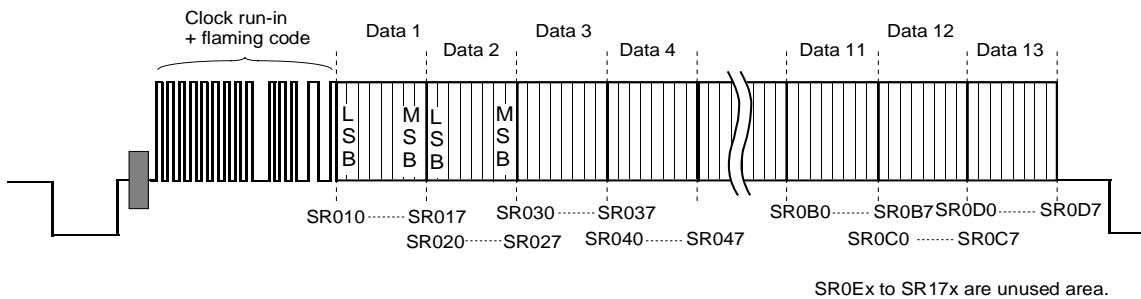


## (2) VPS

In case of the VPS data, 8 bits (a data) are stored for an address from the LSB side.

Low-order 8 bits stores the acquisition data. And, high-order 8 bits become warning bit, when the send data is not recognized as bi-phase type.

The case of bi-phase data = "1,0" or "0,1" (the bi-phase type) becomes "0" for this warning bit, and it becomes "1" in bi-phase data = "0,0" or "1,1" (it is not the bi-phase type). (For example, bi-phase data of SR011 is "0,0" or "1,1", "1" is set to SR019.)



## (3) VBI

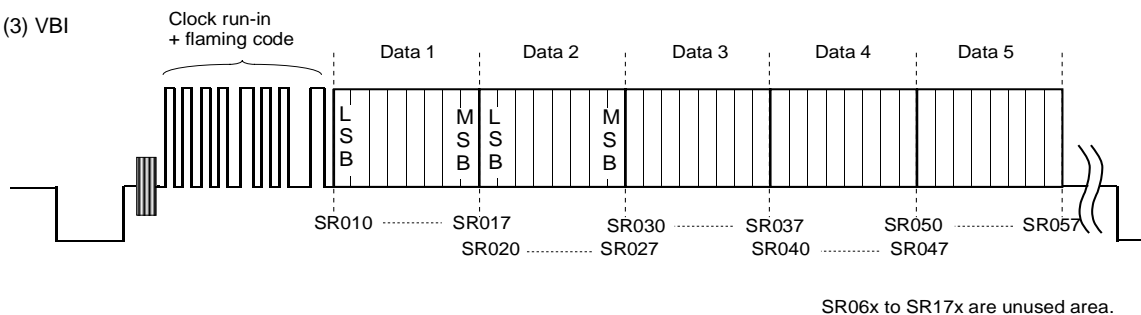


Figure 2.15.2 Slice RAM bit composition

M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

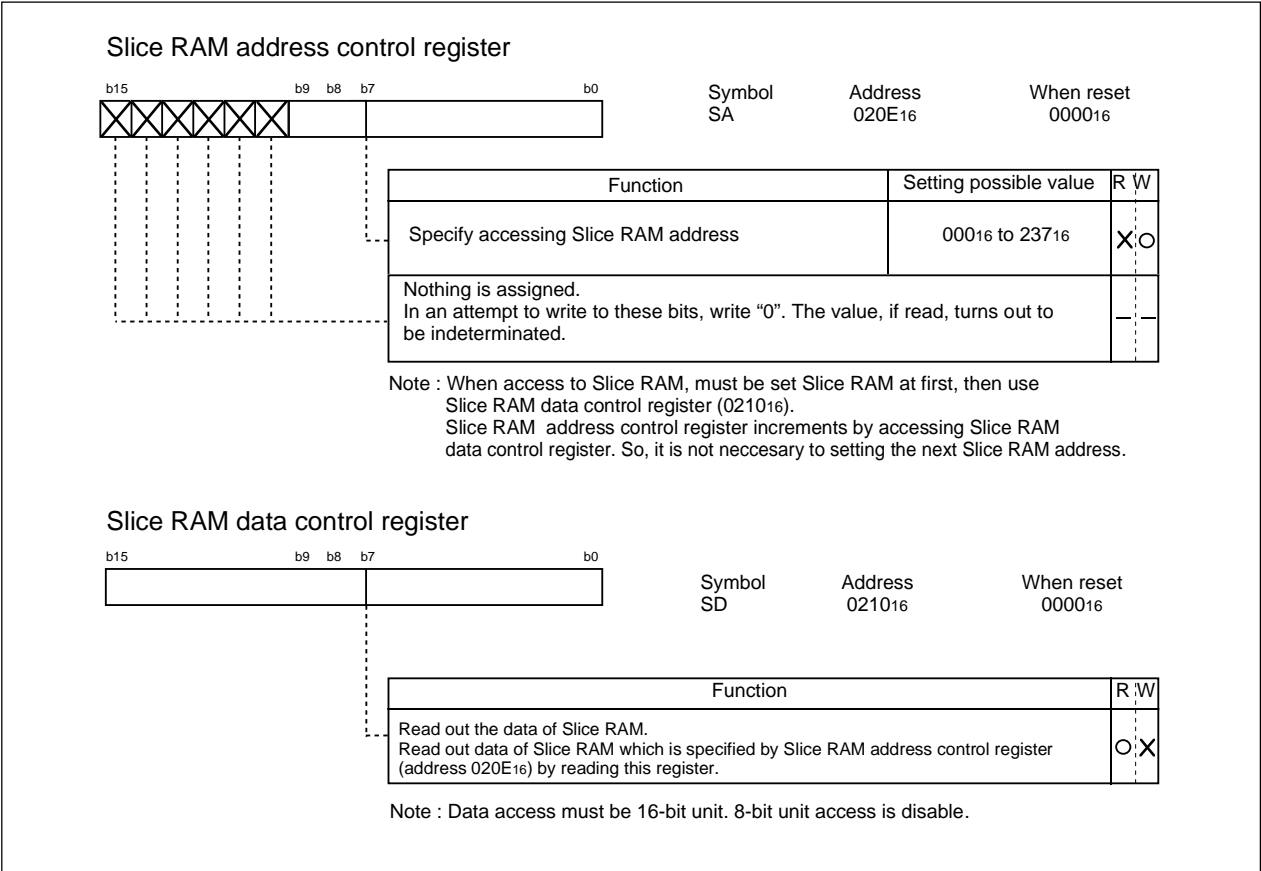


Figure 2.15.3 Slice RAM access registers

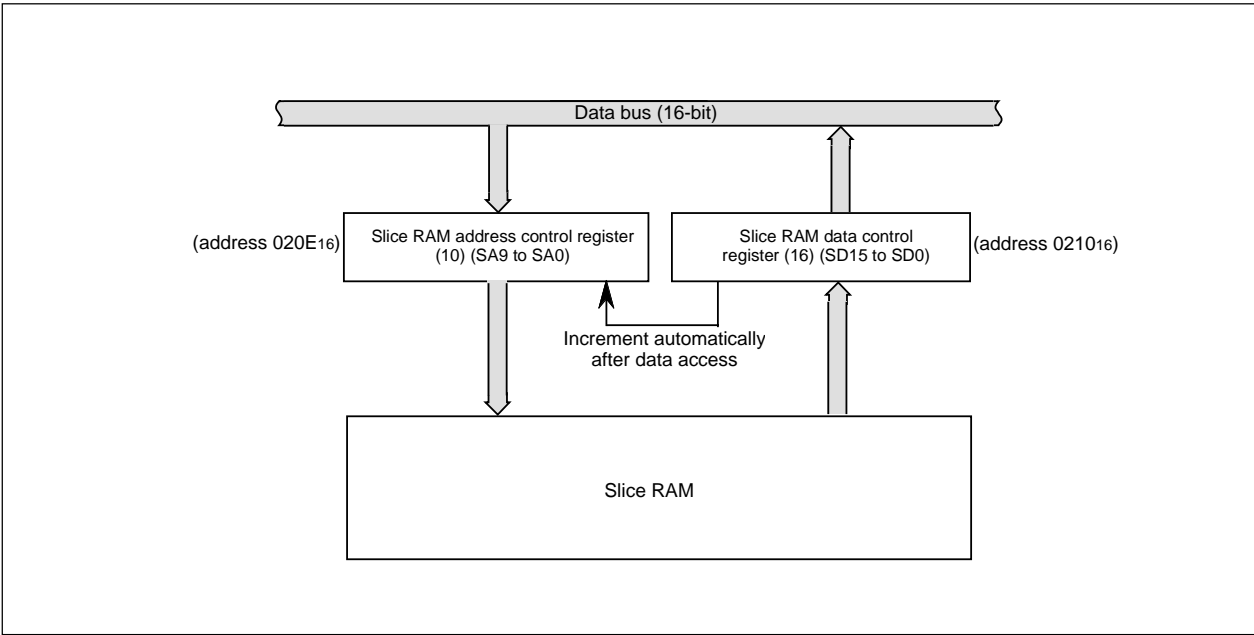


Figure 2.15.4 Slice RAM access block diagram

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.15.4 Expansion Register

Control Data acquisition function. Expansion register composition is shown in Table 2.15.3.

Table 2.15.3 Expansion register composition

DA5 to DA0	DD15	DD14	DD13	DD12	DD11	DD10	DD9	DD8	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0	Remarks
0016	-	-	-	-	-	-	STBY0	-	-	-	-	-	-	-	-	-	-
0116	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0216	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0316	-	-	-	TEST2	TEST1	TEST0	-	-	-	-	-	-	-	-	-	-	Test setting
0416	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0516	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0616	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0716	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0816	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0916	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0A16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0B16	-	PTD8	-	-	-	-	-	-	-	PTC8	-	-	-	-	-	-	Port setting
0C16	-	-	-	TIMBAS	-	-	-	-	-	-	-	-	-	-	-	-	Time base setting
0D16	-	-	-	NXP	-	-	-	-	-	-	-	-	-	-	-	-	Display control setting
0E16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
0F16	-	-	-	-	-	-	ADON	-	-	-	SEL_PDC8	-	-	-	-	-	Slicer control setting
1016	-	-	-	-	-	SYNSEP_0n0	-	-	SLSLVL	SLL_VP2	SLL_VP1	SLL_VP0	-	-	VPS_SUB	-	Sync separation, slice setting
1116	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1216	SEL_VPSH	-	-	-	-	-	-	-	-	-	SEK15	SEK14	SEK13	SEK12	SEK11	SEK10	Acquisition setting
1316	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1416	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1516	-	-	STBY1	-	-	-	VPS_VCO_ON	-	-	PDC_VCO_ON	-	-	XTAL_VCO	-	-	-	Oscillation ON/OFF setting
1616	-	-	-	-	-	PD2	PD1	-	PDC_HP10	PDC_HP9	PDC_HP8	PDC_HP7	PDC_HP6	PDC_HP5	PDC_HP4	PDC_HP3	PDC slice position setting
1716	HGSL	HGSL	-	SOFTSL	-	-	-	-	VPS_HP10	VPS_HP9	VPS_HP8	VPS_HP7	VPS_HP6	VPS_HP5	VPS_HP4	VPS_HP3	VPS slice position setting
1816	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1916	-	-	-	-	VPS_LINE4	VPS_LINE3	VPS_LINE2	VPS_LINE1	VPS_LINE0	-	VBIF2	VBIF1	VPSF2	VPSF1	PDCF2	PDCF1	Acquisition setting
1A16	VPS_FLCT7	VPS_FLCT6	VPS_FLCT5	VPS_FLCT4	VPS_FLCT3	VPS_FLCT2	VPS_FLCT1	VPS_FLCT0	PDC_FLCT7	PDC_FLCT6	PDC_FLCT5	PDC_FLCT4	PDC_FLCT3	PDC_FLCT2	PDC_FLCT1	PDC_FLCT0	PDC, VPS flaming setting
1B16	-	-	-	CHK_VPS	-	-	-	-	-	-	CHK_PDC	-	-	-	-	-	PDC, VPS flaming setting
1C16	-	-	-	SELPEEK	DIV_PDC8	DIV_PDC7	DIV_PDC6	DIV_PDC5	DIV_PDC4	DIV_PDC3	DIV_PDC2	DIV_PDC1	DIV_PDC0	DIV_PDCS2	DIV_PDCS1	DIV_PDCS0	PDC frequency setting
1D16	-	-	-	-	DIV_VPS8	DIV_VPS7	DIV_VPS6	DIV_VPS5	DIV_VPS4	DIV_VPS3	DIV_VPS2	DIV_VPS1	DIV_VPS0	DIV_VPSS2	DIV_VPSS1	DIV_VPSS0	VPS frequency setting
1E16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1F16	-	-	-	-	-	-	-	-	NGSYNC	-	-	FLD	-	-	-	-	Macro, field flag
2016	-	-	MIN5	MIN4	MIN3	MIN2	MIN1	MIN0	-	-	MAX5	MAX4	MAX3	MAX2	MAX1	MAX0	Acquisition setting
2116	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2216	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-



M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

For accessing to expansion register data, set accessing address (DA5 to DA0) (shown in Table 2.15.3) to expansion register address control register (address 0216<sub>16</sub>). Then write data (DD15 to DD0) by expansion register data control register (address 0218<sub>16</sub>). When end the data accessing, expansion register address control register increments address automatically. Then, next address data writing is possible.

Expansion register access registers are shown in Figure 2.15.5, expansion register access block diagram is shown in Figure 2.15.6, and expansion register bit compositions are shown in p153 to 163.

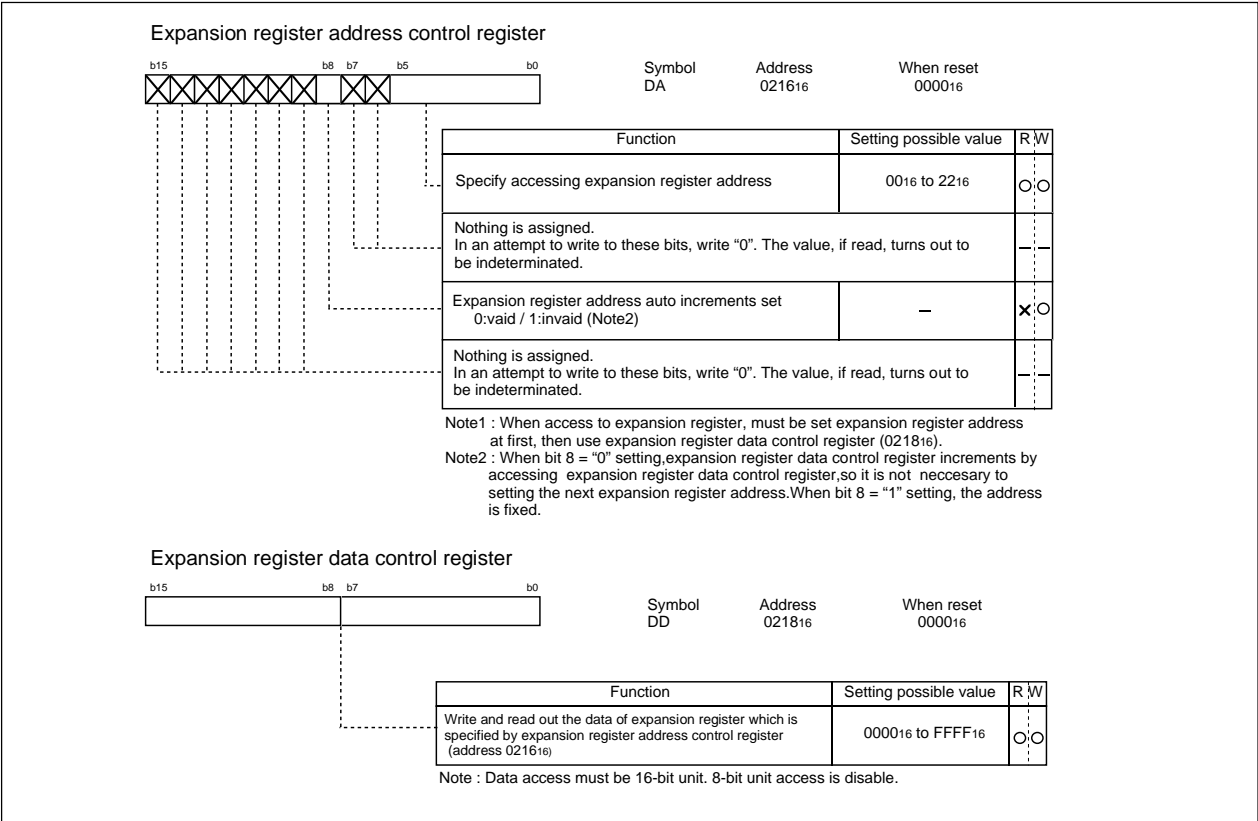


Figure 2.15.5 Expansion register access registers composition

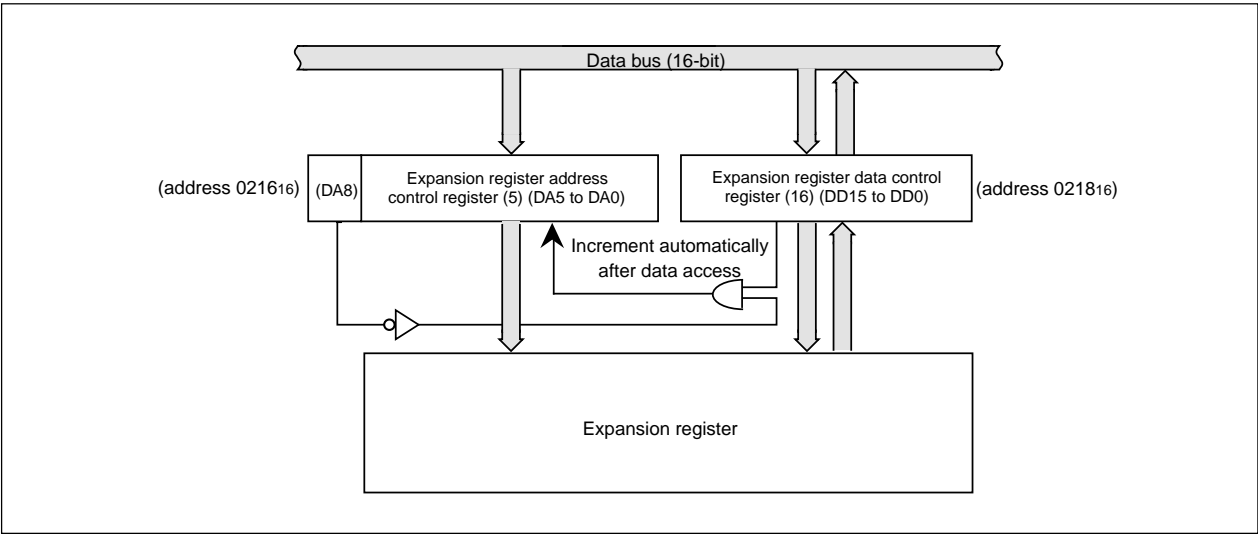


Figure 2.15.6 Expansion register access block diagram

M306H2FCFP


SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

Expansion register construction

(1) Address 0016 (= DA5 to 0)

DD15	0	0	0	0	0	0	0	DD8DD7	0	0	0	0	0	0	0	DD0	0		
Bit symbol		Bit name														Function		R	W
Reserved bit																Must always be set to "0".		X	0
STBY0		Stand-by mode selection bit														0	Normal mode	0	0
																1	Stand-by mode		
Reserved bit																Must always be set to "0".		X	0

(2) Addresses 0116, 0216 (= DA5 to 0)

DD15		DD8DD7								DD0				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	
														
Bit symbol		Bit name									Function		R	W
Reserved bit											Must always be set to "0".		X	0

(3) Address 0316 (= DA5 to 0)

DD15	DD8DD7							DD0																																				
0	0	0					0	0	0	0	0	0	0	0																														
<table><tr><td>Bit symbol</td><td colspan="2">Bit name</td><td>Function</td><td>R</td><td>W</td></tr><tr><td colspan="3">Reserved bit</td><td>Must always be set to "0".</td><td>x</td><td>0</td></tr><tr><td>TEST0</td><td rowspan="3">Test bit</td><td rowspan="3">Must always be set to "0".</td><td>0</td><td>0</td></tr><tr><td>TEST1</td><td>0</td><td>0</td></tr><tr><td>TEST2</td><td>0</td><td>0</td></tr><tr><td colspan="3">Reserved bit</td><td colspan="2">Must always be set to "0".</td><td>x</td><td>0</td></tr></table>															Bit symbol	Bit name		Function	R	W	Reserved bit			Must always be set to "0".	x	0	TEST0	Test bit	Must always be set to "0".	0	0	TEST1	0	0	TEST2	0	0	Reserved bit			Must always be set to "0".		x	0
Bit symbol	Bit name		Function	R	W																																							
Reserved bit			Must always be set to "0".	x	0																																							
TEST0	Test bit	Must always be set to "0".	0	0																																								
TEST1			0	0																																								
TEST2			0	0																																								
Reserved bit			Must always be set to "0".		x	0																																						

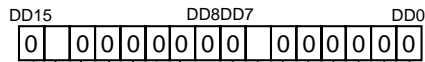
(4) Address 0416 to 0A16 (= DA5 to 0)

DD15	DD8DD7														DD0				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
Bit symbol		Bit name														Function		R	W
Reserved bit																Must always be set to "0".		X	0

# M306H2FCFP

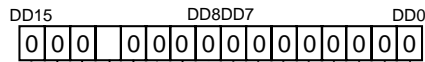
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (5) Address 0B<sub>16</sub> (= DA5 to 0)



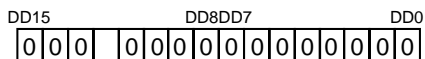
Bit symbol	Bit name		Function	R	W
Reserved bit			Must always be set to "0".	X	○
PTC8	Port P11 output selection bit	0	P11 output	○	○
		1	SLICEON output		
Reserved bit			Must always be set to "0".	X	○
PTD8	Port P11 data selection bit	0	When port output : fixed to "L" when SLICEON output : specified negative polarity	○	○
		1	When port output : fixed to "H" when SLICEON output : specified positive polarity		
Reserved bit			Must always be set to "0".	X	○

## (6) Address 0C<sub>16</sub> (= DA5 to 0)



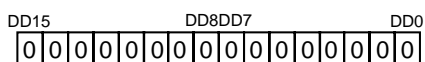
Bit symbol	Bit name	Function		R	W
Reserved bit		Must always be set to "0".		X	○
TIMBAS	Time base selection bit	0	Time base OFF	○	○
		1	Time base ON		
Reserved bit		Must always be set to "0".		X	○

## (7) Address 0D<sub>16</sub> (= DA5 to 0)



Bit symbol	Bit name	Function		R	W
Reserved bit		Must always be set to "0".		X	0
NXP	Broadcast method selection bit	0	NTSC		0
		1	PAL	0	0
Reserved bit		Must always be set to "0".		X	0

## (8) Address 0E<sub>16</sub> (= DA5 to 0)



Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to "0".	X	○

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (9) Address 0F<sub>16</sub> (= DA5 to 0)

DD15 DD8DD7 DD0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

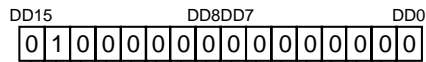
Bit symbol	Bit name	Function	R	W
	Reserved bit	Must always be set to "0".	x	0
SEL_PDCH	PDC clock selection bit	0 Do not set 1 Generates PDC clock in based on FSCIN pin input signal.	0	0
	Reserved bit	Must always be set to "0".	x	0
ADON	Data acquisition control bit	0 Data acquisition OFF 1 Data acquisition ON	0	0
	Reserved bit	Must always be set to "0".	x	0

## (10) Address 10<sub>16</sub> (= DA5 to 0)

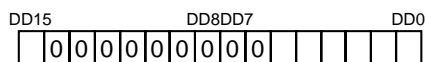
DD15 DD8DD7 DD0

0 0 0 0 0 0 0 0 0 0 0 1 0

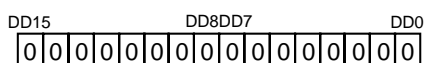
Bit symbol	Bit name	Function	R	W
	Reserved bit	Must always be set to "0".	x	0
VPS_SUB	Flaming code check selection bit for VPS data.	0 Later 8bits of flaming code 16bits 1 Former 4bits and later 4bits of flaming code 16bits (Select 8bits which is set in VPS_FLC0 to 7)	0	0
	Reserved bit	Must always be set to "1".	x	0
	Reserved bit	Must always be set to "0".	x	0
SLI_VP0	Acquisition start line selection bit (Field 1 and 2 are common) Stores data for 18 lines from the 6th line, normally. (SLI_VP2 to SLI_VP0 = "316" fixed)	If the acquisition start line is SLI_VS, <Field 1> $SLI\_VS = \sum_{n=0}^2 2^n SLI\_VPn + 3$ <Field 2> $SLI\_VS = \sum_{n=0}^2 2^n SLI\_VPn + 315$	0	0
SLI_VP1			0	0
SLI_VP2		Stores data for 18 lines from line which is set by this register to slice RAM.	0	0
SLSLVL	Acquisition level control bit	0 Auto level for data acquisition 1 Fix level for data acquisition	0	0
	Reserved bit	Must always be set to "0".	x	0
SYNCSEP_ON0	Synchronous separation control bit	0 Sync-sep circuit OFF 1 Sync-sep circuit ON	0	0
	Reserved bit	Must always be set to "0".	x	0

**M306H2FCFP**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER**(11) Address 11<sub>16</sub> (= DA5 to 0)**

Bit symbol	Bit name	Function	R	W
	Reserved bit	Must always be set to "0".	X	0
	Reserved bit	Must always be set to "1".	X	0
	Reserved bit	Must always be set to "0".	X	0

**(12) Address 12<sub>16</sub> (= DA5 to 0)**

Bit symbol	Bit name	Function			R	W	
SEKI0	Data acquisition control bit 1	SEKI1	SEKI0	N	0	0	
		0	0	5			
		0	1	4			
		1	0	3			
		1	1	2			
SEKI1		N times of the digital value after AD is done.			0	0	
SEKI2	Data acquisition control bit 2	SEKI3	SEKI2	N	0	0	
		0	0	4			
		0	1	3			
		1	0	1			
		1	1	Not differentiate			
SEKI3		It is differentiated for digital value after the SEKI0, 1 operation at digital value in the before N/8 period(clock run-in period).			0	0	
SEKI4	Data acquisition control bit 3	SEKI5	SEKI4	N	0	0	
		0	0	4			
		0	1	3			
		1	0	1			
		1	1	Not differentiate			
SEKI5		It is differentiated for digital value after the SEKI3, 2 operation at digital value in the after N/8 period(clock run-in period).			0	0	
Reserved bit		Must always be set to "0"			x	0	
SEL_VPSH	VPS clock selection bit	0	Do not set			0	0
		1	Generats VPS clock in based on FSCIN pin input signal.				

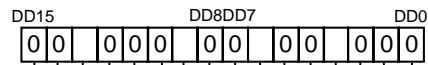
**(13) Addresses 13<sub>16</sub>, 14<sub>16</sub> (= DA5 to 0)**

Bit symbol	Bit name	Function	R	W
	Reserved bit	Must always be set to "0".	X	0

# M306H2FCFP

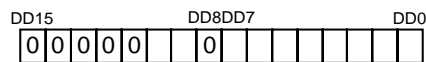
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

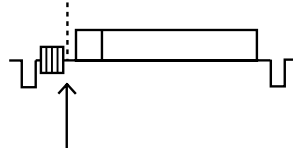
## (14) Address 15<sub>16</sub> (= DA5 to 0)



Bit symbol	Bit name	Function		R	W
Reserved bit		Must always be set to "0".		x	0
XTAL_VCO	Synchronous clock oscillation selection bit	0	Synchronizing clock OFF	0	0
		1	Synchronizing clock oscillation		
Reserved bit		Must always be set to "0".		x	0
PDC_VCO_ON	PDC clock oscillation selection bit	0	PDC clock OFF	0	0
		1	PDC clock oscillation		
Reserved bit		Must always be set to "0".		x	0
VPS_VCO_ON	VPS and VBI clock oscillation selection bit	0	VPS and VBI clock OFF	0	0
		1	VPS and VBI clock oscillation		
Reserved bit		Must always be set to "0".		x	0
STBY1	Stand-by mode selection bit	0	Normal mode	0	0
		1	Stand-by mode.		
Reserved bit		Must always be set to "0".		x	0

## (15) Address 16<sub>16</sub> (= DA5 to 0)

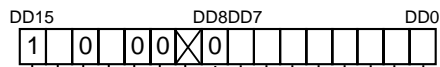


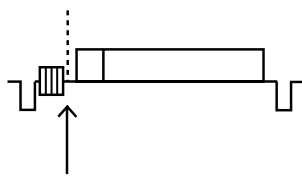
Bit symbol	Bit name	Function	R	W
PDC_HP3	PDC acquisition check start position selection bit	If the PDC acquisition check start position is PDC_HS, $PDC\_HS = T3 \times \sum_{n=3}^{10} 2^{(n-3)} PDC\_HPn$  T3 : PDC clock run-in cycle +2    Set to flaming code check start position  Set by the 144ns (1bit)	<input type="radio"/>	<input type="radio"/>
PDC_HP4			<input type="radio"/>	<input type="radio"/>
PDC_HP5			<input type="radio"/>	<input type="radio"/>
PDC_HP6			<input type="radio"/>	<input type="radio"/>
PDC_HP7			<input type="radio"/>	<input type="radio"/>
PDC_HP8			<input type="radio"/>	<input type="radio"/>
PDC_HP9			<input type="radio"/>	<input type="radio"/>
PDC_HP10			<input type="radio"/>	<input type="radio"/>
Reserved bit		Must always be set to "0".	x	<input type="radio"/>
PD1	PDC, VPS, VBI clock phase control bit	Adjust clock phase for Data slicer. Normally, PD2 to PD1=(10) <sub>2</sub> fixed.	<input type="radio"/>	<input type="radio"/>
PD2			<input type="radio"/>	<input type="radio"/>
Reserved bit		Must always be set to "0".	x	<input type="radio"/>

# M306H2FCFP

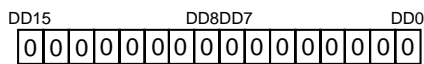
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (16) Address 17<sub>16</sub> (= DA5 to 0)



Bit symbol	Bit name	Function	R	W		
VPS_HP3	VPS and VBI acquisition check start position selection bit	If VPS and VBI acquisition check start position is VPS_HS, $VPS\_HS = T2 \times \sum_{n=3}^{10} 2^{(n-3)} VPS\_HPn$  T2 : VPS or VBI clock run-in cycle ÷2   Set to flaming code check start position  Set by the 200ns (1bit)....VPS Set by the 800ns (1bit)....VBI	○	○		
VPS_HP4			○	○		
VPS_HP5			○	○		
VPS_HP6			○	○		
VPS_HP7			○	○		
VPS_HP8			○	○		
VPS_HP9			○	○		
VPS_HP10			○	○		
Reserved bit			Must always be set to "0".	x	○	
Nothing is assigned.			x	x		
Reserved bit		Must always be set to "0".	x	○		
SOFTSLS	Slicer selection bit	0 PDC, VPS, VBI	○	○		
		1 XDS, WSS, VBI-ID				
Reserved bit		Must always be set to "0".	x	○		
HGSLS	Data slicer control bit	0 PDC, VPS	○	○		
		1 VBI				
HGSL	Data slicer control bit	Must always be set to "1".	○	○		

## (17) Address 18<sub>16</sub> (= DA5 to 0)

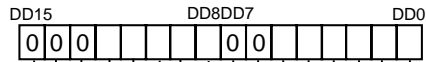


Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to "0".	x	○

# M306H2FCFP

 SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (18) Address 19<sub>16</sub> (= DA<sub>5</sub> to 0)



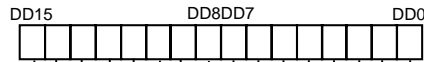
Bit symbol	Bit name	Function		R	W
PDCF1	PDC data acquisition selection bit (field1)	0	Do not acquisition field 1 PDC data	○	○
		1	Acquisition field 1 PDC data		
PDCF2	PDC data acquisition selection bit (field2)	0	Do not acquisition field 2 PDC data	○	○
		1	Acquisition field 2 PDC data		
VPSF1	VPS data acquisition selection bit (field1)	0	Do not acquisition field 1 VPS data	○	○
		1	Acquisition field 1 VPS data		
VPSF2	VPS data acquisition selection bit (field2)	0	Do not acquisition field 2 VPS data	○	○
		1	Acquisition field 2 VPS data		
VBIF1	VBI data acquisition selection bit (field1)	0	Do not acquisition field 1 VBI data	○	○
		1	Acquisition field 1 VBI data		
VBIF2	VBI data acquisition selection bit (field2)	0	Do not acquisition field 2 VBI data	○	○
		1	Acquisition field 2 VBI data		
Reserved bit		Must always be set to "0".		x	○
VPSF_LINE0	VPS data acquisition line selection bit	When VPS data acquisition line is VPS_LINES,  $VPS\_LINES = \sum_{n=0}^4 2^n VPS\_LINE_n + 7$  Fixed to 16th line normally.) (VPS_LINE4 to VPS LINE0 = "010012" fixed) Setting value from 00000 <sub>2</sub> to 10000 <sub>2</sub> (7th line to 23 line)		○	○
○				○	
○				○	
○				○	
○				○	
Reserved bit		Must always be set to "0".		x	○



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (19) Address 1A<sub>16</sub> (= DA<sub>5</sub> to 0)

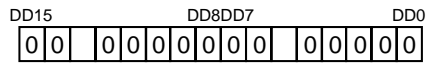


Bit symbol	Bit name	Function	R <sub>1</sub> W
PDC_FLC0	Flaming code selection bit at PDC acquisition	<b>[PDC]</b> 	O:O
PDC_FLC1			O:O
PDC_FLC2			O:O
PDC_FLC3			O:O
PDC_FLC4	Flaming code selection bit at VBI acquisition	<b>[VBI]</b> 	O:O
PDC_FLC5			O:O
PDC_FLC6			O:O
PDC_FLC7			O:O
VPS_FLC0	Flaming code selection bit at VPS and VBI acquisition	<b>[VPS]</b> When VPS_SUB (address <sub>1216</sub> ) = 0 	O:O
VPS_FLC1			O:O
VPS_FLC2			O:O
VPS_FLC3			O:O
VPS_FLC4		VPS_SUB = 1 	O:O
VPS_FLC5			O:O
VPS_FLC6			O:O
VPS_FLC7			O:O

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (20) Address 1B<sub>16</sub> (= DA<sub>5</sub> to 0)



Bit symbol	Bit name	Function		R	W
Reserved bit		Must always be set to "0".		X	0
CHK_PDC5	Flaming code check selection bit	0	PDC_FLC5 valid	X	0
		1	PDC_FLC5 invalid (Note1)		
Reserved bit		Must always be set to "0".		X	0
CHK_VPS5	Flaming code check selection bit	0	VPS_FLC5 valid	X	0
		1	VPS_FLC5 invalid (Note1)		
Reserved bit		Must always be set to "0".		X	0

Note1. At VBI acquisition, must be set to "1".

## (21) Address 1C<sub>16</sub> (= DA<sub>5</sub> to 0)



Bit symbol	Bit name	Function	R	W
DIV_PDCS0	PLL control bit for PDC	Contorl the acquisition clock frequency f <sub>PDC</sub> for PDC.  DIV_PDC8 to DIV_PDC0 = (000010010) <sub>2</sub> DIV_PDCS2 to DIV_PDCS0 = (101) <sub>2</sub>	○	○
DIV_PDCS1			○	○
DIV_PDCS2			○	○
DIV_PDC0	PLL divided value selection bit for PDC		○	○
DIV_PDC1			○	○
DIV_PDC2			○	○
DIV_PDC3			○	○
DIV_PDC4			○	○
DIV_PDC5			○	○
DIV_PDC6			○	○
DIV_PDC7			○	○
DIV_PDC8			○	○
SELPEEK			Peek point detect selection bit	0 Detect from A/D data
	1 Detect from data of digital calculation after normally "1" setting.			
Reserved bit		Must always be set to "0".	x	○

# M306H2FCFP

 SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (22) Address 1D<sub>16</sub> (= DA5 to 0)

DD15				DD8DD7								DD0			
0	0	0	0												
Bit symbol	Bit name		Function		R	W									
DIV_VPSS0	PLL control bit for VPS and VBI		Control the acquisition clock frequency f <sub>VPS</sub> for VPS and VBI.  DIV_VPS8 to DIV_VPS0 = (000001111) <sub>2</sub> DIV_VPSS2 to DIV_VPSS0 = (110) <sub>2</sub>												
DIV_VPSS1															
DIV_VPSS2															
DIV_VPS0	PLL divided value selection bit for VPS and VBI														
DIV_VPS1															
DIV_VPS2															
DIV_VPS3															
DIV_VPS4															
DIV_VPS5															
DIV_VPS6															
DIV_VPS7															
DIV_VPS8															
Reserved bit			Must always be set to "0".	x											

## (23) Address 1E<sub>16</sub> (= DA5 to 0)

DD15	DD8DD7	DD0	Bit symbol	Bit name	Function	R	W
0	0	0	Reserved bit			×	○

## (24) Address 1F<sub>16</sub> (= DA5 to 0)

DD15	DD8DD7	DD0	Bit symbol	Bit name	Function	R	W
			Reserved bit			×	×
			FLD	Fild flag	0 The secound field. 1 The first field.	○	×
			Reserved bit			×	×
			NGSYNC	synchronous signal detected flag (Note 1)	0 Normal 1 Abnormalities	○	×
			Reserved bit			×	×

Note 1: This flag detects unwanted signals during the sync signal (slice period).

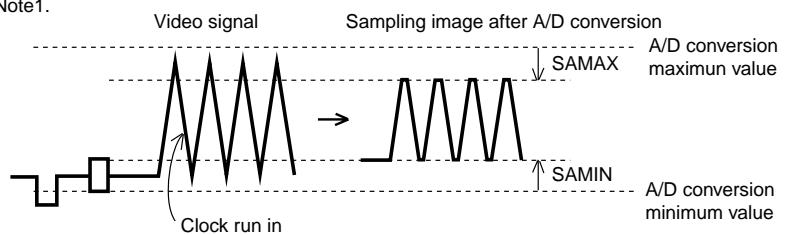
# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## (25) Address 20<sub>16</sub> (= DA5 to 0)

DD15																DD8DD7								DD0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
0	0									0	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			</

Note1.



## (26) Address 21<sub>16</sub>, 22<sub>16</sub> (= DA5 to 0)

DD15	DD8DD7	DD0	Bit symbol	Bit name	Function	R	W
0	0	0	Reserved bit		Must always be set to "0".	X	○

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.15.5 Expansion Register Construction Composition

### (1) Acquisition timing

The SLICEON signal is output in the acquisition possible period.

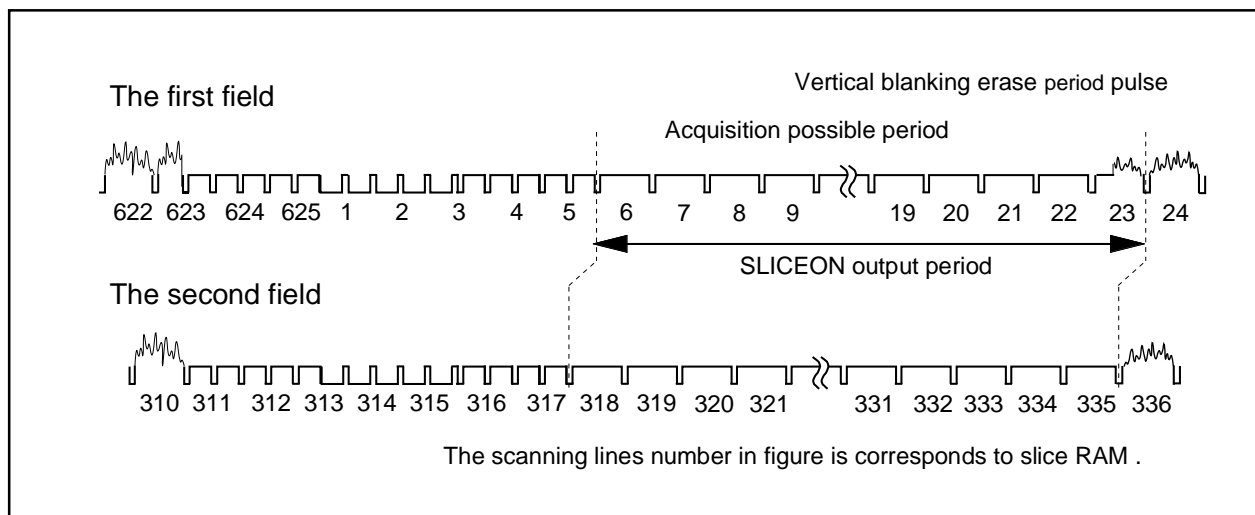


Figure 2.15.7 Acquisition timing

M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

2.15.6 8/4 Humming Decoder

8/4 humming decoder opetates only by written the data which 8/4 humming- decoded to 8/4 humming register (address 021A<sub>16</sub>). 8/4 humming register consists of 16 bits, can decode two data at a time. Can obtain the decoded result by reading 8/4 humming register, and the decoded value and error information are output. Corrects and outputs the decoded value for single error, and outputs only error information for double error. Decoded result is shown in Figure 2.15.8 and humming 8/4 register composition is shown in Figure 2.15.9.

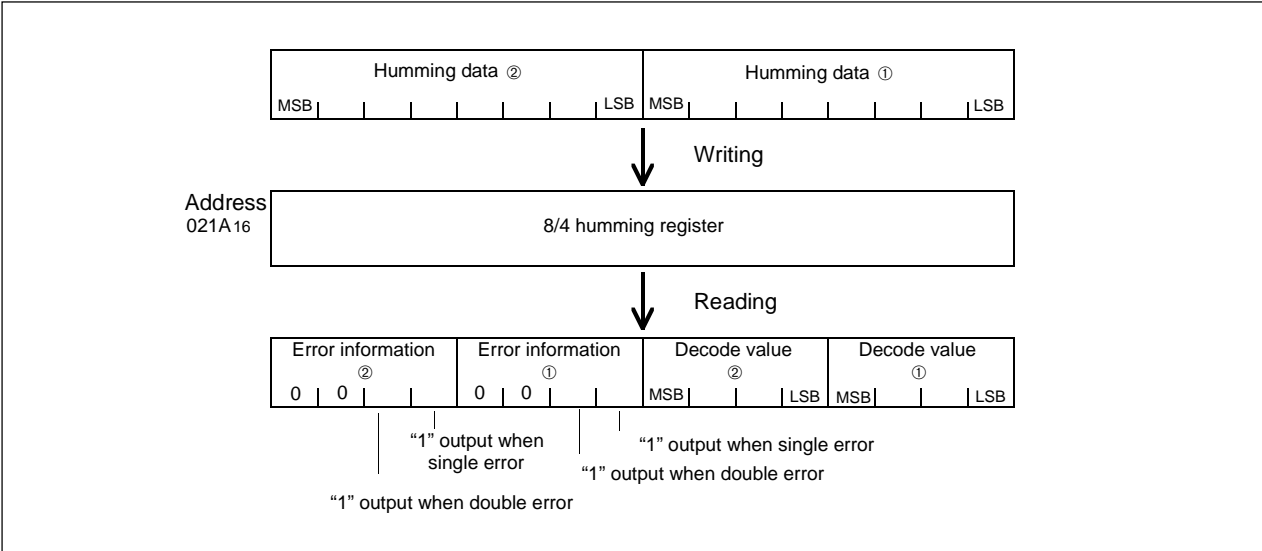


Figure 2.15.8 Decoded result

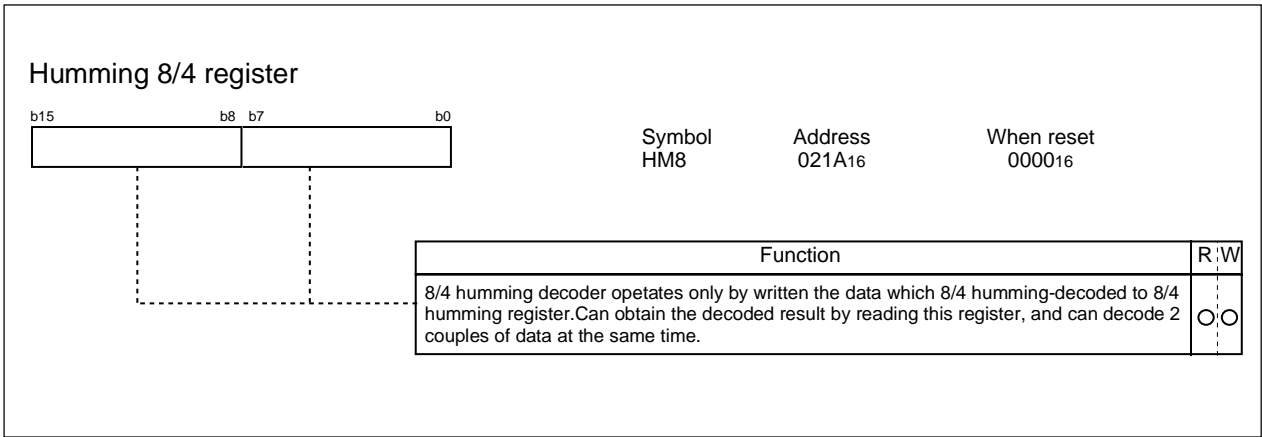


Figure 2.15.9 Humming 8/4 register composition

M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

2.15.7 24/18Humming Decoder

24/18 humming decoder operates only by written the data which 24/18 humming-encoded to 24/18 humming register 0 (address 021C<sub>16</sub>) and 1 (address 021E<sub>16</sub>). Can obtain the decoded result by reading the same 24/18 humming register. Decoded result is shown in Figure 2.15.10 and humming 24/18 register composition is shown in Figure 2.15.11.

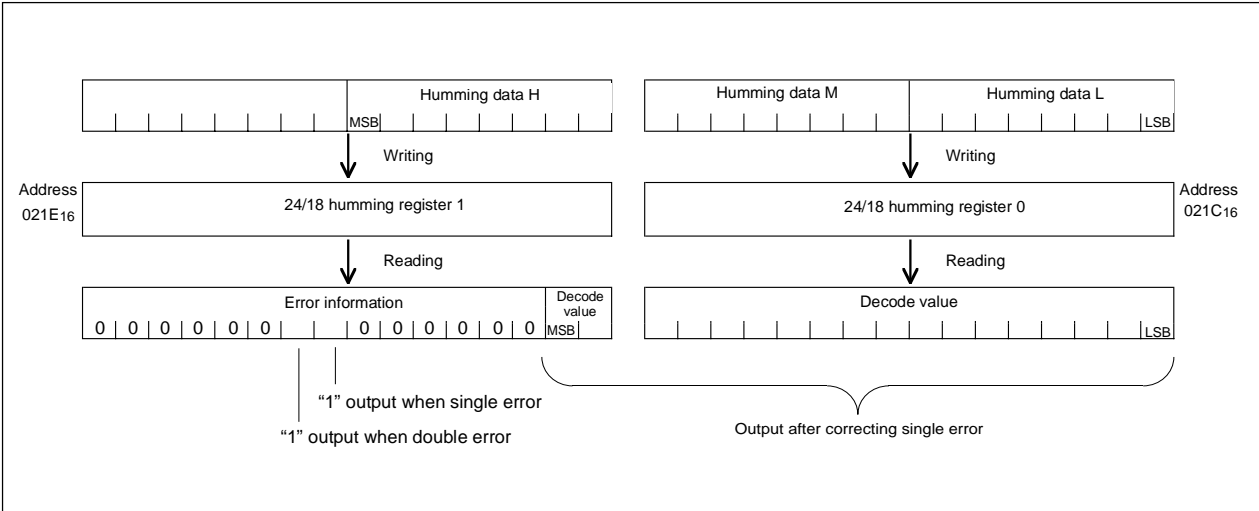


Figure 2.15.10 Decoded result

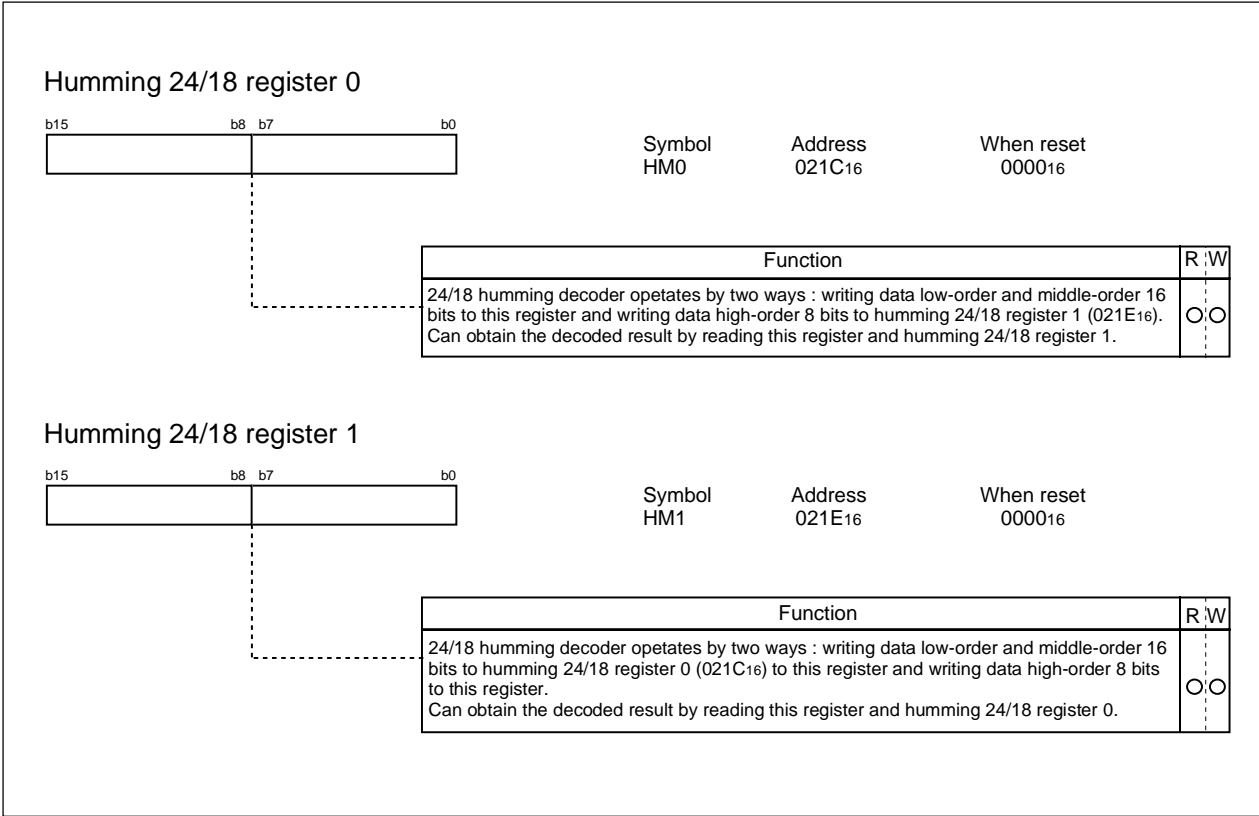


Figure 2.15.11 Humming 24/18 register composition

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## Continuous error correction

When uses humming 8/4 (address 021A<sub>16</sub>) at the same time as humming 24/18, can do the continuous error correction.

Continuous error correction sequence is shown in Figure 2.15.12.

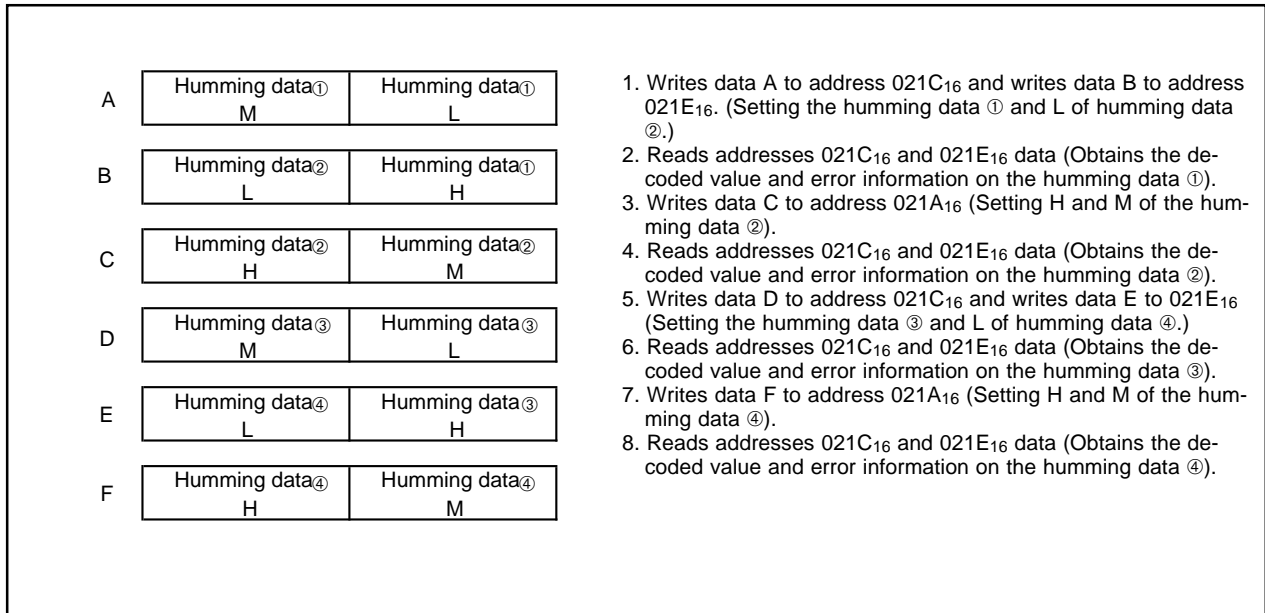


Figure 2.15.12 Continuous error correction sequence

Then, because using a part of circuit of humming 8/4 about this operation, cannot use this operation at the same time.

When using the humming circuit, do the decoded result reading operation at once after the setting data of humming. And do not access other memories (Including the humming circuit) before reading of the decoded result.



## 2.15.8 I/O Composition of pins for Expansion Memory

Figure 2.15.13 and figure 2.15.14 show pins for expansion memory.

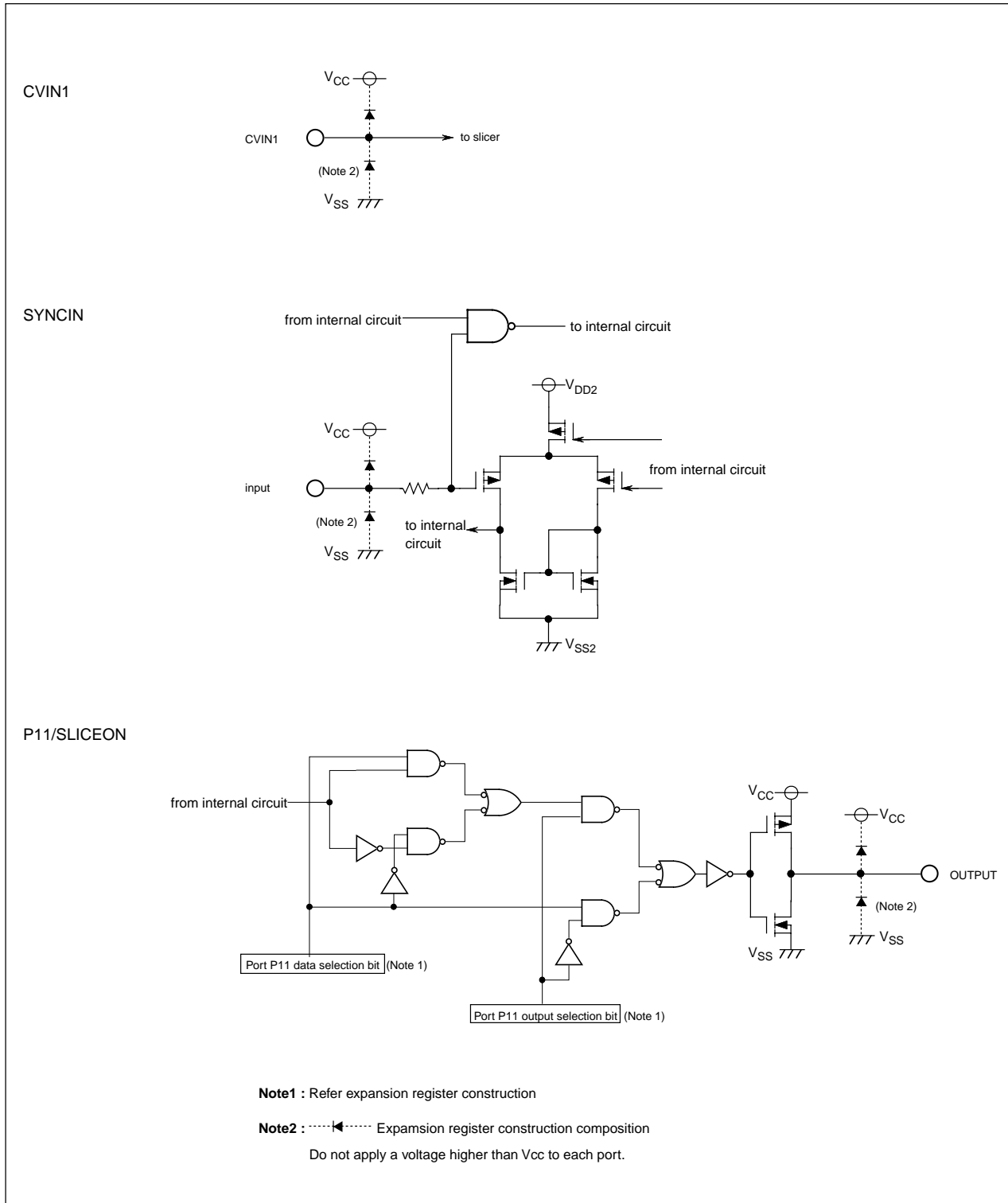


Figure 2.15.13 Pins for expansion memory(1)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

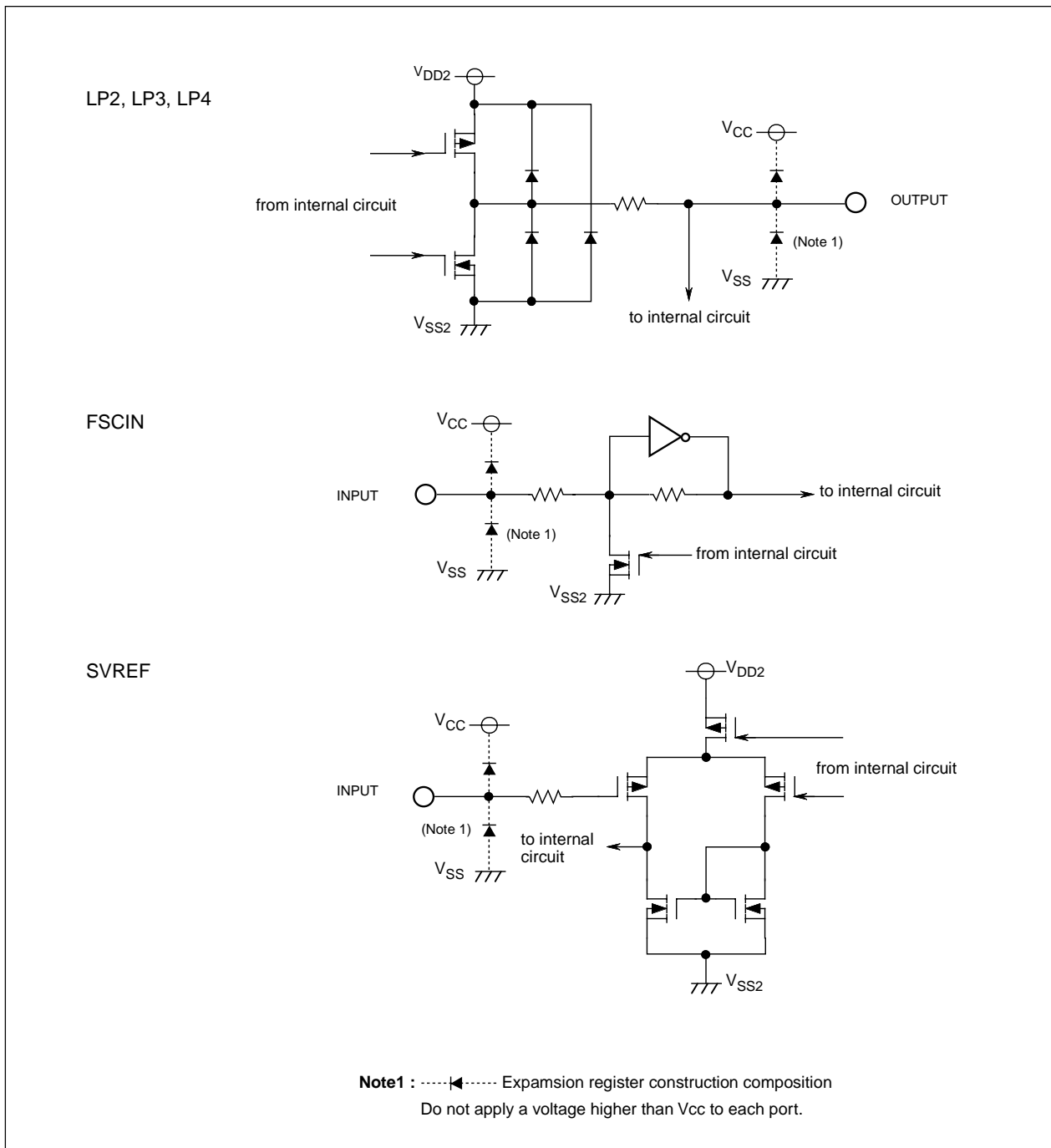


Figure 2.15.14 Pins for expansion memory(2)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 2.16 Programmable I/O Ports

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. P85 is an input-only port and has no built-in pull-up resistance.

Figures 2.16.1 to 2.16.4 show the programmable I/O ports. Figure 2.16.5 shows the I/O 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 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 2.16.6 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.

In memory expansion mode, the contents of corresponding direction register of pins

A0 to A19, D0 to D15,  $\overline{CS0}$  to  $\overline{CS3}$ , RD,  $\overline{WRL/WR}$ ,  $\overline{WRH/BHE}$ , ALE, RDY,  $\overline{HOLD}$ ,  $\overline{HLDA}$  and BCLK cannot be modified.

Note: There is no direction register bit for P85.

### (2) Port registers

Figure 2.16.7 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 mode, the contents of corresponding port register of pins A0 to

A19, D0 to D15,  $\overline{CS0}$  to  $\overline{CS3}$ , RD,  $\overline{WRL/WR}$ ,  $\overline{WRH/BHE}$ , ALE, RDY,  $\overline{HOLD}$ ,  $\overline{HLDA}$  and BCLK cannot be modified.

### (3) Pull-up control registers

Figure 2.16.8 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, 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 2.16.9 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 always.

In memory expansion mode, this register is valid in the following:

- External bus width is 8 bits.
- Port P1 can be used as a port in multiplexed bus for the entire space.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

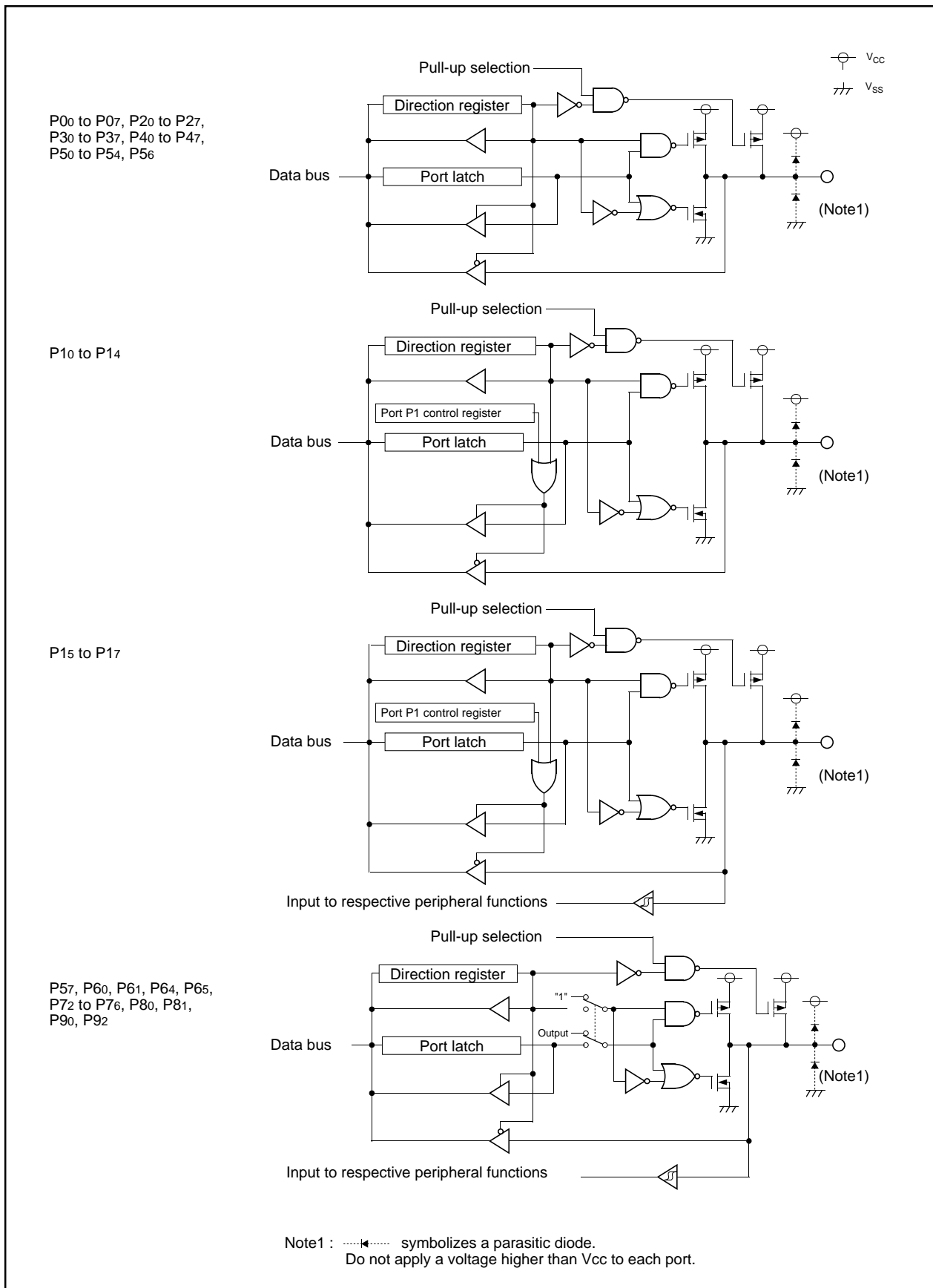
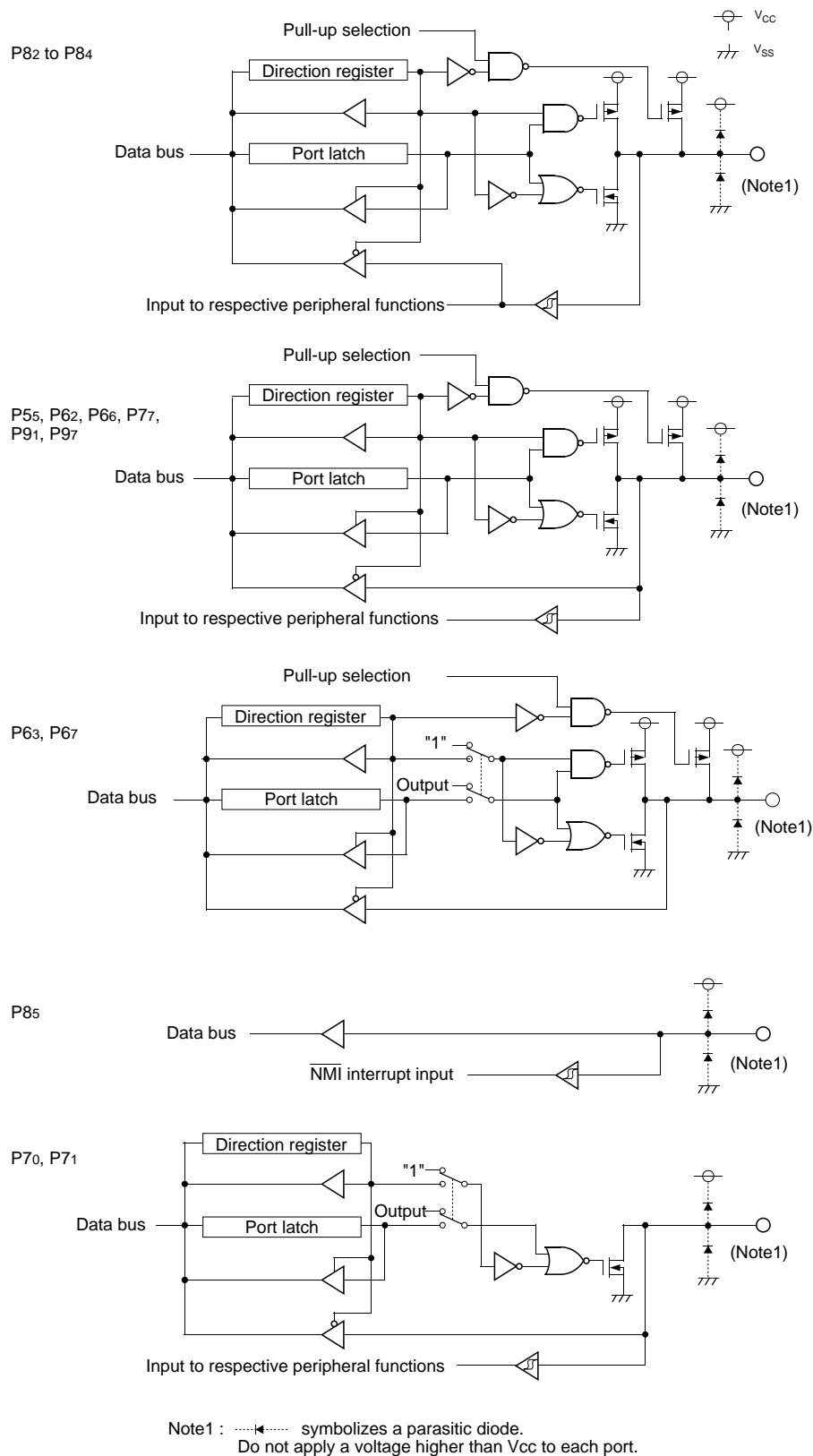
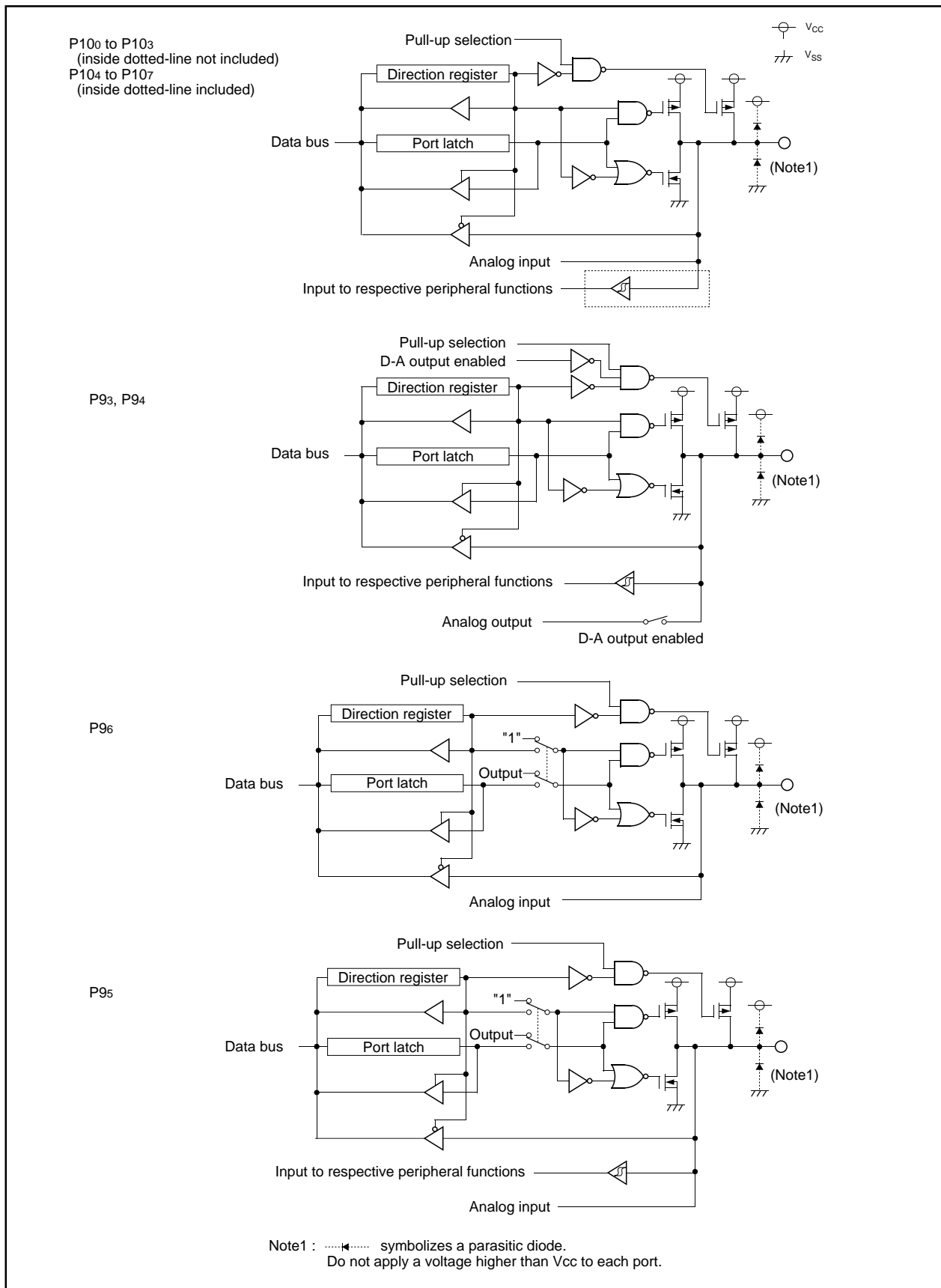


Figure 2.16.1 Programmable I/O ports (1)

**M306H2FCFP**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER**Figure 2.16.2 Programmable I/O ports (2)**

**M306H2FCFP**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER**Figure 2.16.3 Programmable I/O ports (3)**

## M306H2FCFP

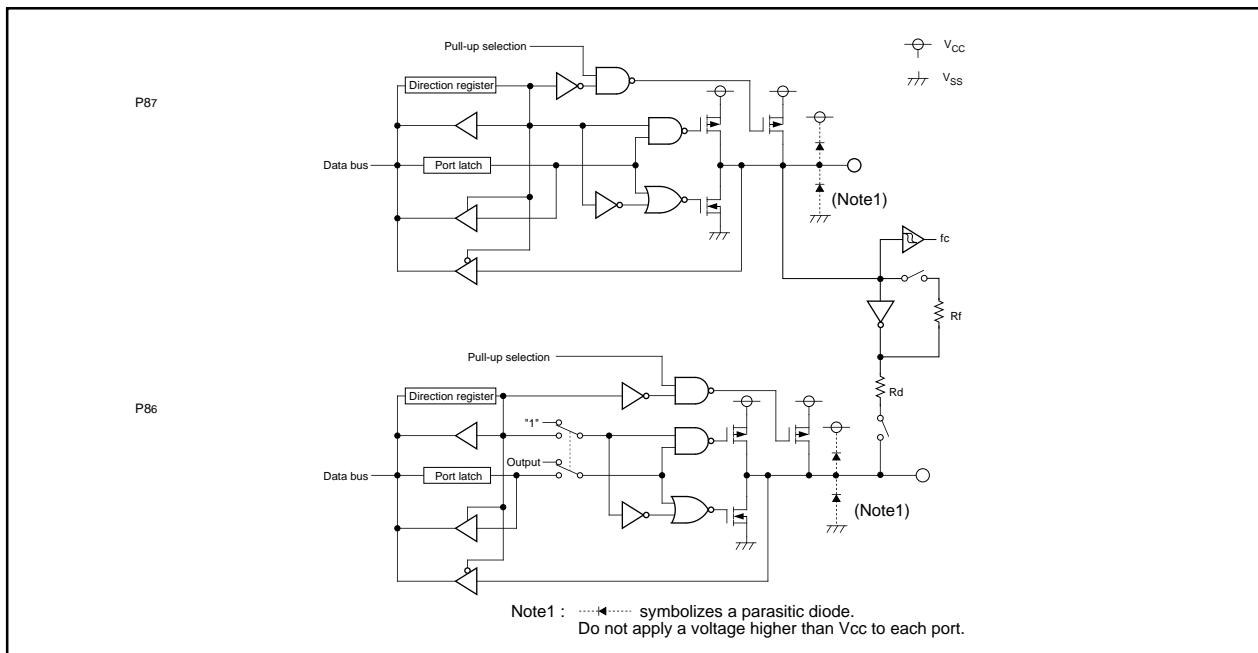
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

Figure 2.16.4 Programmable I/O ports (4)

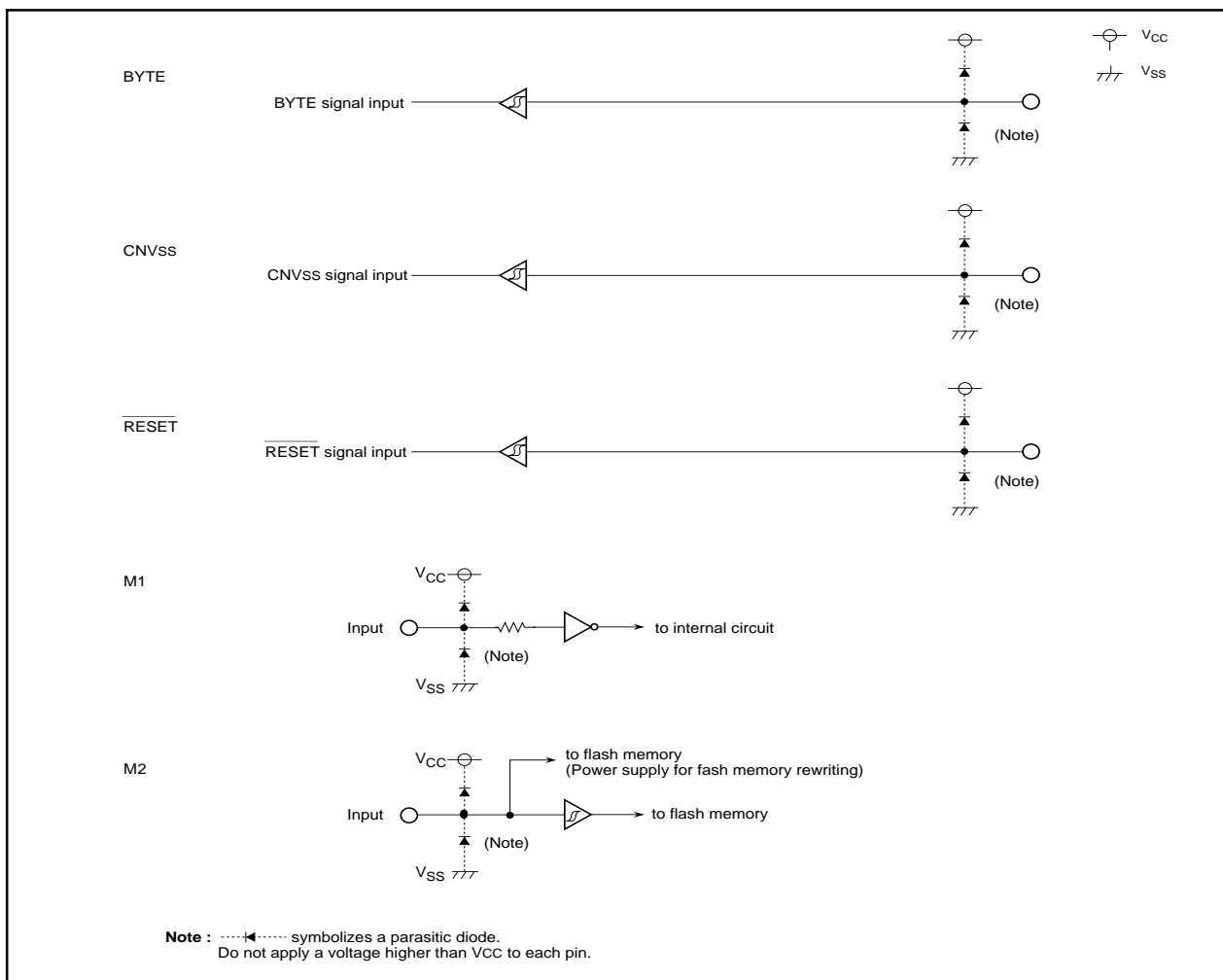


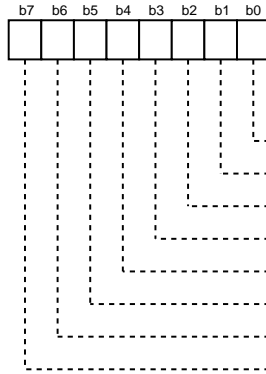
Figure 2.16.5 I/O pins

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## Port Pi direction register (Note 1, 2)

Symbol	Address	When reset
PD <sub>i</sub> (i = 0 to 10, except 8)	03E2 <sub>16</sub> , 03E3 <sub>16</sub> , 03E6 <sub>16</sub> , 03E7 <sub>16</sub> , 03EA <sub>16</sub> 03EB <sub>16</sub> , 03EE <sub>16</sub> , 03EF <sub>16</sub> , 03F3 <sub>16</sub> , 03F6 <sub>16</sub>	00 <sub>16</sub> 00 <sub>16</sub>

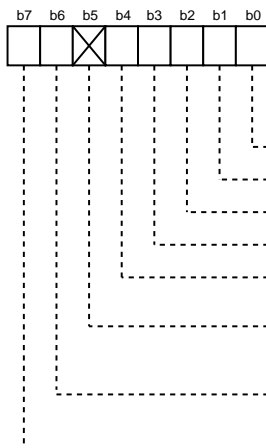


Bit symbol	Bit name	Function	R/W
PD <sub>i</sub> _0	Port Pi0 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (i = 0 to 10 except 8)	○/○
PD <sub>i</sub> _1	Port Pi1 direction register		○/○
PD <sub>i</sub> _2	Port Pi2 direction register		○/○
PD <sub>i</sub> _3	Port Pi3 direction register		○/○
PD <sub>i</sub> _4	Port Pi4 direction register		○/○
PD <sub>i</sub> _5	Port Pi5 direction register		○/○
PD <sub>i</sub> _6	Port Pi6 direction register		○/○
PD <sub>i</sub> _7	Port Pi7 direction register		○/○

Note 1: Set bit 2 of protect register (address 000A<sub>16</sub>) to "1" before rewriting to the port P9 direction register.

Note 2: In memory expansion mode, the contents of corresponding port Pi direction register of pins A<sub>0</sub> to A<sub>19</sub>, D<sub>0</sub> to D<sub>15</sub>,  $\overline{\text{CS}}_0$  to  $\overline{\text{CS}}_3$ ,  $\overline{\text{RD}}$ ,  $\overline{\text{WRL}}/\overline{\text{WR}}$ ,  $\overline{\text{WRH}}/\overline{\text{BHE}}$ ,  $\overline{\text{ALE}}$ ,  $\overline{\text{RDY}}$ ,  $\overline{\text{HOLD}}$ ,  $\overline{\text{HLDA}}$  and  $\overline{\text{BCLK}}$  cannot be modified.

## Port P8 direction register



Symbol	Address	When reset
PD8	03F2 <sub>16</sub>	00X00000 <sub>2</sub>

Bit symbol	Bit name	Function	R/W
PD8_0	Port P80 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○ ○
PD8_1	Port P81 direction register		○ ○
PD8_2	Port P82 direction register		○ ○
PD8_3	Port P83 direction register		○ ○
PD8_4	Port P84 direction register		○ ○
Nothing is assigned. In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.			— —
PD8_6	Port P86 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○ ○
PD8_7	Port P87 direction register		○ ○

Figure 2.16.6 Direction register



M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

Port Pi register (Note 2)

<div><div>b7b6b5b4b3b2b1b0</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div>								Symbol	Address	When reset																										
Pi (i = 0 to 10, except 8)								03E016, 03E116, 03E416, 03E516, 03E816 03E916, 03EC16, 03ED16, 03F116, 03F416	Indeterminate Indeterminate																											
<table><tr><th>Bit symbol</th><th>Bit name</th><th>Function</th><th>R/W</th></tr><tr><td>Pi_0</td><td>Port Pi0 register</td><td rowspan="8">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 (Note)  (i = 0 to 10 except 8)</td><td><input type="radio"/> <input type="radio"/></td></tr><tr><td>Pi_1</td><td>Port Pi1 register</td><td><input type="radio"/> <input type="radio"/></td></tr><tr><td>Pi_2</td><td>Port Pi2 register</td><td><input type="radio"/> <input type="radio"/></td></tr><tr><td>Pi_3</td><td>Port Pi3 register</td><td><input type="radio"/> <input type="radio"/></td></tr><tr><td>Pi_4</td><td>Port Pi4 register</td><td><input type="radio"/> <input type="radio"/></td></tr><tr><td>Pi_5</td><td>Port Pi5 register</td><td><input type="radio"/> <input type="radio"/></td></tr><tr><td>Pi_6</td><td>Port Pi6 register</td><td><input type="radio"/> <input type="radio"/></td></tr><tr><td>Pi_7</td><td>Port Pi7 register</td><td><input type="radio"/> <input type="radio"/></td></tr></table>								Bit symbol	Bit name	Function	R/W	Pi_0	Port Pi0 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 (Note)  (i = 0 to 10 except 8)	<input type="radio"/> <input type="radio"/>	Pi_1	Port Pi1 register	<input type="radio"/> <input type="radio"/>	Pi_2	Port Pi2 register	<input type="radio"/> <input type="radio"/>	Pi_3	Port Pi3 register	<input type="radio"/> <input type="radio"/>	Pi_4	Port Pi4 register	<input type="radio"/> <input type="radio"/>	Pi_5	Port Pi5 register	<input type="radio"/> <input type="radio"/>	Pi_6	Port Pi6 register	<input type="radio"/> <input type="radio"/>	Pi_7	Port Pi7 register	<input type="radio"/> <input type="radio"/>
Bit symbol	Bit name	Function	R/W																																	
Pi_0	Port Pi0 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 (Note)  (i = 0 to 10 except 8)	<input type="radio"/> <input type="radio"/>																																	
Pi_1	Port Pi1 register		<input type="radio"/> <input type="radio"/>																																	
Pi_2	Port Pi2 register		<input type="radio"/> <input type="radio"/>																																	
Pi_3	Port Pi3 register		<input type="radio"/> <input type="radio"/>																																	
Pi_4	Port Pi4 register		<input type="radio"/> <input type="radio"/>																																	
Pi_5	Port Pi5 register		<input type="radio"/> <input type="radio"/>																																	
Pi_6	Port Pi6 register		<input type="radio"/> <input type="radio"/>																																	
Pi_7	Port Pi7 register		<input type="radio"/> <input type="radio"/>																																	

Note 1: Since P70 and P71 are N-channel open drain ports, the data is high-impedance.  
Note 2: In memory expansion mode, the contents of corresponding port Pi direction register of pins A0 to A19, D0 to D15, CS0 to CS3, RD, WRL/WR, WRH/BHE, ALE, RDY, HOLD, HLDA and BCLK cannot be modified.

Port P8 register

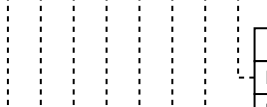
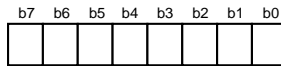
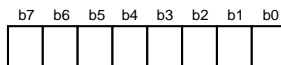
b7	b6	b5	b4	b3	b2	b1	b0	Symbol P8	Address 03F0 <sub>16</sub>	When reset Indeterminate																																				
								<table><thead><tr><th>Bit symbol</th><th>Bit name</th><th>Function</th><th>R</th><th>W</th></tr></thead><tbody><tr><td>P8_0</td><td>Port P8<sub>0</sub> register</td><td rowspan="8">Data is input and output to and from each pin by reading and writing to and from each corresponding bit (except for P8<sub>5</sub>) 0 : "L" level data 1 : "H" level data</td><td><input type="radio"/></td><td><input type="radio"/></td></tr><tr><td>P8_1</td><td>Port P8<sub>1</sub> register</td><td><input type="radio"/></td><td><input type="radio"/></td></tr><tr><td>P8_2</td><td>Port P8<sub>2</sub> register</td><td><input type="radio"/></td><td><input type="radio"/></td></tr><tr><td>P8_3</td><td>Port P8<sub>3</sub> register</td><td><input type="radio"/></td><td><input type="radio"/></td></tr><tr><td>P8_4</td><td>Port P8<sub>4</sub> register</td><td><input type="radio"/></td><td><input type="radio"/></td></tr><tr><td>P8_5</td><td>Port P8<sub>5</sub> register</td><td><input type="radio"/></td><td><input checked="" type="radio"/></td></tr><tr><td>P8_6</td><td>Port P8<sub>6</sub> register</td><td><input type="radio"/></td><td><input type="radio"/></td></tr><tr><td>P8_7</td><td>Port P8<sub>7</sub> register</td><td><input type="radio"/></td><td><input type="radio"/></td></tr></tbody></table>	Bit symbol	Bit name	Function	R	W	P8_0	Port P8 <sub>0</sub> register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit (except for P8 <sub>5</sub> ) 0 : "L" level data 1 : "H" level data	<input type="radio"/>	<input type="radio"/>	P8_1	Port P8 <sub>1</sub> register	<input type="radio"/>	<input type="radio"/>	P8_2	Port P8 <sub>2</sub> register	<input type="radio"/>	<input type="radio"/>	P8_3	Port P8 <sub>3</sub> register	<input type="radio"/>	<input type="radio"/>	P8_4	Port P8 <sub>4</sub> register	<input type="radio"/>	<input type="radio"/>	P8_5	Port P8 <sub>5</sub> register	<input type="radio"/>	<input checked="" type="radio"/>	P8_6	Port P8 <sub>6</sub> register	<input type="radio"/>	<input type="radio"/>	P8_7	Port P8 <sub>7</sub> register	<input type="radio"/>	<input type="radio"/>
Bit symbol	Bit name	Function	R	W																																										
P8_0	Port P8 <sub>0</sub> register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit (except for P8 <sub>5</sub> ) 0 : "L" level data 1 : "H" level data	<input type="radio"/>	<input type="radio"/>																																										
P8_1	Port P8 <sub>1</sub> register		<input type="radio"/>	<input type="radio"/>																																										
P8_2	Port P8 <sub>2</sub> register		<input type="radio"/>	<input type="radio"/>																																										
P8_3	Port P8 <sub>3</sub> register		<input type="radio"/>	<input type="radio"/>																																										
P8_4	Port P8 <sub>4</sub> register		<input type="radio"/>	<input type="radio"/>																																										
P8_5	Port P8 <sub>5</sub> register		<input type="radio"/>	<input checked="" type="radio"/>																																										
P8_6	Port P8 <sub>6</sub> register		<input type="radio"/>	<input type="radio"/>																																										
P8_7	Port P8 <sub>7</sub> register		<input type="radio"/>	<input type="radio"/>																																										

Figure 2.16.7 Port register

**M306H2FCFP**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER**Pull-up control register 0**Symbol  
PUR0Address  
03FC<sub>16</sub>When reset  
00<sub>16</sub>

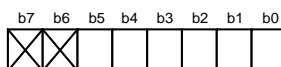
Bit symbol	Bit name	Function	R	W
PU00	P00 to P03 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU01	P04 to P07 pull-up		○	○
PU02	P10 to P13 pull-up		○	○
PU03	P14 to P17 pull-up		○	○
PU04	P20 to P23 pull-up		○	○
PU05	P24 to P27 pull-up		○	○
PU06	P30 to P33 pull-up		○	○
PU07	P34 to P37 pull-up		○	○

**Pull-up control register 1**Symbol  
PUR1Address  
03FD<sub>16</sub>When reset  
00<sub>16</sub>

Bit symbol	Bit name	Function	R	W
PU10	P40 to P43 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU11	P44 to P47 pull-up		○	○
PU12	P50 to P53 pull-up		○	○
PU13	P54 to P57 pull-up		○	○
PU14	P60 to P63 pull-up		○	○
PU15	P64 to P67 pull-up		○	○
PU16	P70 to P73 pull-up (Note 1)		○	○
PU17	P74 to P77 pull-up		○	○

Note 1: Since P70 and P71 are N-channel open drain ports, pull-up is not available for them.

Note 2: In memory expansion mode, the content of these bits can be changed, but the pull-up resistance is not connected.

**Pull-up control register 2**Symbol  
PUR2Address  
03FE<sub>16</sub>When reset  
00<sub>16</sub>

Bit symbol	Bit name	Function	R	W
PU20	P80 to P83 pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU21	P84 to P87 pull-up (Except P85)		○	○
PU22	P90 to P93 pull-up		○	○
PU23	P94 to P97 pull-up		○	○
PU24	P100 to P103 pull-up		○	○
PU25	P104 to P107 pull-up		○	○
Nothing is assigned.			—	—
In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".			—	—

**Figure 2.16.8 Pull-up control register**

M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

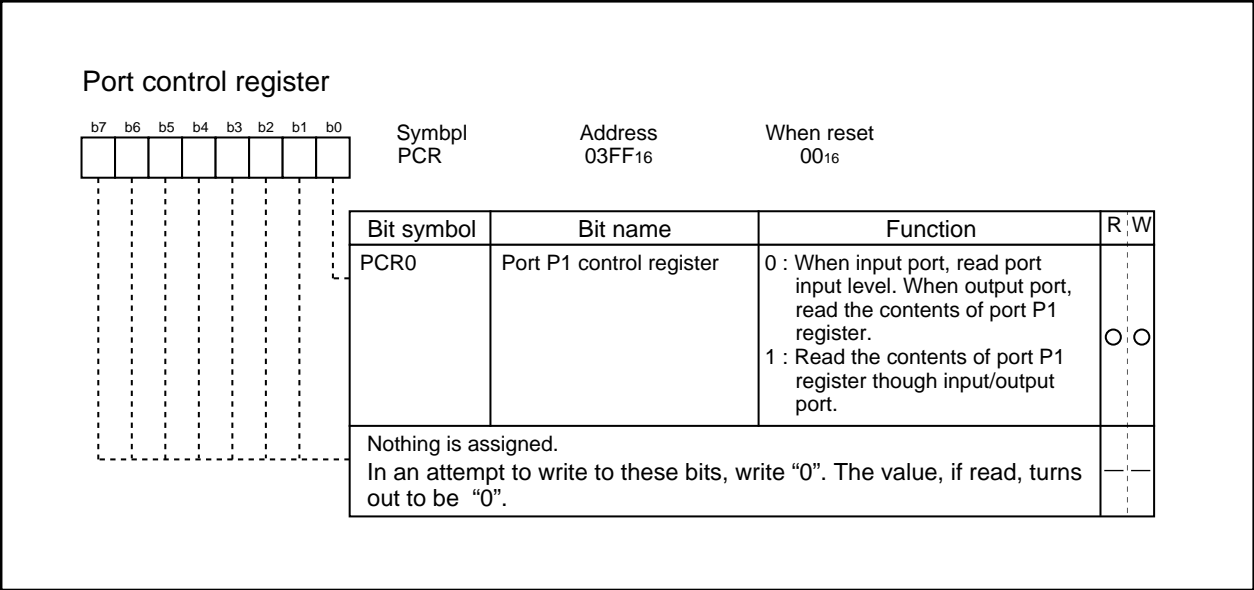


Figure 2.16.9 Port control register

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

### Table 2.16.1 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 or VCC via a resistor; or after setting for output mode, leave these pins open.
XOUT(Note)	Open
$\overline{\text{NMI}}$	Connect via register to VCC (pull-up)
AVCC	Connect to VCC
AVSS, VREF, BYTE	Connect to VSS

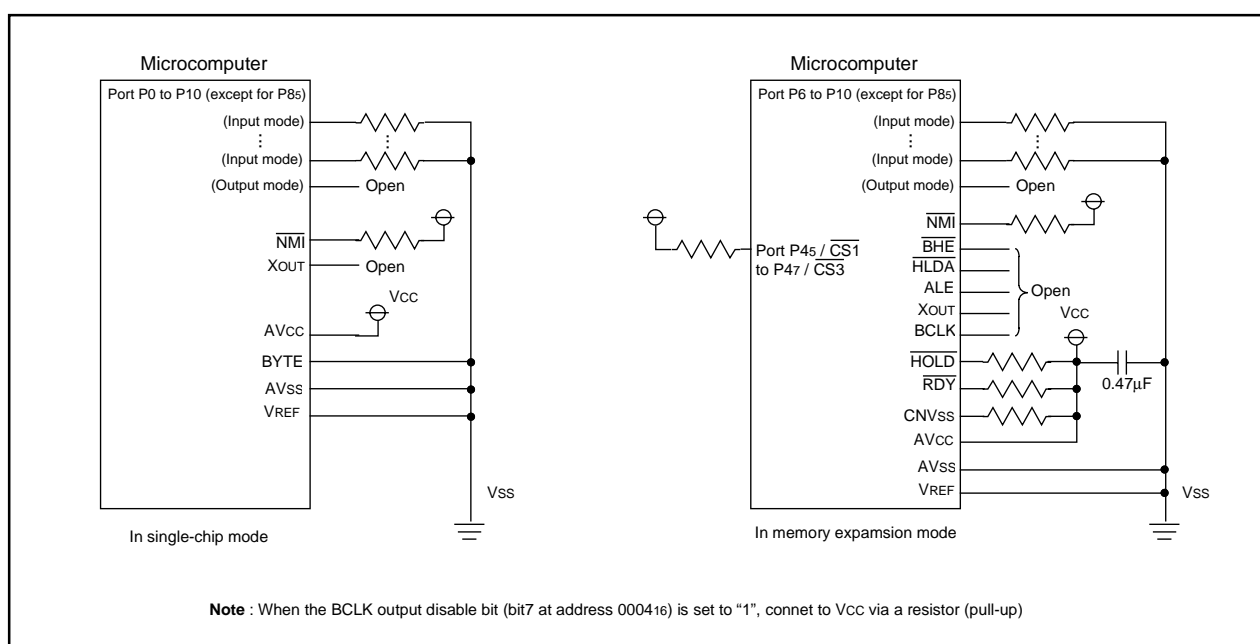
Note: With external clock input to XIN pin.

### Table 2.16.2 Example connection of unused pins in memory expansion mode

Pin name	Connection
Ports P6 to P10 (excluding P8s)	After setting for input mode, connect every pin to Vss or Vcc via a resistor; or after setting for output mode, leave these pins open.
P4s/ $\overline{\text{CS1}}$ to P47/ $\overline{\text{CS3}}$	Sets ports to input mode, sets bits $\overline{\text{CS1}}$ through $\overline{\text{CS3}}$ to 0, and connects to Vcc via resistors (pull-up).
$\overline{\text{BHE}}$ , ALE, $\overline{\text{HLD\bar{A}}}$ , XOUT(Note1), BCLK(Note2)	Open
$\overline{\text{HOLD}}$ , $\overline{\text{RDY}}$ , $\overline{\text{NMI}}$	Connect via 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).



**Figure 2.16.10 Example connection of unused pins**

### 3. USAGE PRECAUTION

#### Timer A (timer mode)

- (1) Reading the timer Ai register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Ai register with the reload timing gets "FFFF<sub>16</sub>". Reading the timer Ai register after setting a value in the timer Ai register with a count halted but before the counter starts counting gets a proper value.

#### Timer A (event counter mode)

- (1) Reading the timer Ai register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Ai register with the reload timing gets "FFFF<sub>16</sub>" by underflow or "0000<sub>16</sub>" by overflow. Reading the timer Ai register after setting a value in the timer Ai register with a count halted but before the counter starts counting gets a proper value.
- (2) When stop counting in free run type, set timer again.
- (3) In the case of using "Event counter mode" as "Free-Run type" for timer A, the timer register contents may be unknown when counting begins. If the timer register is set before counting has started, then the starting value will be unknown.

This issue will occur only for the "Event counter mode" operating as "Free-Run type". The value of the timer register will not be unknown during counting.

#### Timer A (one-shot timer mode)

- (1) Setting the count start flag to "0" while a count is in progress causes as follows:
  - The counter stops counting and a content of reload register is reloaded.
  - The TAIOUT pin outputs "L" level.
  - The interrupt request generated and the timer Ai interrupt request bit goes to "1".
- (2) The timer Ai interrupt request bit goes to "1" if the timer's operation mode is set using any of the following procedures:
  - Selecting one-shot timer mode after reset.
  - Changing operation mode from timer mode to one-shot timer mode.
  - Changing operation mode from event counter mode to one-shot timer mode.

Therefore, to use timer Ai interrupt (interrupt request bit), set timer Ai interrupt request bit to "0" after the above listed changes have been made.

#### Timer A (pulse width modulation mode)

- (1) The timer Ai interrupt request bit becomes "1" if setting operation mode of the timer in compliance with any of the following procedures:
  - Selecting PWM mode after reset.
  - Changing operation mode from timer mode to PWM mode.
  - Changing operation mode from event counter mode to PWM mode.

Therefore, to use timer Ai interrupt (interrupt request bit), set timer Ai interrupt request bit to "0" after the above listed changes have been made.
- (2) Setting the count start flag to "0" while PWM pulses are being output causes the counter to stop counting. If the TAIOUT pin is outputting an "H" level in this instance, the output level goes to "L", and the timer Ai interrupt request bit goes to "1". If the TAIOUT pin is outputting an "L" level in this instance, the level does not change, and the timer Ai interrupt request bit does not become "1".

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## Timer B (timer mode, event counter mode)

- (1) Reading the timer Bi register while a count is in progress allows reading , with arbitrary timing, the value of the counter. Reading the timer Bi register with the reload timing gets "FFFF16". Reading the timer Bi register after setting a value in the timer Bi register with a count halted but before the counter starts counting gets a proper value.

## Timer B (pulse period/pulse width measurement mode)

- (1) If changing the measurement mode select bit is set after a count is started, the timer Bi interrupt request bit goes to "1".
- (2) When the first effective edge is input after a count is started, an indeterminate value is transferred to the reload register. At this time, timer Bi interrupt request is not generated.

## A-D Converter

- (1) Write to each bit (except bit 6) of A-D control register 0, to each bit of A-D control register 1, and to bit 0 of A-D control register 2 when A-D conversion is stopped (before a trigger occurs).  
In particular, when the Vref connection bit is changed from "0" to "1", start A-D conversion after an elapse of 1  $\mu$ s or longer.
- (2) When changing A-D operation mode, select analog input pin again.
- (3) Using one-shot mode or single sweep mode  
Read the correspondence A-D register after confirming A-D conversion is finished. (It is known by A-D conversion interrupt request bit.)
- (4) Using repeat mode, repeat sweep mode 0 or repeat sweep mode 1  
Use the undivided main clock as the internal CPU clock.

## Stop Mode and Wait Mode

- (1) When returning from stop mode by hardware reset,  $\overline{\text{RESET}}$  pin must be set to "L" level until main clock oscillation is stabilized.
- (2) When switching to either wait mode or stop mode, instructions occupying four bytes either from the WAIT instruction or from the instruction that sets the every-clock stop bit to "1" within the instruction queue are prefetched and then the program stops. So put at least four NOPs in succession either to the WAIT instruction or to the instruction that sets the every-clock stop bit to "1".
- (3) When the MCU running in low-speed or low power dissipation mode, do not enter WAIT mode with WAIT peripheral function clock stop bit set to "1".

## Interrupts

- (1) Reading address 0000016
  - When maskable interrupt is occurred, CPU read the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence.  
The interrupt request bit of the certain interrupt written in address 0000016 will then be set to "0".  
Reading address 0000016 by software sets enabled highest priority interrupt source request bit to "0".  
Though the interrupt is generated, the interrupt routine may not be executed.  
Do not read address 0000016 by software.
- (2) Setting the stack pointer
  - The value of the stack pointer immediately after reset is initialized to 000016. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt.  
When using the  $\overline{\text{NMI}}$  interrupt, initialize the stack point at the beginning of a program. Concerning the first in struction immediately after reset, generating any interrupts including the  $\overline{\text{NMI}}$  interrupt is prohib ited.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

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

- As for the  $\overline{\text{NMI}}$  interrupt 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.
- Do not get either into stop mode with the  $\overline{\text{NMI}}$  pin set to "L".

## (4) External interrupt

- 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".

## (5) Rewrite the interrupt control register

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

### Example 1:

```

INT_SWITCH1:
  FCLR  I          ; Disable interrupts.
  AND.B #00h, 0055h ; Clear TA0IC int. priority level and int. request bit.
  NOP                      ; 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 are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

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

Instructions : AND, OR, BCLR, BSET

## Electric Characteristic Differences Between Mask ROM and Flash Memory Version MCUs

There are differences in electric characteristics, operation margin, noise immunity, and noise radiation between Mask ROM and Flash Memory version MCUs due to the difference in the manufacturing processes.

When manufacturing an application system with the Flash Memory version and then switching to use of the Mask ROM version, please perform sufficient evaluations for the commercial samples of the Mask ROM version.

# **M306H2FCFP**

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## **Other Notes**

(1) Timing of power supplying

The power need to supply to VCC, VDD1, VDD2, VDD3 and AVCC at a time. While operating, must set same voltage.

(2) Power supply noise and latch-up

In order to avoid power supply noise and latch-up, connect a bypass capacitor (more than 0.1μF) directly between the VCC pin and Vss pin, VDD1 pin and Vss1 pin, VDD2 pin and Vss2 pin, VDD3 pin and Vss3 pin, AVCC pin and AVss pin using a heavy wire.

(3) When oscillation circuit stop for data slicer

Expansion register XTAL-VCO, PDC\_VCO\_ON, VPS\_VCO\_ON is set at "L", when the data slicer is not used, and the oscillation is stopped. When starting oscillation again, set data at the following order.

- (a) Set expansion register XTAL-VCO = "H".
- (b) Set expansion register PDC\_VCO\_ON, VPS\_VCO\_ON = "H".
- (c) 60 ms or more is a waiting state (stability period of internal oscillation circuit + data slice preparation).

To operate slice RAM, set expansion register XTAL\_VCO = "H". And input 4.43 MHz sub carrier frequency clock from the FSCIN pin.

Access the memories after wating for 20 ms certainly when resuming synchronous oscillation from the off state, and begin to input clock into the FSCIN pin.

(4) At stop mode (clock is stopped)

Set each input pins to as follows.

- (a) Set M1 pin = Vss.
  - (b) Stop the FSCIN pin input.
  - (c) Set expansion register STBY0 and STBY1 = "H".
- Set all expansion registers to "L" except for the superscription register.

(5) When operation start from stand-by mode (clock is stopped)

Input FSCIN pin clock after set "L" to register STBY0 and STBY1.

At next, set expansion register as notes (3).



M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

(6) Notes on operating with a low supply voltage (VCC = 3.0 V)

When in single-chip mode, this product can operate with a low supply voltage (VCC = 3.0 V) only during low power dissipation mode. Before operating with a low supply voltage, always be sure to set the relevant register bits to select low power dissipation mode (BCLK : f(XCIN), main clock XIN : stop, subclock XCIN : oscillating). Then reduce the power supply voltage VCC to 3.0 V.

Also, when returning to normal operation, raise the power supply voltage to 5V while in low power consumption mode before entering normal operation mode.

When moving from any operation mode to another, make sure a state transition occurs according to the state transition diagram (Figure 2.5.5) in Section 2.5.7, "Power control."

The status of the power supply voltage VCC during operation mode transition is shown in Figure 3.1 below.

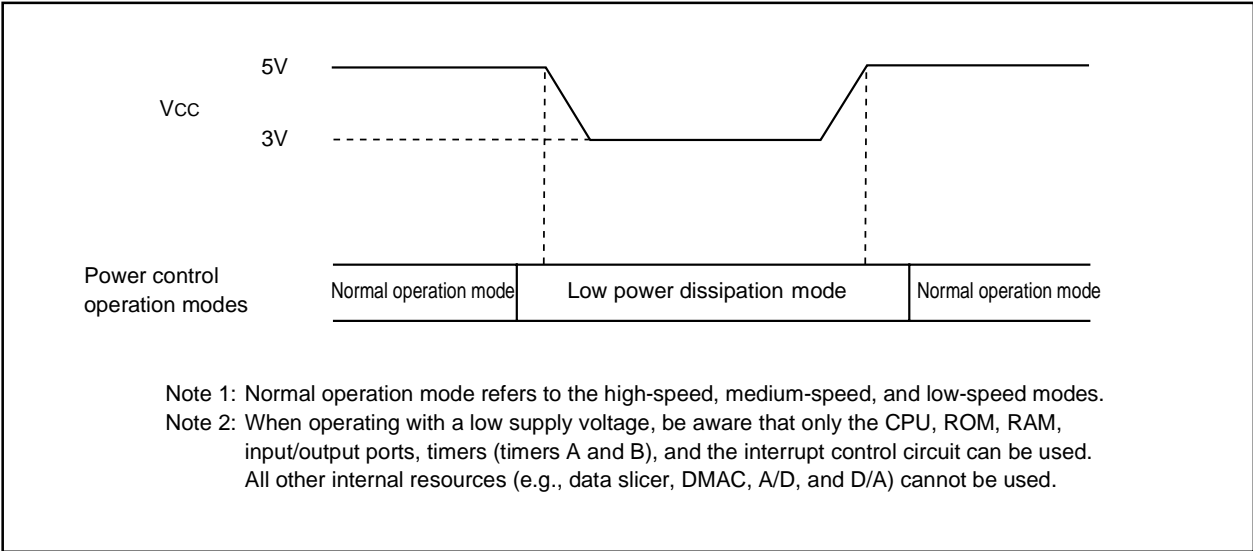


Figure 3.1 Status of the power supply voltage VCC during operation mode transition

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 4. ELECTRICAL CHARACTERISTICS

Table 4.1 Absolute maximum ratings

Symbol	Parameter		Condition	Rated value	Unit
V <sub>cc</sub>	Supply voltage		V <sub>cc</sub> =AV <sub>cc</sub>	-0.3 to 5.75	V
AV <sub>cc</sub>	Analog supply voltage		V <sub>cc</sub> =AV <sub>cc</sub>	-0.3 to 5.75	V
M2	Supply voltage for program/erase			-0.3 to 5.75	V
V <sub>i</sub>	Input voltage	RESET, CNV <sub>ss</sub> , BYTE, P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>2</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , VREF, XIN, M1		-0.3 to V <sub>cc</sub> +0.3	V
		P7 <sub>0</sub> , P7 <sub>1</sub>		-0.3 to 5.75	V
V <sub>o</sub>	Output voltage	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>2</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>4</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , XOUT, P11		-0.3 to V <sub>cc</sub> +0.3	V
		P7 <sub>0</sub> , P7 <sub>1</sub>		-0.3 to 5.75	V
P <sub>d</sub>	Power dissipation		T <sub>a</sub> =25 °C	1000	mW
T <sub>opr</sub>	Operating ambient temperature			-20 to 70	°C
T <sub>stg</sub>	Storage temperature			-40 to 125	°C

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

**Tabl 4.2 Recommended operating conditions (referenced to  $V_{CC} = 4.75V$  to  $5.25V$  at  $T_a = -20$  to  $70^\circ C$  unless otherwise specified)**

Symbol	Parameter		Standard			Unit	
			Min	Typ.	Max.		
Vcc	Supply voltage		4.75	5.0	5.25	V	
AVcc	Analog supply voltage			Vcc		V	
Vss	Supply voltage			0		V	
AVss	Analog supply voltage			0		V	
VIH	HIGH input voltage	P70 to P77, P80 to P87, P90 to P97, P100 to P107, P31 to P37, P40 to P47, P50 to P57, P60 to P67, XIN, RESET, CNVss, BYTE, M1,	0.8Vcc		Vcc	V	
		P00 to P07, P10 to P17, P20 to P27, P30 (during single-chip mode)	0.8Vcc		Vcc	V	
		P00 to P07, P10 to P17, P20 to P27, P30 (data input function during memory expansion mode)	0.5Vcc		Vcc	V	
VIL	LOW input voltage	P70 to P77, P80 to P87, P90 to P97, P100 to P107, P31 to P37, P40 to P47, P50 to P57, P60 to P67, XIN, RESET, CNVss, BYTE, M1	0		0.2Vcc	V	
		P00 to P07, P10 to P17, P20 to P27, P30 (during single-chip mode)	0		0.2Vcc	V	
		P00 to P07, P10 to P17, P20 to P27, P30 (data input function during memory expansion mode)	0		0.16Vcc	V	
VCVIN	Composite video input voltage CVIN, SYNCIN			2V P-P		V	
VFSCIN	Input voltage FSCIN(Note 1)		0.3V P-P		4.0V P-P	V	
IOH (peak)	HIGH peak output current (Note 2.3)	P00to P07, P10 to P17,P20 to P27, P30 to P37, P40 to P47, P50 to P57,P60 to P67, P72 to P77, P80 to P84, P86, P87, P90 to P97, P100 to P107, P11			-10.0	mA	
IOH (avg)	HIGH average output current	P00 to P07, P10 to P17,P20 to P27,P30 to P37, P40 to P47, P50 to P57,P60 to P67,P72 to P77, P80 to P84, P86, P87, P90 to P97, P100 to P107, P11			-5.0	mA	
IOL (peak)	LOW peak output current	P00 to P07, P10 to P17,P20 to P27,P30 to P37, P40 to P47, P50 to P57, P60 to P67,P70 to P77, P80 to P84, P86, P87, P90 to P97, P100 to P107, P11			10.0	mA	
IOL (avg)	LOW average output current	P00 to P07, P10 to P17,P20 to P27,P30 to P37, P40 to P47, P50 to P57,P60 to P67,P70 to P77, P80 to P84, P86, P87, P90 to P97, P100 to P107, P11,			5.0	mA	
f (XIN)	Main clock input oscillation frequency	<div>No wait</div> <div>with wait</div> <div>Vcc=4.75V to 5.25V</div>	0		10	MHz	
f (XCIN)	Subclock oscillation frequency		Vcc=2.80V to 5.25V (see note 4)		32.768	50	kHz
f (FSCIN)	Oscillation frequency for synchronous signal(Duty 40% to 60%)			4.434		MHz	

Note 1: Noise component is within 30mV.

Note 2: The mean output current is the mean value within 100ms.

Note 3: The total IOL (peak) for ports P0, P1, P2, P86, P87, P9, and P10 must be 80mA max. The total IOH (peak) for ports P0, P1, P2, P86, P87, P9, and P10 must be 80mA max. The total IOL (peak) for ports P3, P4, P5, P6, P7, and P80 to P84 must be 80mA max. The total IOH (peak) for ports P3, P4, P5, P6, P72 to P77, and P80 to P84 must be 80mA max.

Note 4: Use the low power dissipation mode.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

**Table 4.3 Electrical characteristics (1)V<sub>CC</sub> = 5V (referenced to V<sub>CC</sub> = 5V, V<sub>SS</sub> = 0V at Ta = 25°C, f(X<sub>IN</sub>) =10MHz unless otherwise specified)**

Symbol	Parameter		Measuring condition	Standard			Unit
				Min	Typ.	Max.	
V <sub>OH</sub>	HIGH output voltage	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>2</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>4</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , P11	I <sub>OH</sub> =-5mA	3.0			V
V <sub>OH</sub>	HIGH output voltage	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>2</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>4</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , P11	I <sub>OH</sub> =-200μA	4.7			V
V <sub>OH</sub>	HIGH output voltage	LP2 to LP4	V <sub>CC</sub> =4.75V, I <sub>OH</sub> =-0.05mA	3.75			V
V <sub>OH</sub>	HIGH output voltage	X <sub>OUT</sub>	HIGHPOWER	I <sub>OH</sub> =-1mA	3.0		V
			LOWPOWER	I <sub>OH</sub> =-0.5mA	3.0		
	HIGH output voltage	X <sub>COU</sub> T	HIGHPOWER	With no load applied		3.0	V
			LOWPOWER	With no load applied		1.6	
V <sub>OL</sub>	LOW output voltage	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>4</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , P11	I <sub>OL</sub> =5mA			2.0	V
V <sub>OL</sub>	LOW output voltage	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>4</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , P11	I <sub>OL</sub> =200μA			0.45	V
V <sub>OL</sub>	LOW output voltage	LP2 to LP4	V <sub>CC</sub> =4.75V, I <sub>OL</sub> =0.05mA			0.4	V
V <sub>OL</sub>	LOW output voltage	X <sub>OUT</sub>	HIGHPOWER	I <sub>OL</sub> =1mA		2.0	V
			LOWPOWER	I <sub>OL</sub> =0.5mA		2.0	
	LOW output voltage	X <sub>COU</sub> T	HIGHPOWER	With no load applied		0	V
			LOWPOWER	With no load applied		0	
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	HOLD, RDY, TA0 <sub>IN</sub> to TA4 <sub>IN</sub> , TB0 <sub>IN</sub> to TB2 <sub>IN</sub> , INT0 to INT5, AD <sub>TRG</sub> , CTS1, CLK1, NMI, TA2 <sub>OUT</sub> to TA4 <sub>OUT</sub> , KI0 to KI3		0.2		0.8	V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	CTS0, CLK0		0.2		1.4	V
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	RESET		0.2		1.8	V
I <sub>IH</sub>	HIGH input current	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE, M1	V <sub>I</sub> =5V			5.0	μA
I <sub>IL</sub>	LOW input current	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>0</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE, M1	V <sub>I</sub> =0V			-5.0	μA
R <sub>PULLUP</sub>	Pull-up resistance	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>2</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>4</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub>	V <sub>I</sub> =0V	30.0	50.0	167.0	kΩ
V <sub>SYN</sub> CIN	Sync voltage amplitude			0.3	0.6	1.2	V
V <sub>dat</sub> (text)	Teletext data voltage amplitude			0.6	0.9	1.4	V
Δ f / f	Range for display oscillator circuit			±7			%
f <sub>H</sub>	Horizontal synchronous signal frequency			14.6	15.625	17.0	kHz

# M306H2FCFP

 SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

**Table 4.4 Electrical characteristics (2)V<sub>CC</sub> = 3V (referenced to V<sub>CC</sub>=3V,V<sub>SS</sub>=0V,T<sub>a</sub>=25°C,  
f(XCIN)=32KHz unless otherwise specified)**

Symbol	Parameter		Measuring condition	Standard			Unit
				Min	Typ.	Max.	
V <sub>OH</sub>	HIGH output voltage	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P72 to P77, P80 to P84, P86, P87, P90 to P97, P100 to P107 P11	I <sub>OH</sub> = -150μA	2.5			V
V <sub>OH</sub>	HIGH output voltage	XCOUT	HIGHPOWER		3.0		V
			LOWPOWER		1.6		
V <sub>OL</sub>	LOW output voltage	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P84, P86, P87, P90 to P97, P100 to P107 P11	I <sub>OL</sub> =150μA			0.5	V
V <sub>OL</sub>	LOW output voltage	XCOUT	HIGHPOWER		0		V
			LOWPOWER		0		
V <sub>T+</sub> -V <sub>T-</sub>	Hysteresis	TA0IN to TA4IN, TB0IN to TB2IN, INT0 to INT5, TA2OUT to TA4OUT, NMI, KI0 to KI3		0.2		0.8	V
I <sub>IH</sub>	HIGH input voltage	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P107, XIN, RESET, CNVss, BYTE, M1	V <sub>I</sub> =3V			4.0	μA
I <sub>IL</sub>	LOW input voltage	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P107, XIN, RESET, CNVss, BYTE, M1	V <sub>I</sub> =0V			-4.0	μA
R <sub>PULLUP</sub>	Pull-up resistance	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P72 to P77, P80 to P84, P86, P87, P90 to P97, P100 to P107	V <sub>I</sub> =0V	66.0	120.0	500.0	kΩ

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

**Table 4.5 Electrical characteristics (referenced to V<sub>CC</sub> = 5V, V<sub>SS</sub> = 0V at Ta = 25°C unless otherwise specified)**

Symbol	Parameter	Measuring condition	Standard			Unit
			Min	Typ.	Max.	
R <sub>IXIN</sub>	Feedback resistance X <sub>IN</sub>			1.0		MΩ
R <sub>IXCIN</sub>	Feedback resistance X <sub>CIN</sub>			6.0		MΩ
V <sub>RAM</sub>	RAM retention voltage	When clock is stopped	2.0			V
I <sub>CC</sub>	Power supply current	At the time of slicer operation f(X <sub>CIN</sub> )=10kHz		150	180	mA
		V <sub>CC</sub> =5.0V f(X <sub>CIN</sub> )=32kHz A rectangular wave (Notes 1, 2)		200		μA
		V <sub>CC</sub> =3.0V f(X <sub>CIN</sub> )=32kHz A rectangular wave (Notes 1, 2)		150		μA
		V <sub>CC</sub> =5.0V f(X <sub>CIN</sub> )=32kHz At the time of weight (Notes 1, 2)		5.0	10.0	μA
		V <sub>CC</sub> =3.0V f(X <sub>CIN</sub> )=32kHz At the time of weight (Notes 1, 2)		3.0	8.0	μA
		At the time of a clock stop (Notes 2)			5.0	μA

Note 1: This is a state where only one timer is operating with fc32 while the oscillation capability is set to LOW and slicer operation is turned OFF.

Note 2: • V<sub>DD1</sub>, V<sub>DD2</sub>, and V<sub>DD3</sub> all are at the same potential level as V<sub>CC</sub>.

- Extension register (address 00<sub>16</sub> DD8) STBY0 and (address 15<sub>16</sub> DD13) STBY1 are set to 1 while all other extension registers (addresses 00<sub>16</sub> through 22<sub>16</sub>) are set to 0.
- Clock input to the FSCIN pin is disabled.
- Inputs to the SYNCIN and CVIN1 pins are disabled.

**Tabl 4.6 Video signal input conditions (V<sub>CC</sub> = 5.0V, Ta = -20 to 70°C)**

Symbol	Parameter	Measuring condition	Standard			Unit
			Min	Typ.	Max.	
V <sub>IN-cu</sub>	Composite video signal input clamp voltage	Sync-chip voltage		1.0		V

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

**Table 4.7 A-D conversion characteristics (referenced to  $V_{CC} = AV_{CC} = V_{REF} = 5V$ ,  $V_{SS} = AV_{SS} = 0V$   
 at  $T_a = 25^{\circ}C$ ,  $f(X_{IN}) = 10MHz$  unless otherwise specified)**

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

**Table 4.8 D-A conversion characteristics (referenced to  $V_{CC} = 5V$ ,  $V_{SS} = AV_{SS} = 0V$ ,  $V_{REF} = 5V$  at  
 $T_a = 25^{\circ}C$ ,  $f(X_{IN}) = 10MHz$  unless otherwise specified)**

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

Note: This applies when using one D-A converter, with the D-A register for the unused D-A converter set to "0016".

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

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

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

Timing requirements (referenced to  $V_{CC} = 5V$ ,  $V_{SS} = 0V$  at  $T_a = 25^{\circ}C$  unless otherwise specified)

Table 4.9 External clock input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_c$	External clock input cycle time	100		ns
$t_{w(H)}$	External clock input HIGH pulse width	40		ns
$t_{w(L)}$	External clock input LOW pulse width	40		ns
$t_r$	External clock rise time		18	ns
$t_f$	External clock fall time		18	ns

Table 4.10 In memory expansion mode

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:

$$t_{ac1(RD-DB)} = \frac{10^9}{f(BCLK) \times 2} - 45 \quad [ns]$$

$$t_{ac2(RD-DB)} = \frac{3 \times 10^9}{f(BCLK) \times 2} - 45 \quad [ns]$$

$$t_{ac3(RD-DB)} = \frac{3 \times 10^9}{f(BCLK) \times 2} - 45 \quad [ns]$$



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

Timing requirements (referenced to  $V_{CC} = 5V$ ,  $V_{SS} = 0V$  at  $T_a = 25^{\circ}C$  unless otherwise specified)

**Table 4.11 Timer A input (counter input in event counter mode)**

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

**Table 4.12 Timer A input (gating input in timer mode)**

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

**Table 4.13 Timer A input (external trigger input in one-shot timer mode)**

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

**Table 4.14 Timer A input (external trigger input in pulse width modulation mode)**

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

**Table 4.15 Timer A input (up/down input in event counter mode)**

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

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

Timing requirements (referenced to  $V_{CC} = 5V$ ,  $V_{SS} = 0V$  at  $T_a = 25^\circ C$  unless otherwise specified)

**Table 4.16 Timer B input (counter input in event counter mode)**

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

**Table 4.17 Timer B input (pulse period measurement mode)**

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

**Table 4.18 Timer B input (pulse width measurement mode)**

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

**Table 4.19 A-D trigger input**

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	ADTRG input cycle time (trigger able minimum)	1000		ns
$t_{w(ADL)}$	ADTRG input LOW pulse width	125		ns

**Table 4.20 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 4.21 External interrupt  $\overline{INT_i}$  inputs**

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

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

**Switching characteristics (referenced to V<sub>CC</sub> = 5V, V<sub>SS</sub> = 0V at Ta = 25°C, CM15 = "1" unless otherwise specified)**

**Table 4.22 In memory expansion mode (No wait)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
t <sub>d</sub> (BCLK-AD)	Address output delay time	Figure 4.1		40	ns
t <sub>h</sub> (BCLK-AD)	Address output hold time (BCLK standard)		4		ns
t <sub>h</sub> (RD-AD)	Address output hold time (RD standard)		0		ns
t <sub>h</sub> (WR-AD)	Address output hold time (WR standard)		0		ns
t <sub>d</sub> (BCLK-CS)	Chip select output delay time			40	ns
t <sub>h</sub> (BCLK-CS)	Chip select output hold time (BCLK standard)		4		ns
t <sub>d</sub> (BCLK-ALE)	ALE signal output delay time			40	ns
t <sub>h</sub> (BCLK-ALE)	ALE signal output hold time		- 4		ns
t <sub>d</sub> (BCLK-RD)	RD signal output delay time			40	ns
t <sub>h</sub> (BCLK-RD)	RD signal output hold time		0		ns
t <sub>d</sub> (BCLK-WR)	WR signal output delay time			40	ns
t <sub>h</sub> (BCLK-WR)	WR signal output hold time		0		ns
t <sub>d</sub> (BCLK-DB)	Data output delay time (BCLK standard)			40	ns
t <sub>h</sub> (BCLK-DB)	Data output hold time (BCLK standard)		4		ns
t <sub>d</sub> (DB-WR)	Data output delay time (WR standard)		(Note1)		ns
t <sub>h</sub> (WR-DB)	Data output hold time (WR standard)(Note2)		0		ns

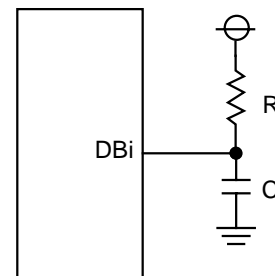
Note 1: Calculated according to the BCLK frequency as follows:

$$t_d(\text{DB} - \text{WR}) = \frac{10^9}{f(\text{BCLK}) \times 2} - 40 \quad [\text{ns}]$$

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<sub>OL</sub> = 0.2V<sub>CC</sub>, C = 30pF, R = 1kΩ, hold time of output "L" level is

$$t = -30\text{pF} \times 1\text{k}\Omega \times \ln(1 - 0.2V_{CC} / V_{CC}) \\ = 6.7\text{ns}.$$



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

Switching characteristics (refer to  $V_{cc} = 5V$ ,  $V_{ss} = 0V$  at  $T_a = 25^{\circ}C$ , CM15 = "1" unless otherwise specified)

Table 4.23 In memory expansion mode (With wait, accessing external memory)

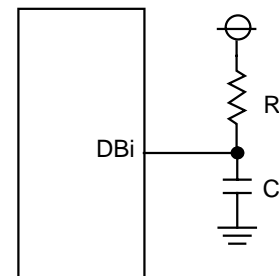
Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_d(BCLK-AD)$	Address output delay time	Figure 4.1		40	ns
$t_h(BCLK-AD)$	Address output hold time (BCLK standard)		4		ns
$t_h(RD-AD)$	Address output hold time (RD standard)		0		ns
$t_h(WR-AD)$	Address output hold time (WR standard)		0		ns
$t_d(BCLK-CS)$	Chip select output delay time			40	ns
$t_h(BCLK-CS)$	Chip select output hold time (BCLK standard)		4		ns
$t_d(BCLK-ALE)$	ALE signal output delay time			40	ns
$t_h(BCLK-ALE)$	ALE signal output hold time		-4		ns
$t_d(BCLK-RD)$	RD signal output delay time			40	ns
$t_h(BCLK-RD)$	RD signal output hold time		0		ns
$t_d(BCLK-WR)$	WR signal output delay time			40	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:

$$t_d(DB - WR) = \frac{10^9}{f(BCLK)} - 40 \quad [ns]$$

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  $V_{OL} = 0.2V_{CC}$ ,  $C = 30pF$ ,  $R = 1k\Omega$ , hold time of output "L" level is

$$t = -30pF \times 1k\Omega \times \ln(1 - 0.2V_{CC} / V_{CC}) = 6.7ns.$$



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

**Switching characteristics (referenced to V<sub>cc</sub> = 5V, V<sub>ss</sub> = 0V at Ta = 25°C, CM15 = "1" unless otherwise specified)**

**Table 4.24 In memory expansion mode (With wait, accessing external memory, multiplex bus area selected)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
t <sub>d</sub> (BCLK-AD)	Address output delay time	Figure 4.1		40	ns
t <sub>h</sub> (BCLK-AD)	Address output hold time (BCLK standard)		4		ns
t <sub>h</sub> (RD-AD)	Address output hold time (RD standard)		(Note)		ns
t <sub>h</sub> (WR-AD)	Address output hold time (WR standard)		(Note)		ns
t <sub>d</sub> (BCLK-CS)	Chip select output delay time			40	ns
t <sub>h</sub> (BCLK-CS)	Chip select output hold time (BCLK standard)		4		ns
t <sub>h</sub> (RD-CS)	Chip select output hold time (RD standard)		(Note)		ns
t <sub>h</sub> (WR-CS)	Chip select output hold time (WR standard)		(Note)		ns
t <sub>d</sub> (BCLK-RD)	RD signal output delay time			40	ns
t <sub>h</sub> (BCLK-RD)	RD signal output hold time		0		ns
t <sub>d</sub> (BCLK-WR)	WR signal output delay time			40	ns
t <sub>h</sub> (BCLK-WR)	WR signal output hold time		0		ns
t <sub>d</sub> (BCLK-DB)	Data output delay time (BCLK standard)			40	ns
t <sub>h</sub> (BCLK-DB)	Data output hold time (BCLK standard)		4		ns
t <sub>d</sub> (DB-WR)	Data output delay time (WR standard)		(Note)		ns
t <sub>h</sub> (WR-DB)	Data output hold time (WR standard)		(Note)		ns
t <sub>d</sub> (BCLK-ALE)	ALE signal output delay time (BCLK standard)			40	ns
t <sub>h</sub> (BCLK-ALE)	ALE signal output hold time (BCLK standard)		- 4		ns
t <sub>d</sub> (AD-ALE)	ALE signal output delay time (Address standard)		(Note)		ns
t <sub>h</sub> (ALE-AD)	ALE signal output hold time (Address standard)		50		ns
t <sub>d</sub> (AD-RD)	Post-address RD signal output delay time		0		ns
t <sub>d</sub> (AD-WR)	Post-address WR signal output delay time		0		ns
t <sub>d</sub> Z(RD-AD)	Address output floating start time			8	ns

Note: Calculated according to the BCLK frequency as follows:

$$t_h(RD - AD) = \frac{10^9}{f(BCLK) \times 2} \quad [ns]$$

$$t_h(WR - AD) = \frac{10^9}{f(BCLK) \times 2} \quad [ns]$$

$$t_h(RD - CS) = \frac{10^9}{f(BCLK) \times 2} \quad [ns]$$

$$t_h(WR - CS) = \frac{10^9}{f(BCLK) \times 2} \quad [ns]$$

$$t_d(DB - WR) = \frac{10^9 \times 3}{f(BCLK) \times 2} - 40 \quad [ns]$$

$$t_h(WR - DB) = \frac{10^9}{f(BCLK) \times 2} \quad [ns]$$

$$t_d(AD - ALE) = \frac{10^9}{f(BCLK) \times 2} - 40 \quad [ns]$$

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

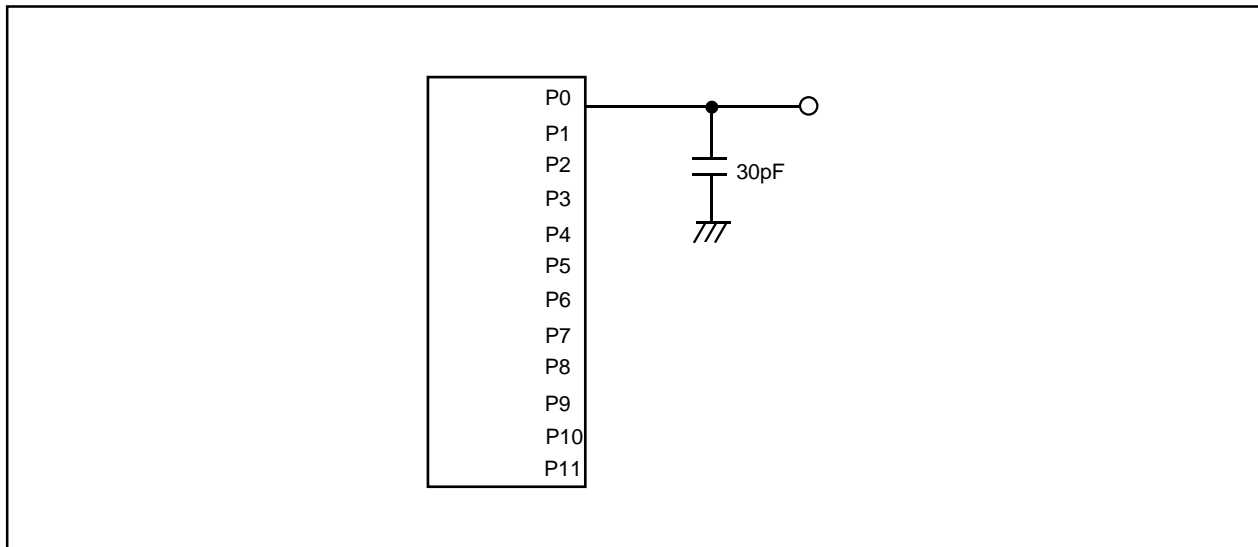


Figure 4.1 Port P0 to P11 measurement circuit

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

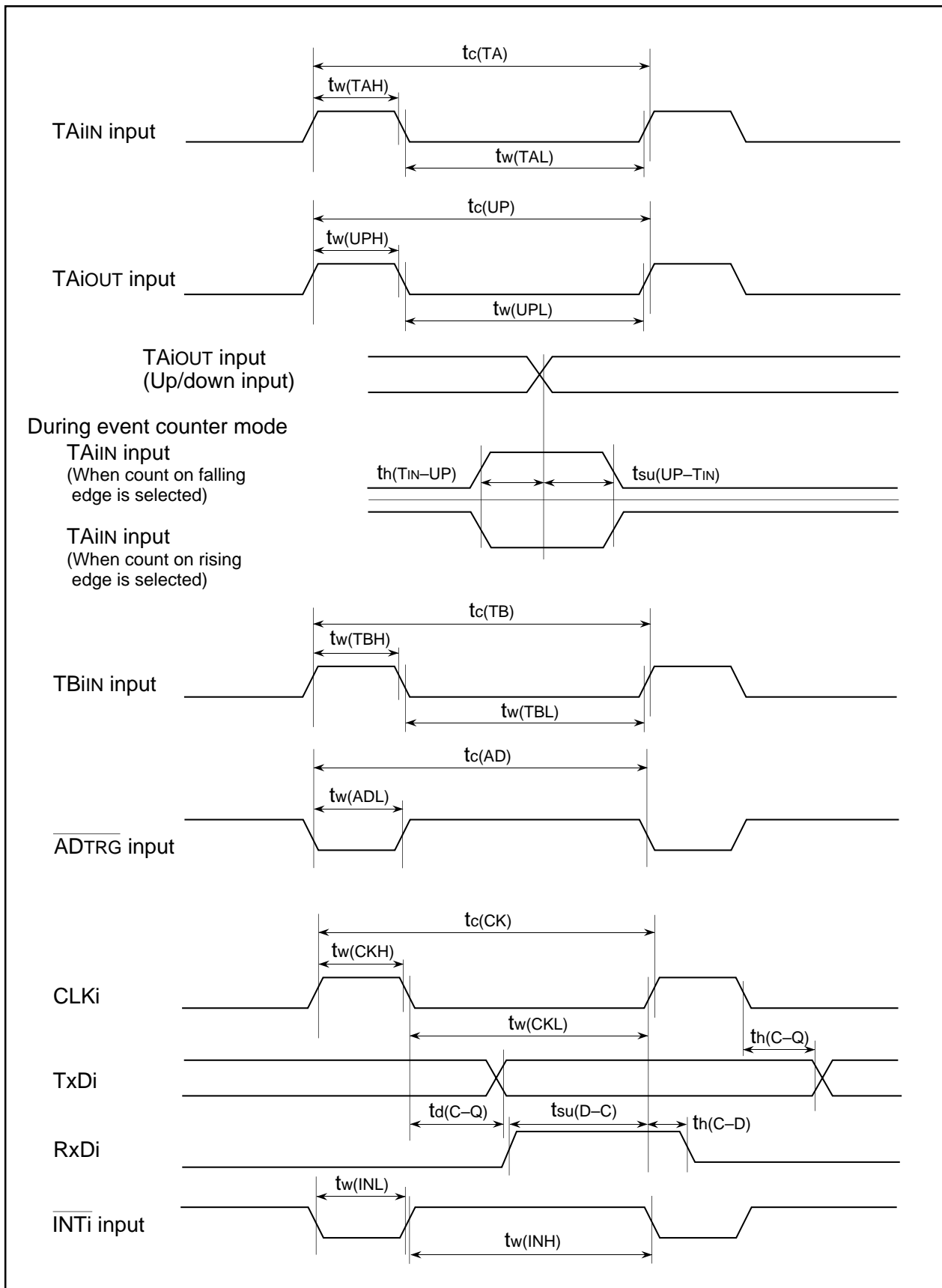


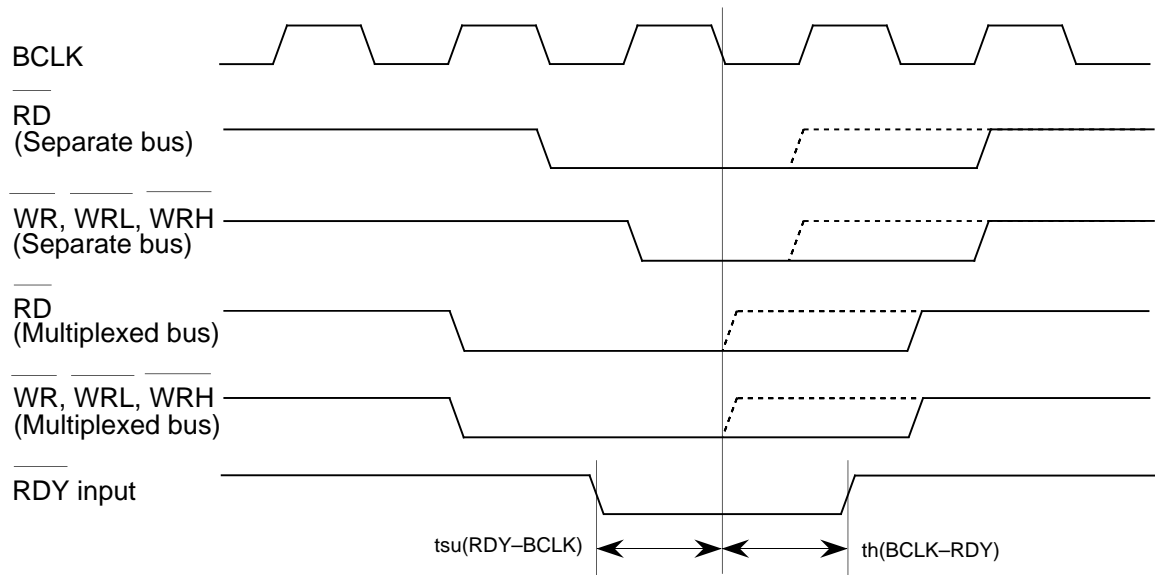
Figure 4.2 Timing diagram (1)

# M306H2FCFP

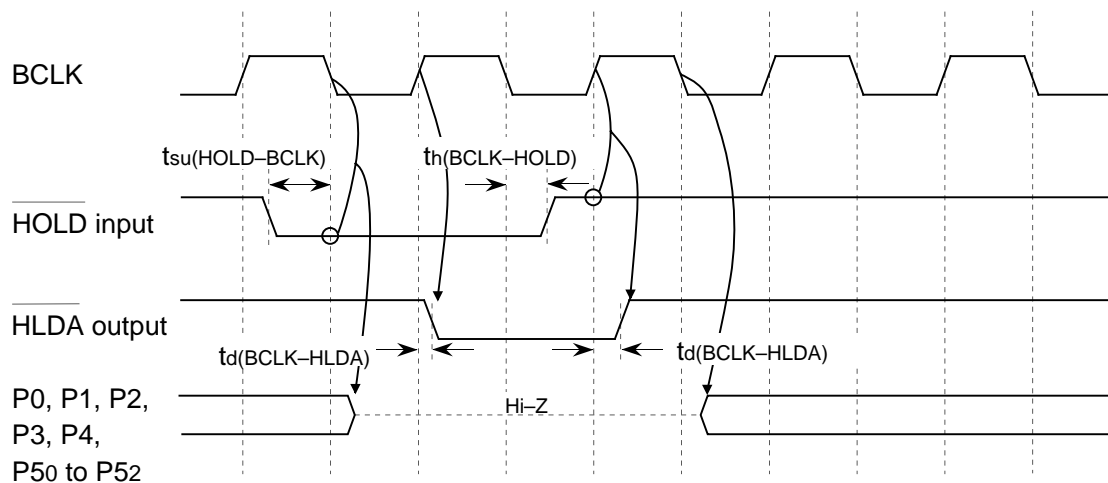
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## In memory expansion 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 :

- $V_{CC}=5V$
- Input timing voltage : Determined with  $V_{IL}=1.0V$ ,  $V_{IH}=4.0V$
- Output timing voltage : Determined with  $V_{OL}=2.5V$ ,  $V_{OH}=2.5V$

Figure 4.3 Timing diagram (2)

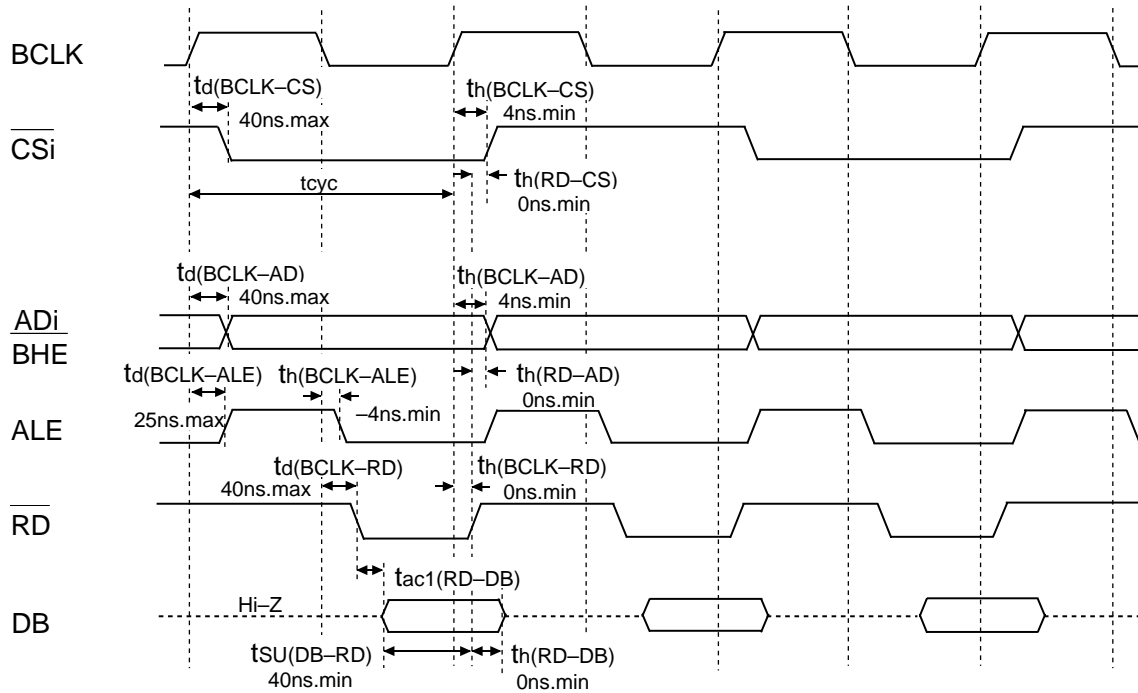


# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## In memory expansion mode (With No Wait)

### Read timing



### Write timing

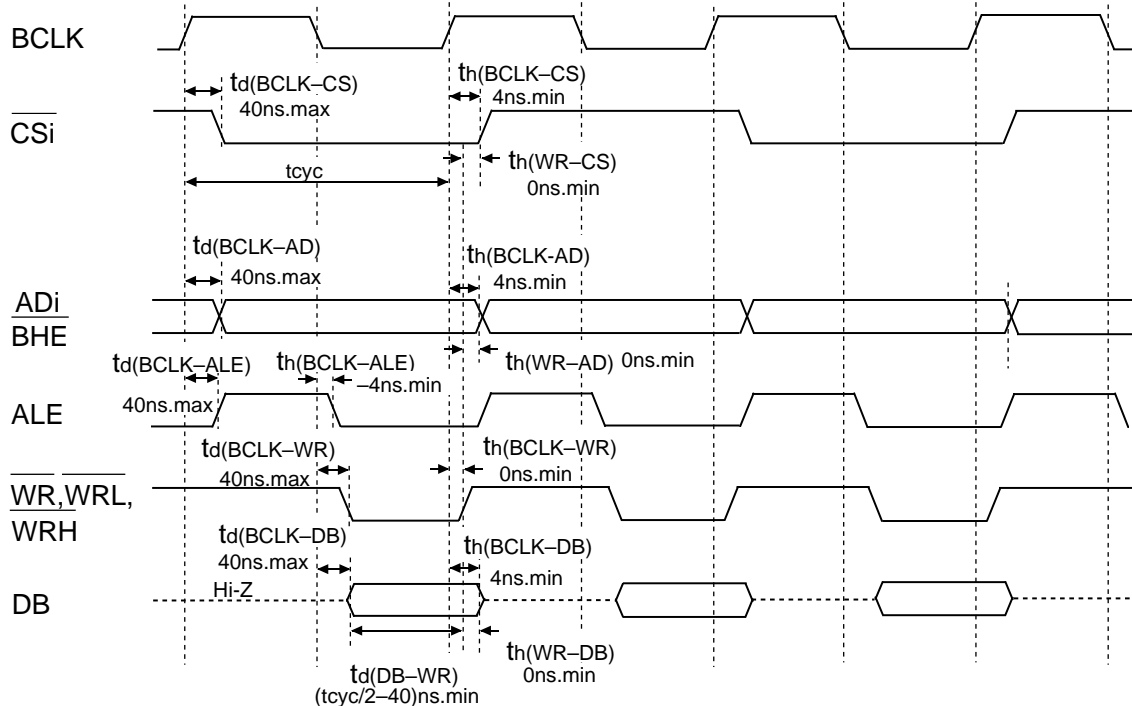


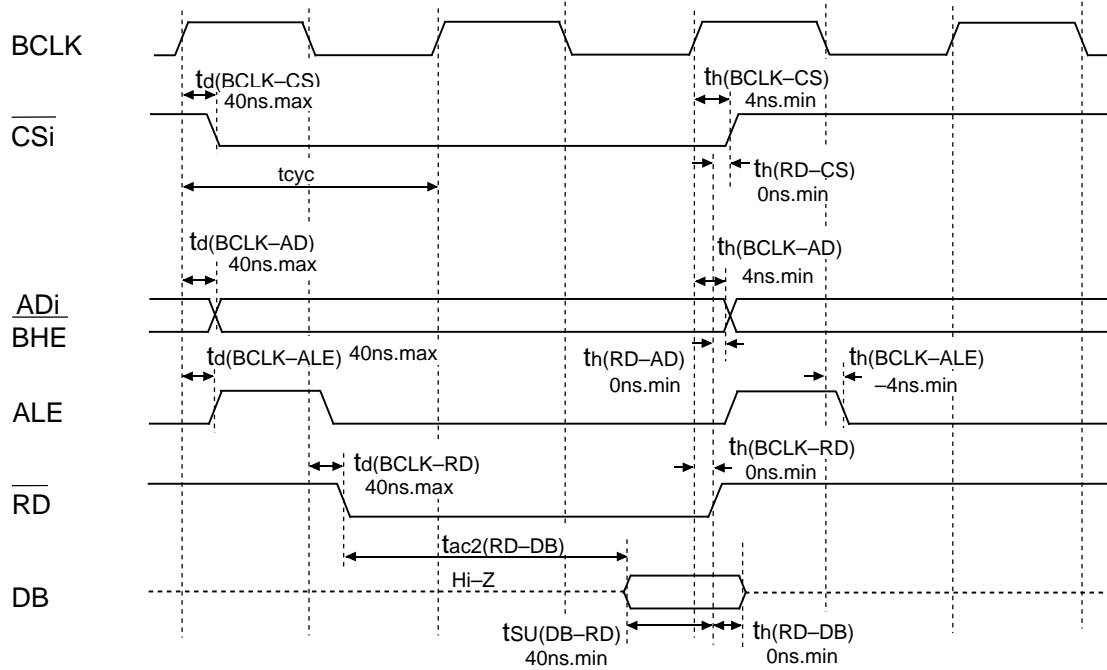
Figure 4.4 Timing diagram (3)

# M306H2FCFP

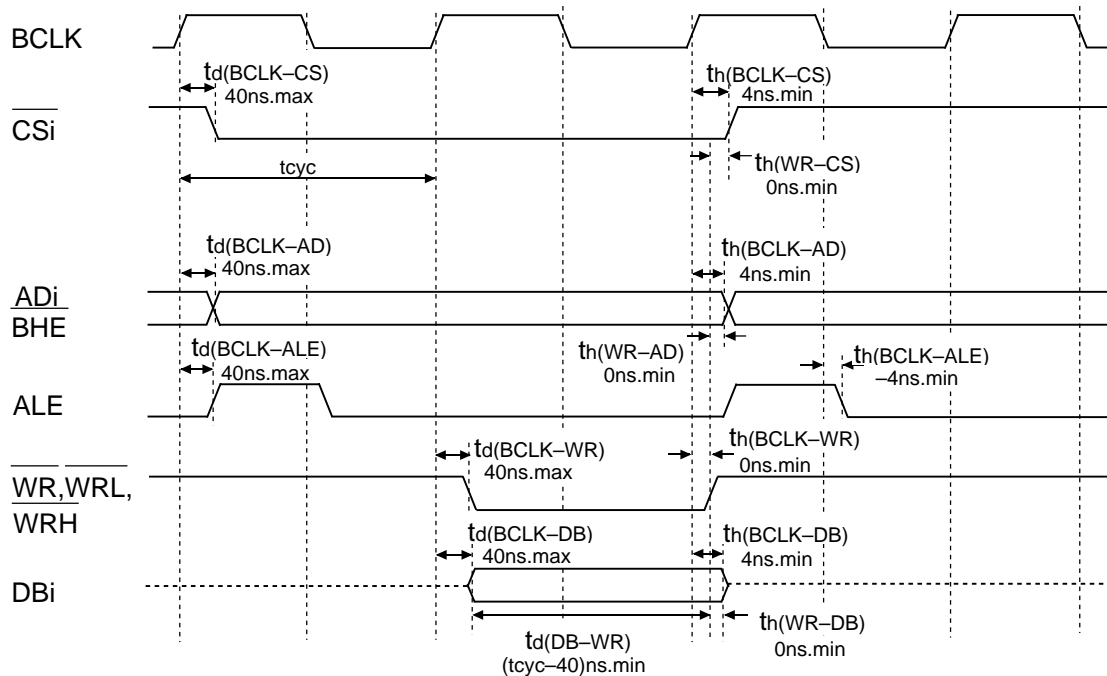
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## In memory expansion mode (When Accessing External Memory Area With Wait)

### Read timing



### Write timing



Measuring conditions :

- VCC=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 4.5 Timing diagram (4)

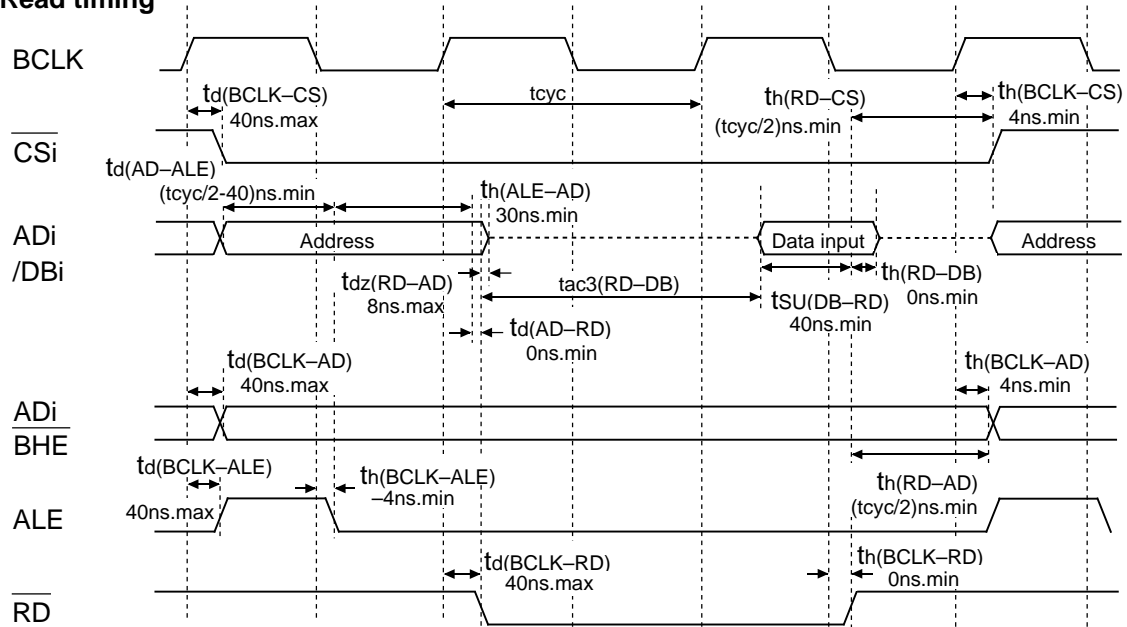
# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

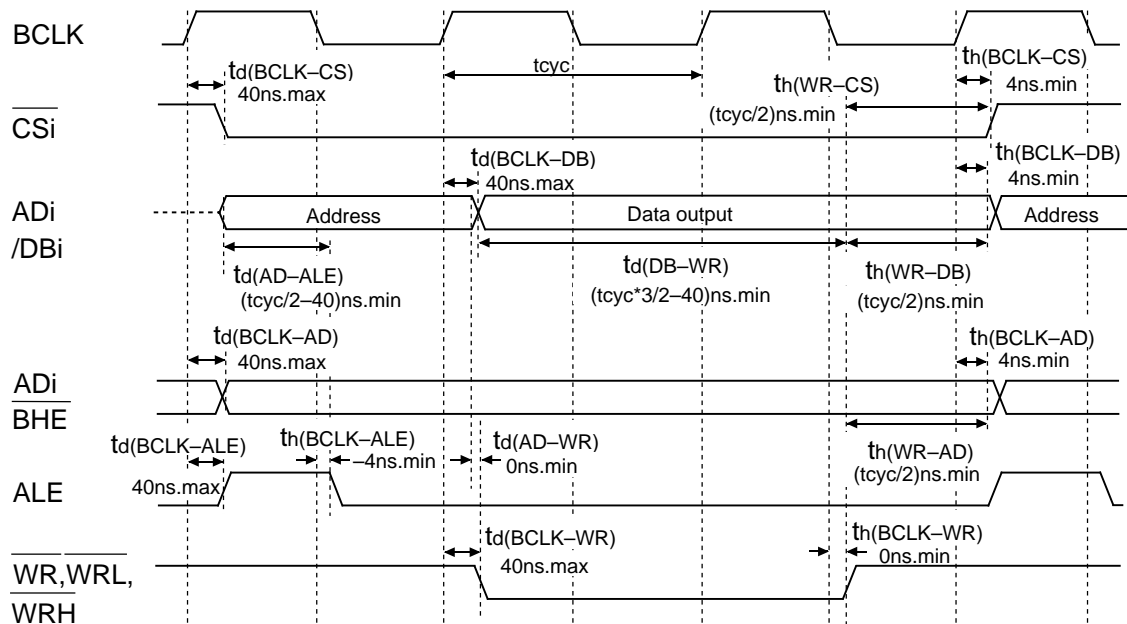
## In memory expansion mode

(When Accessing External Memory Area With Wait, And Select Multiplexed bus)

### Read timing



### Write timing



Measuring conditions :

- $V_{\text{CC}}=5\text{V}$
- Input timing voltage : Determined with  $V_{\text{IL}}=0.8\text{V}$ ,  $V_{\text{IH}}=2.5\text{V}$
- Output timing voltage : Determined with  $V_{\text{OL}}=0.8\text{V}$ ,  $V_{\text{OH}}=2.0\text{V}$

Figure 4.6 Timing diagram (5)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## 5. FLASH MEMORY

### 5.1 Outline Performance

Table 5.1.1 shows the outline performance of the M306H2FCFP (flash memory version).

**Table 5.1.1. Outline performance of the M306H2FCFP (flash memory version)**

Item		Performance
Power supply voltage		4.75V to 5.25 V (at $f(X_{IN}) = 10$ MHz)
Program/erase voltage		4.75V to 5.25 V ( $f(X_{IN}) = 10$ MHz, No wait, $f(X_{IN}) = 5$ MHz, One wait)
Flash memory operation mode		Three modes (parallel I/O, standard serial I/O, CPU rewrite)
Erase block division	User ROM area	Refer to Figure 5.2.1
	Boot ROM area	No division (8 K bytes) (Note 1)
Program method		In units of words
Erase method		Collective erase/block erase
Program/erase control method		Program/erase control by software command
Number of commands		6 commands
Program/erase count		100 times
ROM code protect		Parallel I/O and standard serial I/O modes are supported.

Note : The boot ROM area contains a standard serial I/O mode control program which is stored in it when shipped from the factory. This area can be erased and programmed in only parallel I/O mode.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 5.2 Flash Memory mode

The M306H2FCFP (flash memory version) has an internal new DINOR (Divided bit line NOR) flash memory that can be rewritten with a single power source.

For this flash memory, three flash memory modes are available in which to read, program, and erase: parallel I/O and standard serial I/O modes in which the flash memory can be manipulated using a programmer and a CPU rewrite mode in which the flash memory can be manipulated by the Central Processing Unit (CPU). Each mode is detailed in the pages to follow.

The flash memory is divided into several blocks as shown in Figure 5.2.1, so that memory can be erased one block at a time.

In addition to the ordinary user ROM area to store a microcomputer operation control program, the flash memory has a boot ROM area that is used to store a program to control rewriting in CPU rewrite and standard serial I/O modes. This boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the factory. However, the user can write a rewrite control program in this area that suits the user's application system. This boot ROM area can be rewritten in only parallel I/O mode.

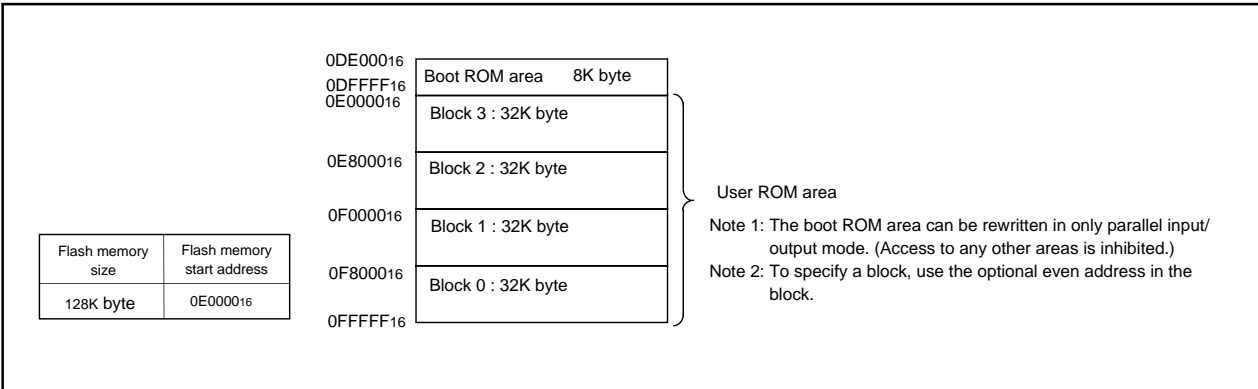


Figure 5.2.1. Block diagram of flash memory version

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 5.3 CPU Rewrite Mode

In CPU rewrite mode, the on-chip flash memory can be operated on (read, program, or erase) under control of the Central Processing Unit (CPU).

In CPU rewrite mode, only the user ROM area shown in Figure 5.2.1 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.

### (1) Microcomputer Mode and Boot Mode

The control program for CPU rewrite mode must be written into the user ROM or boot ROM area in parallel I/O mode beforehand. (If the control program is written into the boot ROM area, the standard serial I/O mode becomes unusable.)

See Figure 5.2.1 for details about the boot ROM area.

Normal microcomputer mode is entered when the microcomputer is reset with pulling CNVSS 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 M1 pin low, the CNVSS pin high, the P50 pin high, the CPU starts operating using the control program in the boot ROM area (program start address is DE000<sub>16</sub> fixation). This mode is called the "boot" mode.

### (2) Block Address

Block addresses refer to the optional even address of each block. These addresses are used in the block erase command.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## 5.3.1 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. This rewrite control program must be transferred to internal RAM before it can be executed.

The CPU rewrite mode is accessed by applying  $5V \pm 5\%$  to the M2 pin and writing "1" for the CPU rewrite mode select bit (bit 1 in address 03B416). Software commands are accepted once the mode is accessed.

In the CPU rewrite mode, write to and read from software commands and data into even-numbered address ("0" for byte address A0) 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 5.3.1 shows the flash memory control register.

Bit 0 is the RY/ $\overline{\text{BY}}$  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 is the CPU rewrite mode select bit. When this bit is set to "1" and  $5V \pm 5\%$  are applied to the M2 pin, the M306H2 accesses the CPU rewrite mode. Software commands are accepted once the mode is accessed. In CPU rewrite mode, the CPU becomes unable to access the internal flash memory directly. Therefore, use the control program in RAM for write to bit 1. 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 is the CPU rewrite mode entry flag. This bit can be read to check whether the CPU rewrite mode has been entered or not.

Bit 3 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". If the control circuit is reset while erasing is in progress, a 5 ms wait is needed so that the flash memory can restore normal operation. Figure 5.3.2 shows a flowchart for setting/releasing the CPU rewrite mode.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

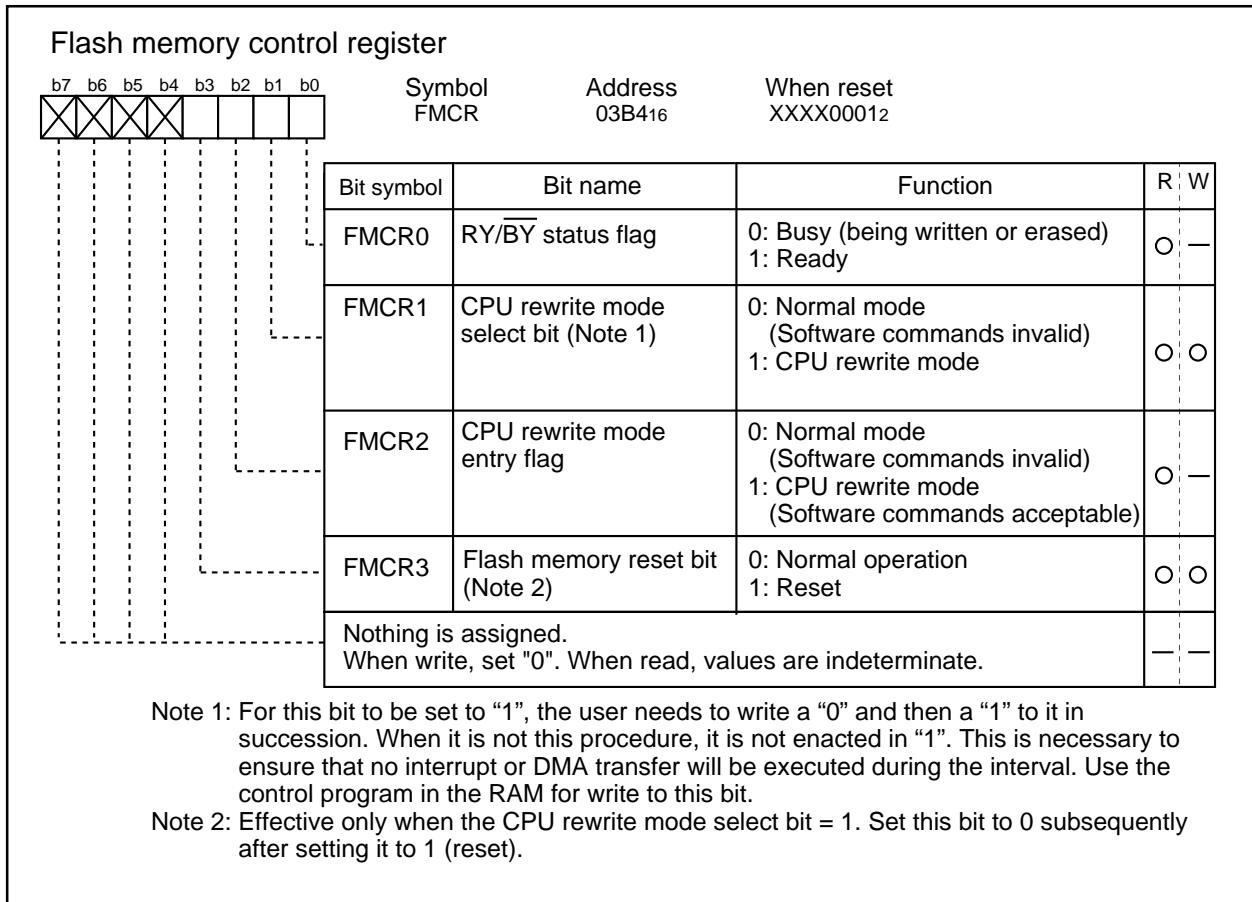


Figure 5.3.1. Flash memory control registers



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

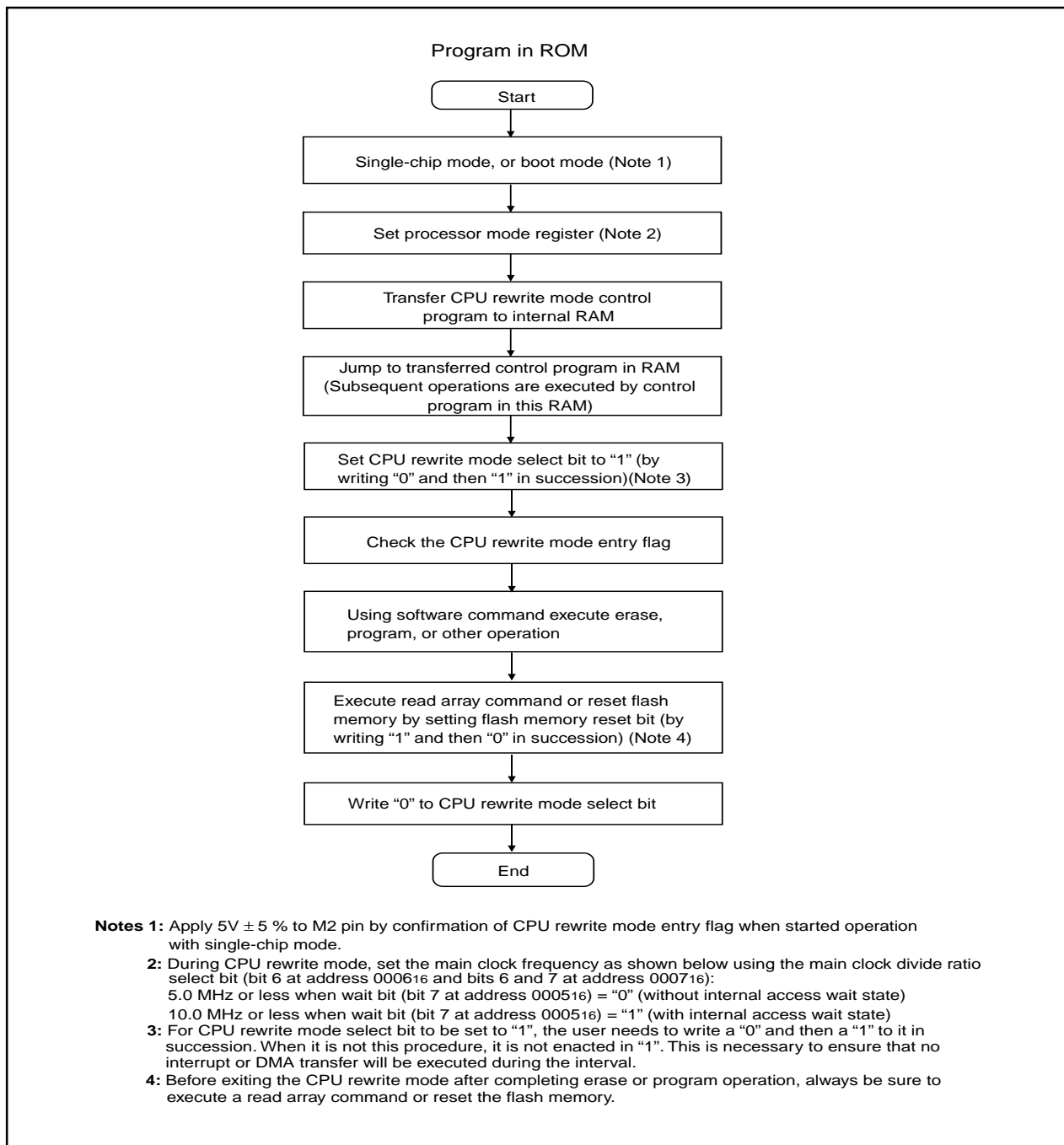


Figure 5.3.2. CPU rewrite mode set/reset flowchart

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

---

## 5.3.2 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 main clock frequency as shown below using the main clock divide ratio select bit (bit 6 at address 0006<sub>16</sub> and bits 6 and 7 at address 0007<sub>16</sub>):

5.0 MHz or less when wait bit (bit 7 at address 0005<sub>16</sub>) = 0 (without internal access wait state)

10.0 MHz or less when wait bit (bit 7 at address 0005<sub>16</sub>) = 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  $\overline{\text{NMI}}$ , address match, and watchdog timer interrupts 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.

### (4) Reset

If the control circuit is reset while erasing is in progress, a 5 ms wait is needed so that the flash memory can restore normal operation. Set a 5 ms wait to release the reset operation.

Also, when the reset has been released, the program execute start address is automatically set to 0DE000<sub>16</sub> at boot mode, therefore program so that the execute start address of the boot ROM is 0DE000<sub>16</sub>.

### (5) 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 or parallel I/O mode to rewrite these blocks.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 5.3.3 Software Commands

Table 5.3.1 lists the software commands available with the M306H2 (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 (D8 to D15) is ignored.

The content of each software command is explained below.

**Table 5.3.1. List of software commands (CPU rewrite mode)**

Command	Cycle number	First bus cycle			Second bus cycle		
		Mode	Address	Data (D0 to D7)	Mode	Address	Data (D0 to D7)
Read array	1	Write	X (Note 5)	FF <sub>16</sub>			
Read status register	2	Write	X	70 <sub>16</sub>	Read	X	SRD (Note 2)
Clear status register	1	Write	X	50 <sub>16</sub>			
Program (Note 3)	2	Write	X	40 <sub>16</sub>	Write	WA (Note 3)	WD (Note 3)
Erase all block	2	Write	X	20 <sub>16</sub>	Write	X	20 <sub>16</sub>
Block erase	2	Write	X	20 <sub>16</sub>	Write	BA (Note 4)	D0 <sub>16</sub>

Note 1: When a software command is input, the high-order byte of data (D8 to D15) is ignored.

Note 2: SRD = Status Register Data

Note 3: WA = Write Address, WD = Write Data

Note 4: BA = Block Address (Enter the optional address of each block that is an even address.)

Note 5: X denotes a given address in the user ROM area (that is an even address).

### Read Array Command (FF<sub>16</sub>)

The read array mode is entered by writing the command code “FF<sub>16</sub>” in the first 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 (D0–D15), 16 bits at a time.

The read array mode is retained intact until another command is written.

### Read Status Register Command (70<sub>16</sub>)

When the command code “70<sub>16</sub>” is written in the first bus cycle, the content of the status register is read out at the data bus (D0–D7) by a read in the second bus cycle.

The status register is explained in the next section.

### Clear Status Register Command (50<sub>16</sub>)

This command is used to clear the bits SR4 to SR5 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<sub>16</sub>” in the first bus cycle.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## Program Command (40<sub>16</sub>)

Program operation starts when the command code “40<sub>16</sub>” is written in the first bus cycle. Then, if the address and data to program are written in the 2nd bus cycle, program operation (data programming and verification) will start.

Whether the write operation is completed can be confirmed by reading the status register or the RY/ $\overline{\text{BY}}$  status flag. When the program starts, the read status register mode is accessed automatically and the content of the status register is read into the data bus (D0 - D7). The status register bit 7 (SR7) is set to 0 at the same time the write operation starts and is returned to 1 upon completion of the write operation. In this case, the read status register mode remains active until the Read Array command (FF<sub>16</sub>) is written.

The RY/ $\overline{\text{BY}}$  status flag is 0 during write operation and 1 when the write operation is completed as is the status register bit 7.

At program end, program results can be checked by reading the status register.

## Erase All Blocks Command (20<sub>16</sub>/20<sub>16</sub>)

By writing the command code “20<sub>16</sub>” in the first bus cycle and the confirmation command code “20<sub>16</sub>” in the second bus cycle that follows, the system starts erase all blocks( erase and erase verify).

Whether the erase all blocks command is terminated can be confirmed by reading the status register or the RY/ $\overline{\text{BY}}$  status flag. When the erase all blocks operation starts, the read status register mode is accessed automatically and 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 erase operation starts and is returned to 1 upon completion of the erase operation. In this case, the read status register mode remains active until the Read Array command (FF<sub>16</sub>) is written.

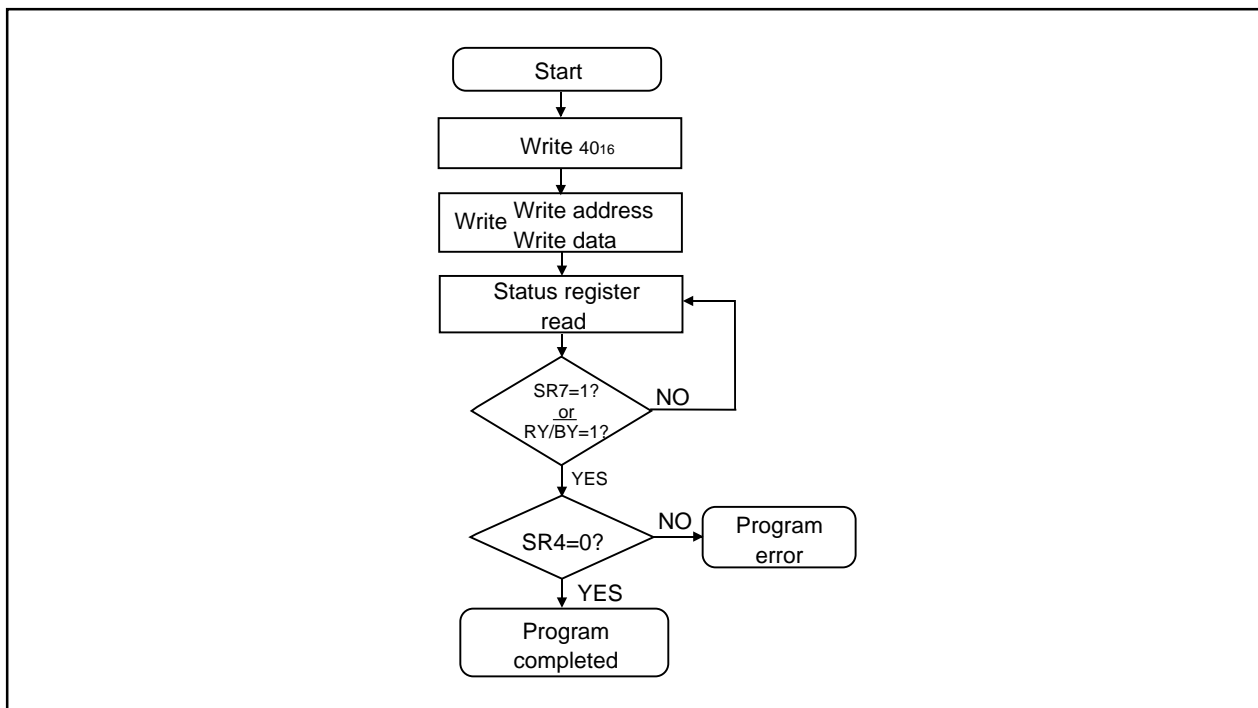


Figure 5.3.3. Program flowchart

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

The RY/ $\overline{\text{BY}}$  status flag is 0 during erase operation and 1 when the erase operation is completed as is the status register bit 7.

At erase all blocks end, erase results can be checked by reading the status register. For details, refer to the section where the status register is detailed.

## Block Erase Command (20<sub>16</sub>/D0<sub>16</sub>)

By writing the command code "20<sub>16</sub>" in the first bus cycle and the confirmation command code "D0<sub>16</sub>" in the second bus cycle that follows to the block address of a flash memory block, the system initiates a block erase (erase and erase verify) operation.

Whether the block erase operation is completed can be confirmed by reading the status register or the RY/ $\overline{\text{BY}}$  status flag. At the same time the block 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 block erase operation starts and is returned to 1 upon completion of the block erase operation. In this case, the read status register mode remains active until the Read Array command (FF<sub>16</sub>).

The RY/ $\overline{\text{BY}}$  status flag is 0 during block erase operation and 1 when the block erase operation is completed as is the status register bit 7.

After the block erase operation is completed, the status register can be read out to know the result of the block erase operation. For details, refer to the section where the status register is detailed.

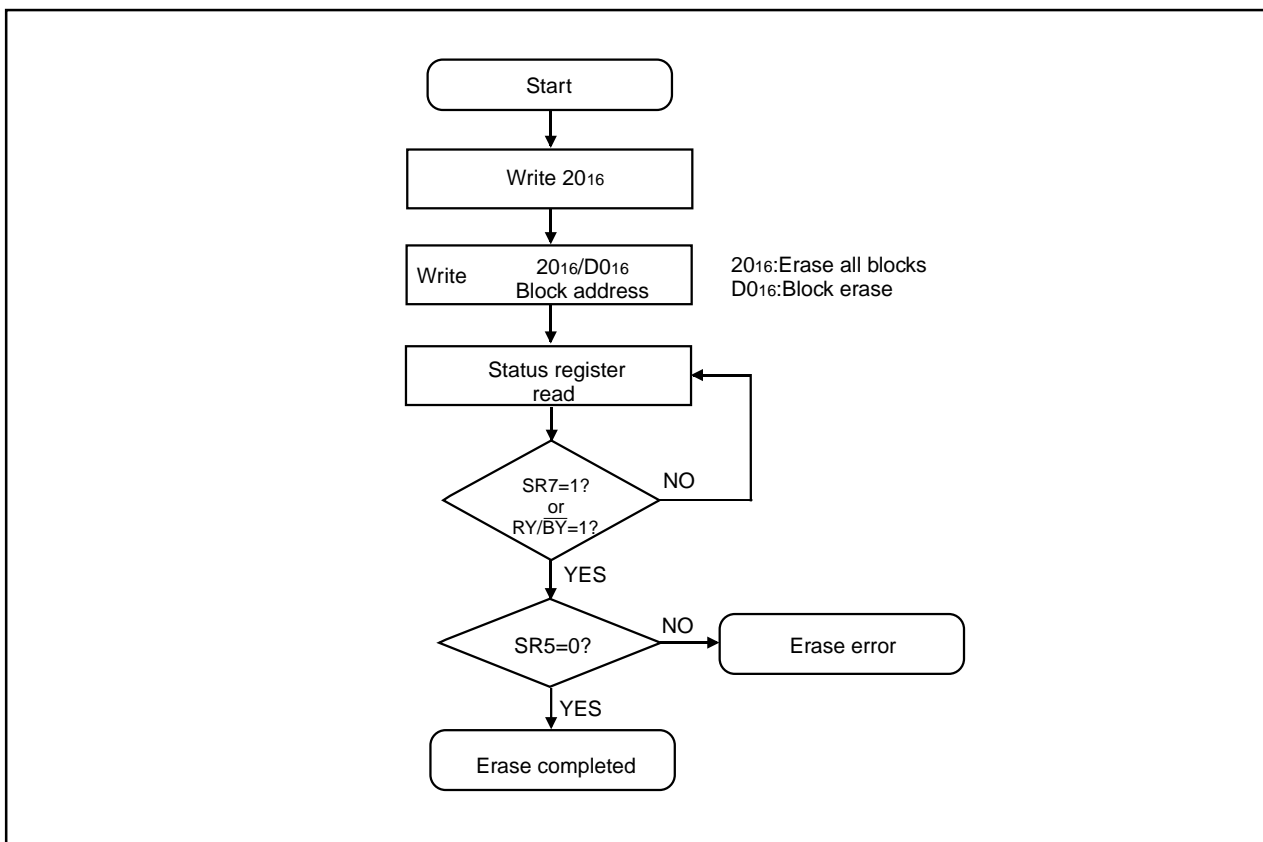


Figure 5.3.4. Erase flowchart

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## 5.3.4 Status Register

The status register shows the operating state of the flash memory and whether erase operations and programs ended successfully or in error. It can be read in the following ways.

- (1) By reading an arbitrary address from the user ROM area after writing the read status register command (70<sub>16</sub>)
- (2) By reading an arbitrary address from the user ROM area in the period from when the program starts or erase operation starts to when the read array command (FF<sub>16</sub>) is input

Table 5.3.2 shows the status register.

Also, the status register can be cleared in the following way.

- (1) By writing the clear status register command (50<sub>16</sub>)  
 After a reset, the status register is set to "80<sub>16</sub>".

Each bit in this register is explained below.

### Sequencer status (SR7)

After power-on, the sequencer status is set to 1 (ready).

The sequencer status indicates the operating status of the device. This status bit is set to 0 (busy) during write or erase operation and is set to 1 upon completion of these operations.

### Erase status (SR5)

The erase status informs the operating status of 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 write operation to the CPU. When a write error occurs, it is set to 1.

The program status is reset to 0 when cleared.

If "1" is written for any of the SR5 or SR4 bits, the program, erase all blocks, and block erase commands are not accepted. Before executing these commands, execute the clear status register command (50<sub>16</sub>) and clear the status register.

Also, any commands are not correct, both SR5 and SR4 are set to 1.

# M306H2FCFP

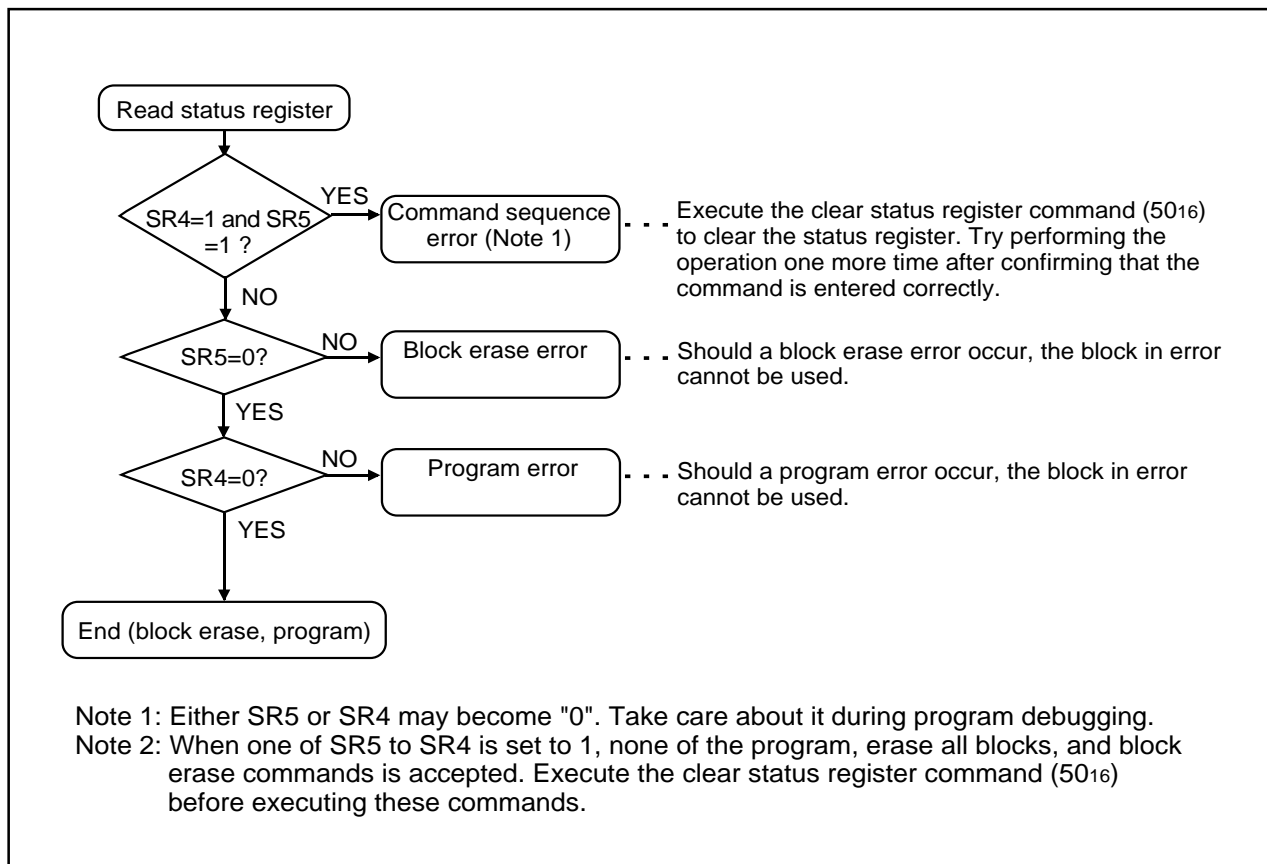
SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

**Table 5.3.2. Definition of each bit in status register**

Each bit of SRD	Status name	Definition	
		"1"	"0"
SR7 (bit7)	Sequencer 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)	Reserved	-	-
SR2 (bit2)	Reserved	-	-
SR1 (bit1)	Reserved	-	-
SR0 (bit0)	Reserved	-	-

## 5.3.5 Full Status Check

By performing full status check, it is possible to know the execution results of erase and program operations. Figure 5.3.5 shows a full status check flowchart and the action to be taken when each error occurs.



**Figure 5.3.5. Full status check flowchart and remedial procedure for errors**

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## 5.4 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.

### 5.4.1 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<sub>16</sub>). Figure 5.4.1 shows the ROM code protect control address (0FFFFF<sub>16</sub>). (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.

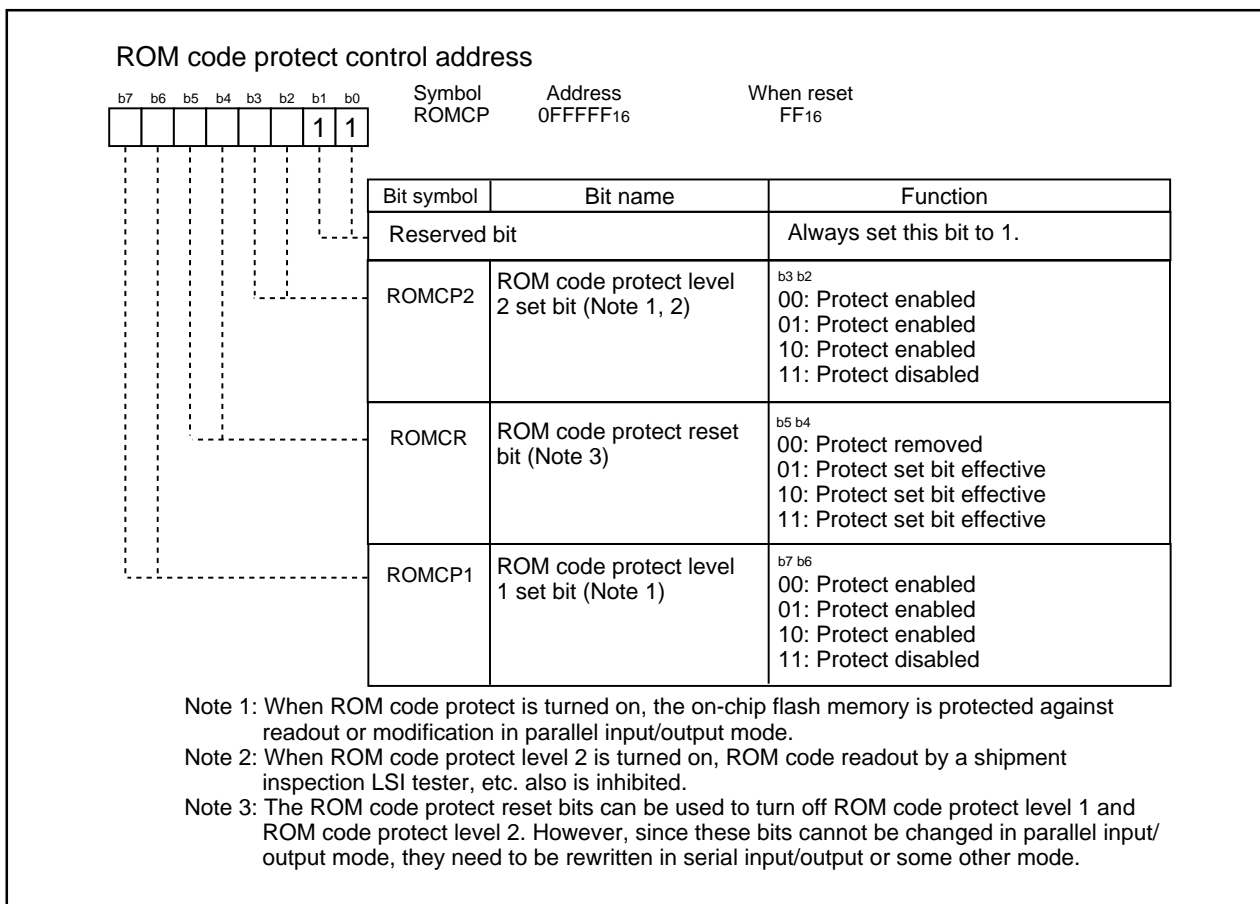


Figure 5.4.1. ROM code protect control address



M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

5.4.2 ID Code Check Function

Use this function in standard serial I/O mode. 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 first byte, are 0FFFD<sub>16</sub>, 0FFFE<sub>316</sub>, 0FFFE<sub>B16</sub>, 0FFFE<sub>F16</sub>, 0FFFF<sub>316</sub>, 0FFFF<sub>716</sub>, and 0FFFF<sub>B16</sub>. Write a program which has had the ID code preset at these addresses to the flash memory.

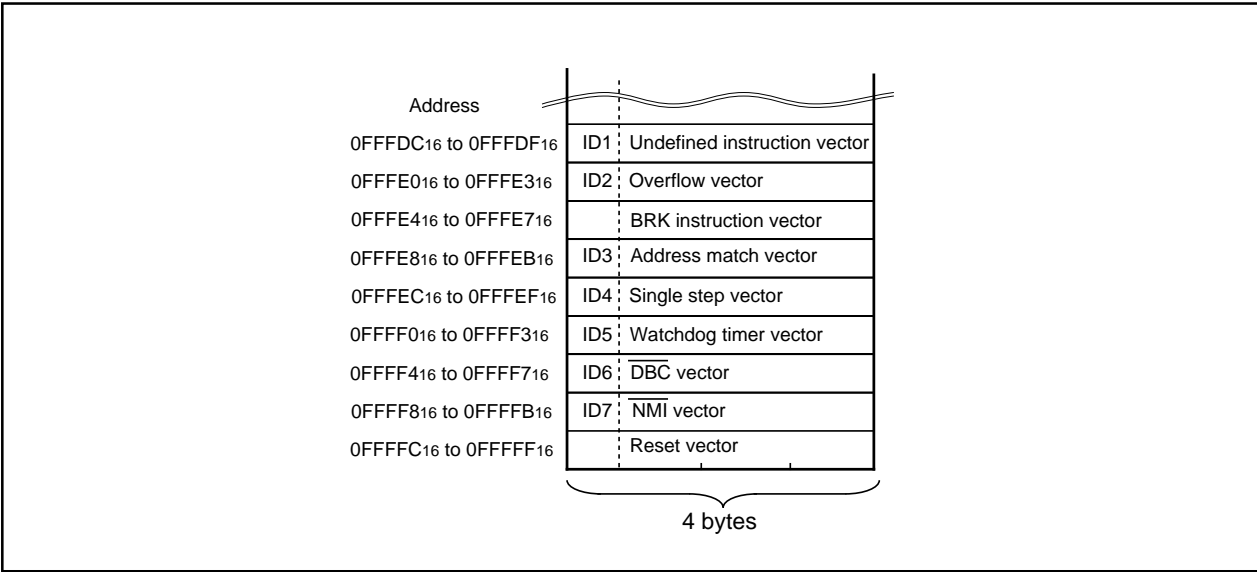


Figure 5.4.2. ID code store addresses

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## 5.5 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 M306H2 (flash memory version).

Refer to the instruction manual of each programmer maker for the details of use.

## User ROM and Boot ROM Areas

In parallel I/O mode, the user ROM and boot ROM areas shown in Figure 5.2.1 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 5.2.1.

The boot ROM area is 8 Kbytes in size. In parallel I/O mode, it is located at addresses 0DE000<sub>16</sub> through 0DFFFF<sub>16</sub>. 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 mode control program stored in it when shipped from the Mitsubishi factory. Therefore, using the device in standard serial input/output mode, you do not need to write to the boot ROM area.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 5.6 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 asynchronous. 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. The standard serial I/O mode is started by connecting "H" to the CNVss pin and P50 pin, connecting "L" to the M1 pin, and releasing the reset operation. (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 5.6.1 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 asynchronous) according to the level of CLK1 pin when the reset is released.

To use standard serial I/O mode 1 (clock synchronized), set the CLK1 pin to "H" level and release the reset. The operation uses the four UART1 pins CLK1, RxD1, TxD1 and RTS1 (BUSY). The CLK1 pin is the transfer clock input pin through which an external transfer clock is input. The TxD1 pin is for CMOS output. The RTS1 (BUSY) pin outputs an "L" level when ready for reception and an "L" level when reception starts.

To use standard serial I/O mode 2 (clock asynchronous), set the CLK1 pin to "L" level and release the reset. The operation uses the two UART1 pins RxD1 and TxD1.

In the standard serial I/O mode, only the user ROM area indicated in Figure 5.2.1 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 (programmer) are not accepted unless the ID code matches.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## Pin functions (Flash memory standard serial I/O mode) (Note 1)

Pin	Name	I/O	Description
Vcc,Vss	Power input		Apply a 4.75 - 5.25 V voltage on the Vcc pin, and 0 V voltage on the Vss pin.
CNVss	CNVss	I	Connect to Vcc.
RESET	Reset input	I	Reset input pin. While reset is "L" level, a 20 cycle or longer clock must be input to X <sub>IN</sub> pin.
M1	M1 input	I	Connect to Vss.
M2	M2 input	I	Apply a 4.75 - 5.25 V voltage on the Vcc pin when built-in flash memory rewriting.
BYTE	BYTE input	I	Connect to Vss.
X <sub>IN</sub>	Clock input	I	Connect a ceramic resonator or crystal oscillator between X <sub>IN</sub> and X <sub>OUT</sub> pins. To input an externally generated clock, input it to X <sub>IN</sub> pin and open X <sub>OUT</sub> pin.
X <sub>OUT</sub>	Clock output	O	
AVcc, AVss	Analog power supply input		Connect AVss to Vss and AVcc to Vcc, respectively.
VREF	Reference voltage input		Enter the reference voltage for AD from this pin.
P0 <sub>0</sub> to P0 <sub>7</sub>	Input port P0	I	Input "H" or "L" level signal or open.
P1 <sub>0</sub> to P1 <sub>7</sub>	Input port P1	I	Input "H" or "L" level signal or open.
P2 <sub>0</sub> to P2 <sub>7</sub>	Input port P2	I	Input "H" or "L" level signal or open.
P3 <sub>0</sub> to P3 <sub>7</sub>	Input port P3	I	Input "H" or "L" level signal or open.
P4 <sub>0</sub> to P4 <sub>7</sub>	Input port P4	I	Input "H" or "L" level signal or open.
P5 <sub>0</sub>	CE input	I	Input "H" level signal or open.
P5 <sub>1</sub> to P5 <sub>7</sub>	Input port P5	I	Input "H" or "L" level signal or open.
P6 <sub>0</sub> to P6 <sub>3</sub>	Input port P6	I	Input "H" or "L" level signal or open.
P6 <sub>4</sub>	BUSY output	O	BUSY signal output pin
P6 <sub>5</sub>	SCLK input	I	Serial clock input pin
P6 <sub>6</sub>	RXD input	I	Serial data input pin
P6 <sub>7</sub>	TXD output	O	Serial data output pin
P7 <sub>0</sub> to P7 <sub>7</sub>	Input port P7	I	Input "H" or "L" level signal or open.
P8 <sub>0</sub> to P8 <sub>4</sub> , P8 <sub>6</sub> , P8 <sub>7</sub>	Input port P8	I	Input "H" or "L" level signal or open.
P8 <sub>5</sub>	NMI input	I	Connect this pin to Vcc.
P9 <sub>0</sub> to P9 <sub>7</sub>	Input port P9	I	Input "H" or "L" level signal or open.
P10 <sub>0</sub> to P10 <sub>7</sub>	Input port P10	I	Input "H" or "L" level signal or open.
P11	Input port P11	O	Open
VDD1, Vss1	Power input		Connect VDD1 pin to Vcc and connect Vss1 pin to Vss.
VDD2, Vss2	Power input		Connect VDD2 pin to Vcc and connect Vss2 pin to Vss.
VDD3, Vss3	Power input		Connect VDD3 pin to Vcc and connect Vss3 pin to Vss.
LP2 to LP4	Filter output	O	Open
FSCIN	fsc input pin for synchronized signal generating	I	Subcarrier (fsc) input pin for synchronized signal generating. Input "L" level signal or open.
CVIN1, SYNCIN	Composite video signal input	I	Input pin of external compound video signal. Input "H" or "L" level signal or open.
SVREF	Synchronous slice level input	I	Slice voltage input pin at the time of slicing synchronized signal.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

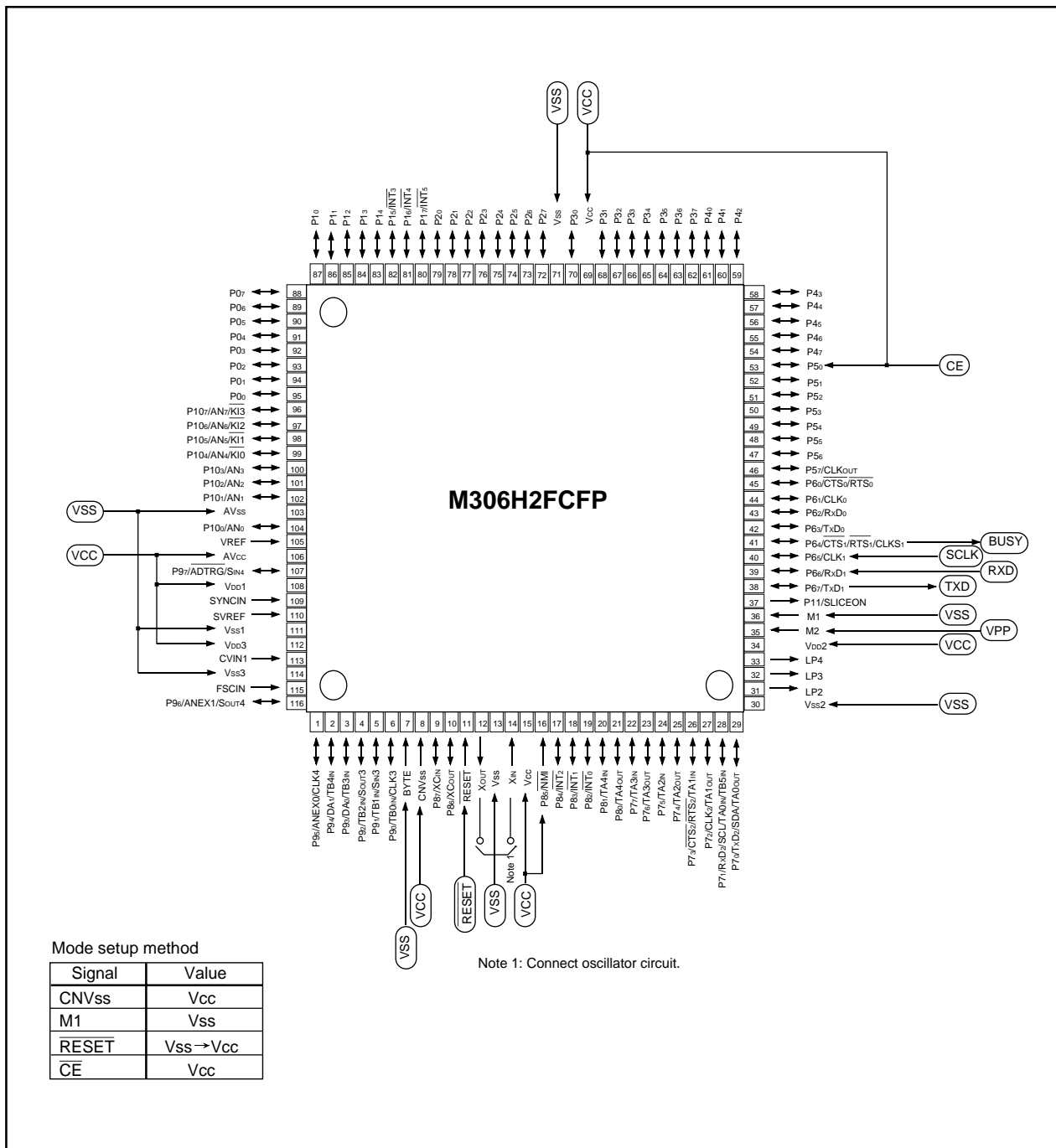


Figure 5.6.1. Pin connections for serial I/O mode (1)

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 5.6.1 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 programmer, etc.) using 4-wire clock-synchronized serial I/O (UART1). Standard serial I/O mode 1 is engaged by releasing the reset with the P65 (CLK1) 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 CLK1 pin, and are then input to the MCU via the RxD1 pin. In transmission, the read data and status are synchronized with the fall of the transfer clock, and output from the TxD1 pin.

The TxD1 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 RTS1 (BUSY) pin is "H" level. Accordingly, always start the next transfer after the RTS1 (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.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 5.6.2 Software Commands

Table 5.6.1 lists software commands. In the standard serial I/O mode 1, erase operations, programs and reading are controlled by transferring software commands via the RxD1 pin. Software commands are explained here below.

**Table 5.6.1. Software commands (Standard serial I/O mode 1)**

	Control command	1st byte transfer	2nd byte	3rd byte	4th byte	5th byte	6th byte		When ID is not verified
1	Page read	FF <sub>16</sub>	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
2	Page program	41 <sub>16</sub>	Address (middle)	Address (high)	Data input	Data input	Data input	Data input to 259th byte	Not acceptable
3	Block erase	20 <sub>16</sub>	Address (middle)	Address (high)	D0 <sub>16</sub>				Not acceptable
4	Erase all blocks	A7 <sub>16</sub>	D0 <sub>16</sub>						Not acceptable
5	Read status register	70 <sub>16</sub>	SRD output	SRD1 output					Acceptable
6	Clear status register	50 <sub>16</sub>							Not acceptable
7	ID check function	F5 <sub>16</sub>	Address (low)	Address (middle)	Address (high)	ID size	ID1	To ID7	Acceptable
8	Download function	FA <sub>16</sub>	Size (low)	Size (high)	Check-sum	Data input	To required number of times		Not acceptable
9	Version data output function	FB <sub>16</sub>	Version data output	Version data output	Version data output	Version data output	Version data output	Version data output to 9th byte	Acceptable
10	Boot ROM area output function	FC <sub>16</sub>	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
11	Read check data	FD <sub>16</sub>	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.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## 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<sub>16</sub>" command code with the 1st byte.
- (2) Transfer addresses A<sub>8</sub> to A<sub>15</sub> and A<sub>16</sub> to A<sub>23</sub> with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, data (D<sub>0</sub>–D<sub>7</sub>) for the page (256 bytes) specified with addresses A<sub>8</sub> to A<sub>23</sub> will be output sequentially from the smallest address first in sync with the fall of the clock.

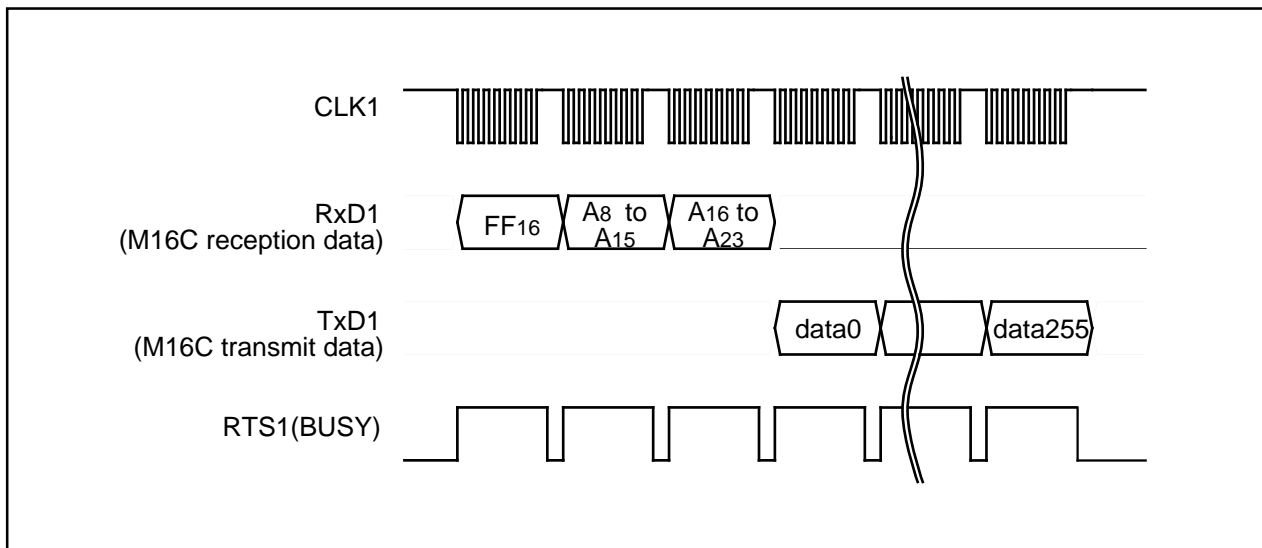


Figure 5.6.2. Timing for page read

## Read Status Register Command

This command reads status information. When the "70<sub>16</sub>" 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.

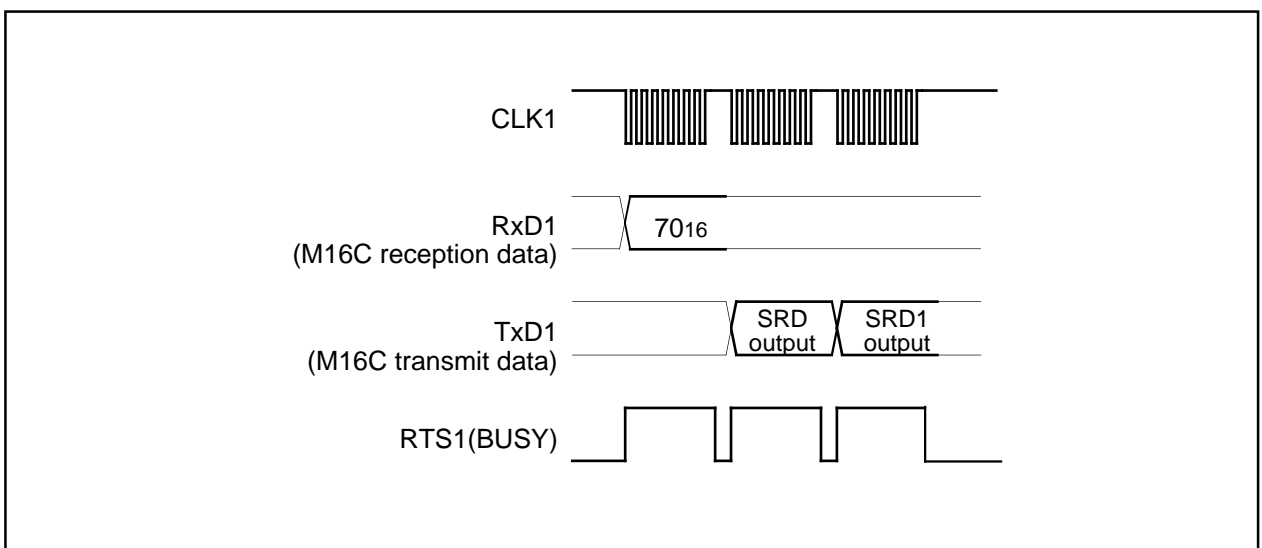


Figure 5.6.3. Timing for reading the status register



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## Clear Status Register Command

This command clears the bits (SR4–SR5) which are set when the status register operation ends in error. When the “50<sub>16</sub>” command code is sent with the 1st byte, the aforementioned bits are cleared. When the clear status register operation ends, the RTS<sub>1</sub> (BUSY) signal changes from the “H” to the “L” level.

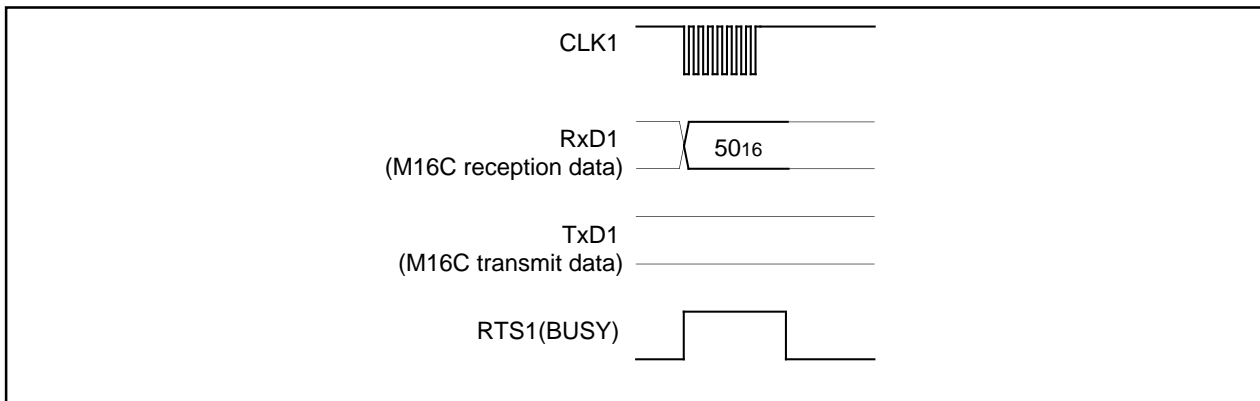


Figure 5.6.4. Timing for clearing the status register

## Page Program Command

This command writes the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page program command as explained here following.

- (1) Transfer the “41<sub>16</sub>” command code with the 1st byte.
- (2) Transfer addresses A<sub>8</sub> to A<sub>15</sub> and A<sub>16</sub> to A<sub>23</sub> with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, as write data (D<sub>0</sub>–D<sub>7</sub>) for the page (256 bytes) specified with addresses A<sub>8</sub> to A<sub>23</sub> is input sequentially from the smallest address first, that page is automatically written.

When reception setup for the next 256 bytes ends, the RTS<sub>1</sub> (BUSY) signal changes from the “H” to the “L” level. The result of the page program can be known by reading the status register. For more information, see the section on the status register.

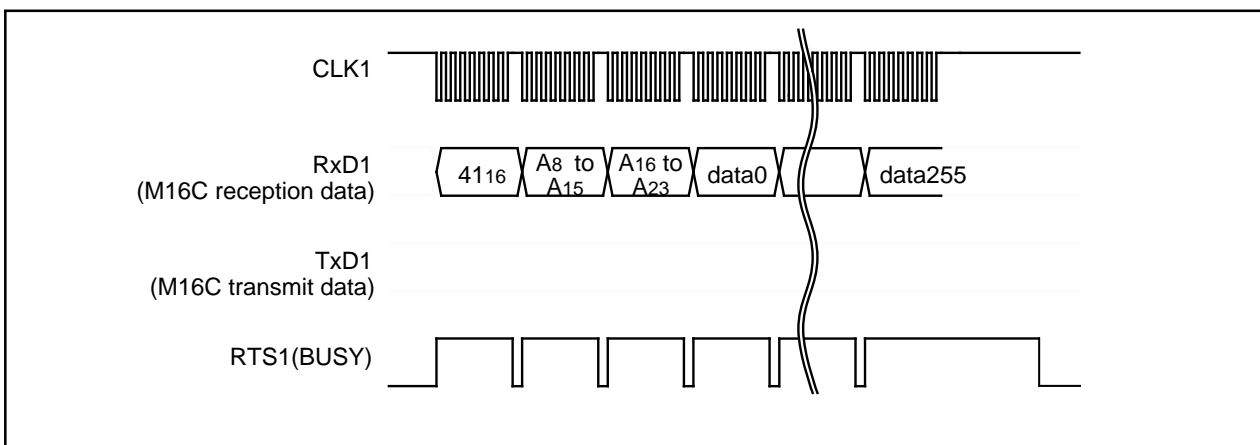


Figure 5.6.5. Timing for the page program

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## Block Erase Command

This command erases the data in the specified block. Execute the block erase command as explained here following.

- (1) Transfer the "2016" command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) Transfer the verify command code "D016" with the 4th byte. With the verify command code, the erase operation will start for the specified block in the flash memory. Write the optional even address of the specified block for addresses A8 to A23.

When block erasing ends, the RTS1 (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.

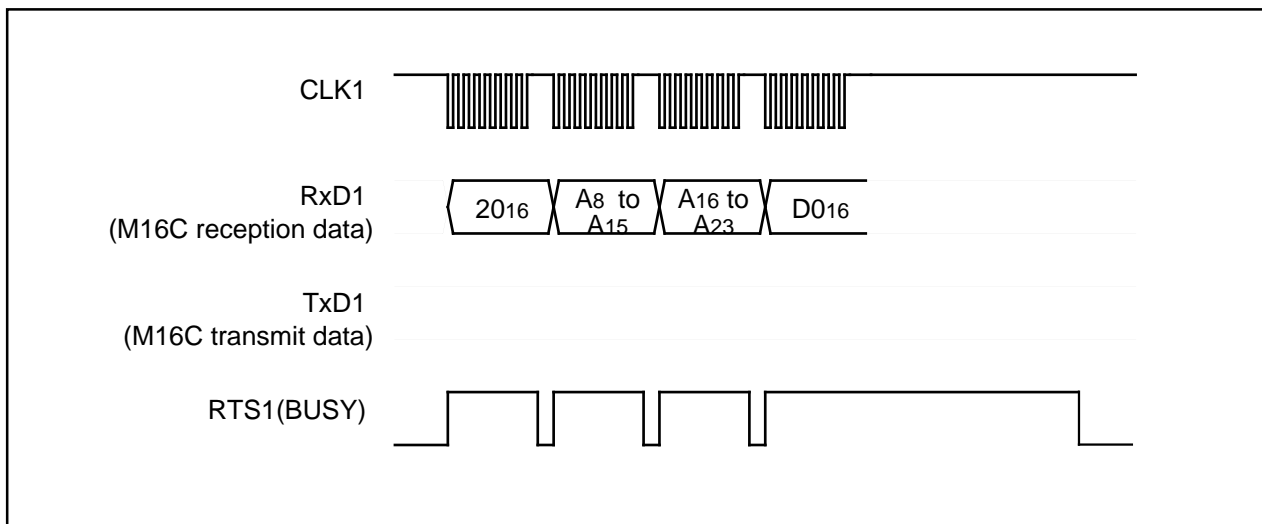


Figure 5.6.6. Timing for block erasing

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## Erase All Blocks Command

This command erases the content of all blocks. Execute the erase all blocks command as explained here following.

- (1) Transfer the "A716" command code with the 1st byte.
  - (2) Transfer the verify command code "D016" 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 RTS1 (BUSY) signal changes from the "H" to the "L" level. The result of the erase operation can be known by reading the status register.

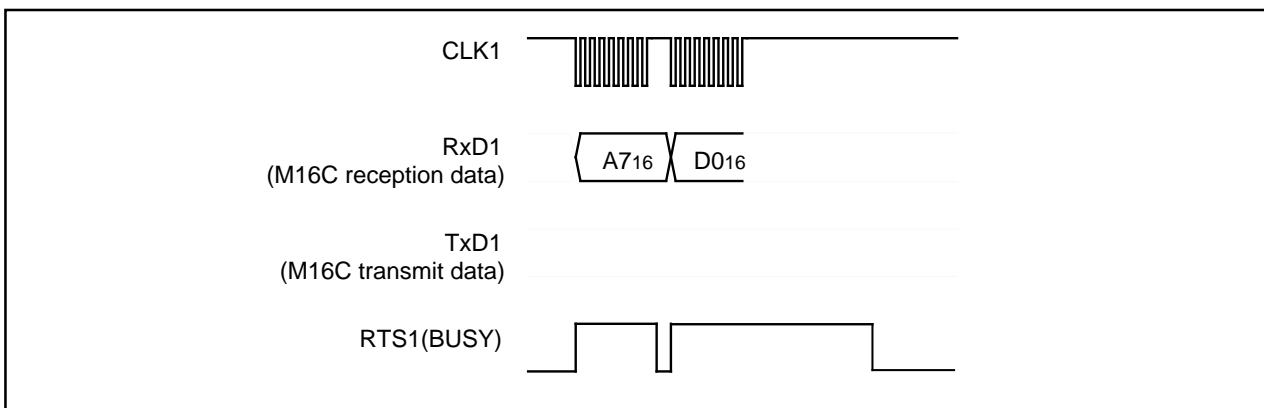


Figure 5.6.7. Timing for erasing all blocks

## Download Command

This command downloads a program to the RAM for execution. Execute the download command as explained here following.

- (1) Transfer the "FA16" 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.

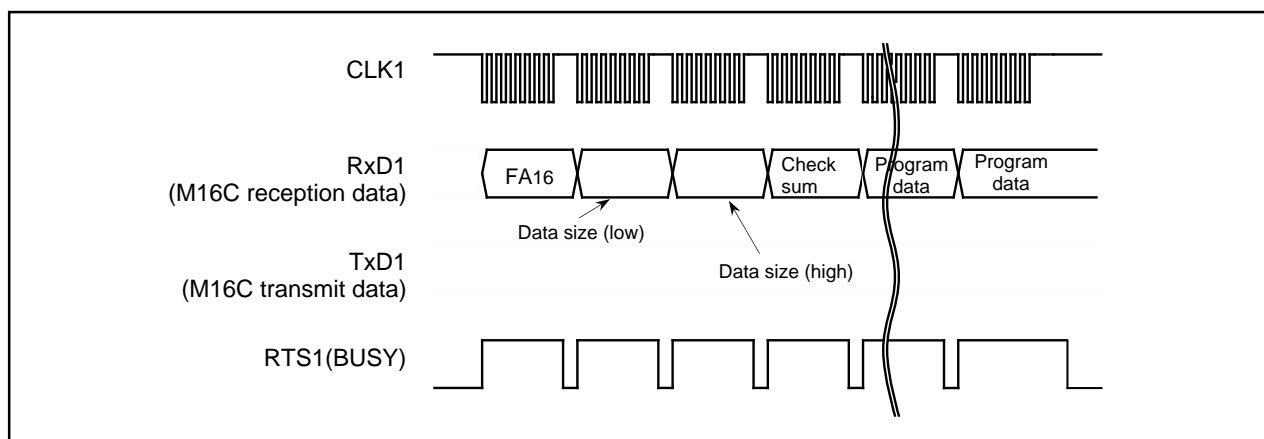


Figure 5.6.8. Timing for download

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## Version Information Output Command

This command outputs the version information of the control program stored in the boot area. Execute the version information output command as explained here following.

- (1) 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.

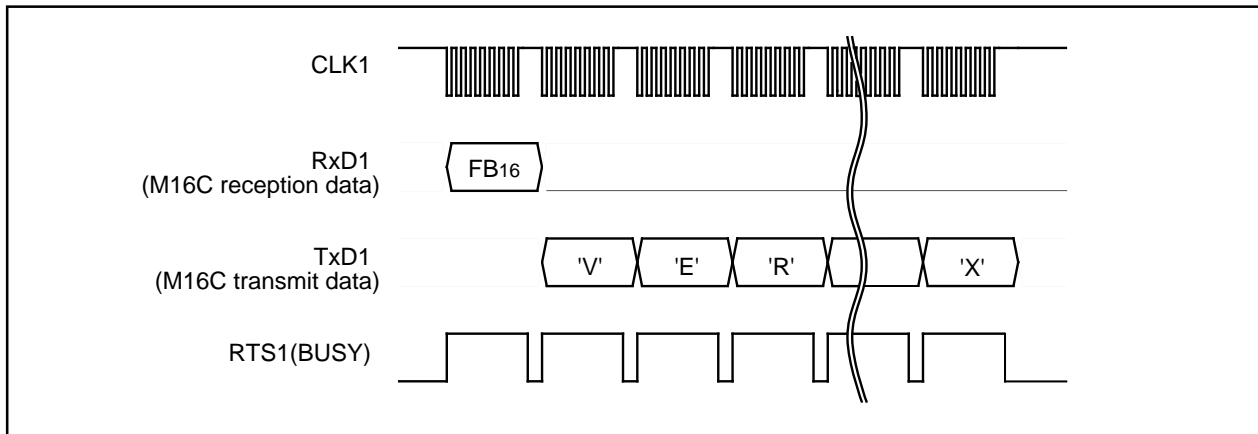


Figure 5.6.9. Timing for version information output

## Boot ROM Area Output 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–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.

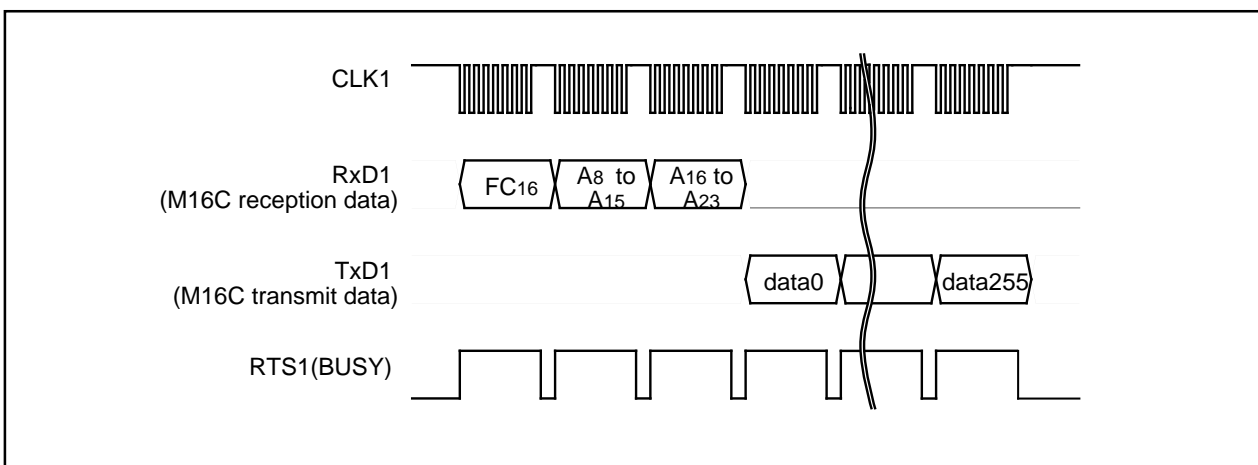


Figure 5.6.10. Timing for boot ROM area output

M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

ID Check

This command checks the ID code. Execute the boot ID check command as explained here following.

- (1) Transfer the “F5<sub>16</sub>” command code with the 1st byte.
- (2) Transfer addresses A<sub>0</sub> to A<sub>7</sub>, A<sub>8</sub> to A<sub>15</sub> and A<sub>16</sub> to A<sub>23</sub> 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.

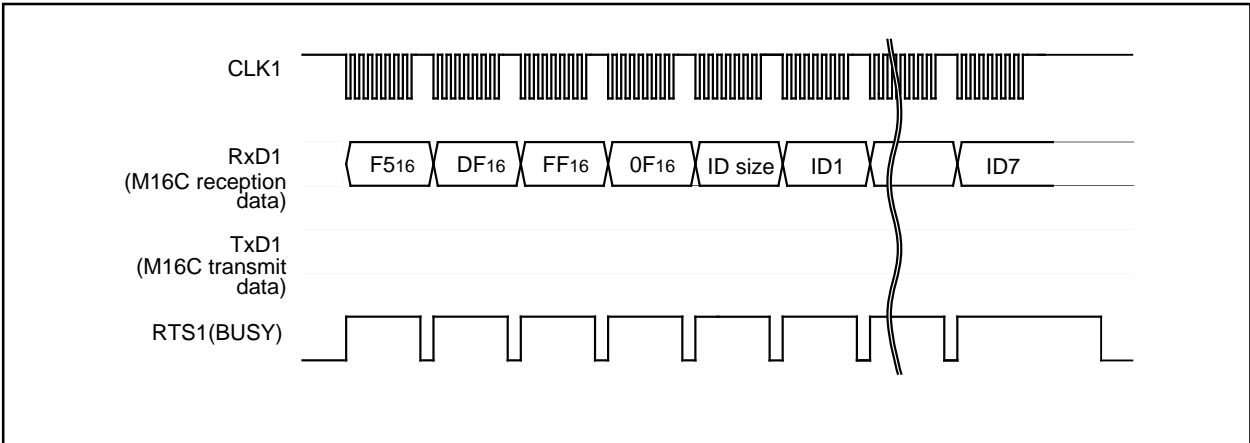


Figure 5.6.11. Timing for the ID check

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 0FFFDF<sub>16</sub>, 0FFFE3<sub>16</sub>, 0FFFE8<sub>16</sub>, 0FFFEF<sub>16</sub>, 0FFFF3<sub>16</sub>, 0FFFF7<sub>16</sub> and 0FFFFB<sub>16</sub>. Write a program into the flash memory, which already has the ID code set for these addresses.

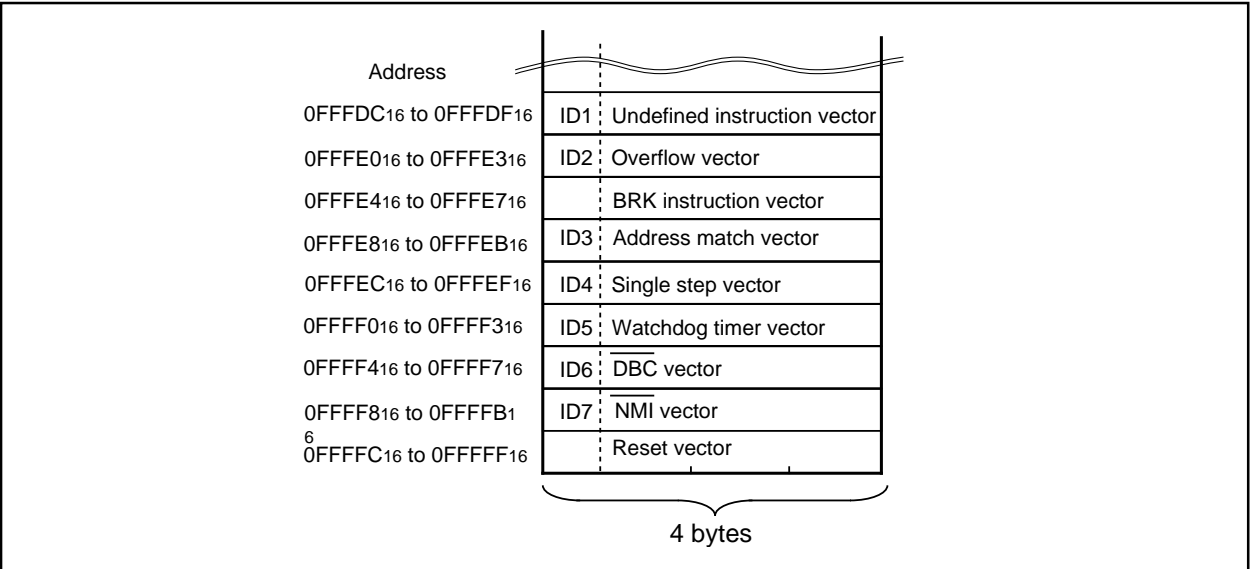


Figure 5.6.12. ID code storage addresses

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## Read Check Data

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. Check data adds write data in 1 byte units and obtains the two's-compliment of the insignificant 2 bytes of the accumulated data.

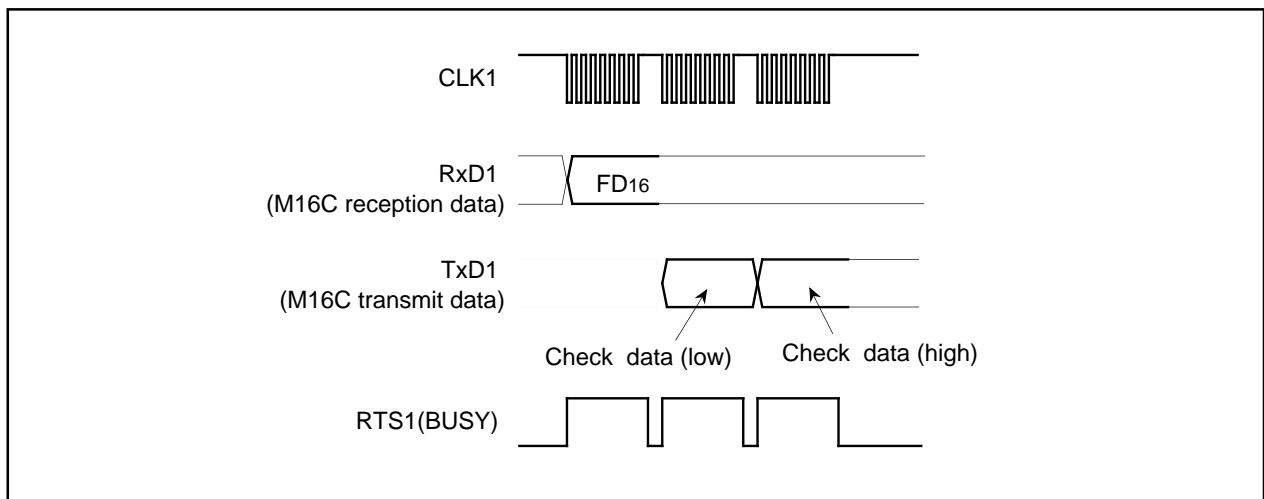


Figure 5.6.13. Timing for the read check data

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 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<sub>16</sub>). Also, the status register is cleared by writing the clear status register command (50<sub>16</sub>).

Table 5.6.2 gives the definition of each status register bit. After clearing the reset, the status register outputs "80<sub>16</sub>".

**Table 5.6.2. Status register (SRD)**

SRD0 bits	Status name	Definition	
		"1"	"0"
SR7 (bit7)	Sequencer 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)	Reserved	-	-
SR2 (bit2)	Reserved	-	-
SR1 (bit1)	Reserved	-	-
SR0 (bit0)	Reserved	-	-

## Sequencer status (SR7)

After power-on, the sequencer status is set to 1(ready).

The sequencer status indicates the operating status of the device. This status bit is set to 0 (busy) during write or erase operation and is set to 1 upon completion of these operations.

## 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".

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 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<sub>16</sub>). Also, status register 1 is cleared by writing the clear status register command (50<sub>16</sub>). Table 5.6.3 gives the definition of each status register 1 bit. "00<sub>16</sub>" is output when power is turned ON and the flag status is maintained even after the reset.

**Table 5.6.3. 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.

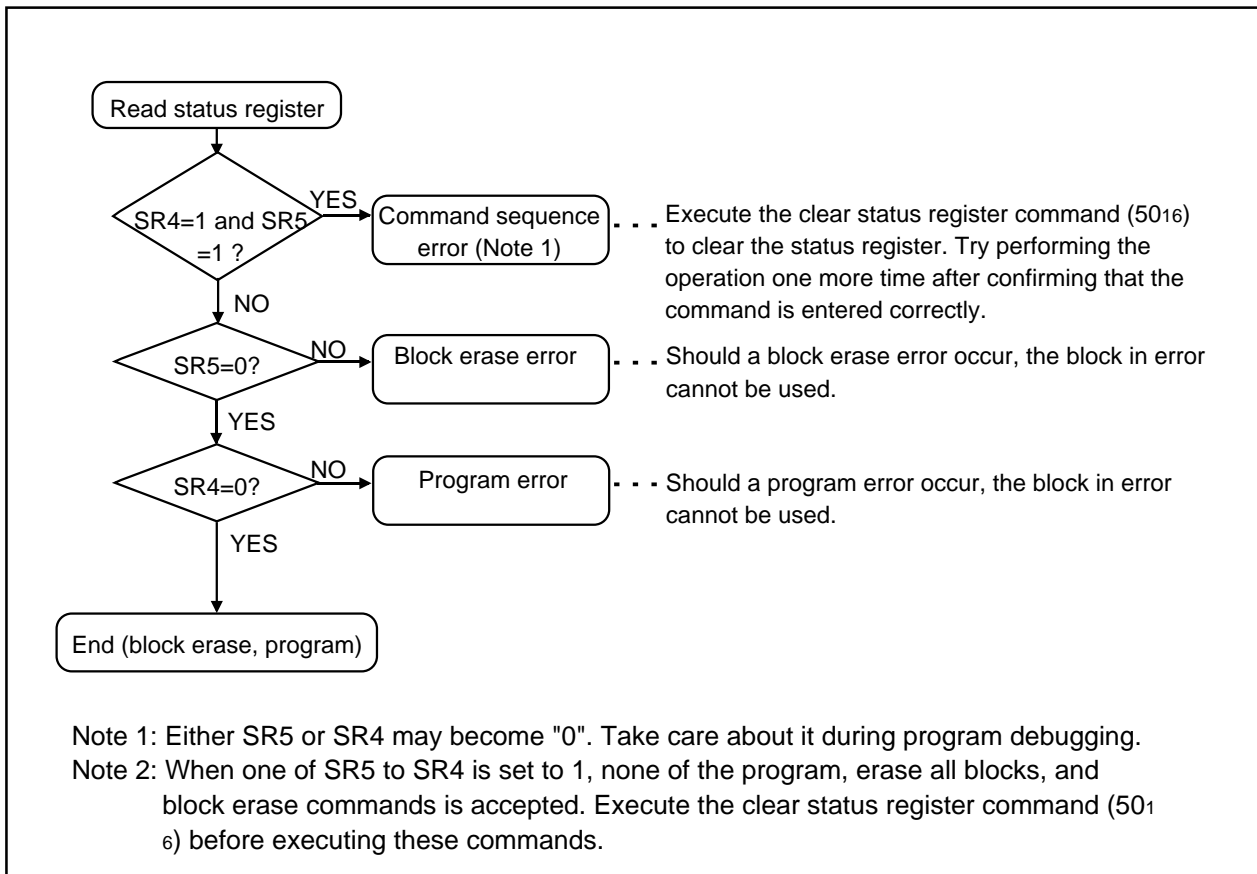
### Data Receive Time Out (SR9)

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.



**M306H2FCFP**SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER**Full Status Check**

Results from executed erase and program operations can be known by running a full status check. Figure 5.6.14 shows a flowchart of the full status check and explains how to remedy errors which occur.



**Figure 5.6.14. Full status check flowchart and remedial procedure for errors**

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 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.

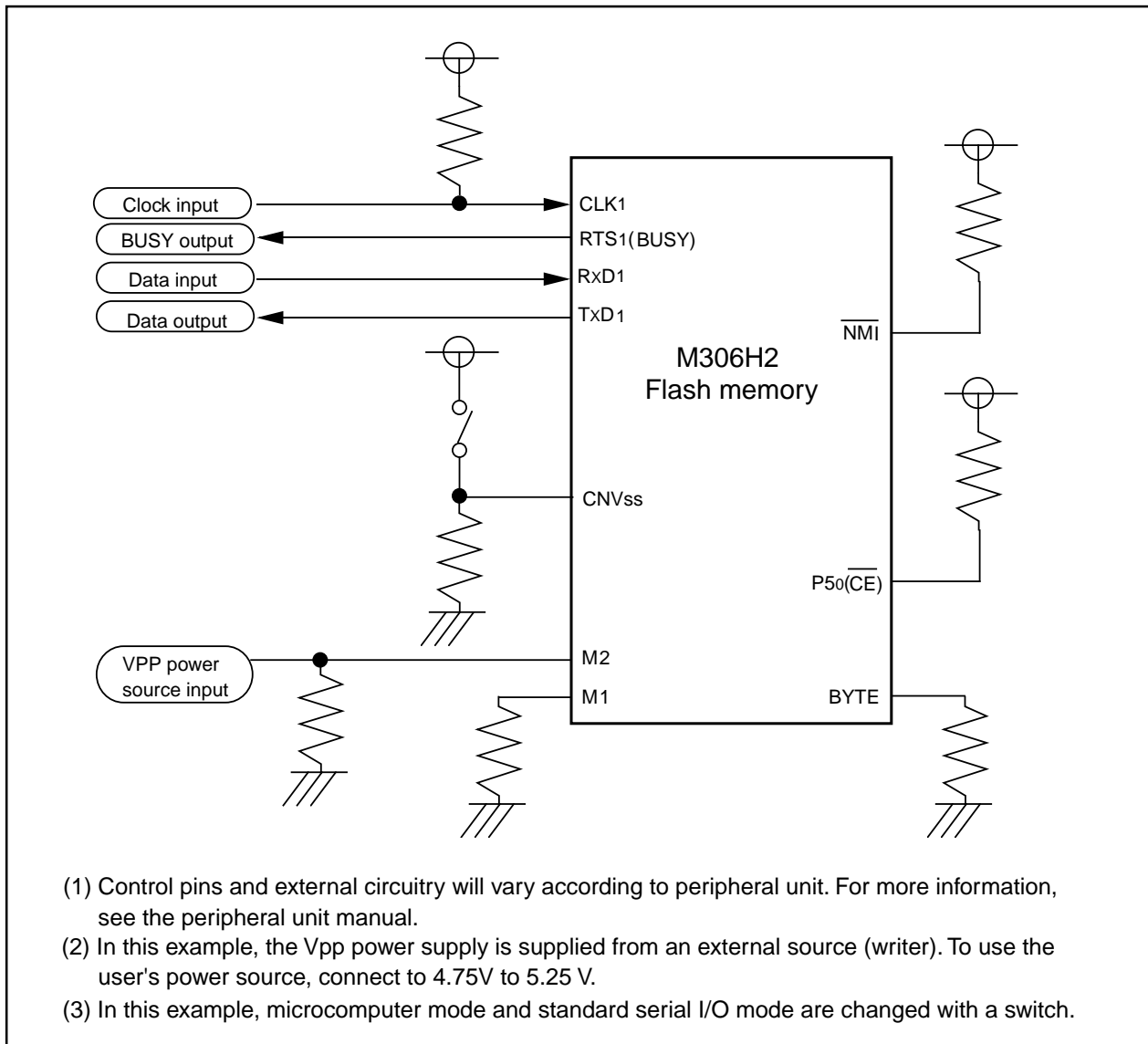


Figure 5.6.15. Example circuit application for the standard serial I/O mode 1

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 5.6.4 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 programmer, etc.) using 2-wire clock-asynchronized serial I/O (UART1). Standard serial I/O mode 2 is engaged by releasing the reset with the P65 (CLK1) pin "L" level.

The TxD1 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 (Figure 5.6.16) are made with a peripheral unit. However, this requires a main clock with a minimum 2 MHz input oscillation frequency. Baud rate can also be changed from 9,600 bps to 19,200, 38,400 or 57,600 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 initial communications with peripheral units, how frequency is identified and software commands.

### Initial communications with peripheral units

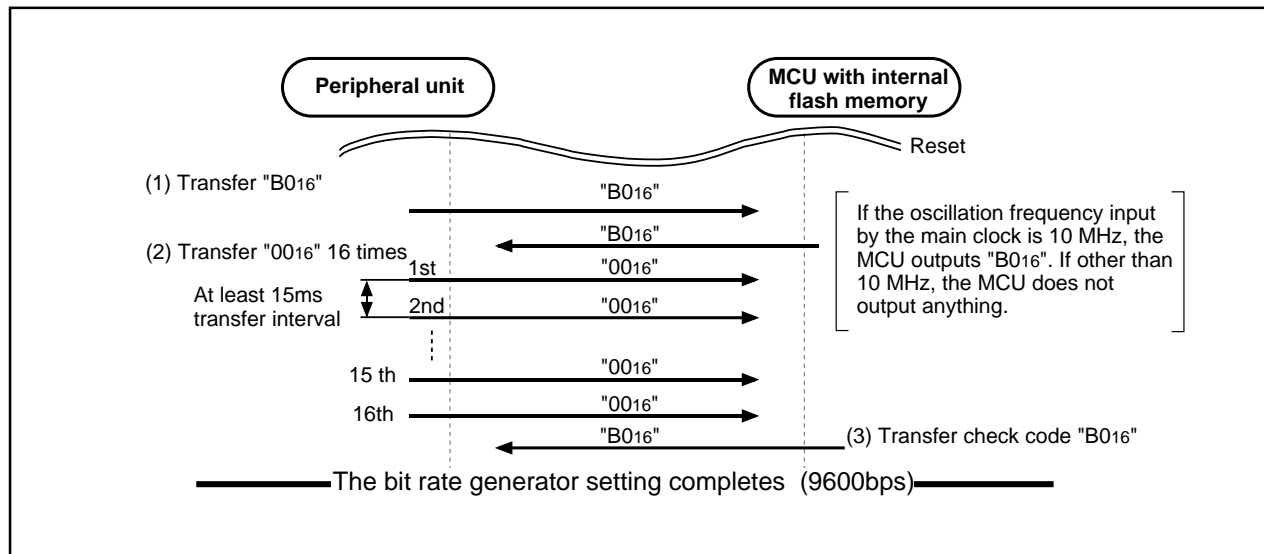
After the reset is released, the bit rate generator is adjusted to 9,600 bps to match the oscillation frequency of the main clock, by sending the code as prescribed by the protocol for initial communications with peripheral units (Figure 5.6.16).

- (1) Transmit "B016" from a peripheral unit. If the oscillation frequency input by the main clock is 10 MHz, the MCU with internal flash memory outputs the "B016" check code. If the oscillation frequency is anything other than 10 MHz, the MCU does not output anything.
- (2) Transmit "0016" from a peripheral unit 16 times. (The MCU with internal flash memory sets the bit rate generator so that "0016" can be successfully received.)
- (3) The MCU with internal flash memory outputs the "B016" check code and initial communications end successfully \*1. Initial communications must be transmitted at a speed of 9,600 bps and a transfer interval of a minimum 15 ms. Also, the baud rate at the end of initial communications is 9,600 bps.

\*1. If the peripheral unit cannot receive "B016" successfully, change the oscillation frequency of the main clock.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER



**Figure 5.6.16. Peripheral unit and initial communication**

## How frequency is identified

When "0016" data is received 16 times from a peripheral unit at a baud rate of 9,600 bps, the value of the bit rate generator is set to match the operating frequency (2 - 10 MHz). The highest speed is taken from the first 8 transmissions and the lowest from the last 8. These values are then used to calculate the bit rate generator value for a baud rate of 9,600 bps.

Baud rate cannot be attained with some operating frequencies. Table 5.6.4 gives the operation frequency and the baud rate that can be attained for.

**Table 5.6.4 Operation frequency and the baud rate**

Operation frequency (MHz)	Baud rate 9,600bps	Baud rate 19,200bps	Baud rate 38,400bps	Baud rate 57,600bps
10MHz	✓	✓	—	✓
8MHz	✓	✓	—	✓
7.3728MHz	✓	✓	✓	✓
6MHz	✓	✓	✓	—
5MHz	✓	✓	—	—
4.5MHz	✓	✓	—	✓
4.194304MHz	✓	✓	✓	—
4MHz	✓	✓	—	—
3.58MHz	✓	✓	✓	✓
3MHz	✓	✓	✓	—
2MHz	✓	—	—	—

✓ : Communications possible

— : Communications not possible

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 5.6.5 Software Commands

Table 5.6.5 lists software commands. In the standard serial I/O mode 2, erase operations, programs and reading are controlled by transferring software commands via the RxD1 pin. Standard serial I/O mode 2 adds four transmission speed commands - 9,600, 19,200, 38,400 and 57,600 bps - to the software commands of standard serial I/O mode 1. Software commands are explained here below.

**Table 5.6.5. Software commands (Standard serial I/O mode 2)**

	Control command	1st byte transfer	2nd byte	3rd byte	4th byte	5th byte	6th byte		When ID is not verified
1	Page read	FF <sub>16</sub>	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
2	Page program	41 <sub>16</sub>	Address (middle)	Address (high)	Data input	Data input	Data input	Data input to 259th byte	Not acceptable
3	Block erase	20 <sub>16</sub>	Address (middle)	Address (high)	D0 <sub>16</sub>				Not acceptable
4	Erase all unlocked blocks	A7 <sub>16</sub>	D0 <sub>16</sub>						Not acceptable
5	Read status register	70 <sub>16</sub>	SRD output	SRD1 output					Acceptable
6	Clear status register	50 <sub>16</sub>							Not acceptable
7	ID check function	F5 <sub>16</sub>	Address (low)	Address (middle)	Address (high)	ID size	ID1	To ID7	Acceptable
8	Download function	FA <sub>16</sub>	Size (low)	Size (high)	Check-sum	Data input	To required number of times		Not acceptable
9	Version data output function	FB <sub>16</sub>	Version data output	Version data output	Version data output	Version data output	Version data output	Version data output to 9th byte	Acceptable
10	Boot ROM area output function	FC <sub>16</sub>	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
11	Read check data	FD <sub>16</sub>	Check data (low)	Check data (high)					Not acceptable
12	Baud rate 9600	B0 <sub>16</sub>	B0 <sub>16</sub>						Acceptable
13	Baud rate 19200	B1 <sub>16</sub>	B1 <sub>16</sub>						Acceptable
14	Baud rate 38400	B2 <sub>16</sub>	B2 <sub>16</sub>						Acceptable
15	Baud rate 57600	B3 <sub>16</sub>	B3 <sub>16</sub>						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.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## 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<sub>16</sub>" command code with the 1st byte.
- (2) Transfer addresses A<sub>8</sub> to A<sub>15</sub> and A<sub>16</sub> to A<sub>23</sub> with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, data (D<sub>0</sub>–D<sub>7</sub>) for the page (256 bytes) specified with addresses A<sub>8</sub> to A<sub>23</sub> will be output sequentially from the smallest address first.

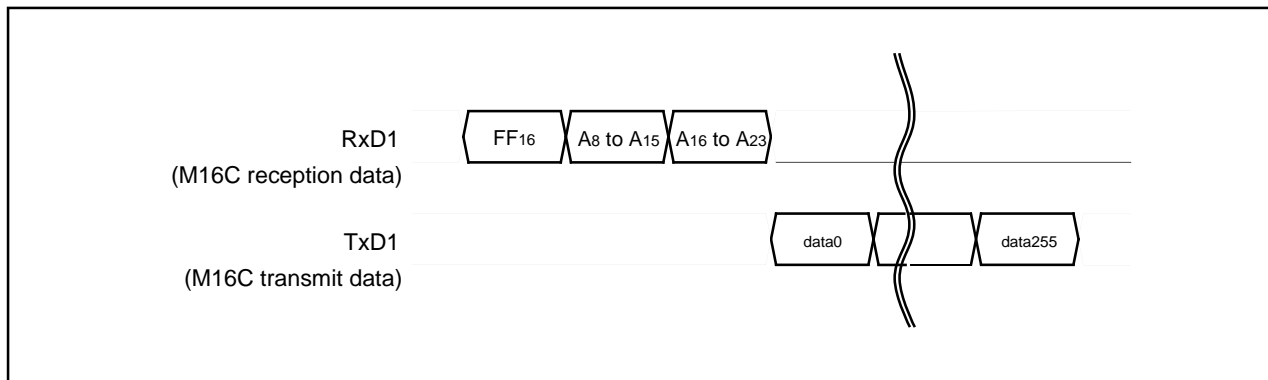


Figure 5.6.17. Timing for page read

## Read Status Register Command

This command reads status information. When the "70<sub>16</sub>" 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.

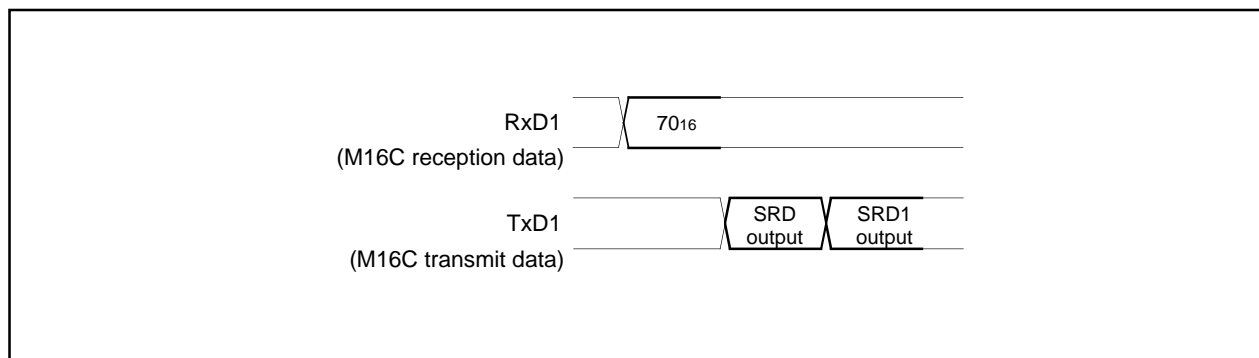


Figure 5.6.18. Timing for reading the status register

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## Clear Status Register Command

This command clears the bits (SR4–SR5) which are set when the status register operation ends in error. When the “50<sub>16</sub>” command code is sent with the 1st byte, the aforementioned bits are cleared.

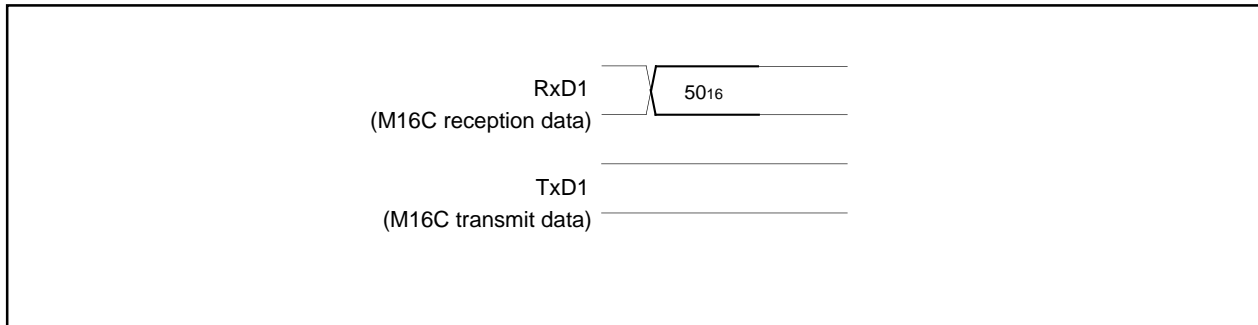


Figure 5.6.19. Timing for clearing the status register

## Page Program Command

This command writes the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page program command as explained here following.

- (1) Transfer the “41<sub>16</sub>” command code with the 1st byte.
- (2) Transfer addresses A<sub>8</sub> to A<sub>15</sub> and A<sub>16</sub> to A<sub>23</sub> with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, as write data (D<sub>0</sub>–D<sub>7</sub>) for the page (256 bytes) specified with addresses A<sub>8</sub> to A<sub>23</sub> 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.

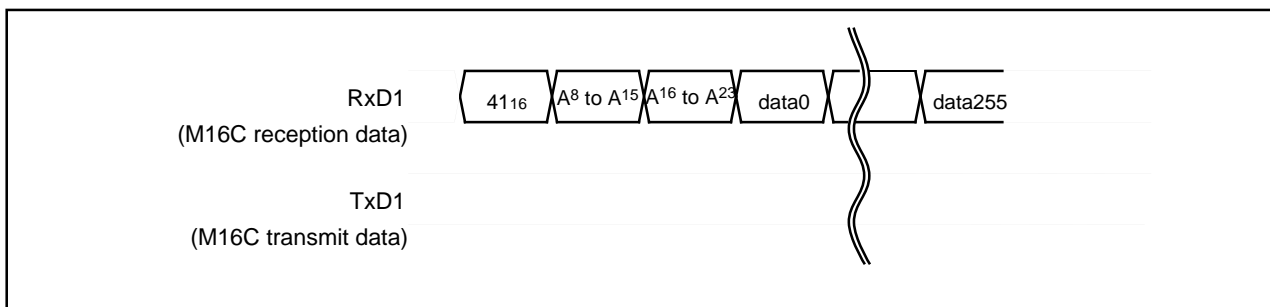


Figure 5.6.20. Timing for the page program

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## 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<sub>16</sub>" command code with the 1st byte.
- (2) Transfer addresses A<sub>8</sub> to A<sub>15</sub> and A<sub>16</sub> to A<sub>23</sub> with the 2nd and 3rd bytes respectively.
- (3) Transfer the verify command code "D0<sub>16</sub>" with the 4th byte. With the verify command code, the erase operation will start for the specified block in the flash memory. Write the optional even address of the specified block for addresses A<sub>8</sub> to A<sub>23</sub>.

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.

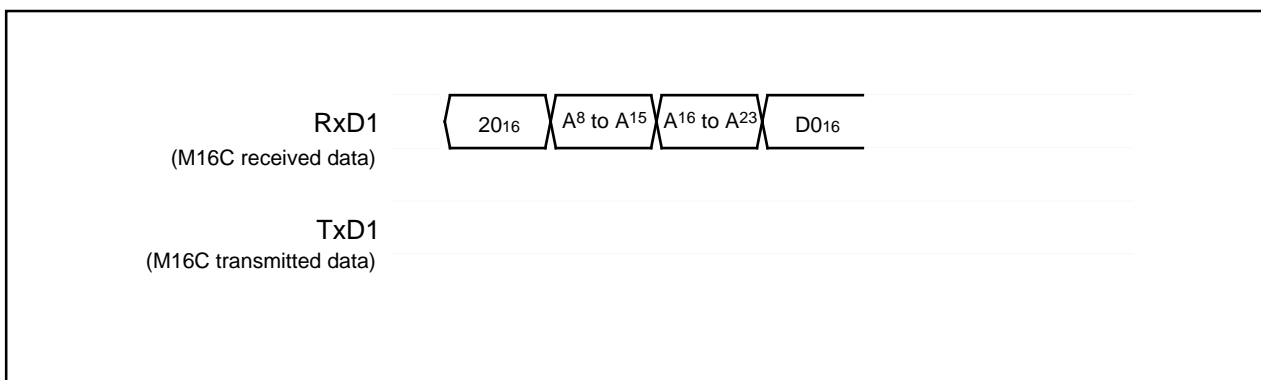


Figure 5.6.21. Timing for block erasing

## Erase All Blocks Command

This command erases the content of all blocks. Execute the erase all blocks command as explained here following.

- (1) Transfer the "A7<sub>16</sub>" command code with the 1st byte.
- (2) Transfer the verify command code "D0<sub>16</sub>" 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.

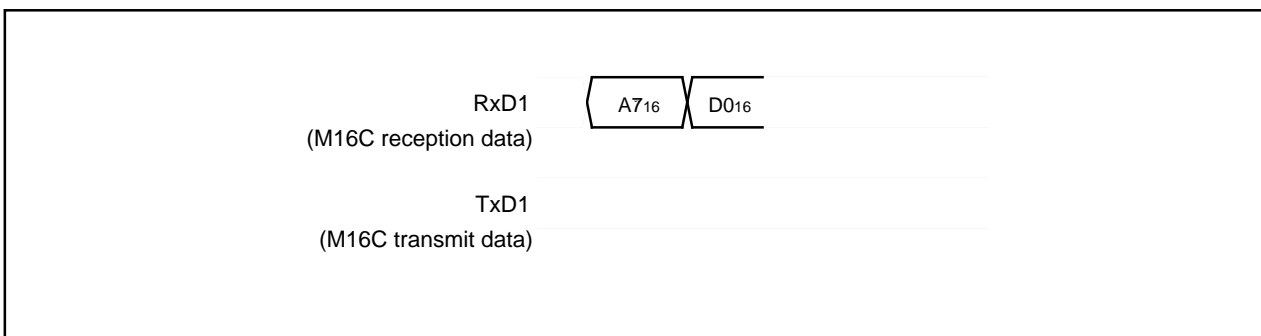


Figure 5.6.22. Timing for erasing all blocks



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## Download Command

This command downloads a program to the RAM for execution. Execute the download command as explained here following.

- (1) Transfer the "FA<sub>16</sub>" 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.

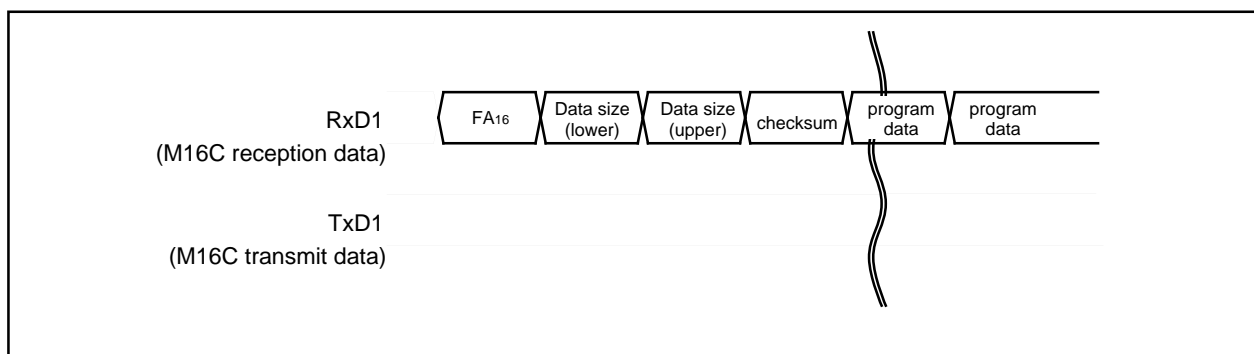


Figure 5.6.23. Timing for download

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## Version Information Output Command

This command outputs the version information of the control program stored in the boot area. Execute the version information output command as explained here following.

- (1) 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.

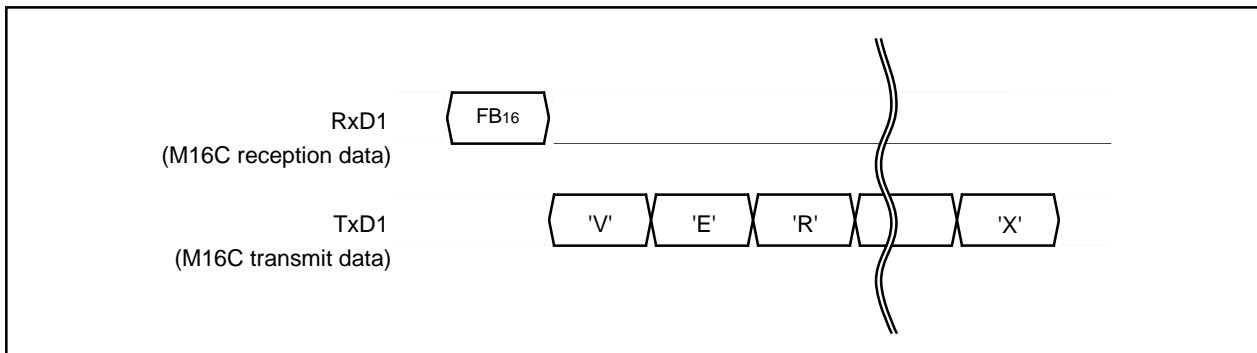


Figure 5.6.24. Timing for version information output

## Boot ROM Area Output 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–D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first.

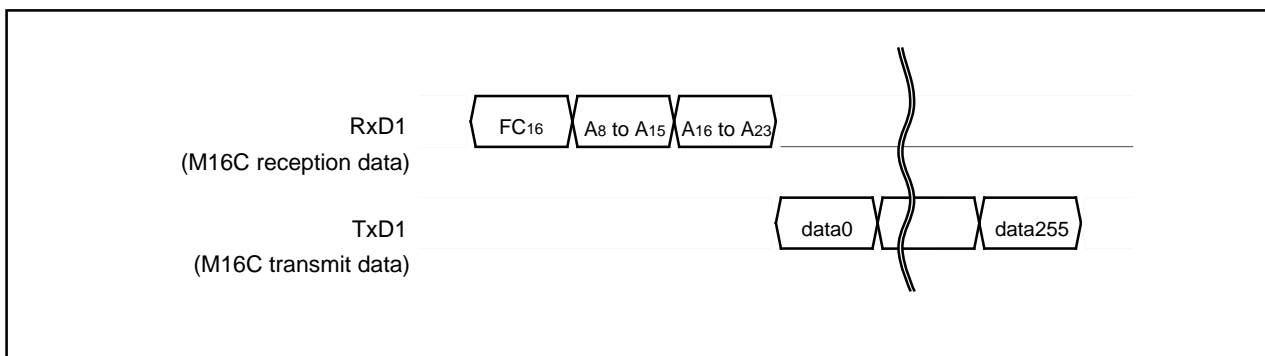


Figure 5.6.25. Timing for boot ROM area output

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

### ID Check

This command checks the ID code. Execute the boot ID check command as explained here following.

- (1) Transfer the “F5<sub>16</sub>” command code with the 1st byte.
- (2) Transfer addresses A0 to A7, A8 to A15 and A16 to A23 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.

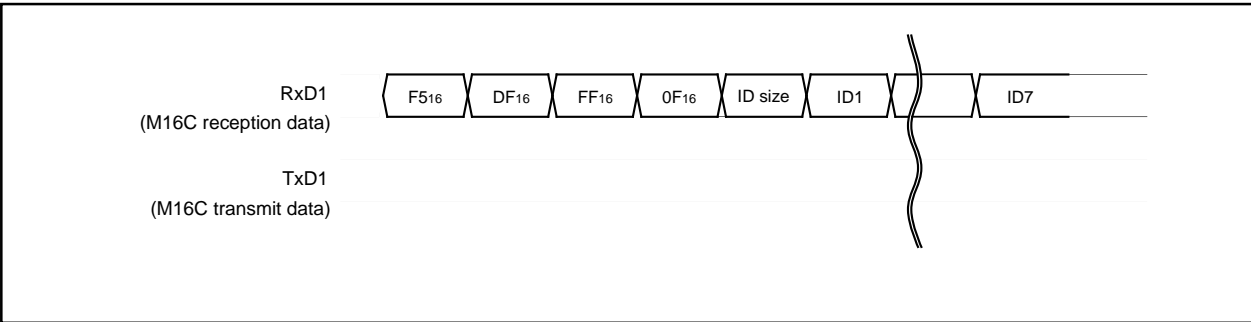


Figure 5.6.26. Timing for the ID check

### 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 0FFFD<sub>16</sub>, 0FFFE<sub>316</sub>, 0FFFE<sub>B16</sub>, 0FFFE<sub>F16</sub>, 0FFFF<sub>316</sub>, 0FFFF<sub>716</sub> and 0FFFF<sub>B16</sub>. Write a program into the flash memory, which already has the ID code set for these addresses.

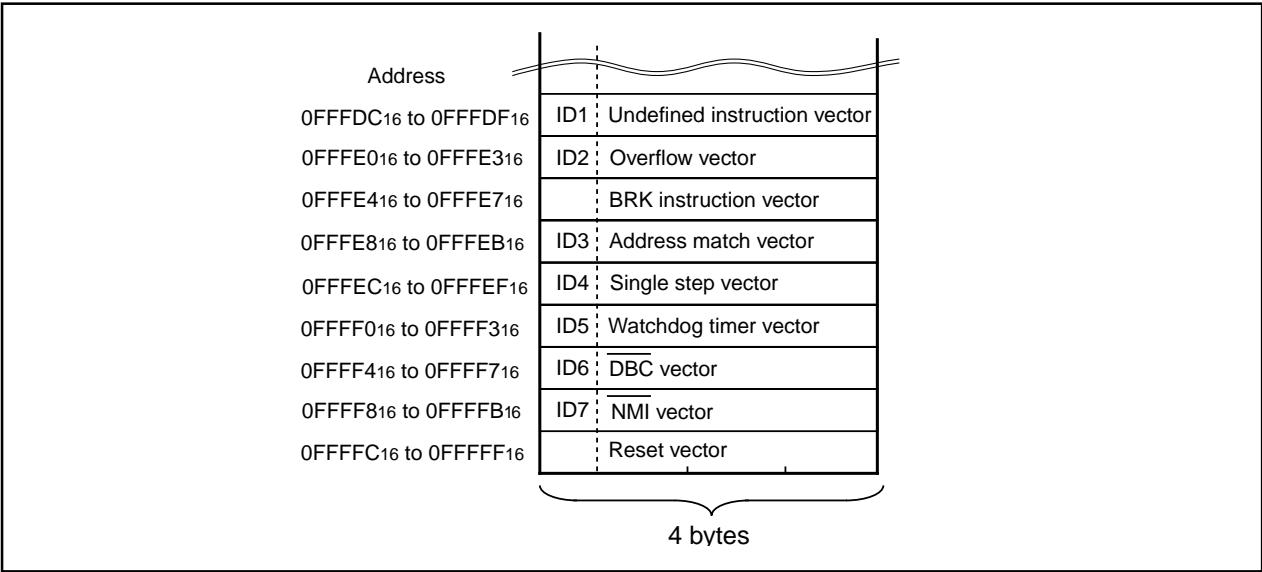


Figure 5.6.27. ID code storage addresses

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## Read Check Data

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. Check data adds write data in 1 byte units and obtains the two's-compliment of the insignificant 2 bytes of the accumulated data.

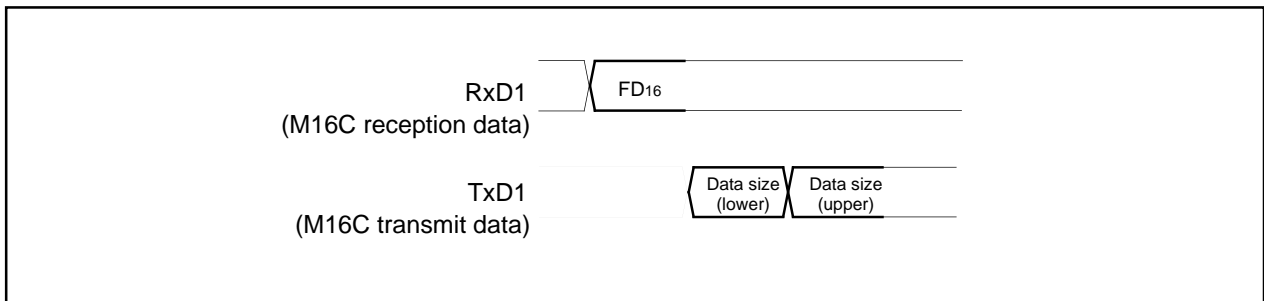


Figure 5.6.28. Timing for the read check data

## Baud Rate 9600

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.

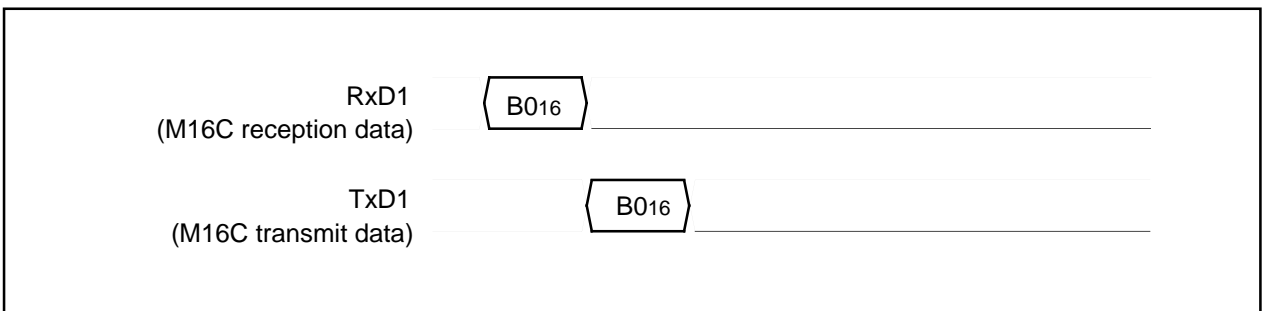


Figure 5.6.29. Timing of baud rate 9600

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## Baud Rate 19200

This command changes baud rate to 19,200 bps. Execute it as follows.

- (1) Transfer the "B116" command code with the 1st byte.
- (2) After the "B116" check code is output with the 2nd byte, change the baud rate to 19,200 bps.

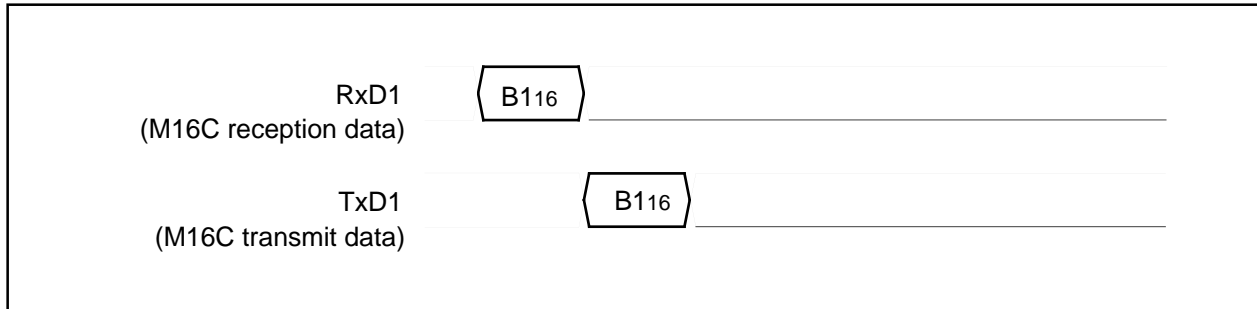


Figure 5.6.30. Timing of baud rate 19200

## Baud Rate 38400

This command changes baud rate to 38,400 bps. Execute it as follows.

- (1) Transfer the "B216" command code with the 1st byte.
- (2) After the "B216" check code is output with the 2nd byte, change the baud rate to 38,400 bps.

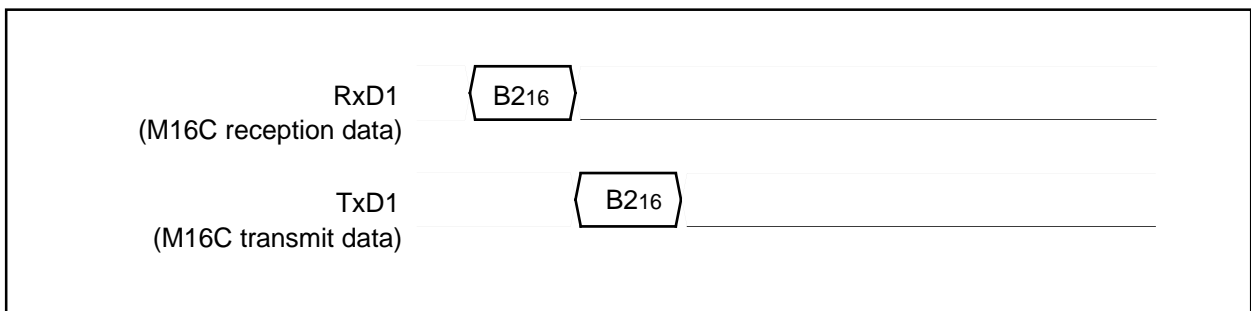


Figure 5.6.31. Timing of baud rate 38400

## Baud Rate 57600

This command changes baud rate to 57,600 bps. Execute it as follows.

- (1) Transfer the "B316" command code with the 1st byte.
- (2) After the "B316" check code is output with the 2nd byte, change the baud rate to 57,600 bps.

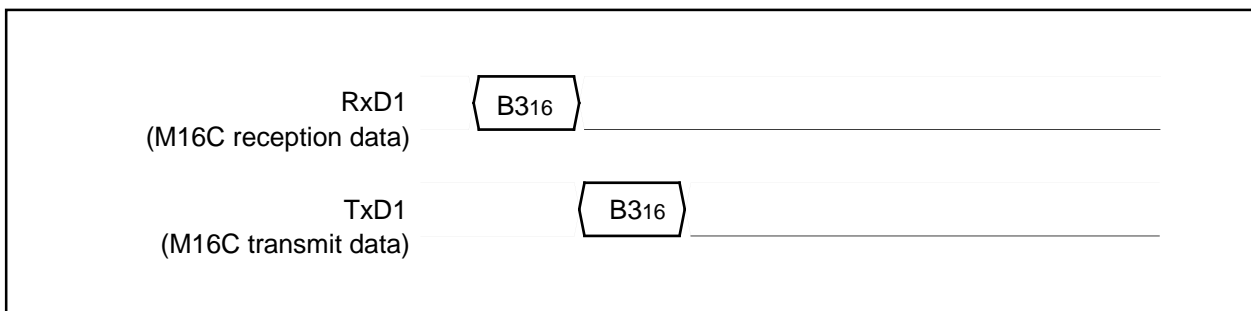


Figure 5.6.32. Timing of baud rate 57600

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 5.6.6 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.

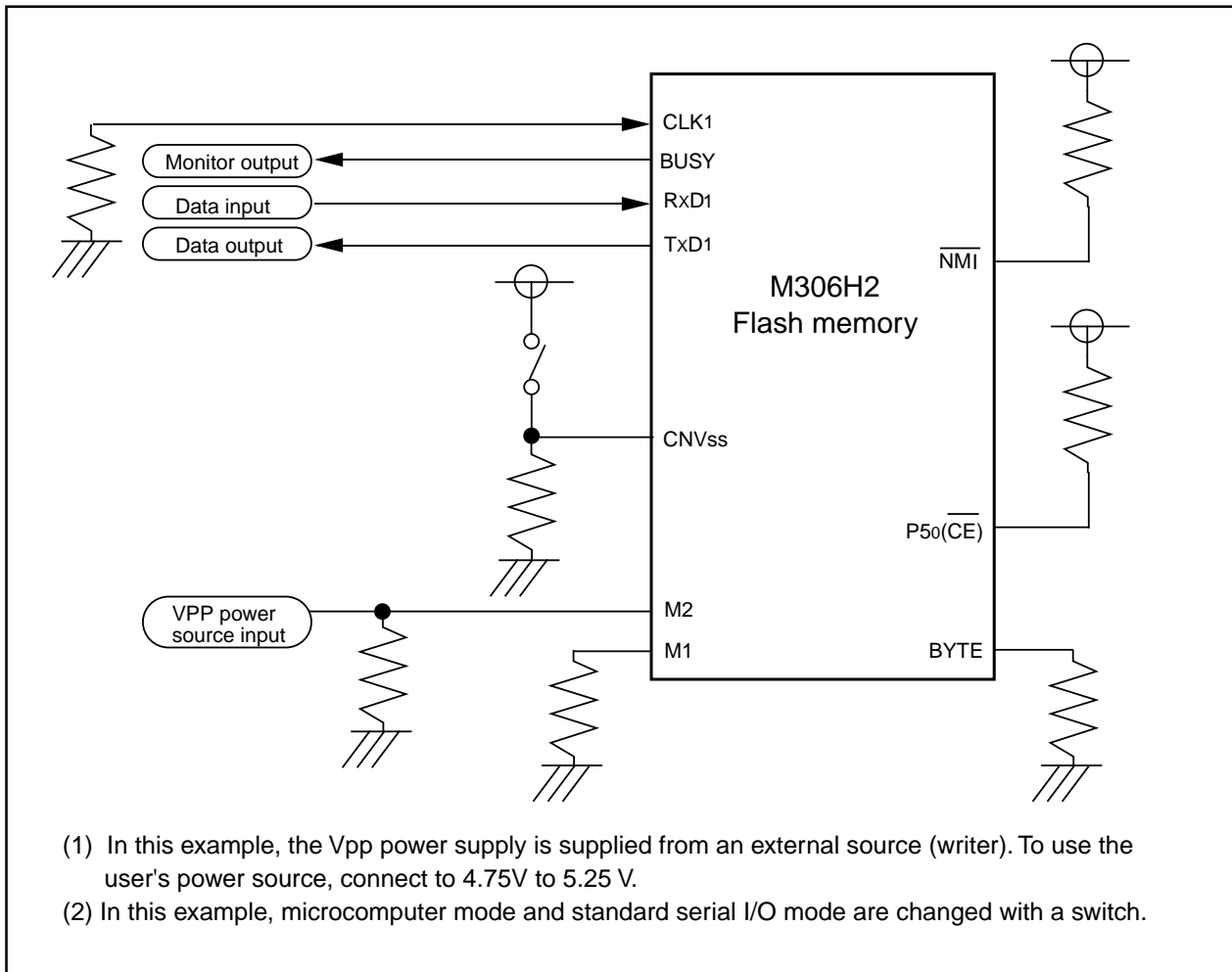


Figure 5.6.23. Example circuit application for the standard serial I/O mode 2

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

## 5.7 Electrical Characteristics

**Table 5.7.1 Absolute maximum ratings**

Symbol	Parameter		Condition	Rated value	Unit
V <sub>cc</sub>	Supply voltage		V <sub>cc</sub> =AV <sub>cc</sub>	- 0.3 to 5.75	V
AV <sub>cc</sub>	Analog supply voltage		V <sub>cc</sub> =AV <sub>cc</sub>	- 0.3 to 5.75	V
M2	Supply voltage for program/erase			- 0.3 to 5.75	V
V <sub>i</sub>	Input voltage	RESET, CNV <sub>ss</sub> , BYTE, P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>2</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , VREF, XIN, M1		- 0.3 to V <sub>cc</sub> +0.3	V
		P7 <sub>0</sub> , P7 <sub>1</sub>		- 0.3 to 5.75	V
V <sub>o</sub>	Output voltage	P0 <sub>0</sub> to P0 <sub>7</sub> , P1 <sub>0</sub> to P1 <sub>7</sub> , P2 <sub>0</sub> to P2 <sub>7</sub> , P3 <sub>0</sub> to P3 <sub>7</sub> , P4 <sub>0</sub> to P4 <sub>7</sub> , P5 <sub>0</sub> to P5 <sub>7</sub> , P6 <sub>0</sub> to P6 <sub>7</sub> , P7 <sub>2</sub> to P7 <sub>7</sub> , P8 <sub>0</sub> to P8 <sub>4</sub> , P8 <sub>6</sub> , P8 <sub>7</sub> , P9 <sub>0</sub> to P9 <sub>7</sub> , P10 <sub>0</sub> to P10 <sub>7</sub> , XOUT, P11		- 0.3 to V <sub>cc</sub> +0.3	V
		P7 <sub>0</sub> , P7 <sub>1</sub> ,		- 0.3 to 5.75	V
P <sub>d</sub>	Power dissipation		T <sub>a</sub> =25 °C	1000	mW
T <sub>opr</sub>	Operating ambient temperature (Note)			25 ± 5	°C
T <sub>stg</sub>	Storage temperature			- 40 to 125	°C

Note: It is a value in flash memory mode. Other parameter becomes same as a value in microcomputer mode.

# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

**Table 5.7.2 DC electrical characteristics (referenced to  $V_{CC} = 5.0V$  at  $T_a = 25^{\circ}C$  unless otherwise specified)**

Symbol	Parameter	Condition	Rated value			Unit
			Min.	Typ.	Max.	
$I_{PP1}$	$V_{PP}$ power supply current (at read)	$V_{PP} = V_{CC}$			100	$\mu A$
$I_{PP2}$	$V_{PP}$ power supply current (at program)	$V_{PP} = V_{CC}$			60	mA
$I_{PP3}$	$V_{PP}$ power supply current (at erase)	$V_{PP} = V_{CC}$			70	mA
$V_{PP}$	$V_{PP}$ power supply voltage		4.75		5.25	V



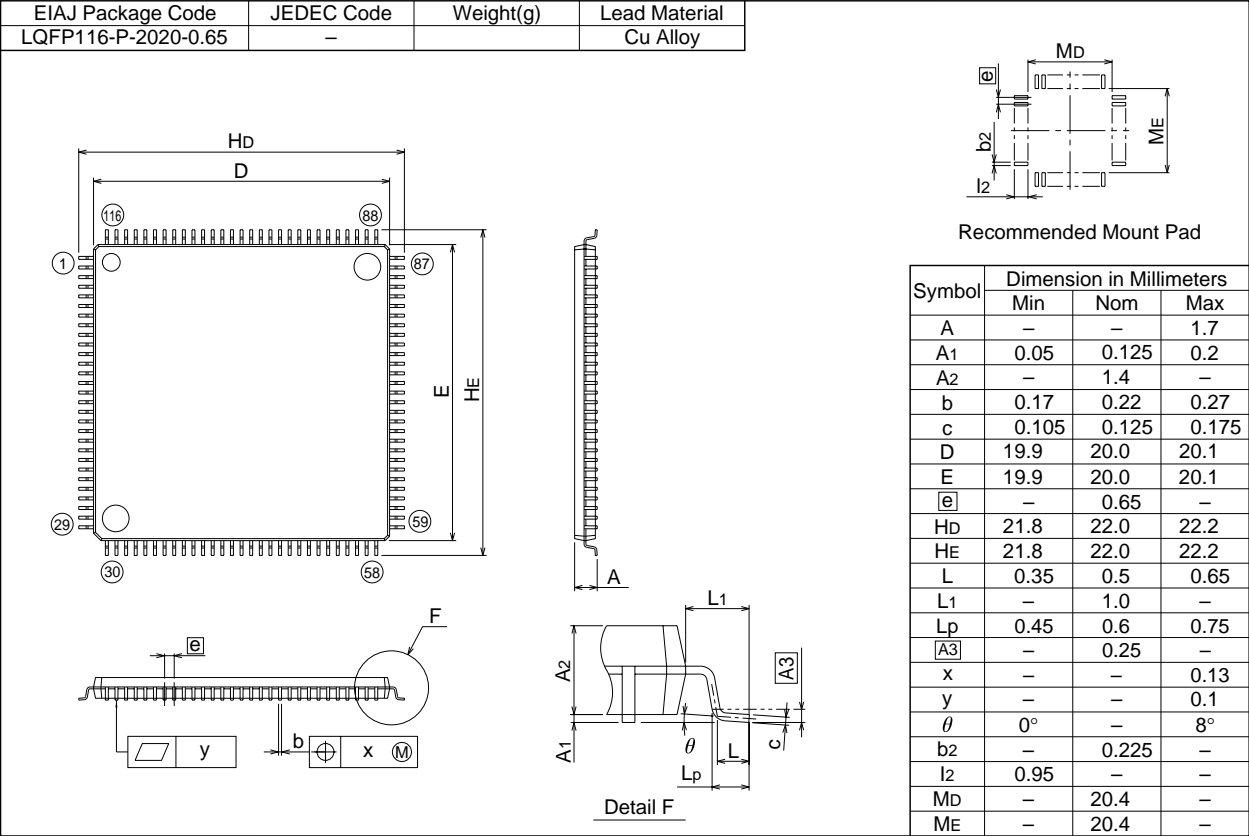
M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
with DATA ACQUISITION CONTROLLER

6. PACKAGE OUTLINE

116P6A-A (MMP)

Plastic 116pin 20X20mm body LQFP



# M306H2FCFP

SINGLE-CHIP 16-BIT CMOS MICROCOMPUTER  
 with DATA ACQUISITION CONTROLLER

## 7. DIFFERENCES BETWEEN M306H2MC-XXXFP AND M306H2FCFP

Item	M306H2MC-XXXFP	M306H2FCFP
Processor mode	Single-chip mode Memory extension mode Microprocessor mode(Note 1)	Single-chip mode Memory expansion mode
ROM type	Mask ROM	Flash memory
M1/M2 pin function	M1: Test input (Connect it to the Vss pin.)	M1: Chip mode setting input (Note 2)
	M2: Test input (Connect it to the Vss pin.)	M2: Flash memory rewriting power supply input
CNVss pin function	This pin switches between processor modes.	Normally connect it to the Vss pin.(Note 2)
BYTE pin function	This pin switches between external data buses.	This pin switches between external data buses. (Note 3)

Notes 1: Microprocessor mode can be used only on M306H2MC-XXXFP.

2: These pins are used to control flash memory mode. For more information, consult M306H2FCFP flash memory specifications.

3 :When use flash memory mode (parallel I/O, standard serial I/O and CPU rewriting mode), connect with VSS pin.

4 :Since it differs in part about an electrical characteristics, check the specification of M306H2MC-XXXFP and M306H2FCFP.

5 :There are differences in electric characteristics, operation margin, noise immunity, and noise radiation between Mask ROM and Flash Memory version MCUs due to the difference in the manufacturing processes. When manufacturing an application system with the Flash Memory version and then switching to use of the Mask ROM version, please perform sufficient evaluations for the commercial samples of the Mask ROM version.

