What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	-
Core Size	-
Speed	-
Connectivity	-
Peripherals	-
Number of I/O	-
Program Memory Size	-
Program Memory Type	-
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	-
Data Converters	-
Oscillator Type	-
Operating Temperature	-
Mounting Type	-
Package / Case	-
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/m30291fathp-u3aaj3

Table 1.12 Pin Characteristics for 80-Pin Package

Pin No.	Control Pin	Port	Interrupt Pin	Timer Pin	Timer S Pin	UART/CAN Pin	Multi-master I²C bus Pin	Analog Pin
1		$P9_5$				CLK_4		$AN2_5$
2		$P9_3$				CTX		$AN2_4$
3		$P9_2$		TB_{2IN}		CRX		$AN3_2$
4		$P9_1$		TB_{1IN}				$AN3_1$
5	CLK_{OUT}	$P9_0$		TB_{0IN}				$AN3_0$
6	CNVss							
7	X_{CIN}	$P8_7$						
8	X_{COUT}	$P8_6$						
9	$\overline{RESET}$							
10	X_{OUT}							
11	Vss							
12	X_{IN}							
13	Vcc							
14		$P8_5$	$\overline{NMI}$	SD				
15		$P8_4$	$\overline{INT_2}$	ZP				
16		$P8_3$	$\overline{INT_1}$					
17		$P8_2$	$\overline{INT_0}$					
18		$P8_1$		TA_{4IN} / $\overline{U}$				
19		$P8_0$		TA_{4OUT} / U				
20		$P7_7$		TA_{3IN}				
21		$P7_6$		TA_{3OUT}				
22		$P7_5$		TA_{2IN} / $\overline{W}$				
23		$P7_4$		TA_{2OUT} / W				
24		$P7_3$		TA_{1IN} / $\overline{V}$		$\overline{CTS_2}$ / $\overline{RTS_2}$ / TxD_1		
25		$P7_2$		TA_{1OUT} / V		CLK_2 / RxD_1		
26		$P7_1$		TA_{0IN}		RxD_2 / SCL_2 / CLK_1		
27		$P7_0$		TA_{0OUT}		TxD_2 / SDA_2 / $\overline{RTS_1}$ / $\overline{CTS_1}$ / $\overline{CTS_0}$ / $CLKS_1$		
28		$P6_7$				TxD_1		
29		$P6_6$				RxD_1		
30		$P6_5$				CLK_1		
31		$P6_4$				$\overline{RTS_1}$ / $\overline{CTS_1}$/ $\overline{CTS_0}$ / $CLKS_1$		
32		$P3_7$						
33		$P3_6$						
34		$P3_5$						
35		$P3_4$						
36		$P3_3$						
37		$P3_2$				S_{OUT3}		
38		$P3_1$				S_{IN3}		
39		$P3_0$				CLK_3		
40		$P6_3$				TxD_0		

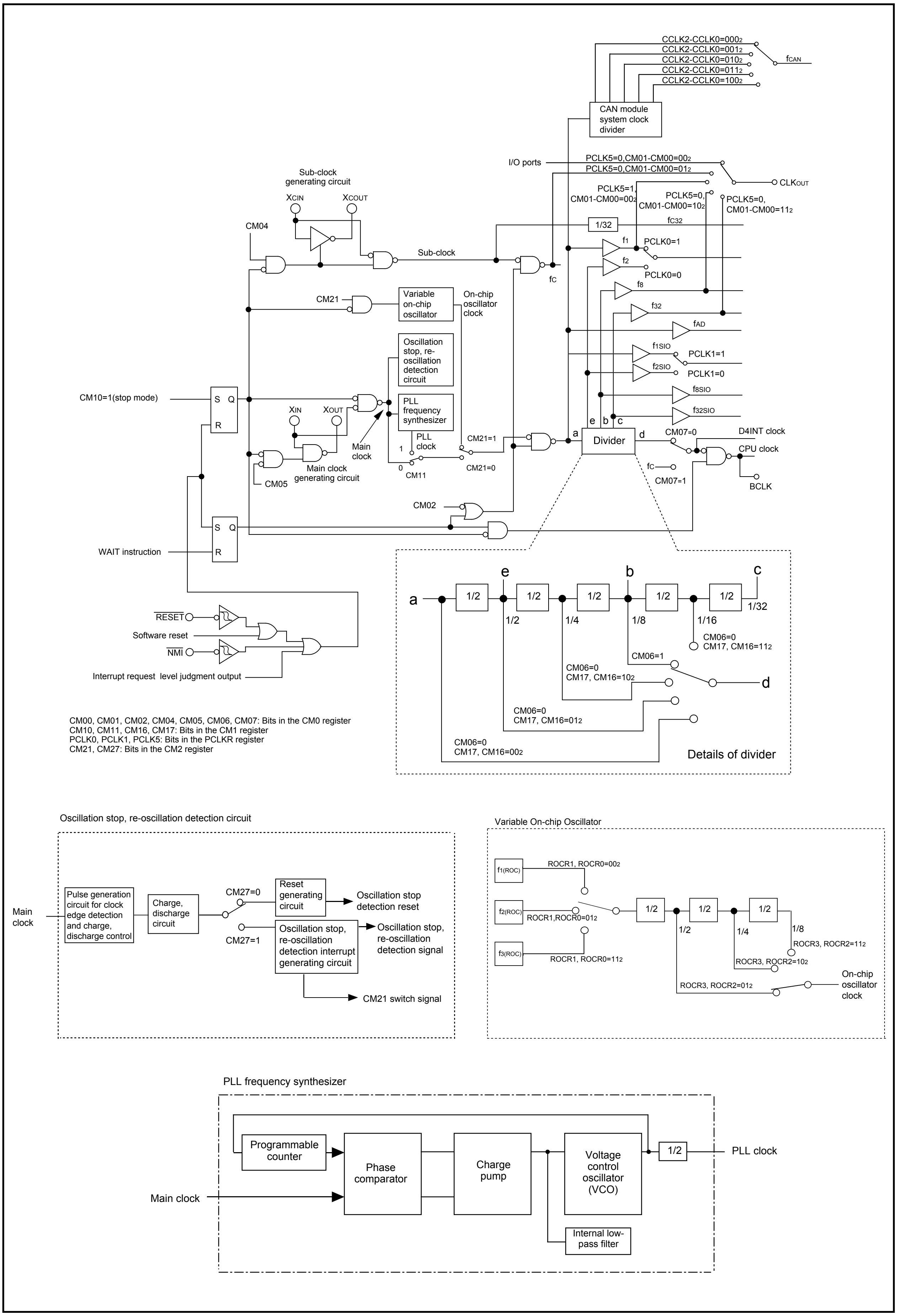
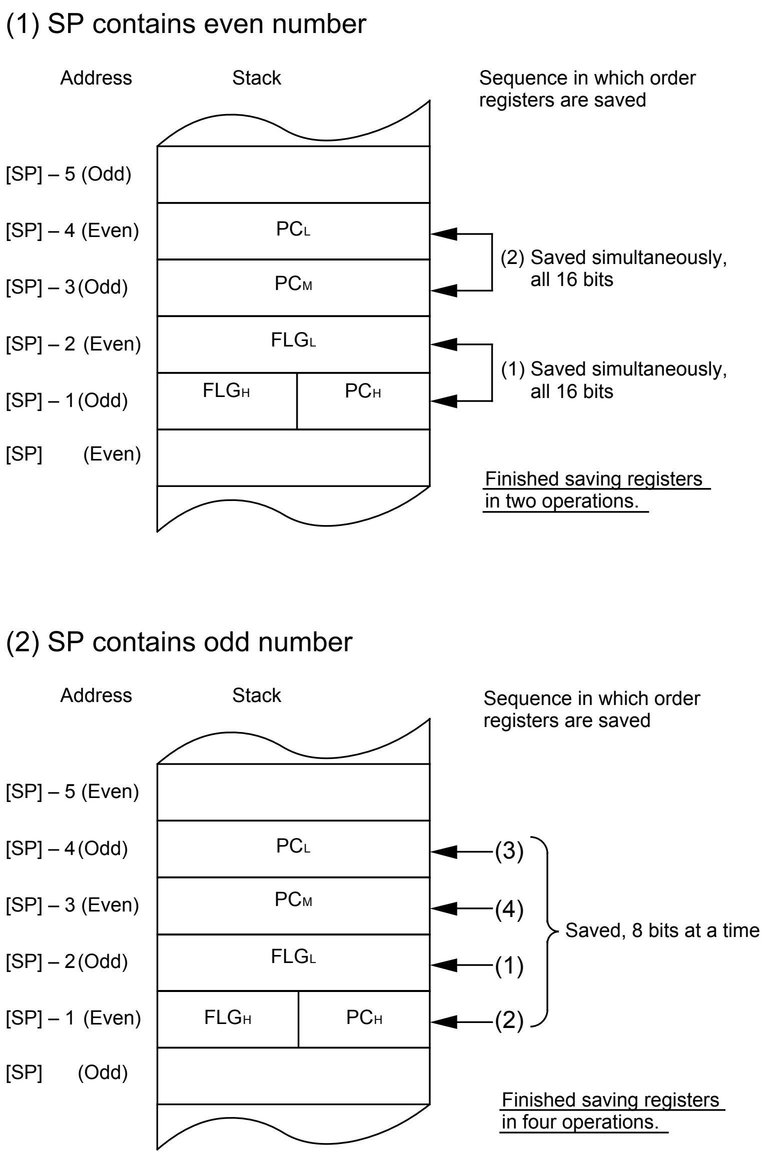


Figure 7.1 Clock Generation Circuit

The operation of saving registers carried out in the interrupt sequence is dependent on whether the SP(1), at the time of acceptance of an interrupt request, is even or odd. If the stack pointer (1) is even, the FLG register and the PC are saved, 16 bits at a time. If odd, they are saved in two steps, 8 bits at a time. **Figure 9.8** shows the operation of the saving registers.

NOTE:
1. When any INT instruction in software numbers 32 to 63 has been executed, this is the SP indicated by the U flag. Otherwise, it is the ISP.



NOTE:
1. [SP] denotes the initial value of the SP when interrupt request is acknowledged. After registers are saved, the SP content is [SP] minus 4.

Figure 9.8 Operation of Saving Register

9.4.4 Returning from an Interrupt Routine

The FLG register and PC in the state in which they were immediately before entering the interrupt sequence are restored from the stack by executing the REIT instruction at the end of the interrupt routine. Thereafter the CPU returns to the program which was being executed before accepting the interrupt request.

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

9.5 Interrupt Priority

If two or more interrupt requests are generated while executing one instruction, the interrupt request that has the highest priority is accepted.

For maskable interrupts (peripheral functions), any desired priority level can be selected using bits ILVL2 to ILVL0. However, if two or more maskable interrupts have the same priority level, their interrupt priority is resolved by hardware, with the highest priority interrupt accepted.

The watchdog timer and other special interrupts have their priority levels set in hardware. **Figure 9.9** 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.

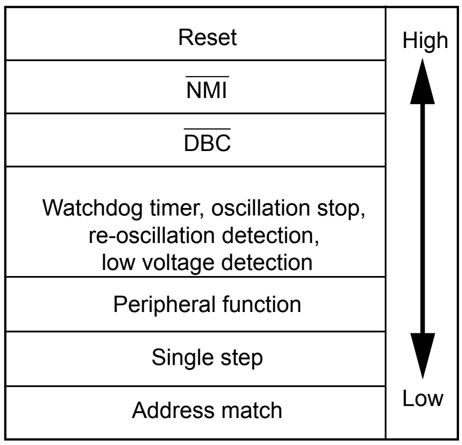


Figure 9.9 Hardware Interrupt Priority

9.5.1 Interrupt Priority Resolution Circuit

The interrupt priority resolution circuit is used to select the interrupt with the highest priority among those requested.

Figure 9.10 shows the circuit that judges the interrupt priority level.

12.1 Timer A

Figure 12.3 shows a block diagram of the timer A. **Figures 12.4** to **12.6** show registers related to the timer A. The timer A supports the following four modes. Except in event counter mode, timers A0 to A4 all have the same function. Use bits TMOD1 to TMOD0 in the TAiMR register (i = 0 to 4) to select the desired mode.

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external device or overflows and underflows of other timers.
- One-shot timer mode: The timer outputs a pulse only once before it reaches the minimum count 0000_{16}.
- Pulse width modulation (PWM) mode: The timer outputs pulses in a given width successively.

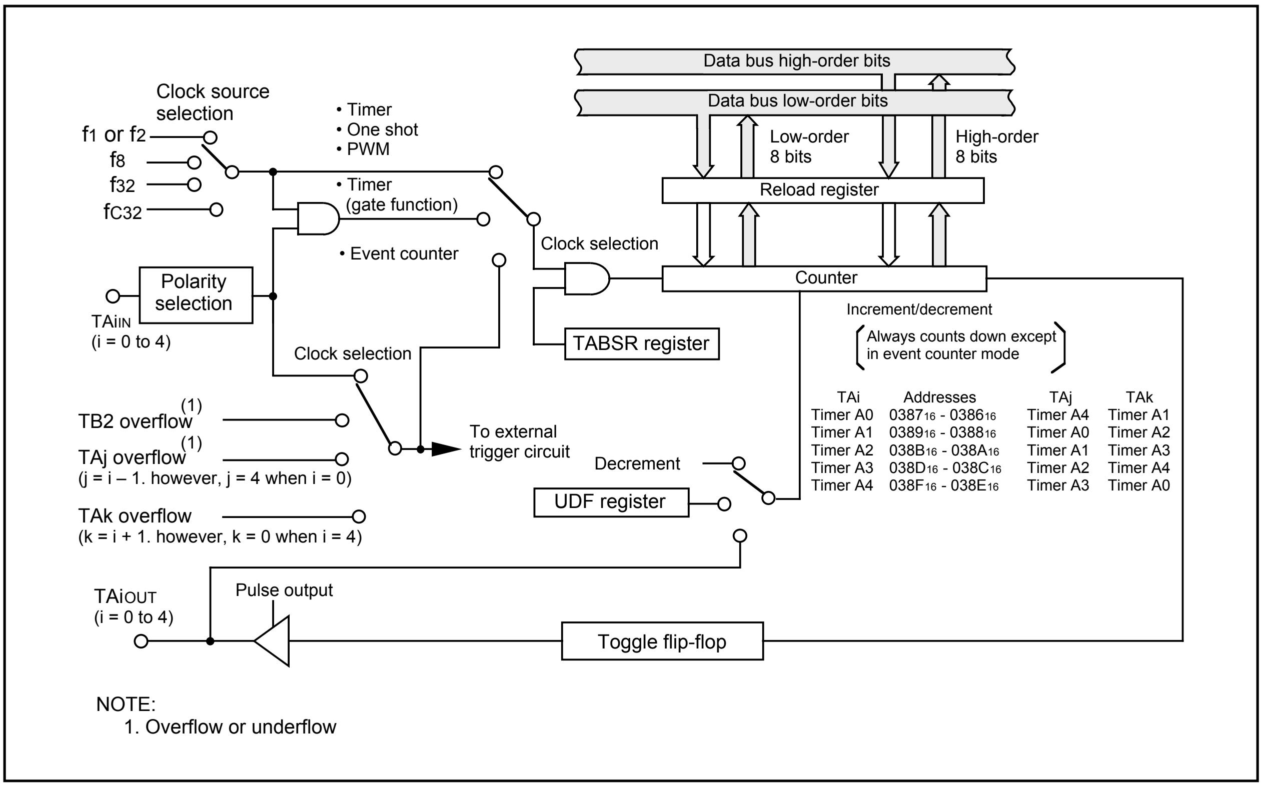


Figure 12.3 Timer A Block Diagram

Timer Ai Mode Register (i=0 to 4)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
TA0MR to TA4MR	0396_{16} to $039A_{16}$	00_{16}

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode 1 1 : Pulse width modulation (PWM) mode	RW
TMOD1			RW
MR0	————	Function varies with each operation mode	RW
MR1			RW
MR2			RW
MR3			RW
TCK0	Count source select bit	Function varies with each operation mode	RW
TCK1			RW

Figure 12.4 TA0MR to TA4MR Registers

12.2.1 Timer Mode

In timer mode, the timer counts a count source generated internally (see **Table 12.6**). **Figure 12.18** shows TBiMR register in timer mode.

Table 12.6 Specifications in Timer Mode

Item	Specification
Count source	f_1, f_2, f_8, f_{32}, f_{C32}
Count operation	• Decrement • When the timer underflows, it reloads the reload register contents and continues counting
Divide ratio	1/(n+1) n: set value of TBi register (i= 0 to 2) 0000_{16} to $FFFF_{16}$
Count start condition	Set TBiS bit(1) to 1 (start counting)
Count stop condition	Set TBiS bit to 0 (stop counting)
Interrupt request generation timing	Timer underflow
TBiIN pin function	I/O port
Read from timer	Count value can be read by reading TBi register
Write to timer	• When not counting and until the 1st count source is input after counting start Value written to TBi register is written to both reload register and counter • When counting (after 1st count source input) Value written to TBi register is written to only reload register (Transferred to counter when reloaded next)

NOTE:
1. Bits TB0S to TB2S are assigned to the bit 7 to bit 5 in the TABSR register.

Timer Bi Mode Register (i= 0 to 2)

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

Symbol	Address	After Reset
TB0MR to TB2MR	$039B_{16}$ to $039D_{16}$	$00XX0000_2$

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0: Timer mode or A/D trigger mode	RW
TMOD1			RW
MR0	No effect in timer mode Can be set to 0 or 1		RW
MR1			RW
MR2	TB0MR register Set to 0 in timer mode		RW
	TB1MR, TB2MR registers Nothing is assigned. If necessary, set to 0. When read, its content is undefined		—
MR3	When write in timer mode, set to 0. When read in timer mode, its content is undefined		RO
TCK0	Count source select bit	b7 b6 0 0: f_1 or f_2 0 1: f_8 1 0: f_{32} 1 1: f_{C32}	RW
TCK1			RW

Figure 12.18 TBiMR Register in Timer Mode

Figures 13.2 to **13.10** show registers associated with the IC/OC base timer, the time measurement function, and the waveform generating function.

Base Timer Register[1]

b15 (b7) b8 (b0) b7 b0

Symbol	Address	After Reset
G1BT	0321_{16} - 0320_{16}	Undefined

Function	Setting Range	RW
• When the base timer is operating: When read, the value of base timer plus 1 can be read. When write, the counter starts counting from the value written. When the base timer is reset, this register is set to 0000_{16}. [2] • When the base timer is reset: This register is set to 0000_{16} but a value read is undefined. No value is written. [2]	0000_{16} to $FFFF_{16}$	RW

NOTES:
1. The G1BT register reflects the value of the base timer, synchronizing with the count source f_{BT1} cycles.
2. This base timer stops only when bits BCK1 to BCK0 in the G1BCR0 register are set to 00_2 (count source clock stop). The base timer operates when bits BCK1 to BCK0 are set to other than 00_2. When the BTS bit in the G1BCR1 register is set to 0, the base timer is reset continuously, and remaining set to 0000_{16}. When the BTS bit is set to 1, this state is cleared and the timer starts counting.

Base Timer Control Register 0

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	0			

Symbol	Address	After Reset
G1BCR0	0322_{16}	00_{16}

Bit Symbol	Bit Name	Function	RW
BCK0	Count source select bit	b1b0 0 0: Clock stop 0 1: Do not set to this value 1 0: Two-phase input [1] 1 1: f_1 or f_2 [2]	RW
BCK1			RW
RST4	Base timer reset cause select bit 4	0: Do not reset Base timer by matching G1BTRR 1: Reset Base timer by matching G1BTRR[3]	RW
— (b5-b3)	Reserved bit	Set to 0	RW
CH7INSEL	Channel 7 input select bit	0: $P2_7$/OUTC1_7/INPC1_7 pin 1: $P1_7$/$\overline{INT5}$/INPC1_7/IDU pin	RW
IT	Base timer interrupt select bit	0: Bit 15 in the base timer overflows 1: Bit 14 in the base timer overflows	RW

NOTES:
1. This setting can be used when bits UD1 to UD0 in the G1BCR1 register are set to 10_2 (two-phase signal processing mode). Do not set bits BCK1 and BCK0 to 10_2 in other modes.
2. When the PCLK0 bit in the PCLKR register is set to 0, the count source is f_2 cycles. And when the PCLK0 bit is set to set to 1, the count source is f_1 cycles.
3. When the RST4 bit is set to 1, set the RST1 bit in the G1BCR1 register to 0.

Figure 13.2 G1BT and G1BCR0 Registers

Interrupt Enable Register 0

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
G1IE0	0331_{16}	00_{16}

Bit Symbol	Bit Name	Function	RW
G1IE00	Interrupt enable 0, CH0	0 : IC/OC interrupt 0 request disable 1 : IC/OC interrupt 0 request enable	RW
G1IE01	Interrupt enable 0, CH1		RW
G1IE02	Interrupt enable 0, CH2		RW
G1IE03	Interrupt enable 0, CH3		RW
G1IE04	Interrupt enable 0, CH4		RW
G1IE05	Interrupt enable 0, CH5		RW
G1IE06	Interrupt enable 0, CH6		RW
G1IE07	Interrupt enable 0, CH7		RW

Interrupt Enable Register 1

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	After Reset
G1IE1	0332_{16}	00_{16}

Bit Symbol	Bit Name	Function	RW
G1IE10	Interrupt enable 1, CH0	0 : IC/OC interrupt 1 request disable 1 : IC/OC interrupt 1 request enable	RW
G1IE11	Interrupt enable 1, CH1		RW
G1IE12	Interrupt enable 1, CH2		RW
G1IE13	Interrupt enable 1, CH3		RW
G1IE14	Interrupt enable 1, CH4		RW
G1IE15	Interrupt enable 1, CH5		RW
G1IE16	Interrupt enable 1, CH6		RW
G1IE17	Interrupt enable 1, CH7		RW

Figure 13.10 G1IE0 and G1IE1 Registers

14.1.1.1 Counter Measure for Communication Error Occurs

If a communication error occurs while transmitting or receiving in clock synchronous serial I/O mode, follow the procedures below.

•Resetting the UiRB register (i=0 to 2)

(1) Set the RE bit in the UiC1 register to 0 (reception disabled)
(2) Set bits SMD2 to SMD0 in the UiMR register to 000_2 (Serial I/O disabled)
(3) Set bits SMD2 to SMD0 in the UiMR register to 001_2 (Clock synchronous serial I/O mode)
(4) Set the RE bit in the UiC1 register to 1 (reception enabled)

•Resetting the UiTB register (i=0 to 2)

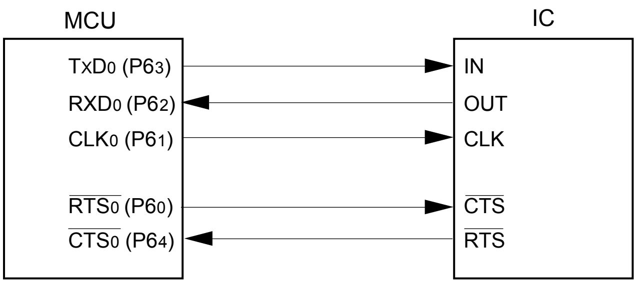
(1) Set bits SMD2 to SMD0 in the UiMR register to 000_2 (Serial I/O disabled)
(2) Set bits SMD2 to SMD0 in the UiMR register to 001_2 (Clock synchronous serial I/O mode)
(3) 1 is written to TE bit in the UiC1 register (reception enabled), regardless to the TE bit.

14.1.1.7 $\overline{CTS}/\overline{RTS}$ separate function (UART0)

This function separates $\overline{CTS_0}/\overline{RTS_0}$, outputs $\overline{RTS_0}$ from the $P6_0$ pin, and accepts as input the $\overline{CTS_0}$ from the $P6_4$ pin or $P7_0$ pin. To use this function, set the register bits as shown below.

- The CRD bit in the U0C0 register is set to 0 (enables UART0 $\overline{CTS}/\overline{RTS}$)
- The CRS bit in the U0C0 register is set to 1 (outputs UART0 $\overline{RTS}$)
- The CRD bit in the U1C0 register is set to 0 (enables UART1 $\overline{CTS}/\overline{RTS}$)
- The CRS bit in the U1C0 register is set to 0 (inputs UART1 $\overline{CTS}$)
- The RCSP bit in the UCON register is set to 1 (inputs $\overline{CTS_0}$ from the $P6_4$ pin or $P7_0$ pin)
- The CLKMD1 bit in the UCON register is set to 0 ($CLKS_1$ not used)

Note that when using the $\overline{CTS}/\overline{RTS}$ separate function, UART1 $\overline{CTS}/\overline{RTS}$ separate function cannot be used.



NOTE:
1. This applies to the case where the U1MAP bit in the PACR register is set to 0 ($P6_7$ to $P6_4$).

Figure 14.15 CTS/RTS separate function usage

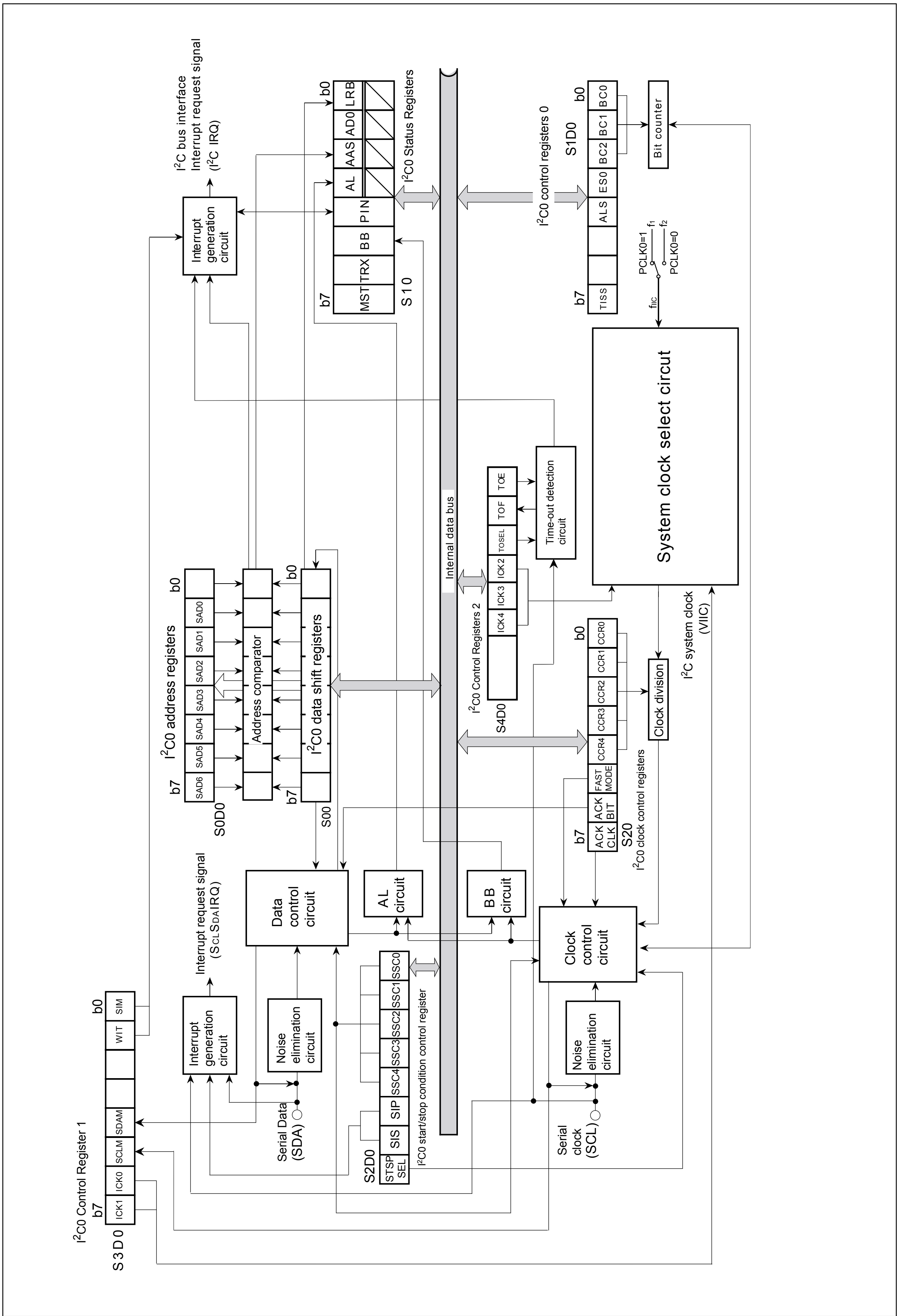


Figure 16.1 Block diagram of multi-master I²C bus interface

16.13 Address Data Communication

This section describes data transmit control when a master transferes data or a slave receives data in 7-bit address format. **Figure 16.20 (1)** shows a master transmit format.

(1) A master transmit device transmits data to a receive device

S	Slave address	R/$\overline{W}$	A	Data	A	Data	A/$\overline{A}$	P
	7 bits	0		1 - 8 bits		1 - 8 bits		

(2) A master receive device receives data from a transmit device

S	Slave address	R/$\overline{W}$	A	Data	A	Data	$\overline{A}$	P
	7 bits	1		1 - 8 bits		1 - 8 bits		

S: START condition P: STOP condition
A: ACK bit R/$\overline{W}$: Read/$\overline{\text{Write}}$ bit

Figure 16.20 Address data communication format

16.13.1 Example of Master Transmit

For example, a master transmits data as shown below when following conditions are met: standard clock mode, SCL clock frequency of 100kHz and ACK clock added.

1) Set s slave address to the 7 high-order bits in the S0D0 register
2) Set 85_{16} to the S20 register, 000_2 to bits ICK4 to ICK2 in the S4D0 register and 00_{16} to the S3D0 registe to generate an ACK clock and set SCL clock frequency t 100 kHz (f_1=8MHz, f_{IIC}=f1)
3) Set 00_{16} to the S10 register to reset transmit/receive
4) Set 08_{16} to the S1D0 register to enable data communication
5) Confirm whether the bus is free by BB flag setting in the S10 register
6) Set $E0_{16}$ to the S10 register to enter START condition standby mode
7) Set the destination address in 7 high-order bits and 0 to a least significant bit in the S00 register to generate START condition. At this time, the first byte consisting of SCL and ACK clock are automatically generated
8) Set a transmit data to the S00 register. At this time, SCL and an ACK clock are automatically generated
9) When transmitting more than 1-byte control data, repeat the above step 8).
10) Set $C0_{16}$ in the S10 register to enter STOP condition standby mode if ACK is not returned from the slave receiver or if the transmit is completed
11) Write dummy data to the S00 regiser to generate STOP condition

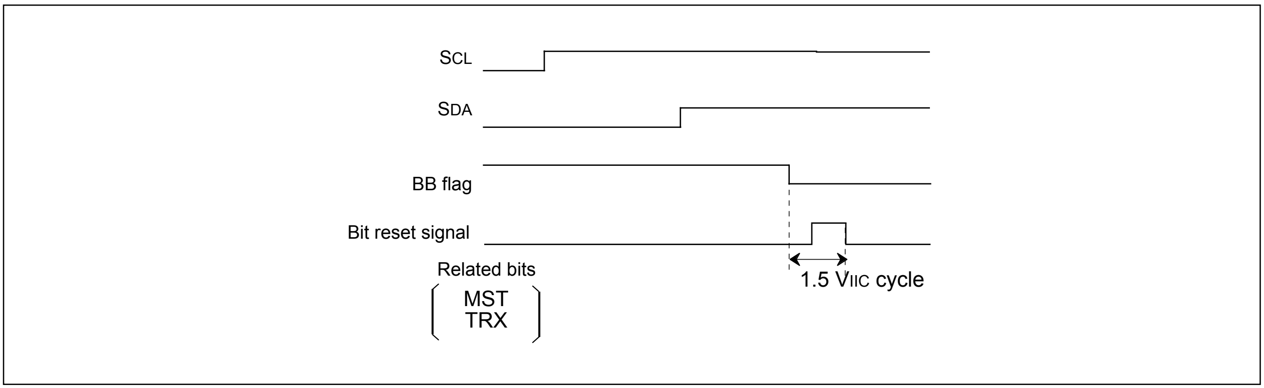


Figure 16.21 The bit reset timing (The STOP condition detection)

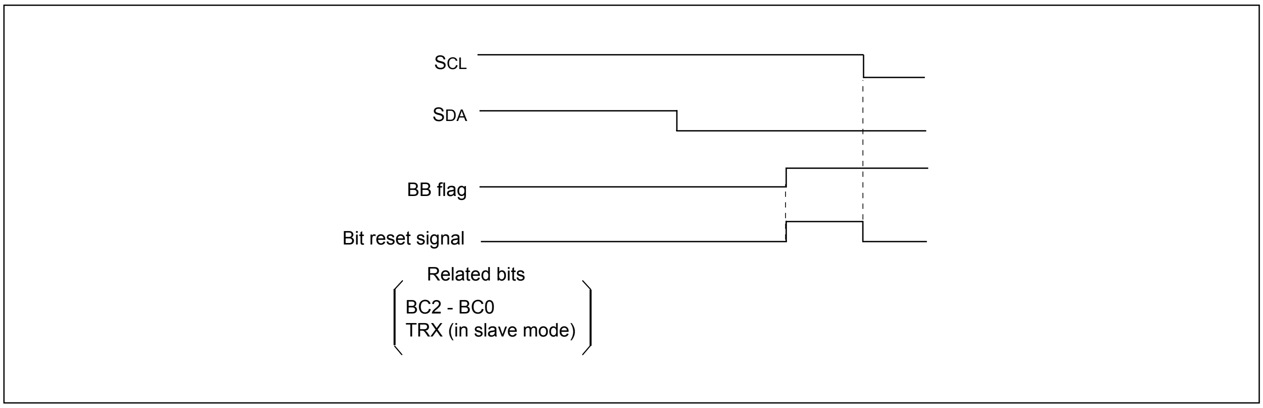


Figure 16.22 The bit reset timing (The START condition detection)

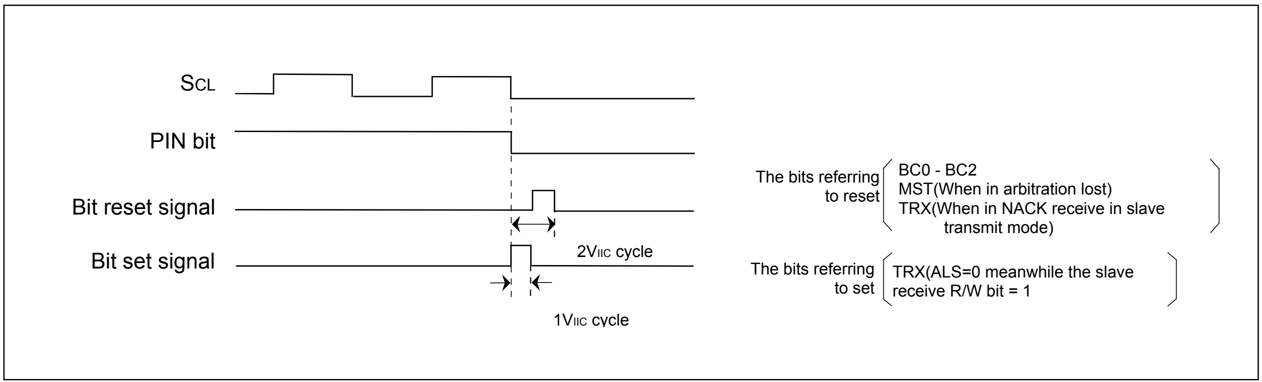


Figure 16.23 Bit set/reset timing (at the completion of data transfer)

19. Programmable I/O Ports

Note
Ports P04 to P07, P10 to P14 , P34 to P37 and P95 to P97 are not available in 64-pin package.

The programmable input/output ports (hereafter referred to simply as “I/O ports”) consist of 71 lines P0, P1, P2, P3, P6, P7, P8, P9, P10 (except P94) for the 80-pin package, or 55 lines P00 to P03, P15 to P17, P2, P30 to P33, P6, P7, P8, P90 to P93, P10 for the 64-pin package. Each port can be set for input or output every line by using a direction register, and can also be chosen to be or not be pulled high in sets of 4 lines.
Figures 19.1 to **19.4** show the I/O ports. **Figure 19.5** shows the I/O pins.
Each pin functions as an I/O port, a peripheral function input/output.
For details on how to set peripheral functions, refer to each functional description in this manual. If any pin is used as a peripheral function input, set the direction bit for that pin to 0 (input mode). Any pin used as an output pin for peripheral functions is directed for output no matter how the corresponding direction bit is set.

19.1 Port Pi Direction Register (PDi Register, i = 0 to 3, 6 to 10)

Figure 19.6 shows the direction registers.
This register selects whether the I/O port is to be used for input or output. The bits in this register correspond one for one to each port.

19.2 Port Pi Register (Pi Register, i = 0 to 3, 6 to 10)

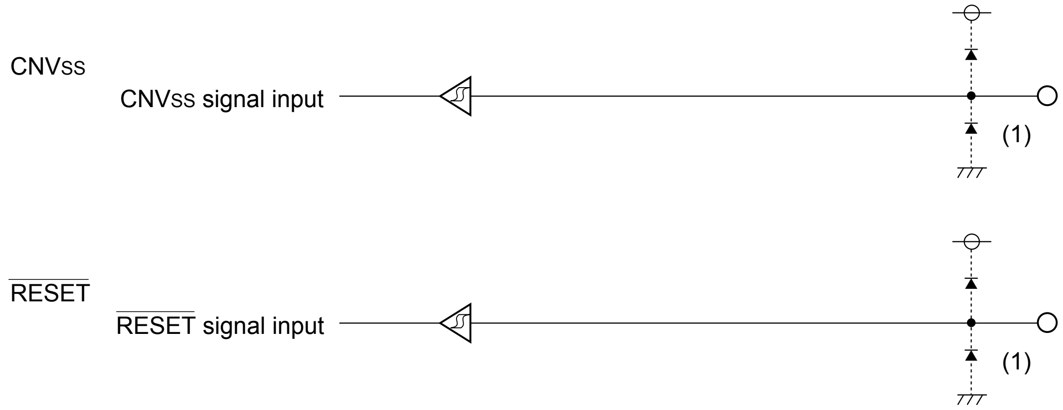
Figure 19.7 shows the Pi registers.
Data input/output to and from external devices are accomplished by reading and writing to the Pi register. The Pi register consists of a port latch to hold the output data and a circuit to read the pin status. For ports set for input mode, the input level of the pin can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register.
For ports set for output mode, the port latch can be read by reading the corresponding Pi register, and data can be written to the port latch by writing to the Pi register. The data written to the port latch is output from the pin. The bits in the Pi register correspond one for one to each port.

19.3 Pull-up Control Register 0 to 2 (PUR0 to PUR2 Registers)

Figure 19.8 shows registers PUR0 to PUR2.
Registers PUR0 to PUR2 select whether the pins, divided into groups of four pins, are pulled up or not. The pins, selected by setting the bits in registers PUR0 to PUR2 to 1 (pull-up), are pulled up when the direction registers are set to 0 (input mode). The pins are pulled up regardless of the pins’ function.

19.4 Port Control Register (PCR Register)

Figure 19.9 shows the port control register.
When the P1 register is read after setting the PCR0 bit in the PCR register to 1, the corresponding port latch can be read no matter how the PD1 register is set.



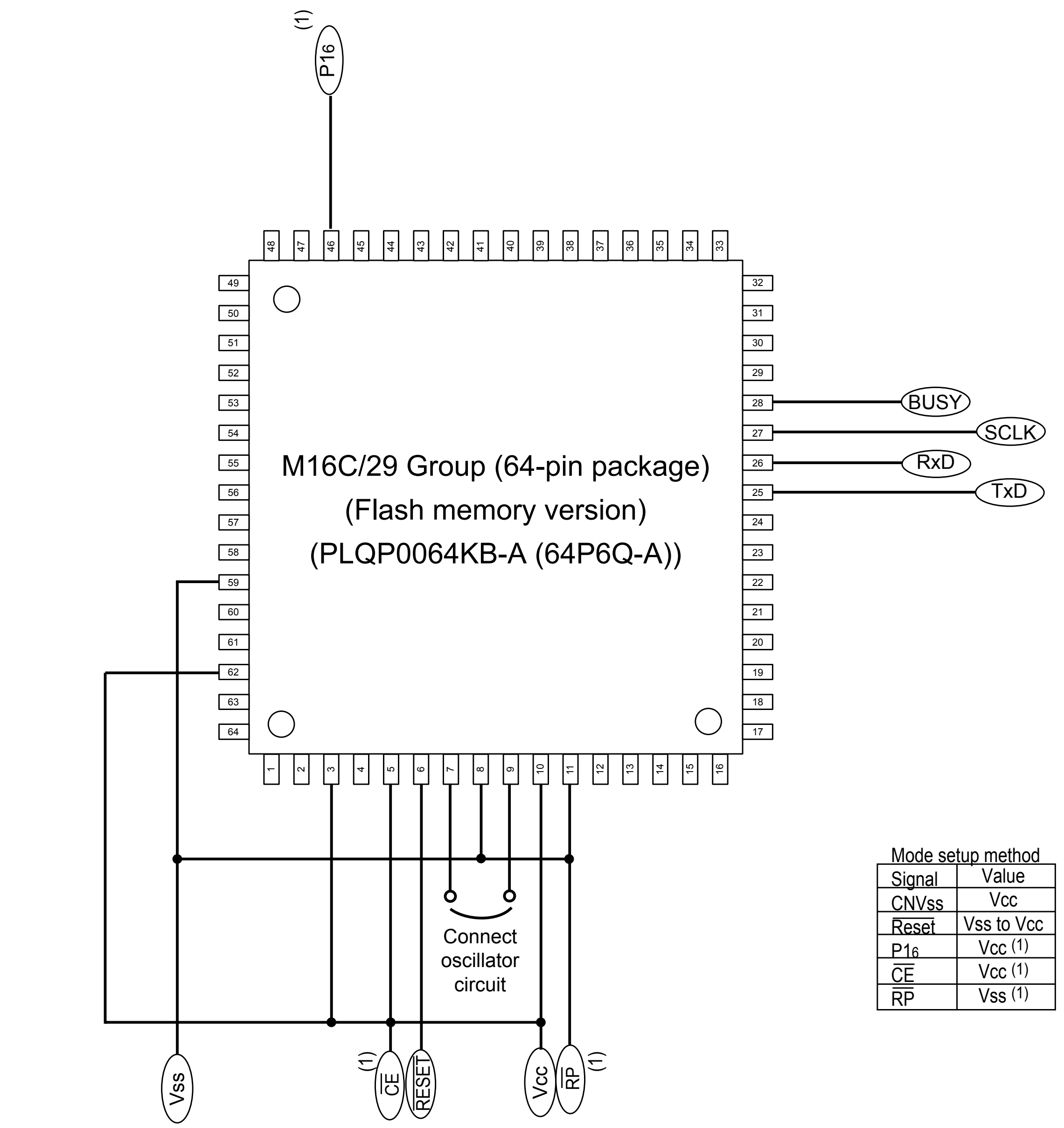
NOTE:
1. symbolizes a parasitic diode.
Make sure the input voltage on each port will not exceed Vcc.

Figure 19.5 I/O Pins

Table 20.8 Pin Descriptions (Flash Memory Standard Serial I/O Mode)

Pin	Name	I/O	Descriptio
VCC,VSS	Power input		Apply the voltage guaranteed for Program and Erase to Vcc pin and 0 V to Vss pin.
CNVSS	CNVSS	I	Connect to Vcc pin.
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. While $\overline{\text{RESET}}$ pin is “L”, wait for td(ROC).
XIN	Clock input	I	Connect a ceramic resonator or crystal oscillator between XIN and XOUT pins. To input an externally generated clock, input it to XIN pin and open XOUT pin.
XOUT	Clock output	O	
AVCC, AVSS	Analog power supply input		Connect AVss to Vss and AVcc to Vcc, respectively.
VREF	Reference voltage input	I	Enter the reference voltage for AD conversion.
P00 to P07	Input port P0	I	Input “H” or “L” signal or leave open.
P10 to P15, P17	Input port P1	I	Input “H” or “L” signal or leave open.
P16	Input port P1	I	Connect this pin to Vcc while $\overline{\text{RESET}}$ pin is “L”. (2)
P20 to P27	Input port P2	I	Input "H" or “L” level signal or leave open.
P30 to P37	Input port P3	I	Input "H" or “L” level signal or leave open.
P60 to P63	Input port P6	I	Input "H" or “L” level signal or leave open.
P64	BUSY output	O	Standard serial I/O mode 1: BUSY signal output pin Standard serial I/O mode 2: Monitor signal output pin for boot program operation check
P65	SCLK input	I	Standard serial I/O mode 1: Serial clock input pin Standard serial I/O mode 2: Input “L”.
P66	RxD input	I	Serial data input pin
P67	TxD output	O	Serial data output pin (1)
P70 to P77	Input port P7	I	Input “H” or “L” signal or leave open.
P80 to P84, P87	Input port P8	I	Input “H” or “L” signal or leave open.
P85	$\overline{\text{RP}}$ input	I	Connect this pin to Vss while $\overline{\text{RESET}}$ pin is “L”. (2)
P86	$\overline{\text{CE}}$ input	I	Connect this pin to Vcc while $\overline{\text{RESET}}$ pin is “L”. (2)
P90 to P92, P95 to P97	Input port P9	I	Input “H” or “L” signal or leave open.
P93	Input port P93 — Normal-ver.	I/O	“H” signal is output for specific time. Input “H” signal or leave open.
	Input port P93 — T-ver./V-ver.	I	Input “H” or “L” signal or leave open.
P100 to P107	Input port P10	I	Input “H” or “L” signal or leave open.

NOTES:
1. When using standard serial I/O mode 1, to input “H” to the TxD pin is necessary while the $\overline{\text{RESET}}$ pin is held “L”. Therefore, connect this pin to VCC via a resistor. Adjust the pull-up resistor value on a system not to affect a data transfer after reset, because this pin changes to a data-output pin
2. Set the following, either or both.
   -Connect the $\overline{\text{CE}}$ pin to VCC.
   -Connect the $\overline{\text{RP}}$ pin to VSS and P16 pin to VCC.



NOTE:

1. Set the following, either or both, in serial I/O Mode, while the $\overline{\text{RESET}}$ pin is applied a low-level ("L") signal.
   - Connect the $\overline{\text{CE}}$ pin to Vcc.
   - Connect the $\overline{\text{RP}}$ pin to Vss and the P16 pin to Vcc.

Figure 20.15 Pin Connections for Serial I/O Mode (1)

21.2 T version

Table 21.40 Absolute Maximum Ratings

<table>
<tr><th>Symbol</th><th colspan="3">Parameter</th><th>Condition</th><th>Value</th><th>Unit</th></tr>
<tr><td>V_{CC}</td><td colspan="3">Supply Voltage</td><td>$V_{CC}=AV_{CC}$</td><td>-0.3 to 6.5</td><td>V</td></tr>
<tr><td>AV_{CC}</td><td colspan="3">Analog Supply Voltage</td><td>$V_{CC}=AV_{CC}$</td><td>-0.3 to 6.5</td><td>V</td></tr>
<tr><td>V_I</td><td>Input Voltage</td><td colspan="2">$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$,
$P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$,
$P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$,
$P10_0$ to $P10_7$,
X_{IN}, V_{REF}, $\overline{RESET}$, CNV_{SS}</td><td></td><td>-0.3 to V_{CC}+0.3</td><td>V</td></tr>
<tr><td>V_O</td><td>Output Voltage</td><td colspan="2">$P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$,
$P3_0$ to $P3_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$,
$P8_0$ to $P8_7$, $P9_0$ to $P9_3$, $P9_5$ to $P9_7$,
$P10_0$ to $P10_7$,
X_{OUT}</td><td></td><td>-0.3 to V_{CC}+0.3</td><td>V</td></tr>
<tr><td>Pd</td><td colspan="3">Power Dissipation</td><td>$-40 \leq Topr \leq 85°$ C</td><td>300</td><td>mW</td></tr>
<tr><td rowspan="3">Topr</td><td rowspan="3">Operating Ambient Temperature</td><td colspan="2">during CPU operation</td><td></td><td>-40 to 85</td><td>° C</td></tr>
<tr><td rowspan="2">during flash memory program and erase operation</td><td>Program Space (Block 0 to Block 5)</td><td></td><td>0 to 60</td><td>° C</td></tr>
<tr><td>Data Space (Block A, Block B)</td><td></td><td>-40 to 85</td><td>° C</td></tr>
<tr><td>Tstg</td><td colspan="3">Storage Temperature</td><td></td><td>-65 to 150</td><td>° C</td></tr>
</table>

22.12 Programmable I/O Ports

1. If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register is set to 1 (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the $P7_2$ to $P7_5$, $P8_0$ and $P8_1$ pins go to a high-impedance state.

2. The input threshold voltage of pins differs between programmable input/output ports and peripheral functions.
Therefore, if any pin is shared by a programmable input/output port and a peripheral function and the input level at this pin is outside the range of recommended operating conditions V_{IH} and V_{IL} (neither “high” nor “low”), the input level may be determined differently depending on which side—the programmable input/output port or the peripheral function—is currently selected.

3. When the SM32 bit in the S3C register is set to 1, the $P3_2$ pin goes to high-impedance state. When the SM42 bit in the S4C register is set to 1, the $P9_6$ pin goes to high-imepdance state.

4. When the INV03 bit in the INVC0 register is 1(three-phase motor control timer output enabled), an "L" input on the $P8_5/\overline{NMI}/\overline{SD}$ pin, has the following effect.

   - When the TB2SC register IVPCR1 bit is set to 1 (three-phase output forcible cutoff by input on $\overline{SD}$ pin enabled), the U/ $\overline{U}$/ V/ $\overline{V}$/ W/ $\overline{W}$ pins go to a high-impedance state.
   - When the TB2SC register IVPCR1 bit is set to 0 (three-phase output forcible cutoff by input on $\overline{SD}$ pin disabled), the U/ $\overline{U}$/ V/ $\overline{V}$/ W/ $\overline{W}$ pins go to a normal port.

   Therefore, the $P8_5$ pin can not be used as programmable I/O port when the INV03 bit is set to 1.
   When the $\overline{SD}$ function isn't used, set to 0 (Input) in $PD8_5$ and pullup to H in the $P8_5/\overline{NMI}/\overline{SD}$ pin from outside.

Rev.	Date	Description: Page	Description: Summary
			note 1 is added
		160	• **Figure 13.21 Prescaler Function and Gate Function** Note 1 modified
		166	• **Table 13.10 SR Waveform Output Mode Specifications** Specification modified
		167	• **Figure 13.24 Set/Reset Waveform Output Mode** Description for (1) Free-running operation modified, register names modified
		168	• **Table 13.11 Pin Setting for Time Measurement and Waveform Generating Functions** Description of port direction modified
			Serial I/O
		170	• Note is modified
		171	• **Figure 14.1 Block Diagram of UARTi (i = 0 to 2)** PLL clock is added to the upper portion of diagram
		174	• **Figure 14.4 U0TB to U2TB, U0RB to U2RB, U0BRG to U2BRG Registers** Note 2 is modified, note 3 is newly added
		175	• **Figure 14.5 U0MR Register, U1MR Register** Bit map is modified
		176	• **Figure 14.6 U0C0 Register** Note 3 modified, Note 4 to 7 are added
			• **Figure 14.6 U2C0 Register** Note 2 is added
		177	• **Figure 14.7 PACR Register** added
		180	• **Table 14.1 Clock Synchronous Serial I/O Mode Specifications** Select function modified, note 2 modified
		182	• **Table 14.3 Pin Functions** Note 1 added
			• **Table 14.4 $P6_4$ Pin Functions** Note 1 added
		183	• **Figure 14.10 Typical transmit/receive timings in clock synchronous serial I/O mode** Example of receive timing: figure modified
		184	• **14.1.1.1 Counter Measre for Communication Error Occurs** newly added
		185	• **14.1.1.2 CLK Polarity Select Function** Newly added
		186	• **Figure 14.14 Transfer Clock Output From Multiple Pins** Note 2 added
		187	• **14.1.1.7 $\overline{CTS}/\overline{RTS}$ separate function (UART0)** modified
			• **Figure 14.15 CTS/RTS Separate Function Usage** Note 1 added
		188	• **Table 14.5 UART Mode Specifications** Select function modified, note 1 modified
		190	• **Table 14.7 I/O Pin Functions in UART Mode** Note 1 added
			• **Table 14.8 $P6_4$ Pin Functions in UART Mode** Note 2 added
		192	• **Figure 14.17 Receive Operation** $\overline{RTSi}$ line is modified
			• **14.1.2.1 Bit Rates** newly added
			• **Table 14.9 Example of Bit Rates and Settings** newly added
		193	• **14.1.2.2 Counter Measure for Communication Error** newly added
		195	• **14.1.2.6 CTS/RTS Separate Function (UART0)** $P7_0$ pin is added
			• **Figure 14.21 CTS/RTS Separate Function** Note 1 added
		196	• **Table 14.10 I^2C mode Specifications** Note 2 modified