NEC

MOS INTEGRATED CIRCUIT

μPD17717, 17718, 17719

4-BIT SINGLE-CHIP MICROCONTROLLERS WITH DEDICATED HARDWARE FOR DIGITAL TUNING SYSTEM

The μPD17717, 17718, and 17719 are 4-bit single-chip CMOS microcontrollers containing hardware for digital tuning systems.

Provided with a wealth of hardware, these microcontrollers are available in many variations of ROM and RAM capacities to support various applications.

Therefore, a high-performance, multi-function digital tuning system can be configured with only one chip.

In addition, a one-time PROM model, μPD17P719, which can be written only once and therefore is ideal for program evaluation and small-scale production of a μPD17717, 17718, or 17719 system, is also available.

FEATURES

- Program memory (ROM)
  μPD17717 : 24K bytes (12288 × 16 bits)
  μPD17718, 17719: 32K bytes (16384 × 16 bits)
- General-purpose data memory (RAM)
  μPD17717, 17718: 1120 × 4 bits
  μPD17719 : 1776 × 4 bits
- Instruction execution time
  1.78 μs (with f_X = 4.5-MHz crystal oscillator)
- PLL frequency synthesizer
  Dual modulus prescaler (130 MHz MAX.), programmable divider, phase comparator, charge pump
- Abundant peripheral hardware units
  General-purpose I/O ports, serial interfaces, A/D converter, D/A converter (PWM output), BEEP output, frequency counter
- Many interrupts
  External: 6 sources
  Internal : 6 sources
- Power-ON reset, CE reset, and power failure detection circuit
- Supply voltage: V_{DD} = 5 V ± 10 %

ORDERING INFORMATION

Part Number	Package
μPD17717GC-×××-3B9	80-pin plastic QFP (14 × 14 mm, 0.65 mm pitch)
μPD17718GC-×××-3B9	80-pin plastic QFP (14 × 14 mm, 0.65 mm pitch)
μPD17719GC-×××-3B9	80-pin plastic QFP (14 × 14 mm, 0.65 mm pitch)

Remark ××× indicates a ROM code suffix.

Unless otherwise specified, the μPD17719 is treated as the representative model in this document.

Document No. U12330EJ1V0DS00 (1st edition)
Date Published May 1997 N
Printed in Japan

FUNCTIONAL OUTLINE

Item \ Part Number		μPD17717	μPD17718	μPD17719
Program memory (ROM)		24K bytes (12288 × 16 bits)	32K bytes (16384 × 16 bits)	
General-purpose data memory (RAM)		1120 × 4 bits		1776 × 4 bits
Instruction execution time		1.78 μs (with f_X = 4.5-MHz crystal oscillator)		
General-purpose port		• I/O port : 46 pins • Input port : 12 pins • Output port: 4 pins		
Stack level		• Address stack: 15 levels • Interrupt stack: 4 levels • DBF stack : 4 levels (can be manipulated via software)		
Interrupt		• External: 6 sources (falling edge of CE pin, INT0 through INT4) • Internal : 6 sources (timers 0 through 3, serial interfaces 2 and 3)		
Timer		5 channels • Basic timer (clock: 10, 20, 50, 100 Hz) : 1 channel • 8-bit timer with gate counter (clock: 1 k, 2 k, 10 k, 100 kHz) : 1 channel • 8-bit timer (clock: 1 k, 2 k, 10 k, 100 kHz) : 2 channels • 8-bit timer multiplexed with PWM (clock: 440 Hz, 4.4 kHz) : 1 channel		
A/D converter		8 bits × 6 channels (hardware mode and software mode selectable)		
D/A converter (PWM)		3 channels (8-bit or 9-bit resolution selectable by software) Output frequency: 4.4 kHz, 440 Hz (with 8-bit PWM selected) 2.2 kHz, 220 Hz (with 9-bit PWM selected)		
Serial interface		2 units (4 channels) • 3-wire serial I/O mode, SBI mode, or 2-wire serial I/O/I^2C[Note] bus mode selectable • 3-wire serial I/O/UART mode selectable		
PLL frequency synthesizer	Division mode	• Direct division mode (VCOL pin (MF mode) : 0.5 to 3 MHz) • Pulse swallow mode (VCOL pin (HF mode) : 10 to 40 MHz) (VCOH pin (VHF mode): 60 to 130 MHz)		
	Reference frequency	13 types selectable (1, 1.25, 2.5, 3, 5, 6.25, 9, 10, 12.5, 18, 20, 25, 50 kHz)		
	Charge pump	Two error-out output pins (EO0, EO1)		
	Phase comparator	Unlock status detectable by program		
Frequency counter		• Intermediate frequency (IF) measurement P1C0/FMIFC pin: in FMIF mode 10 to 11 MHz in AMIF mode 0.4 to 0.5 MHz P1C1/AMIFC pin: in AMIF mode 0.4 to 0.5 MHz • External gate width measurement P2A1/FCG1, P2A0/FCG0 pin		
BEEP output		2 pins Output frequency: 1 kHz, 3 kHz, 4 kHz, 6.7 kHz (BEEP0 pin) 67 Hz, 200 Hz, 3 kHz, 4 kHz (BEEP1 pin)		

Note When the I^2C bus mode is used (including when it is implemented in software without using the peripheral hardware), inform NEC when you place an order for mask ROM.

Item \ Part Number	μPD17717	μPD17718	μPD17719
Reset	• Power-ON reset (on power application) • Reset by $\overline{\text{RESET}}$ pin • Watchdog timer reset Can be set only once on power application: 65536 instruction, 131072 instruction, or no-use selectable • Stack pointer overflow/underflow reset Can be set only once on power application: interrupt stack or address stack selectable • CE reset (CE pin low → high level) CE reset delay timing can be set. • Power failure detection function		
Standby	• Clock stop mode (STOP) • Halt mode (HALT)		
Supply voltage	• PLL operation: V_{DD} = 4.5 to 5.5 V • CPU operation: V_{DD} = 3.5 to 5.5 V		
Package	80-pin plastic QFP (14 × 14 mm, 0.65 mm pitch)		

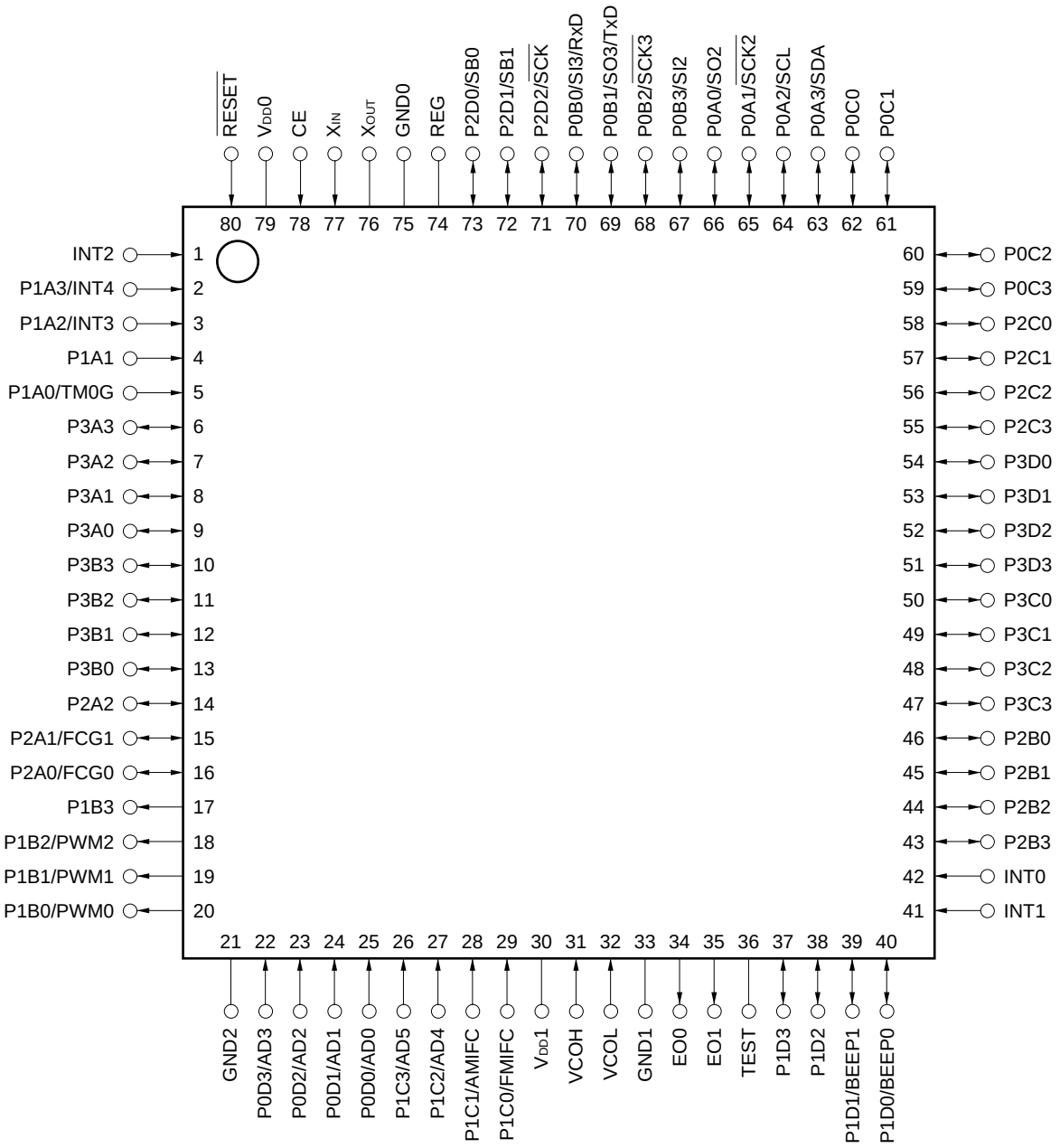
PIN CONFIGURATION (Top View)

80-pin plastic QFP (14 × 14 mm, 0.65 mm pitch)

μPD17717GC-×××-3B9

μPD17718GC-×××-3B9

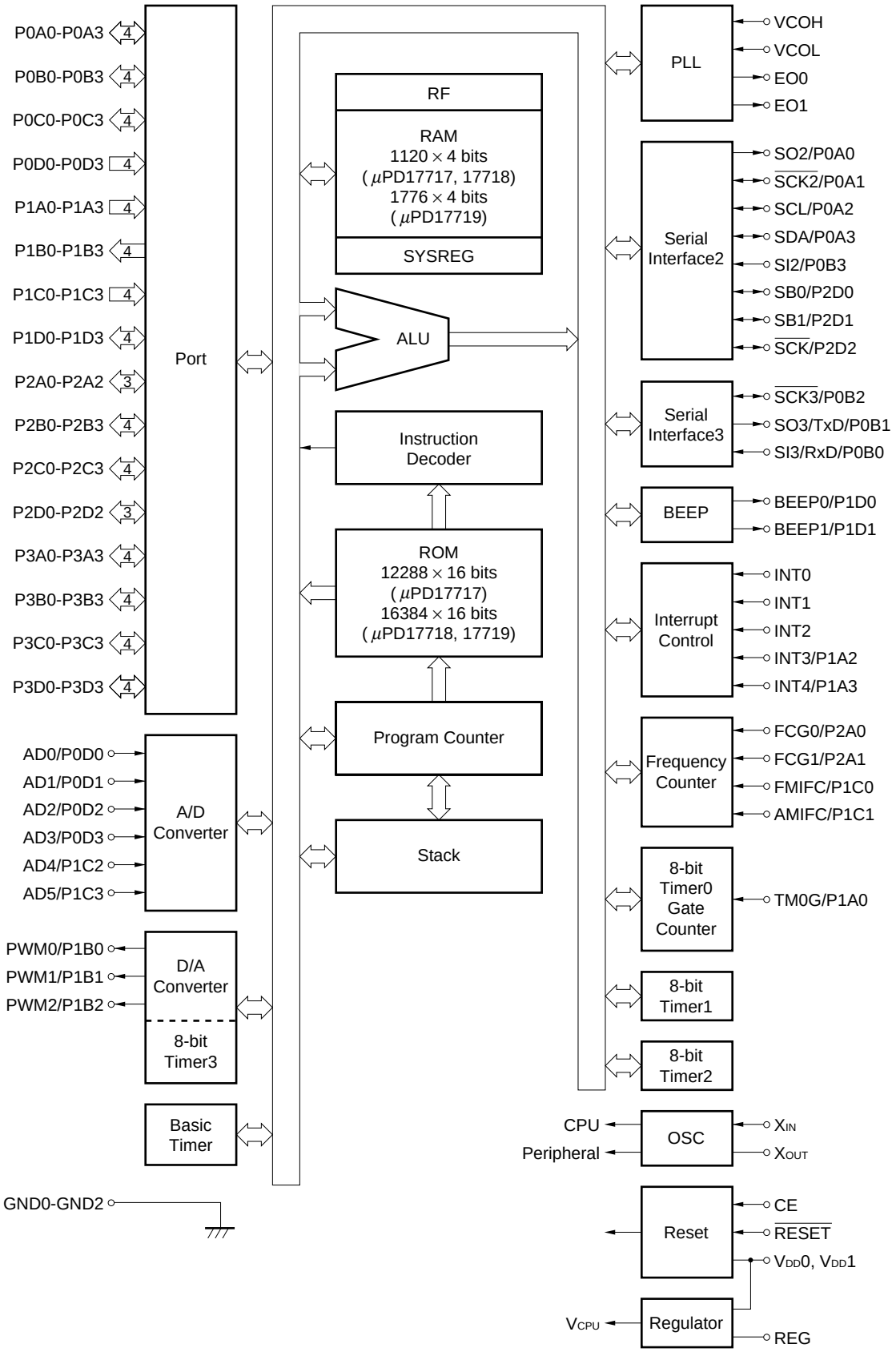
μPD17719GC-×××-3B9



PIN NAME

AD0-AD5	: A/D converter input
AMIFC	: AM frequency counter input
BEEP0, BEEP1	: BEEP output
CE	: Chip enable
EO0, EO1	: Error-out output
FCG0, FGC1	: Frequency counter gate input
FMIFC	: FM frequency counter input
GND0-GND2	: Ground 0 to 2
INT0-INT4	: External interrupt input
PWM0-PWM2	: D/A converter output
P0A0-P0A3	: Port 0A
P0B0-P0B3	: Port 0B
P0C0-P0C3	: Port 0C
P0D0-P0D3	: Port 0D
P1A0-P1A3	: Port 1A
P1B0-P1B3	: Port 1B
P1C0-P1C3	: Port 1C
P1D0-P1D3	: Port 1D
P2A0-P2A2	: Port 2A
P2B0-P2B3	: Port 2B
P2C0-P2C3	: Port 2C
P2D0-P2D2	: Port 2D
P3A0-P3A3	: Port 3A
P3B0-P3B3	: Port 3B
P3C0-P3C3	: Port 3C
P3D0-P3D3	: Port 3D
REG	: CPU regulator
$\overline{\text{RESET}}$	: Reset input
RxD	: UART serial data input
SB0, SB1	: SBI serial data I/O
$\overline{\text{SCK}}$	: SBI serial clock I/O
$\overline{\text{SCK2}}$, $\overline{\text{SCK3}}$	: 3-wire serial clock I/O
SCL	: 2-wire serial clock I/O
SDA	: 2-wire serial data I/O
SI2, SI3	: 3-wire serial data input
SO2, SO3	: 3-wire serial data output
TEST	: Test input
TM0G	: Timer 0 gate input
TxD	: UART serial data output
VCOH	: Local oscillation high input
VCOL	: Local oscillation low input
$V_{DD}0$, $V_{DD}1$	: Power supply
X_{IN}, X_{OUT}	: Main clock oscillation

BLOCK DIAGRAM



Contents

1. PIN FUNCTIONS

1.1 Pin Function List

<table>
<tr><th>Pin No.</th><th>Symbol</th><th>Function</th><th>Output Form</th></tr>
<tr><td>1
41
42</td><td>INT2
INT1
INT0</td><td>Edge-detectable vectored interrupt input pins. Rising or falling edge can be specified.</td><td>–</td></tr>
<tr><td>2
3
4
5</td><td>P1A3/INT4
P1A2/INT3
P1A1
P1A0/TM0G</td><td>Port 1A multiplexed with external interrupt request signal input and event signal input pins.
• P1A3 through P1A0
• 4-bit input port
• INT4, INT3
• Edge-detectable vectored interrupt
• TM0G
• Input for gate of 8-bit timer 0
<table><tr><td colspan="3">At reset</td><td>With clock stopped</td></tr><tr><td>Power-ON reset</td><td>WDT&SP reset</td><td>CE reset</td><td></td></tr><tr><td>Input
(P1A3 through P1A0)</td><td>Input
(P1A3 through P1A0)</td><td>Retained</td><td>Retained</td></tr></table></td><td>–</td></tr>
<tr><td>6
|
9</td><td>P3A3
|
P3A0</td><td>4-bit I/O port.
Can be set in input or output mode in 4-bit units.
<table><tr><td colspan="3">At reset</td><td>With clock stopped</td></tr><tr><td>Power-ON reset</td><td>WDT&SP reset</td><td>CE reset</td><td></td></tr><tr><td>Input</td><td>Input</td><td>Retained</td><td>Retained</td></tr></table></td><td>CMOS
push-pull</td></tr>
<tr><td>10
|
13</td><td>P3B3
|
P3B0</td><td>4-bit I/O port.
Can be set in input or output mode in 4-bit units.
<table><tr><td colspan="3">At reset</td><td>With clock stopped</td></tr><tr><td>Power-ON reset</td><td>WDT&SP reset</td><td>CE reset</td><td></td></tr><tr><td>Input</td><td>Input</td><td>Retained</td><td>Retained</td></tr></table></td><td>CMOS
push-pull</td></tr>
<tr><td>14
15
16</td><td>P2A2
P2A1/FCG1
P2A0/FCG0</td><td>Port 2A multiplexed with external gate counter input pins.
• P2A2 through P2A0
• 3-bit I/O port
• Can be set in input or output mode in 1-bit units.
• FCG1, FCG0
• Input for external gate counter
<table><tr><td colspan="3">At reset</td><td>With clock stopped</td></tr><tr><td>Power-ON reset</td><td>WDT&SP reset</td><td>CE reset</td><td></td></tr><tr><td>Input
(P2A2 through P2A0)</td><td>Input
(P2A2 through P2A0)</td><td>Retained
(P2A2 through P2A0)</td><td>Retained
(P2A2 through P2A0)</td></tr></table></td><td>CMOS
push-pull</td></tr>
</table>

<table>
<tr><th>Pin No.</th><th>Symbol</th><th>Function</th><th>Output Form</th></tr>
<tr><td>17
18
|
20</td><td>P1B3
P1B2/PWM2
|
P1B0/PWM0</td><td>Port 1B multiplexed with D/A converter output pins.
• P1B3 through P1B0
• 4-bit output port
• PWM2 through P2M0
• 8- or 9-bit D/A converter output
<table><tr><td colspan="3">At reset</td><td rowspan="2">With clock stopped</td></tr><tr><td>Power-ON reset</td><td>WDT&SP reset</td><td>CE reset</td></tr><tr><td>Outputs low level
(P1B3 through P1B0)</td><td>Outputs low level
(P1B3 through P1B0)</td><td>Retained</td><td>Retained
(P1B3 through P1B0)</td></tr></table></td><td>N-ch
open-drain
(12 V)</td></tr>
<tr><td>21
33
75</td><td>GND2
GND1
GND0</td><td>Ground</td><td>–</td></tr>
<tr><td>22
|
25</td><td>P0D3/AD3
|
P0D0/AD0</td><td>Port 0D multiplexed with A/D converter input pins
• P0D3 through P0D0
• 4-bit input port
• Can be connected with pull-down resistor in 1-bit units.
• AD3 through AD0
• Analog input of A/D converter with 8-bit resolution
<table><tr><td colspan="3">At reset</td><td rowspan="2">With clock stopped</td></tr><tr><td>Power-ON reset</td><td>WDT&SP reset</td><td>CE reset</td></tr><tr><td>Input with pull-down resistor
(P0D3 through P0D0)</td><td>Input with pull-down resistor
(P0D3 through P0D0)</td><td>Retained</td><td>Retained</td></tr></table></td><td>–</td></tr>
<tr><td>26
27
28
29</td><td>P1C3/AD5
P1C2/AD4
P1C1/AMIFC
P1C0/FMIFC</td><td>Port 1C multiplexed with A/D converter input and IF counter input pins.
• P1C3 through P1C0
• 4-bit input port
• AD5, AD4
• Analog input to A/D converter with 8-bit resolution
• FMIFC, AMIFC
• Input to frequency counter
<table><tr><td colspan="3">At reset</td><td rowspan="2">With clock stopped</td></tr><tr><td>Power-ON reset</td><td>WDT&SP reset</td><td>CE reset</td></tr><tr><td>Input
(P1C3 through P1C0)</td><td>Input
(P1C3 through P1C0)</td><td>• P1C3/AD5, P1C2/AD4 retained
• P1C1/AMIFC, P1C0/FMIFC input (P1C1, P1C0)</td><td>• P1C3/AD5, P1C2/AD4 retained
• P1C1/AMIFC, P1C0/FMIFC input (P1C1, P1C0)</td></tr></table></td><td>–</td></tr>
<tr><td>30
79</td><td>$V_{DD}1$
$V_{DD}0$</td><td>Power supply. Supply the same voltage to these pins.
• With CPU and peripheral function operating : 4.5 to 5.5 V
• With CPU operating : 3.5 to 5.5 V
• With clock stopped : 2.2 to 5.5 V</td><td>–</td></tr>
</table>

<table>
<tr><th>Pin No.</th><th>Symbol</th><th>Function</th><th>Output Form</th></tr>
<tr><td>31
32</td><td>VCOH
VCOL</td><td>PLL local oscillation (VCO) frequency input.
• VCOH
• Active with VHF mode selected by program; otherwise, pulled down.
• VCOL
• Active with HF or MW mode selected by program; otherwise, pulled down.

Because the input of these pins goes into an AC amplifier, cut the DC component of the input signal with a capacitor.</td><td>–</td></tr>
<tr><td rowspan="3">34
35</td><td rowspan="3">EO0
EO1</td><td>Output from charge pump of PLL frequency synthesizer. Outputs the divided frequency of local oscillation and the result of comparison of the phase difference of reference frequency.</td><td rowspan="3">CMOS
3-state</td></tr>
<tr><td>At reset
Power-ON reset | WDT&SP reset | CE reset
High-impedance output | High-impedance output | High-impendance output
With clock stopped: High-impedance output</td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td>36</td><td>TEST</td><td>Test input pin.
Be sure to connect this pin to GND.</td><td>–</td></tr>
<tr><td rowspan="3">37
38
39
40</td><td rowspan="3">P1D3
P1D2
P1D1/BEEP1
P1D0/BEEP0</td><td>Port 1D and BEEP output.
• P1D3 through P1D0
• 4-bit I/O port
• Can be set in input or output mode in 1-bit units.
• BEEP1, BEEP0
• BEEP output</td><td rowspan="3">CMOS
push-pull</td></tr>
<tr><td>At reset
Power-ON reset | WDT&SP reset | CE reset
Input (P1D3 through P1D0) | Input (P1D3 through P1D0) | Retained (P1D3 through P1D0)
With clock stopped: Retained (P1D3 through P1D0)</td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td>43
|
46</td><td>P2B3
|
P2B0</td><td>4-bit I/O Port.
Can be set in input or output mode in 1-bit units.
At reset
Power-ON reset | WDT&SP reset | CE reset
Input | Input | Retained
With clock stopped: Retained</td><td>CMOS
push-pull</td></tr>
<tr><td>47
|
50</td><td>P3C3
|
P3C0</td><td>4-bit I/O Port.
Can be set in input or output mode in 4-bit units.
At reset
Power-ON reset | WDT&SP reset | CE reset
Input | Input | Retained
With clock stopped: Retained</td><td>CMOS
push-pull</td></tr>
<tr><td>51
|
54</td><td>P3D3
|
P3D0</td><td>4-bit I/O Port.
Can be set in input or output mode in 4-bit units.
At reset
Power-ON reset | WDT&SP reset | CE reset
Input | Input | Retained
With clock stopped: Retained</td><td>CMOS
push-pull</td></tr>
</table>

<table>
<tr><th>Pin No.</th><th>Symbol</th><th>Function</th><th>Output Form</th></tr>
<tr><td>55
|
58</td><td>P2C3
|
P2C0</td><td>4-bit I/O Port.
Can be set in input or output mode in 1-bit units.
At reset | With clock stopped
Power-ON reset | WDT&SP reset | CE reset
Input | Input | Retained | Retained</td><td>CMOS push-pull</td></tr>
<tr><td>59
|
62</td><td>P0C3
|
P0C0</td><td>4-bit I/O Port.
Can be set in input or output mode in 1-bit units.
At reset | With clock stopped
Power-ON reset | WDT&SP reset | CE reset
Input | Input | Retained | Retained</td><td>CMOS push-pull</td></tr>
<tr><td>63
64</td><td>P0A3/SDA
P0A2/SCL</td><td rowspan="3">Ports P0A, P0B, and P2D are multiplexed with I/O of serial interface.
• P0A3 through P0A0
• 4-bit I/O port
• Can be set in input or output mode in 1-bit units.
• P0B3 through P0B0
• 4-bit I/O port
• Can be set in input or output mode in 1-bit units.
• P2D2-P2D0
• 3-bit I/O port
• Can be set in input or output mode in 1-bit units.
• SDA, SCL
• Serial data and serial clock I/O of serial interface 2 in 2-wire serial I/O or I²C bus mode
• $\overline{SCK2}$, SO2, SI2
• Serial clock I/O, serial data output, and serial data input of serial interface 2 in 3-wire serial I/O mode
• $\overline{SCK3}$, SO3, SI3
• Serial clock I/O, serial data output, serial data input of serial interface 3 in 3-wire serial I/O mode
• TxD, RxD
• Serial data output and serial data input when UART of serial interface 3 is selected
• $\overline{SCK}$, SB1, SB0
• Serial clock and serial data I/O when SBI of serial interface 2 is selected
At reset | With clock stopped
Power-ON reset | WDT&SP reset | CE reset
Input (P0A3 through P0A0, P0B3 through P0B0, P2D2 through P2D0) | Input (P0A3 through P0A0, P0B3 through P0B0, P2D2 through P2D0) | Retained (P0A3 through P0A0, P0B3 through P0B0, P2D2 through P2D0) | Retained (P0A3 through P0A0, P0B3 through P0B0, P2D2 through P2D0)</td><td>N-ch open-drain</td></tr>
<tr><td>65
66
67
68
69

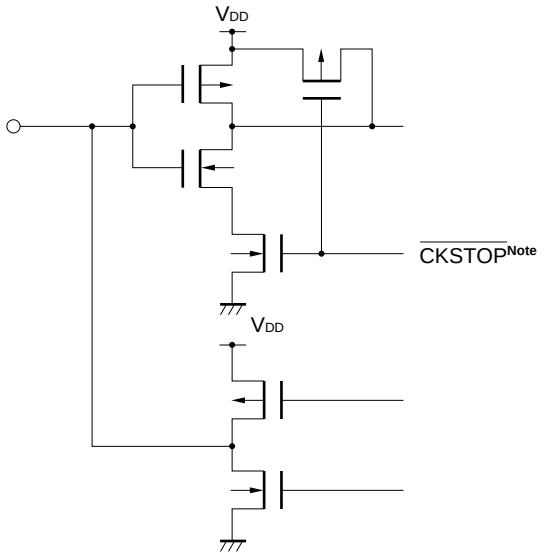
70

71</td><td>P0A1/$\overline{SCK2}$
P0A0/SO2
P0B3/SI2
P0B2/$\overline{SCK3}$
P0B1/SO3/
TxD
P0B0/SI3/
RxD
P2D2/$\overline{SCK}$</td><td>CMOS push-pull</td></tr>
<tr><td>72
73</td><td>P2D1/SB1
P2D0/SB0</td><td>N-ch open-drain</td></tr>
</table>

Pin No.	Symbol	Function	Output Form
74	REG	CPU regulator. Connect this pin to GND via 0.1-μF capacitor.	–
76 77	X_{OUT} X_{IN}	Ground pins of crystal resonator.	–
78	CE	Device operation-selection, CE reset, and interrupt signal input pin. • Device operation-select When CE is high, PLL frequency synthesizer can operate. When CE is low, PLL frequency synthesizer is automatically disabled internally. • CE reset When CE goes high, device is reset at rising edge of internal basic timer setting pulse. This pin also has reset timing delay function. • Interrupt Vectored interrupt occurs at falling edge of this pin.	–
80	$\overline{\text{RESET}}$	Reset input	–

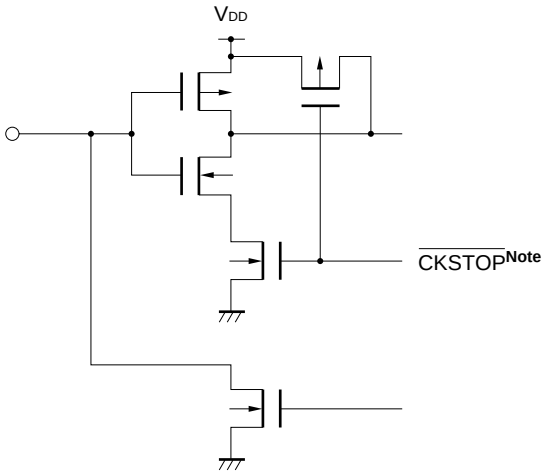
1.2 Equivalent Circuits of Pins

(1) P0A (P0A1/$\overline{SCK2}$, P0A0/SO2)
P0B (P0B3/SI2, P0B2/$\overline{SCK3}$, P0B1/SO3/TxD, P0B0/SI3/RxD)
P0C (P0C3, P0C2, P0C1, P0C0)
P1D (P1D3, P1D2, P1D1/BEEP1, P1D0/BEEP0)
P2A (P2A2, P2A1/FCG1, P2A0/FCG0)
P2B (P2B3, P2B2, P2B1, P2B0)
P2C (P2C3, P2C2, P2C1, P2C0)
P2D (P2D2/$\overline{SCK}$)
P3A (P3A3, P3A2, P3A1, P3A0)
P3B (P3B3, P3B2, P3B1, P3B0)
P3C (P3C3, P3C2, P3C1, P3C0)
P3D (P3D3, P3D2, P3D1, P3D0)
} (I/O)



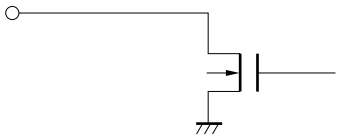
Note This is an internal signal that is output when the clock stop instruction is executed, and its circuit is designed not to increase the current consumption due to noise even if it is floated.

(2) P0A (P0A3/SDA, P0A2/SCL)
P2D (P2D1/SB1, P2D0/SB0) } **(I/O)**

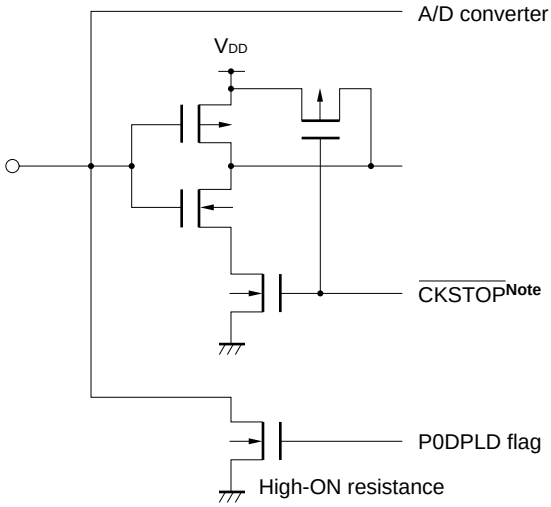


Note This is an internal signal that is output when the clock stop instruction is executed, and its circuit is designed not to increase the current consumption due to noise even if it is floated.

(3) P1B (P1B3, P1B2/PWM2, P1B1/PWM1, P1B0/PWM0) (output)

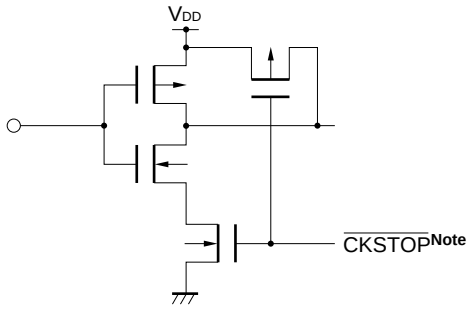


(4) P0D (P0D3/AD3, P0D2/AD2, P0D1/AD1, P0D0/AD0) (input)



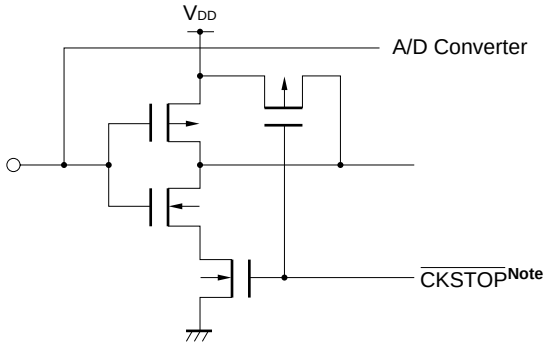
Note This is an internal signal that is output when the clock stop instruction is executed, and its circuit is designed not to increase the current consumption due to noise even if it is floated.

(5) P1A (P1A1) (input)



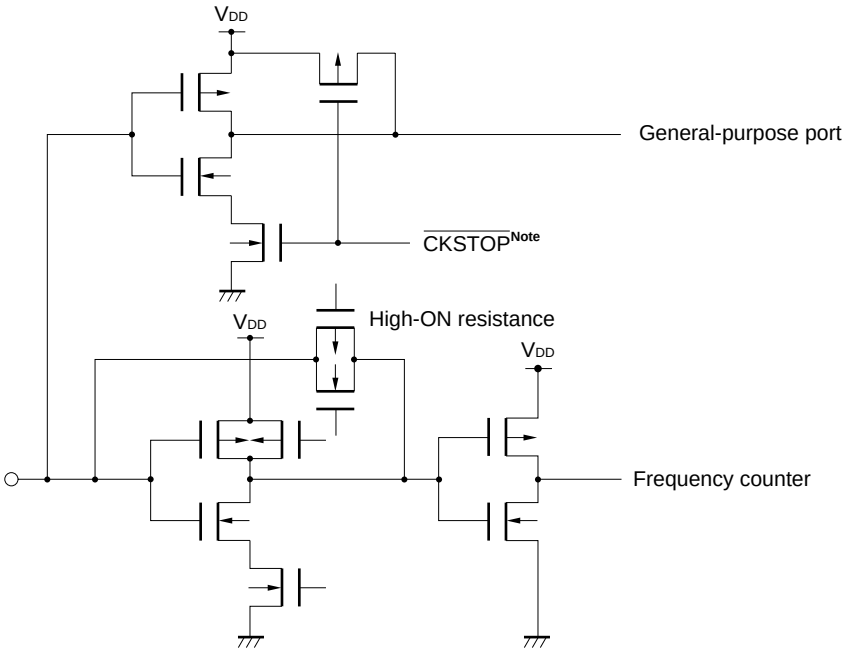
Note This is an internal signal that is output when the clock stop instruction is executed, and its circuit is designed not to increase the current consumption due to noise even if it is floated.

(6) P1C (P1C3/AD5, P1C2/AD4) (input)



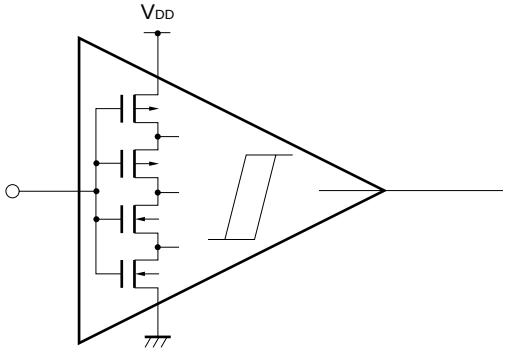
Note This is an internal signal that is output when the clock stop instruction is executed, and its circuit is designed not to increase the current consumption due to noise even if it is floated.

(7) P1C (P1C1/AMIFC, P1C0/FMIFC) (input)

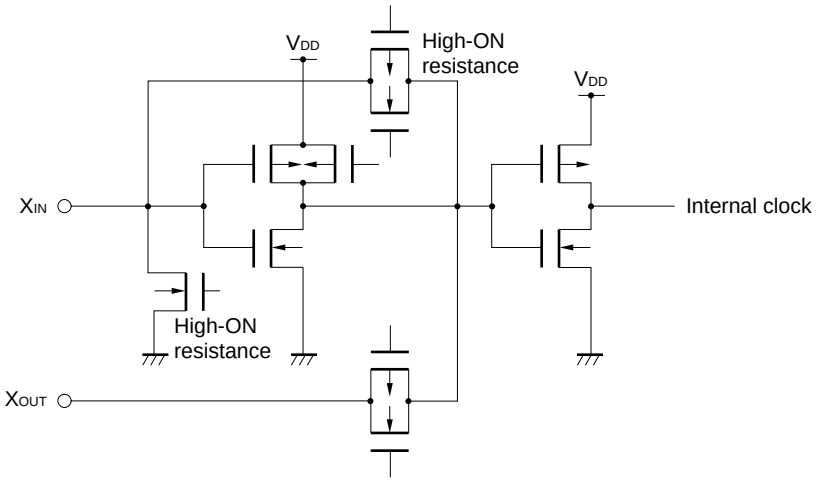


Note This is an internal signal that is output when the clock stop instruction is executed, and its circuit is designed not to increase the current consumption due to noise even if it is floated.

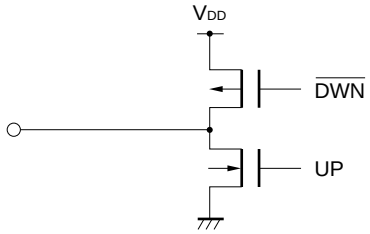
(8) CE
$\overline{\text{RESET}}$
INT0, INT1, INT2
P1A (P1A3/INT4, P1A2/INT3, P1A0/TM0G)
(Schmitt trigger input)



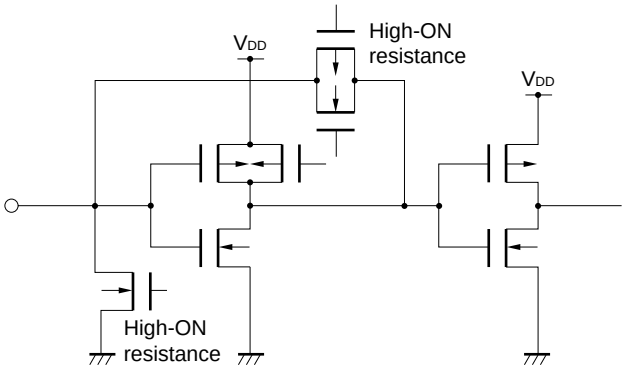
(9) X_{OUT} (output), X_{IN} (input)



(10) EO1, EO0 (output)



(11) VCOH, VCOL (Input)



1.3 Connections of Unused Pins

It is recommended to connect unused pins as follows:

Table 1-1. Connections of Unused Pins (1/2)

Pin Name		I/O Mode	Recommended Connections of Unused Pins
Port pin	P0D3/AD3-P0D0/AD0	Input	Individually connect to GND via resistor[Note 1].
	P1C3/AD5 P1C2/AD4		
	P1C1/AMIFC[Note 2] P1C0/FMIFC[Note 2]		Set in port mode and individually connect to V_{DD} or GND via resistor[Note 1].
	P1A3/INT4 P1A2/INT3 P1A1 P1A0/TM0G		Individually connect to GND via resistor[Note 1].
	P1B3 P1B2/PWM2-P1B0/PWM0	N-ch open-drain output	Set to low-level output by software and then open.
	P0A3/SDA P0A2/SCL P0A1/$\overline{SCK2}$ P0A0/SO2	I/O[Note 3]	Set in general-purpose input port mode by software and individually connect to V_{DD} or GND via resistor[Note 1].
	P0B3/SI2 P0B2/$\overline{SCK3}$ P0B1/SO3/TxD P0B0/SI3/RxD		
	P0C3-P0C0		
	P1D3 P1D2 P1D1/BEEP1 P1D0/BEEP0		
	P2A2 P2A1/FCG1 P2A0/FCG0		
	P2B3-P2B0		
	P2C3-P2C0		
	P2D2/$\overline{SCK}$ P2D1/SB1 P2D0/SB0		

Notes 1. If a pin is externally pulled up (connected to V_{DD} via resistor) or pulled down (connected to GND via resistor) with a high resistance, the pin almost enters a high-impedance state, increasing the current (through-current) consumption of the port. Generally, the resistance of a pull-up or pull-down resistor is several 10 kΩ, though it depends on the application circuit.

2. Do not set these pins as AMIFC and FMIFC pins; otherwise, the current consumption will increase.

3. The I/O ports are set in the general-purpose I/O port mode at power-ON reset, when reset by the $\overline{RESET}$ pin, or when reset due to overflow or underflow of the watchdog timer or the stack.

Table 1-1. Connections of Unused Pins (2/2)

Pin Name		I/O Mode	Recommended Connections of Unused Pins
Port pin	P3A3-P3A0	I/O[Note 2]	Set in general-purpose input port mode by software and individually connect to V_{DD} or GND via resistor[Note 1].
	P3B3-P3B0		
	P3C3-P3C0		
	P3D3-P3D0		
Pins other than port pins	CE	Input	Connect to V_{DD} via resistor[Note 1].
	EO1	Output	Open
	EO0		
	INT0-INT2	Input	Individually connect to GND via resistor[Note 1].
	$\overline{\text{RESET}}$	Input	Connect to V_{DD} via resistor[Note 1].
	TEST	–	Directly connect to GND.
	VCOH	Input	Disable PLL via software and open.
	VCOL		

Notes 1. If a pin is externally pulled up (connected to V_{DD} via resistor) or pulled down (connected to GND via resistor) with a high resistance, the pin almost enters a high-impedance state, increasing the current (through-current) consumption of the port. Generally, the resistance of a pull-up or pull-down resistor is several 10 kΩ, though it depends on the application circuit.

2. The I/O ports are set in the general-purpose input port mode at power-ON reset, when reset by the $\overline{\text{RESET}}$ pin, or when reset due to overflow or underflow of the watchdog timer or the stack.

1.4 Cautions on Using CE, INT0 through INT4, and $\overline{\text{RESET}}$ Pins

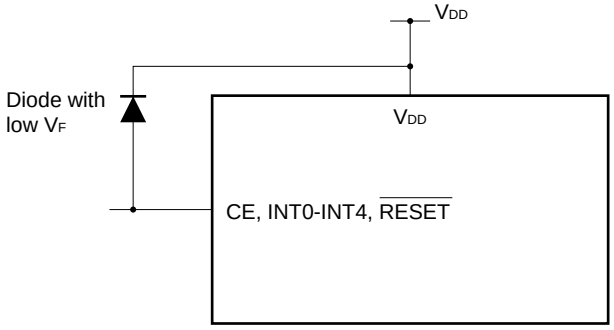
The CE, INT0 through INT4, and $\overline{\text{RESET}}$ pins have a function to set a test mode in which the internal operations of the μPD17719 are tested (IC test), in addition to the functions listed in 1.1 Pin Function List.

When a voltage exceeding V_{DD} is applied to any of these pins, the device is set in the test mode. If a noise exceeding V_{DD} is superimposed during normal operation, therefore, the test mode is set by mistake, hindering the normal operation.

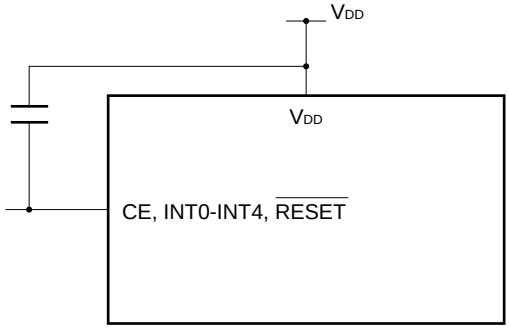
Especially if the wiring length of pins is too long, noise is superimposed on these pins. In consequence, the above problem occurs.

Therefore, keep the wiring length as short as possible to prevent noise from being superimposed. If superimposition of noise is unavoidable, connect an external component as illustrated below to suppress the noise.

- **Connect a diode with low V_F between a pin and V_{DD}.**



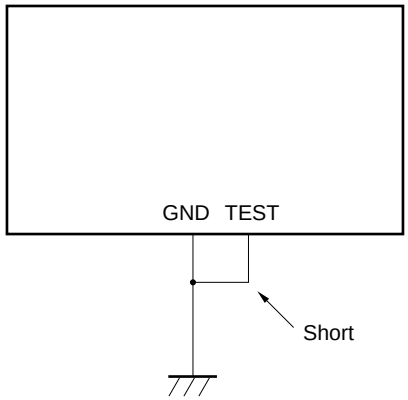
- **Connect a capacitor between a pin and V_{DD}.**



1.5 Cautions on Using TEST Pin

When V_{DD} is applied to the TEST pin, the device is set in the test mode. Therefore, be sure to keep the wiring length of this pin as short as possible, and directly connect it to the GND pin.

If the wiring length between the TEST pin and GND pin is too long, or if external noise is superimposed on the TEST pin, generating a potential difference between the TEST pin and GND pin, your program may not run normally.



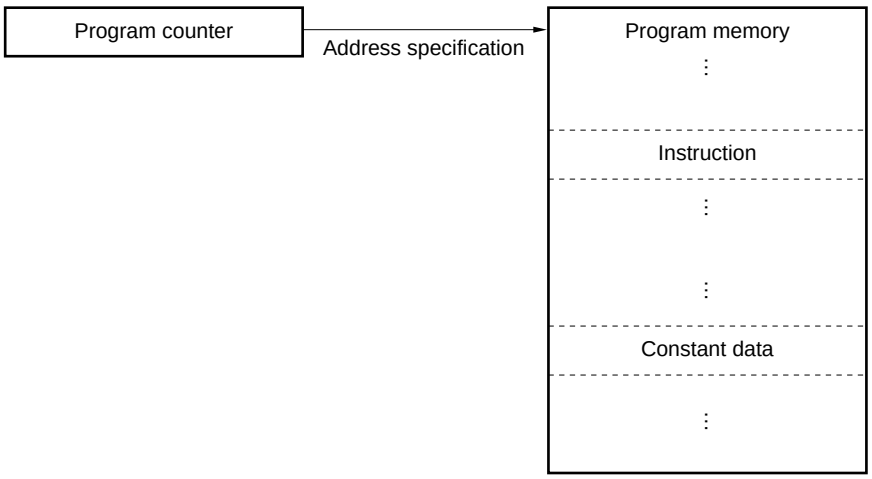
2. PROGRAM MEMORY (ROM)

2.1 Outline of Program Memory

Figure 2-1 outlines the program memory.
As shown in this figure, the addresses of the program memory are specified by the program counter.
The program memory has the following two major functions.

- To store programs
- To store constant data

Figure 2-1. Outline of Program Memory



2.2 Program Memory

Figure 2-2 shows the configuration of the program memory.

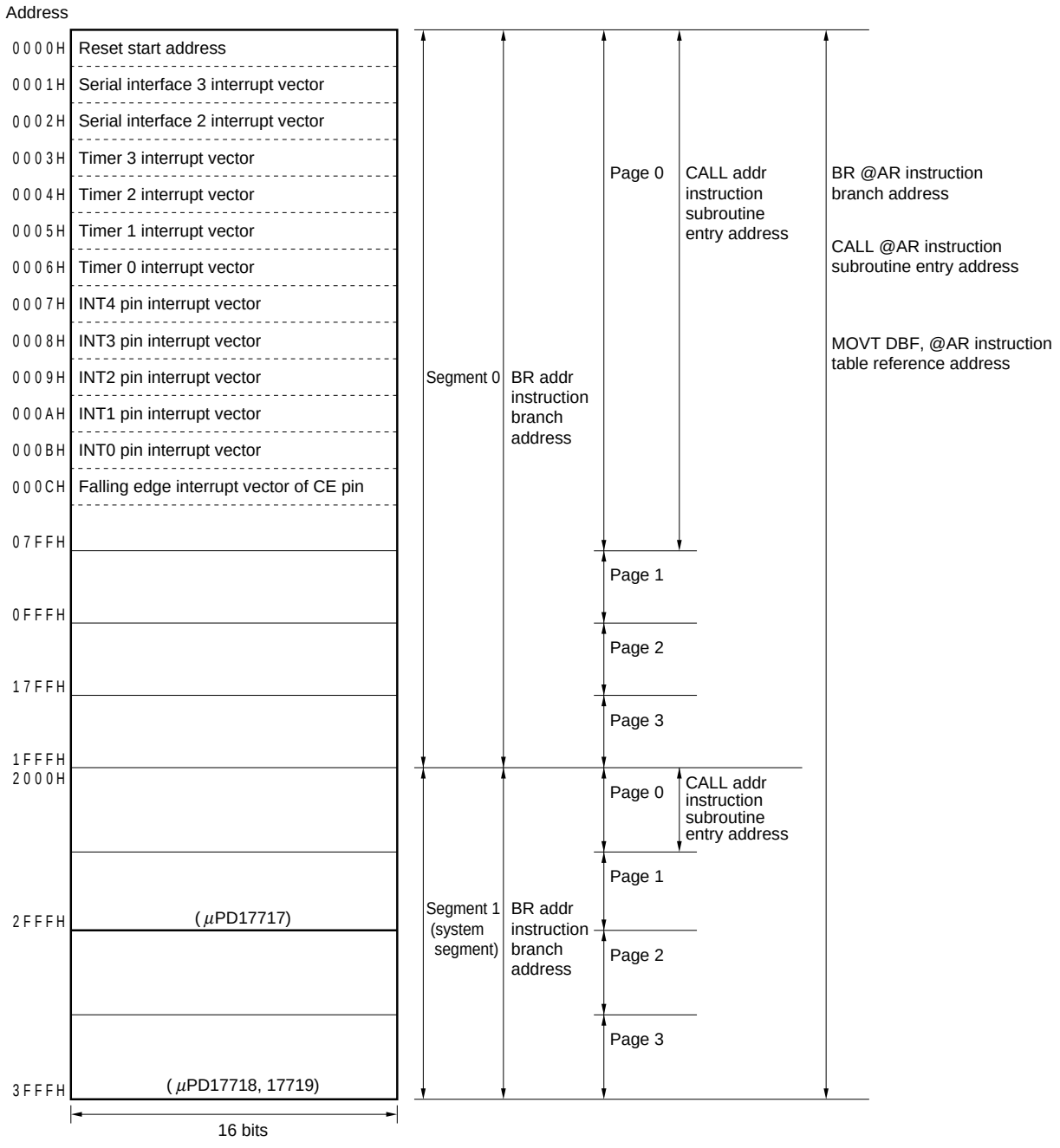
As shown in this figure, the μPD17717 has 24K bytes (12288 x 16 bits) of program memory, and the μPD17718 and 17719 have 32K bytes (16384 × 16 bits).

Therefore, the program memory addresses of the μPD17717 are 0000H through 2FFFH, and those of the μPD17718 and 17719 are 0000H through 3FFFH.

Because all “instructions” are “one-word instructions”, one instruction can be stored to one address of the program memory.

As constant data, the contents of the program memory are read to the data buffer by using a table reference instruction.

Figure 2-2. Configuration of Program Memory



2.3 Program Counter

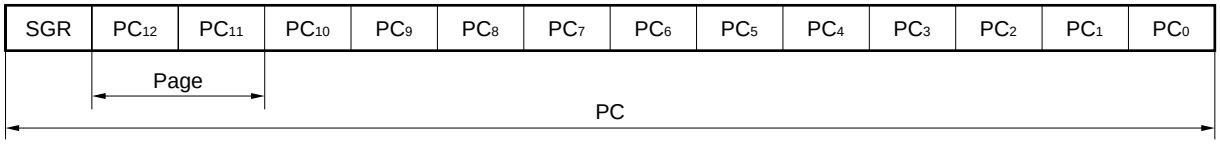
2.3.1 Configuration of program counter

Figure 2-3 shows the configuration of the program counter.

As shown in this figure, the program counter consists of a 13-bit binary counter and a 1-bit segment register (SGR). Bits 11 and 12 of the program counter indicate a page.

The program counter specifies an address of the program memory.

Figure 2-3. Configuration of Program Counter



2.3.2 Segment register (SGR)

The segment register specifies a segment of the program memory.

Table 2-1 shows the relationships between the segment register and program memory.

The segment register is set only when the SYSCAL entry instruction is executed.

Table 2-1. Relationships between Segment Register and Program Memory

Value of Segment Register	Segment of Program Memory
0	Segment 0
1	Segment 1

2.4 Flow of Program

The flow of the program is controlled by the program counter that specifies an address of the program memory.

The program flow when each instruction is executed is described below.

Figure 2-5 shows the value that is set to the program counter when each instruction is executed.

Table 2-2 shows the vector address when an interrupt is accepted.

2.4.1 Branch instruction

(1) Direct branch (“BR addr”)

The branch destination address of the direct branch instruction is in the same segment of the program memory. In other words, a branch cannot be executed exceeding a segment.

(2) Indirect branch (“BR @AR”)

The branch destination addresses of the indirect branch instruction are all the addresses of the program memory, i.e., addresses 0000H through 2FFFH for the μPD17717 and 0000H through 3FFFH for the μPD17718 and 17719.

For further information, also refer to **5.3 Address Register (AR)**.

2.4.2 Subroutine

(1) Direct subroutine call (“CALL addr”)

The first address of a subroutine that can be called by the direct subroutine instruction is in page 0 of each segment (addresses 0000H through 07FFH).

(2) Indirect subroutine call (CALL @AR)

The first addresses of a subroutine that can be called by the indirect subroutine call instruction are all the addresses of the program memory, i.e., addresses 0000H through 2FFFH for the μPD17717 and 0000H through 3FFFH for the μPD17718 and 17719.

For further information, also refer to **5.3 Address Register (AR).**

2.4.3 Table reference

The addresses that can be referenced by the table reference instruction (“MOVT DBF, @AR”) are all the addresses of the program memory, i.e., addresses 0000H through 2FFFH for the μPD17717 and 0000H through 3FFFH for the μPD17718 and 17719.

For further information, also refer to **5.3 Address Register (AR)** and **9.2.2 Table reference instruction (MOVT, DBF, @AR)**.

2.4.4 System call

The first address of a subroutine that can be called by the system call instruction (“SYSCAL entry”) is the first 16 steps of each block (block 0 to 7) in page 0 of segment 1 (system segment).

Figure 2-4. Outline of System Call Instruction

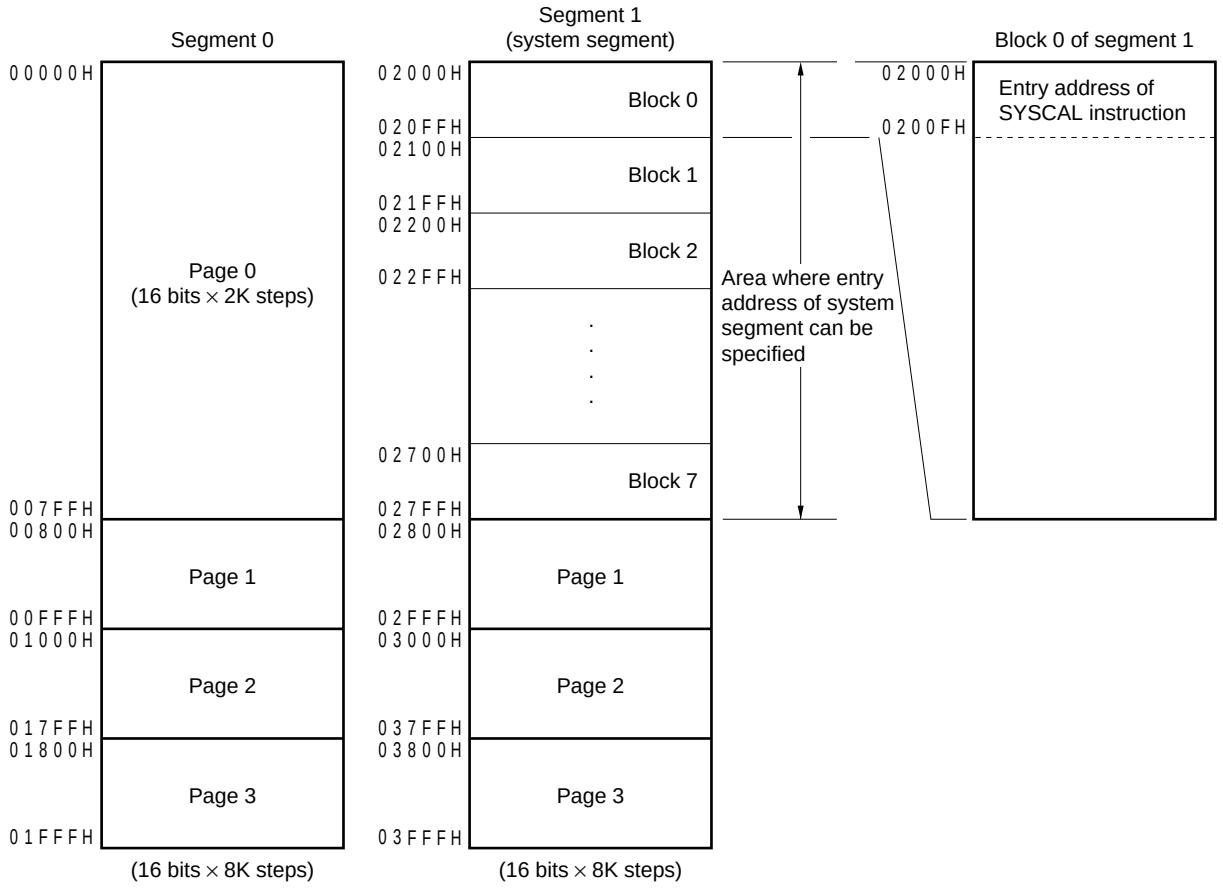


Figure 2-5. Value of Program Counter Upon Execution of Instruction

Instruction \ Program counter		Contents of Program Counter (PC)														
		SGR	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
BR addr	Page 0	Re-tained	0	0	Operand of instruction (addr)										
	Page 1		0	1											
	Page 2		1	0											
	Page 3		1	1											
CALL addr		Re-tained	0	0	Operand of instruction (addr)										
SYSCAL entry		1	0	0	$entry_H$			0	0	0	0	$entry_L$			
BR @AR CALL @AR MOVT DBF, @AR		Contents of address register													
RET RETSK RETI		Contents of address stack register (ASR) (return address) specified by stack pointer (SP)													
Other instructions (including skip instruction)		Re-tained	Increment												
When interrupt is accepted		0	Vector address of each interrupt												
Power-ON reset, watchdog timer reset, $\overline{\text{RESET}}$ pin, CE reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0

$entry_H$: high-order 3 bits of entry

$entry_L$: low-order 4 bits of entry

Table 2-2. Interrupt Vector Address

Order	Internal/External	Interrupt Source	Vector Address
1	External	Falling edge of CE pin	00CH
2	External	INT0 pin	00BH
3	External	INT1 pin	00AH
4	External	INT2 pin	009H
5	External	INT3 pin	008H
6	External	INT4 pin	007H
7	Internal	Timer 0	006H
8	Internal	Timer 1	005H
9	Internal	Timer 2	004H
10	Internal	Timer 3	003H
11	Internal	Serial interface 2	002H
12	Internal	Serial interface 3	001H

2.5 Cautions on Using Program Memory

2.5.1 Last address in each segment

The segment register is not connected to the binary counter.

Therefore, address 0000H of segment 0 is specified next to address 1FFFH, which is the last address of segment 0.

To specify between segments, a dedicated instruction such as an indirect branch, indirect subroutine call, or system call instruction is used.

3. ADDRESS STACK (ASK)

3.1 Outline of Address Stack

Figure 3-1 outlines the address stack.

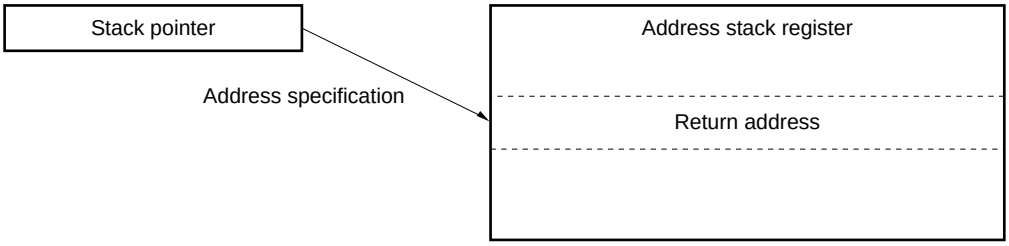
The address stack consists of a stack pointer and address stack registers.

The address of an address stack register is specified by the stack pointer.

The address stack saves a return address when a subroutine call instruction is executed or when an interrupt is accepted.

The address stack is also used when the table reference instruction is executed.

Figure 3-1. Outline of Address Stack



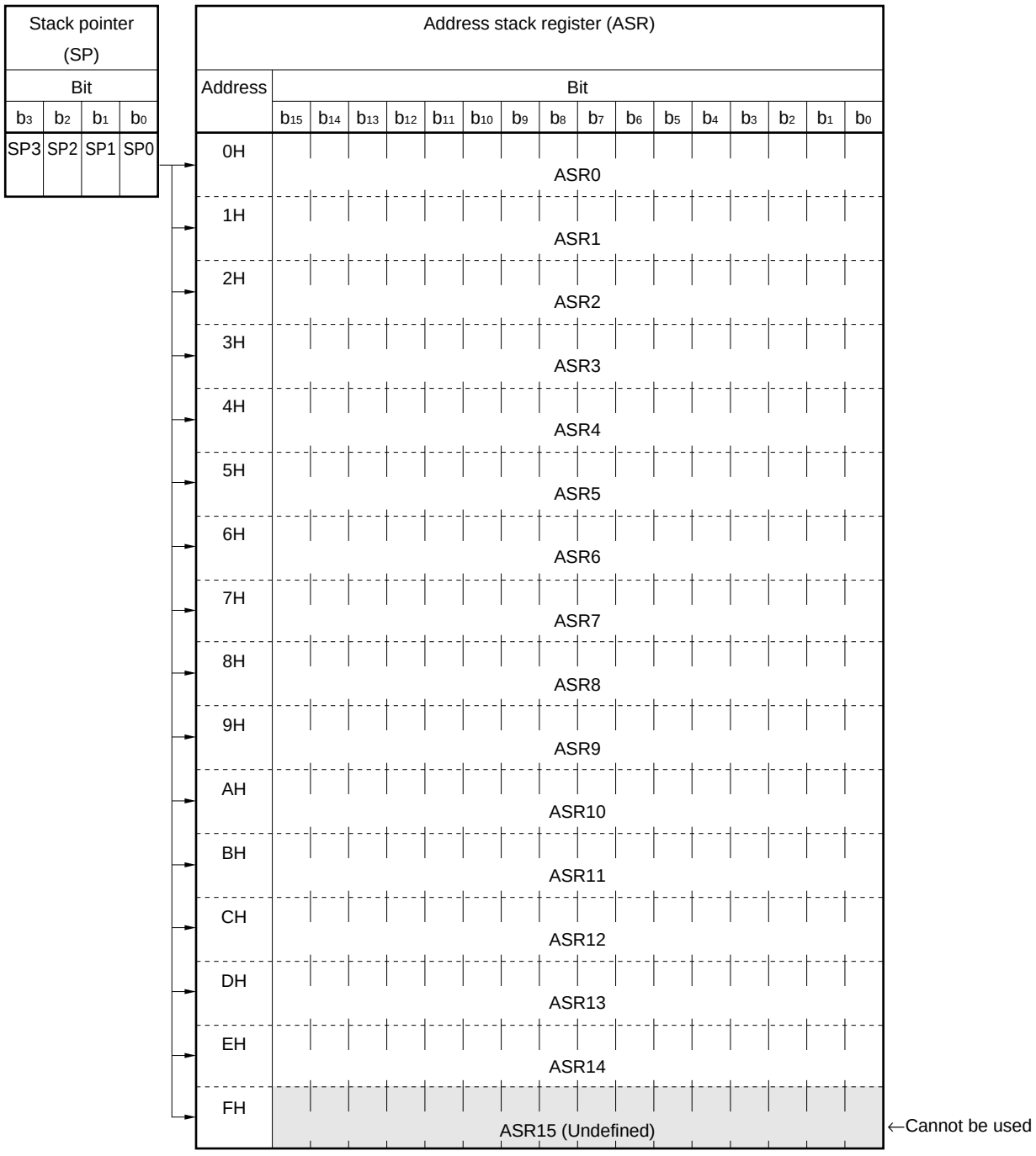
3.2 Address Stack Register (ASR)

Figure 3-2 shows the configuration of the address stack register.

The address stack register consists of sixteen 16-bit registers ASR0 through ASR15. Actually, however, it consists of fifteen 16-bit registers (ASR0 through ASR14) because no register is allocated to ASR15.

The address stack saves a return address when a subroutine is called, when an interrupt is accepted, and when the table reference instruction is executed.

Figure 3-2. Configuration of Address Stack Register



3.3 Stack Pointer (SP)

3.3.1 Configuration and function of stack pointer

Figure 3-3 shows the configuration and functions of the stack pointer.

The stack pointer consists of a 4-bit binary counter.

It specifies the address of an address stack register.

A value can be directly read from or written to the stack pointer by using a register manipulation instruction.

Figure 3-3. Configuration and Function of Stack Pointer

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Stack pointer (SP)	SP3	SP2	SP1	SP0	01H	R/W

b_3	b_2	b_1	b_0	Specifies address of address stack register (ASR)
0	0	0	0	Address 0 (ASR0)
0	0	0	1	Address 1 (ASR1)
0	0	1	0	Address 2 (ASR2)
0	0	1	1	Address 3 (ASR3)
0	1	0	0	Address 4 (ASR4)
0	1	0	1	Address 5 (ASR5)
0	1	1	0	Address 6 (ASR6)
0	1	1	1	Address 7 (ASR7)
1	0	0	0	Address 8 (ASR8)
1	0	0	1	Address 9 (ASR9)
1	0	1	0	Address 10 (ASR10)
1	0	1	1	Address 11 (ASR11)
1	1	0	0	Address 12 (ASR12)
1	1	0	1	Address 13 (ASR13)
1	1	1	0	Address 14 (ASR14)
1	1	1	1	Setting prohibited

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	1	1	1	1
	WDT&SP reset	1	1	1	1
	CE reset	1	1	1	1
Clock stop		Retained			

Power-ON reset : Reset by $\overline{\text{RESET}}$ pin up on power application

WDT&SP reset : Reset by watchdog timer and stack pointer

CE reset : CE reset

Clock stop : Upon execution of clock stop instruction

3.4 Operation of Address Stack

3.4.1 Subroutine call instruction (“CALL addr”, “CALL @AR”) and return instruction (“RET”, “RETSK”)

When a subroutine call instruction is executed, the value of the stack pointer is decremented by one, and the return address is stored to an address stack register specified by the stack pointer.

When the return instruction is executed, the contents of the address stack register (return address) specified by the stack pointer are restored to the program counter, and the value of the stack pointer is incremented by one.

3.4.2 Table reference instruction (“MOVT DBF, @AR”)

When the table reference instruction is executed, the value of the stack pointer is incremented by one, and the return address is stored to an address stack register specified by the stack pointer.

Next, the contents of the program memory specified by the address register are read to the data buffer, the contents of the address stack register (return value) specified by the stack pointer are restored to the program counter, and the value of the stack pointer is incremented by one.

3.4.3 When interrupt is accepted and on execution of return instruction (“RETI”)

When an interrupt is accepted, the value of the stack pointer is decremented by one, and the return address is stored to an address stack register specified by the stack pointer.

When the return instruction is executed, the contents of an address stack register (return value) specified by the stack pointer are restored to the program counter, and the value of the stack pointer is incremented by one.

3.4.4 Address stack manipulation instruction (“PUSH AR”, “POP AR”)

When the “PUSH” instruction is executed, the value of the stack pointer is decremented by one, and the contents of the address register are transferred to an address stack register specified by the stack pointer.

When the “POP” instruction is executed, the contents of an address stack register specified by the stack pointer are transferred to the address register, and the value of the stack pointer is incremented by one.

3.4.5 System call instruction (“SYSCAL entry”) and return instruction (“RET”, “RETSK”)

When the “SYSCAL entry” instruction is executed, the value of the stack pointer is decremented by one, and the return address and the value of the segment register are stored to an address stack register specified by the stack pointer.

When the return instruction is executed, the contents of an address stack register (return value) specified by the stack pointer are restored to the program counter and segment register, and the value of the stack pointer is incremented by one.

3.5 Cautions on Using Address Stack

3.5.1 Nesting level and operation on overflow

The value of address stack register (ASR15) is “undefined” when the value of the stack pointer is 0FH.

Accordingly, if a subroutine call or system call exceeding 15 levels, or an interrupt is used without manipulating the stack, execution returns to an “undefined” address.

3.5.2 Reset on detection of overflow or underflow of address stack

Whether the device is reset on detection of overflow or underflow of the address stack can be specified by program. At reset, the program is started from address 0, and some control registers are initialized.

This reset function is valid at power-ON reset or reset by the $\overline{\text{RESET}}$ pin. For details, refer to **21. RESET**.

4. DATA MEMORY (RAM)

4.1 Outline of Data Memory

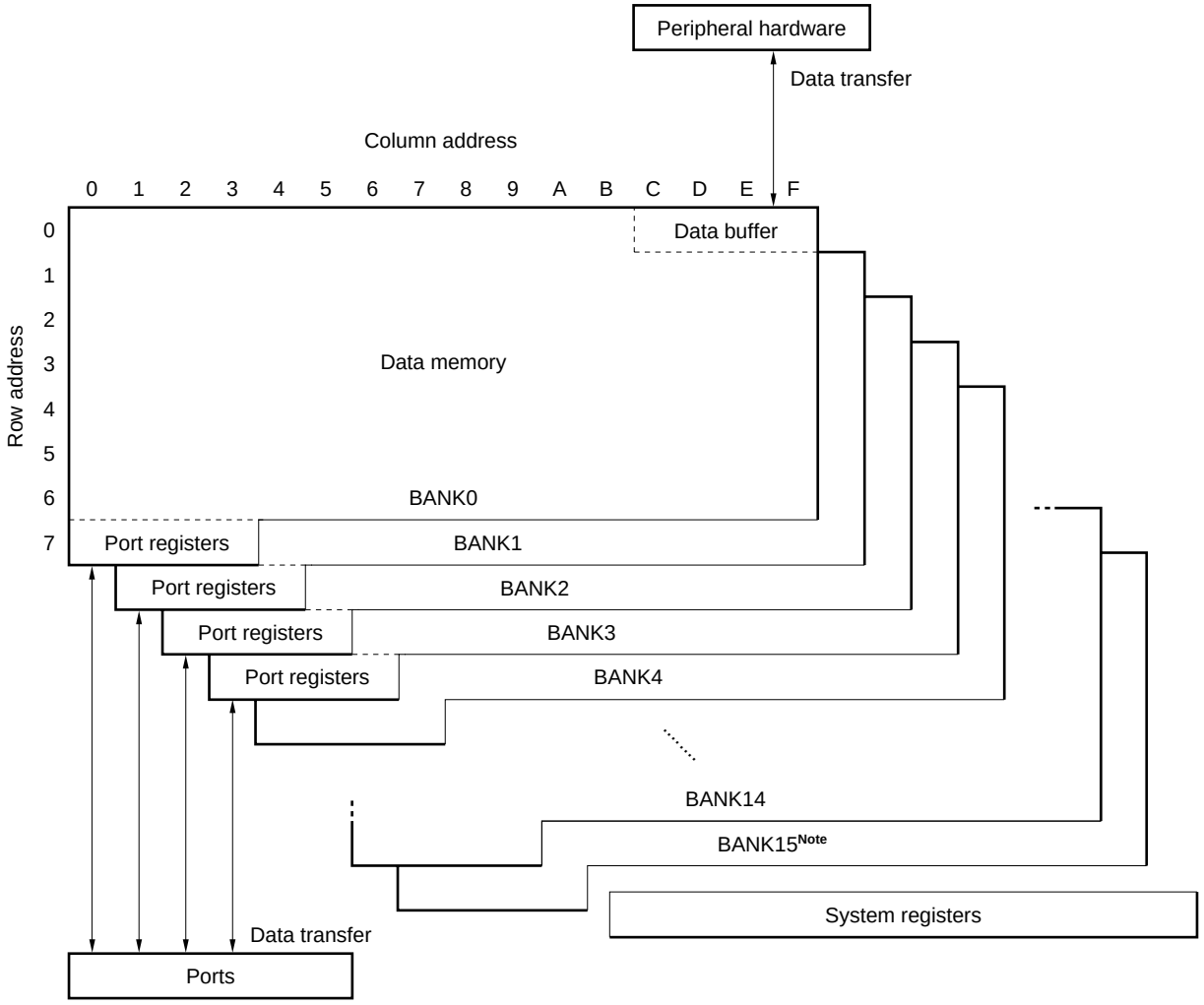
Figure 4-1 outlines the data memory.

As shown in the figure, system registers, a data buffer, port registers, and port input/output selection registers are located on the data memory.

The data memory stores data, transfers data with the peripheral hardware or ports, and controls the CPU.

Figure 4-1. Outline of Data Memory (1/2)

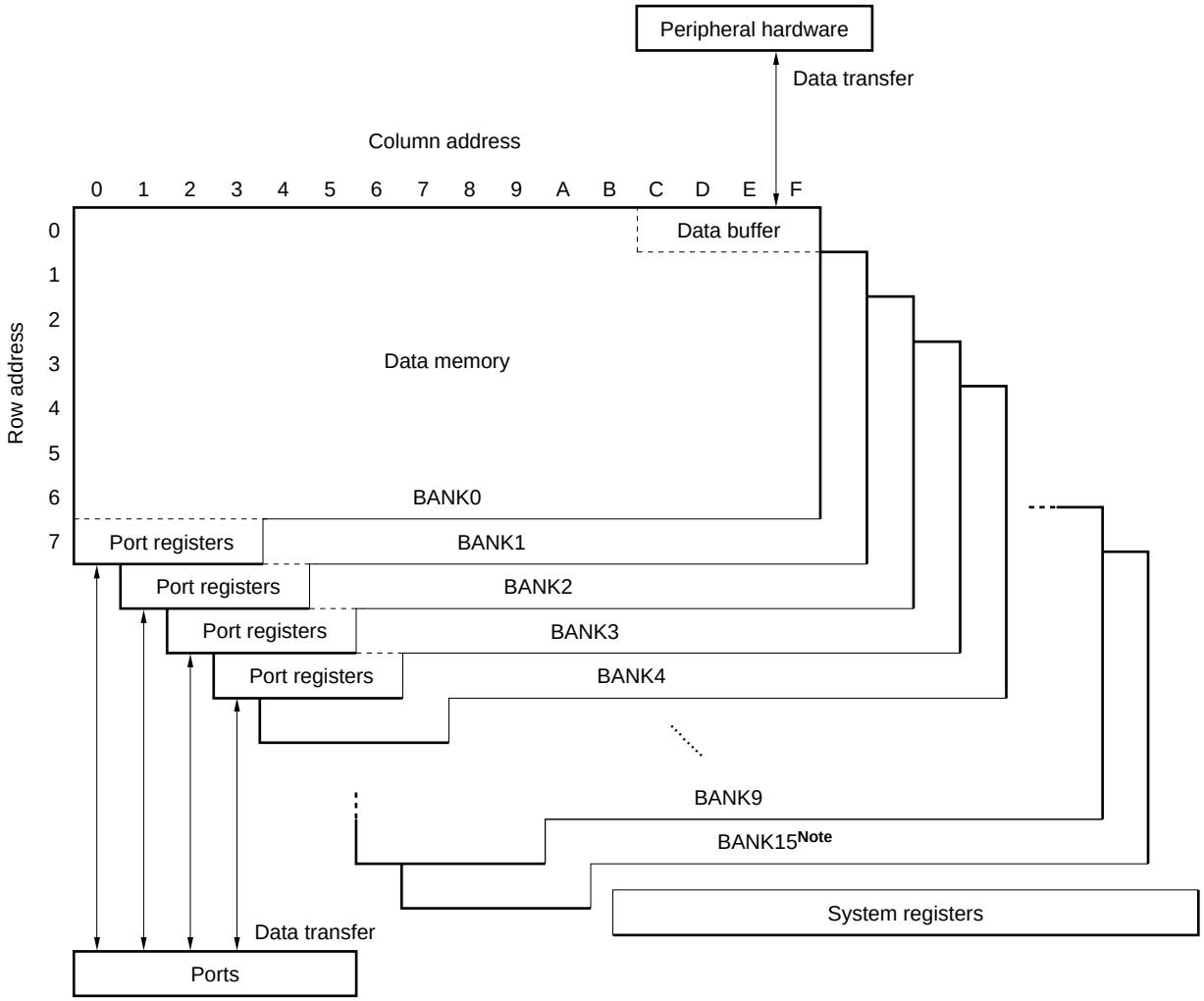
(a) μPD17719



Note Port input/output selection registers are allocated to addresses 60H through 6FH of BANK 15.

Figure 4-1. Outline of Data Memory (2/2)

(b) μPD17717, 17718



Note Port input/output selection registers are allocated to addresses 60H through 6FH of BANK 15.

Cautions 1. The μPD17717 and 17718 do not have BANKs 10 through 14.

2. Nothing is allocated to addresses 00H through 5FH of BANK15.

4.2 Configuration and Function of Data Memory

Figure 4-2 shows the configuration of the data memory.

As shown in this figure, the data memory is divided into several banks with each bank made up of a total of 128 nibbles with 7H row addresses and 0FH column addresses.

The data memory can be divided into five functional blocks. Each block is described in 4.2.1 through 4.2.5 below.

The contents of the data memory can be operated on, compared, judged, and transferred in 4-bit units with a single data memory manipulation instruction.

Table 4-1 lists the data memory manipulation instructions.

4.2.1 System registers (SYSREG)

The system registers are allocated to addresses 74H through 7FH.

Because the system registers are allocated to all banks, the same system registers exist at addresses 74H through 7FH of any bank.

For details, refer to **5. SYSTEM REGISTER (SYSREG)**.

4.2.2 Data buffer (DBF)

The data buffer is allocated to addresses 0CH through 0FH of BANK 0.

For details, refer to **9. DATA BUFFER (DBF)**.

4.2.3 Port registers

The port registers are allocated to addresses 70H through 73H of BANKs 0 through 3.

For details, refer to **11. GENERAL-PURPOSE PORTS**.

4.2.4 Port input/output selection registers

Port input/output selection registers are allocated to addresses 60H through 6FH of BANK15.

For details, refer to **8.4 Port Input/Output Selection Register**.

4.2.5 General-purpose data memory

The general-purpose data memory is allocated to the addresses of the data memory excluding those of the system registers, port registers, and port input/output selection registers.

(a) μPD17719

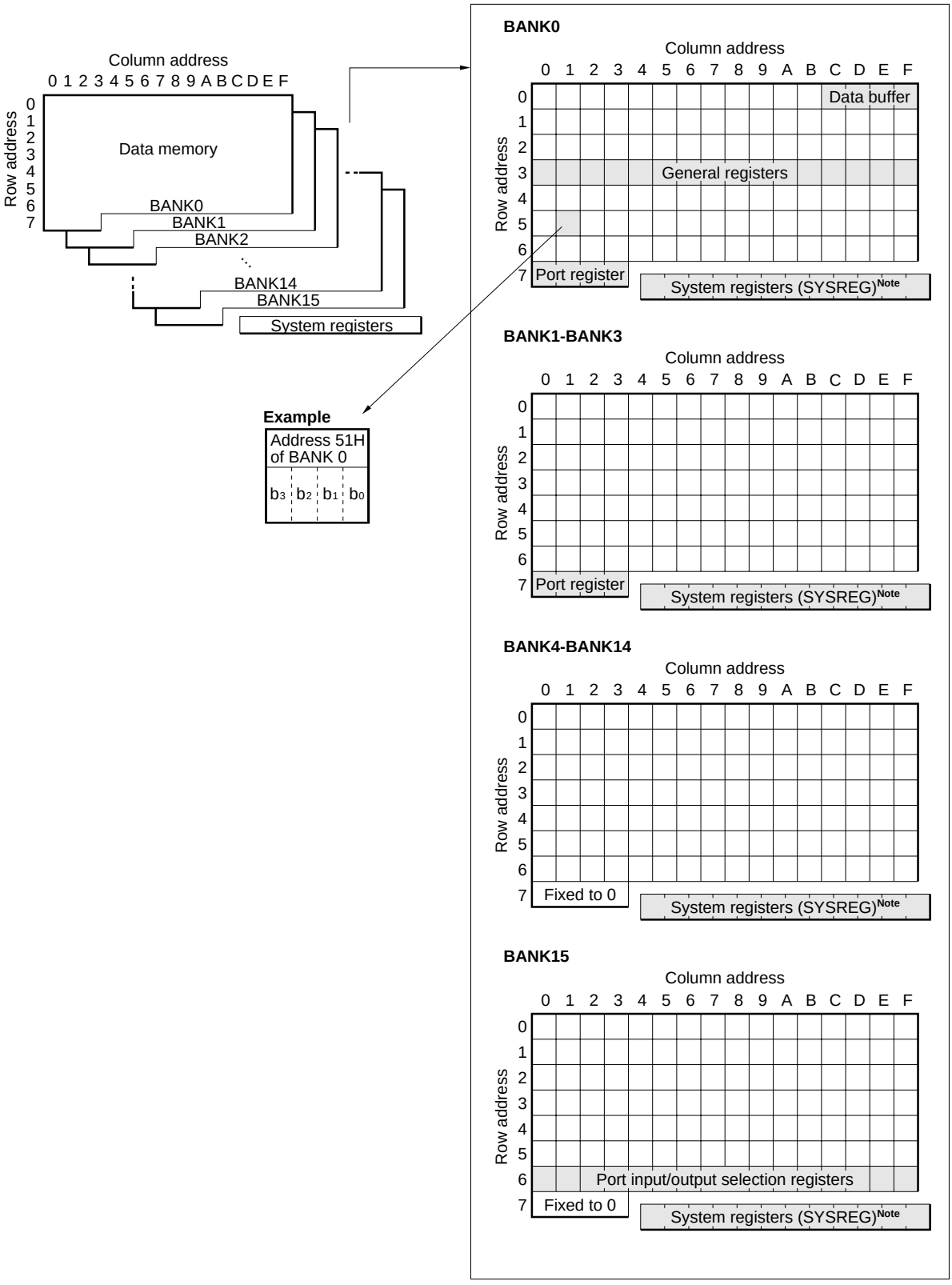
The general-purpose data memory of the μPD17709 consists of a total of 1776 nibbles of the 112 nibbles each of BANKs 0 through 15 (BANK15 only has 96 nibbles).

(b) μPD17717, 17718

The general-purpose data memory of the μPD17707 and 17708 consists of a total of 1120 nibbles of the 112 nibbles each of BANKs 0 through 9.

Figure 4-2. Configuration of Data Memory (1/2)

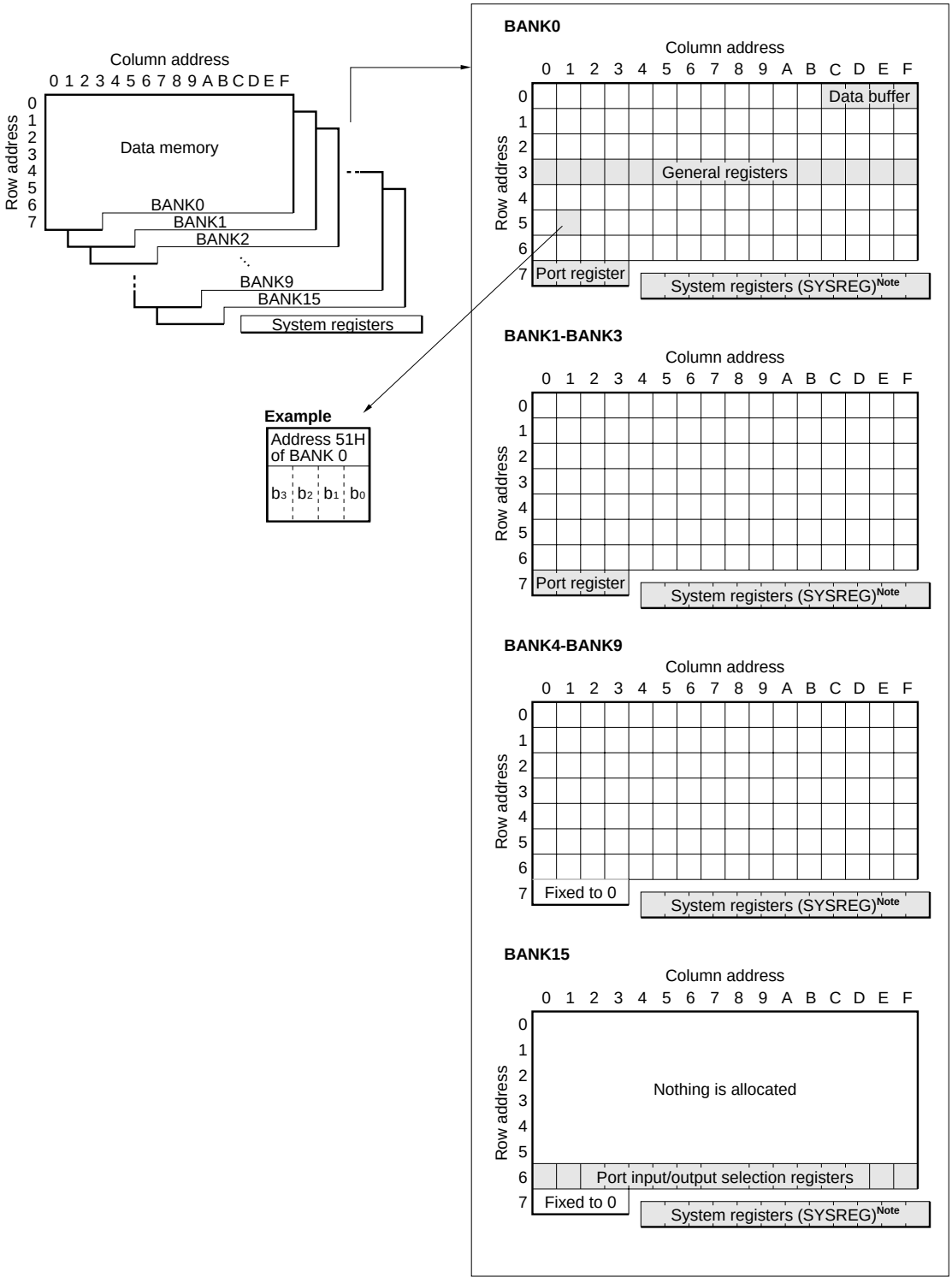
(a) μPD17719



Note An identical system register exists.

Figure 4-2. Configuration of Data Memory (2/2)

(b) μPD17717, 17718



Note An identical system register exists.

Cautions 1. The μPD17717 and 17718 do not have BANKs 10 through 14.

2. Nothing is allocated to addresses 00H through 5FH of BANK15.

Table 4-1. Data Memory Manipulation Instructions

Function		Instruction
Operation	Add	ADD ADDC
	Subtract	SUB SUBC
	Logic	AND OR XOR
Compare		SKE SKGE SKLT SKNE
Transfer		MOV LD ST
Judge		SKT SKF

4.3 Data Memory Addressing

Figure 4-3 shows address specification of the data memory.

An address of the data memory is specified by a bank, row address, and column address.

A row address and a column address are directly specified by a data memory manipulation instruction. However, a bank is specified by the contents of a bank register.

For the details of the bank register, refer to **5. SYSTEM REGISTER (SYSREG)**.

Figure 4-3. Address Specification of Data Memory

	Bank				Row address			Column address			
	b_3	b_2	b_1	b_0	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data memory address	Bank register				Instruction operand						

4.4 Cautions on Using Data Memory

4.4.1 At power-ON reset

The contents of the general-purpose data memory are “undefined” at power-ON reset.

Initialize the data memory as necessary.

4.4.2 Cautions on data memory not provided

If a data memory manipulation instruction that reads the data memory is executed to a data memory address not provided, undefined data is read.

Nothing is changed even if data is written to such an address.

5. SYSTEM REGISTERS (SYSREG)

5.1 Outline of System Registers

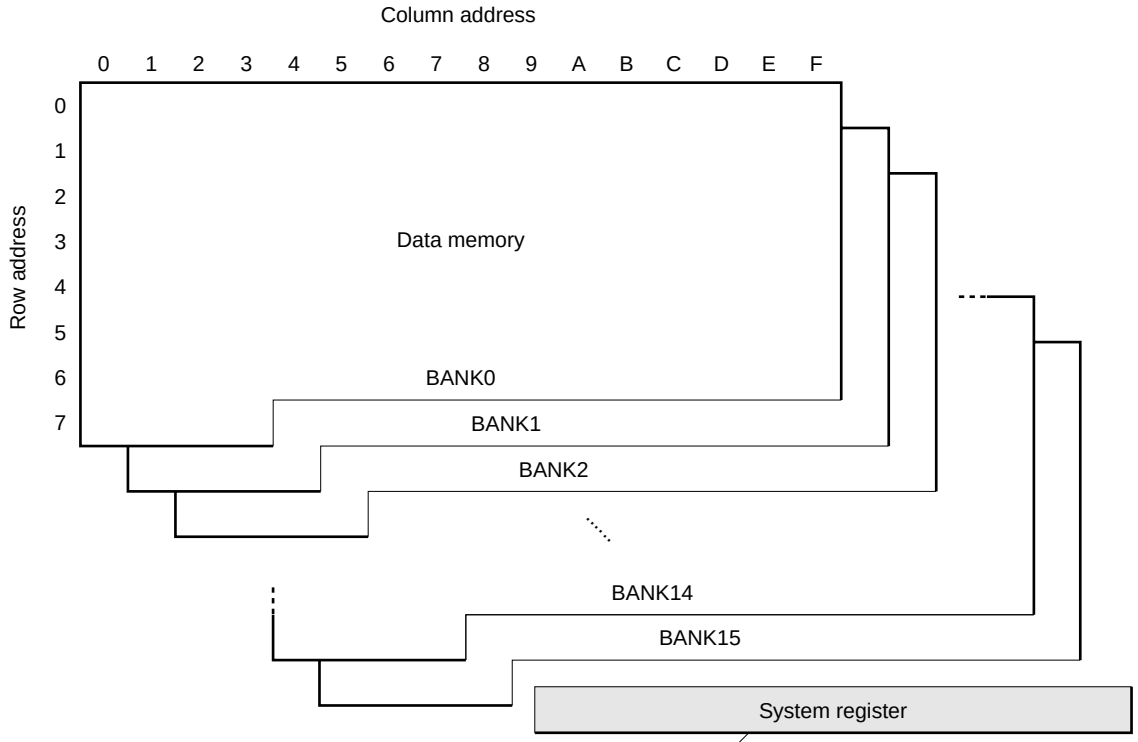
Figure 5-1 shows the location of the system registers on the data memory and their outline.

As shown in the figure, the system registers are allocated to addresses 74H through 7FH of all the banks of the data memory. Therefore, identical system registers exist at addresses 74H through 7FH of any bank.

Because the system registers are located on the data memory, they can be manipulated by all data memory manipulation instructions.

Seven types of system registers are available depending on function.

Figure 5-1. Location and Outline of System Registers on Data Memory



Remark The μPD17717 and 17718 do not have BANKs 10 through 14.

Address	74H	75H	76H	77H	78H	79H	7AH	7BH	7CH	7DH	7EH	7FH
Name	Address register (AR)				Window register (WR)	Bank register (BANK)	Index register (IX)			General register pointer (RP)		Program status word (PSWORD)
							Data memory row address pointer (MP)					
Function	Controls program memory address				Transfers data with register file	Specifies bank of data memory	Modifies address of data memory			Specifies address of general register		Controls operation

5.2 System Register List

Figure 5-2 shows the configurations of the system registers.

Figure 5-2. Configuration of System Registers

Address	74H	75H	76H	77H	78H	79H	7AH	7BH	7CH	7DH	7EH	7FH
Name	System registers											
	Address register (AR)				Window register (WR)	Bank register (BANK)	Index register (IX)			General register pointer (RP)		Program status word (PSWORD)
							Data memory row address pointer (MP)					
Symbol	AR3	AR2	AR1	AR0	WR	BANK	IXH	IXM	IXL	RPH	RPL	PSW
							MPH	MPL				
Bit	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0
Data	MPE						0	(MP)	(IX)	(RP)	BCDP	CMPY ZIXE

5.3 Address Register (AR)

5.3.1 Configuration of address register

Figure 5-3 shows the configuration of the address register.

As shown in the figure, the address register consists of 16 bits of system register addresses 74H through 77H (AR3 through AR0).

Figure 5-3. Configuration of Address Register

Address		74H				75H				76H				77H			
Name		Address register (AR)															
Symbol		AR3				AR2				AR1				AR0			
Bit		b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data		MSB															LSB
At reset	Power-ON reset	0				0				0				0			
	WDT&SP reset	0				0				0				0			
	CE reset	0				0				0				0			
Clock stop		Retained				Retained				Retained				Retained			

Power-ON reset : Reset by $\overline{\text{RESET}}$ pin on power application

WDT&SP reset : Reset by watchdog timer and stack pointer

CE reset : CE reset

Clock stop : On execution of clock stop instruction

5.3.2 Function of address register

The address register specifies a program memory address when the table reference instruction (“MOVT DBF, @AR”), stack manipulation instruction (“PUSH AR”, “POP AR”), indirect branch instruction (“BR @AR”), or indirect subroutine call instruction (“CALL @AR”) is executed.

A dedicated instruction (“INC AR”) is available that can increment the contents of the address instruction by one.

The following paragraphs (1) through (5) describe the operation of the address register when the respective instructions are executed.

(1) Table reference instruction (“MOVT DBF, @AR”)

When the table reference instruction is executed, the constant data (16 bits) of a program memory address specified by the contents of the address register are read to the data buffer.

The constant data that can be specified by the address register is stored to address 0000H to 2FFFH in the case of the μPD17717, and address 0000H to 3FFFH in the case of the μPD17718 and 17719.

(2) Stack manipulation instruction (“PUSH AR”, “POP AR”)

When the “PUSH AR” instruction is executed, the value of the stack pointer is decremented by one, and the contents of the address register (AR) are transferred to an address stack register specified by the stack pointer whose value has been decremented by one.

When the “POP AR” instruction is executed, the contents of an address stack register specified by the stack pointer are transferred to the address register, and the value of the stack pointer is incremented by one.

(3) Indirect branch instruction (“BR @AR”)

When this instruction is executed, the program branches to a program memory address specified by the contents of the address register.

The branch address that can be specified by the address register is 0000H to 2FFFH in the case of the μPD17717, and 0000H to 3FFFH in the case of the μPD17718 and 17719.

(4) Indirect subroutine call instruction (“CALL @AR”)

The subroutine at a program memory address specified by the contents of the address register can be called.

The first address of the subroutine that can be specified by the address register is 0000H to 2FFFH in the case of the μPD17717, and 0000H to 3FFFH in the case of the μPD17718 and 17719.

(5) Address register increment instruction (“INC AR”)

This instruction increments the contents of the address register by one.

5.3.3 Address register and data buffer

The address register can transfer data as part of the peripheral hardware via the data buffer.

For details, refer to **9. DATA BUFFER (DBF)**.

5.3.4 Cautions on Using Address Register

Because the address register is configured in 16 bits, it can specify an address up to FFFFH.

However, the program memory exists at addresses 0000H through 2FFFH in the case of the μPD17717 and addresses 0000H through 3FFFH in the case of the μPD17718 and 17719.

Therefore, the maximum value that can be set to the address register of the μPD17717 is address 2FFFH. In the case of the μPD17718 and 17719, it is address 3FFFH.

5.4 Window Register (WR)

5.4.1 Configuration of window register

Figure 5-4 shows the configuration of the window register.

As shown in the figure, the window register consists of 4 bits of system register address 78H (WR).

Figure 5-4. Configuration of Window Register

Address		78H			
Name		Window register (WR)			
Symbol		WR			
Bit		b_3	b_2	b_1	b_0
Data		MSB			LSB
At reset	Power-ON reset	Undefined			
	WDT&SP reset	Retained			
	CE reset				
Clock stop					

5.4.2 Function of window register

The window register is used to transfer data with the register file (RF) to be described later.

Data transfer between the window register and register file is manipulated by using dedicated instructions “PEEK WR, rf” and “POKE, rf WR” (rf: address of register file).

The following paragraphs (1) and (2) describe the operation of the window register when these instructions are executed.

For further information, also refer to **8. REGISTER FILE (RF)**.

(1) “PEEK WR, rf” instruction

When this instruction is executed, the contents of the register file addressed by “rf” are transferred to the window register.

(2) “POKE rf, WR” instruction

When this instruction is executed, the contents of the window register are transferred to the register file addressed by “rf”.

5.5 Bank Register (BANK)

5.5.1 Configuration of bank register

Figure 5-5 shows the configuration of the bank register.

As shown in the figure, the bank register consists of 4 bits of system register address 79H (BANK).

Figure 5-5. Configuration of Bank Register

Address		79H			
Name		Bank register (BANK)			
Symbol		BANK			
Bit		b_3	b_2	b_1	b_0
Data		MSB			LSB
At reset	Power-ON reset	0			
	WDT&SP reset	0			
	CE reset	0			
Clock stop		Retained			

5.5.2 Function of bank register

The bank register specifies a bank of the data memory.

Table 5-1 shows the relationships between the value of the bank register and a bank of the data memory that is specified.

Because the bank register is one of the system registers, its contents can be rewritten regardless of the bank currently specified.

When manipulating a bank register, therefore, the status of the bank at that time is irrelevant.

Table 5-1. Data Memory Bank Specification

Bank Register (BANK)				Bank of Data Memory
b_3	b_2	b_1	b_0	
0	0	0	0	BANK0
0	0	0	1	BANK1
0	0	1	0	BANK2
0	0	1	1	BANK3
0	1	0	0	BANK4
0	1	0	1	BANK5
0	1	1	0	BANK6
0	1	1	1	BANK7

Bank Register (BANK)				Bank of Data Memory
b_3	b_2	b_1	b_0	
1	0	0	0	BANK8
1	0	0	1	BANK9
1	0	1	0	BANK10[Note]
1	0	1	1	BANK11[Note]
1	1	0	0	BANK12[Note]
1	1	0	1	BANK13[Note]
1	1	1	0	BANK14[Note]
1	1	1	1	BANK15

Note Do not set BANKs 10 through 14 in the μPD17717 and 17718 because these banks are not provided.

Caution The area to which the data memory is allocated differs depending on the model. For details, refer to Figure 4-2 Configuration of Data Memory.

5.6 Index Register (IX) and Data Memory Row Address Pointer (MP: memory pointer)

5.6.1 Configuration of index register and data memory row address pointer

Figure 5-6 shows the configuration of the index register and data memory row address pointer.

As shown in the figure, the index register consists of an index register (IX) made up of 11 bits (the low-order 3 bits (IXH) of system register address 7AH, and 7BH and 7CH (IXM, IXL)) and an index enable flag (IXE) at the lowest bit position of 7FH (PSW).

The data memory row address pointer (memory pointer) consists of a data memory row address pointer (MP) that is made up of 7 bits of the low-order 3 bits of 7AH (MPH) and 7BH (MPL), and a data memory row address pointer enable flag (memory pointer enable flag: MPE) at the lowest bit position of 7AH (MPH).

In other words, the high-order 7 bits of the index register are shared with the data memory row address pointer

Figure 5-6. Configuration of Index Register and Data Memory Row Address Pointer

Address		7AH				7BH				7CH				7EH				7FH			
Name		Index register (IX) / Memory pointer (MP)															Program status word (PSWORD)				
Symbol		IXH / MPH				IXM / MPL				IXL								PSW			
Bit		b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data		MPE	$\overline{MSB}$										$\overline{LSB}$								IXE
			$\overline{MSB}$ (MP)						$\overline{LSB}$ (MP)												
At reset	Power-ON reset	0				0				0											0
	WDT&SP reset	0				0				0											0
	CE reset	0				0				0											0
Clock stop		Retained				Retained				Retained											R

R: retained

5.6.2 Functions of index register and data memory row address pointer

The index register and data memory row address pointer modify the addresses of the data memory.

The following paragraphs (1) and (2) describe their functions.

A dedicated instruction (“INC IX”) that increments the value of the index register by one is available.

For the details of address modification, refer to **7. ALU (Arithmetic Logic Unit) BLOCK**.

(1) Index register (IX)

When a data memory manipulation instruction is executed, the data memory address is modified by the contents of the index register.

This modification, however, is valid only when the IXE flag is set to 1.

To modify the address, the bank, row address, and column address of the data memory are ORed with the contents of the index register, and the instruction is executed to a data memory address (called real address) specified by the result of this OR operation.

All data memory manipulation instructions are subject to address modification by the index register.

The following instructions, however, are not subject to address modification by the index register.

INC	AR	RORC	r
INX	IX	CALL	addr
MOVT	DBF, @AR	CALL	@AR
PUSH	AR	RET	
POP	AR	RETSK	
PEEK	WR,rf	RETI	
POKE	rf,WR	EI	
GET	DBF,p	DI	
PUT	p, DBF	STOP	s
BR	addr	HALT	h
BR	@AR	NOP	

(2) Data memory row address pointer (MP)

When the general register indirect transfer instruction (“MOV @r,m” or “MOV m,@r”) is executed, the indirect transfer destination address is modified.

This modification, however, is valid only when the MPE flag is set to 1.

To modify the address, the bank and row address at the indirect transfer destination are replaced by the contents of the data memory row address pointer.

Instructions other than the general register indirect transfer instruction are not subject to address modification.

(3) Index register increment instruction (“INC IX”)

This instruction increments the contents of the index register by one.

Because the index register is configured of 10 bits, its contents are incremented to “000H” if the “INC IX” instruction is executed when the contents of the index register are “3FFH”.

5.7 General Register Pointer (RP)

5.7.1 Configuration of General Register Pointer

Figure 5-7 shows the configuration of the general register pointer.

As shown in the figure, the general register pointer consists of 7 bits including 4 bits of system register address 7DH (RPH) and the high-order 3 bits of address 7EH (RPL).

Figure 5-7. Configuration of General Register Pointer

Address		7DH				7EH			
Name		General register pointer (RP)							
Symbol		RPH				RPL			
Bit		b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data		MSB						LSB	BCD
At reset	Power-ON reset	0				0			
	WDT&SP reset	0				0			
	CE reset	0				0			
Clock stop		Retained				Retained			

5.7.2 Function of general register pointer

The general register pointer specifies a general register on the data memory.

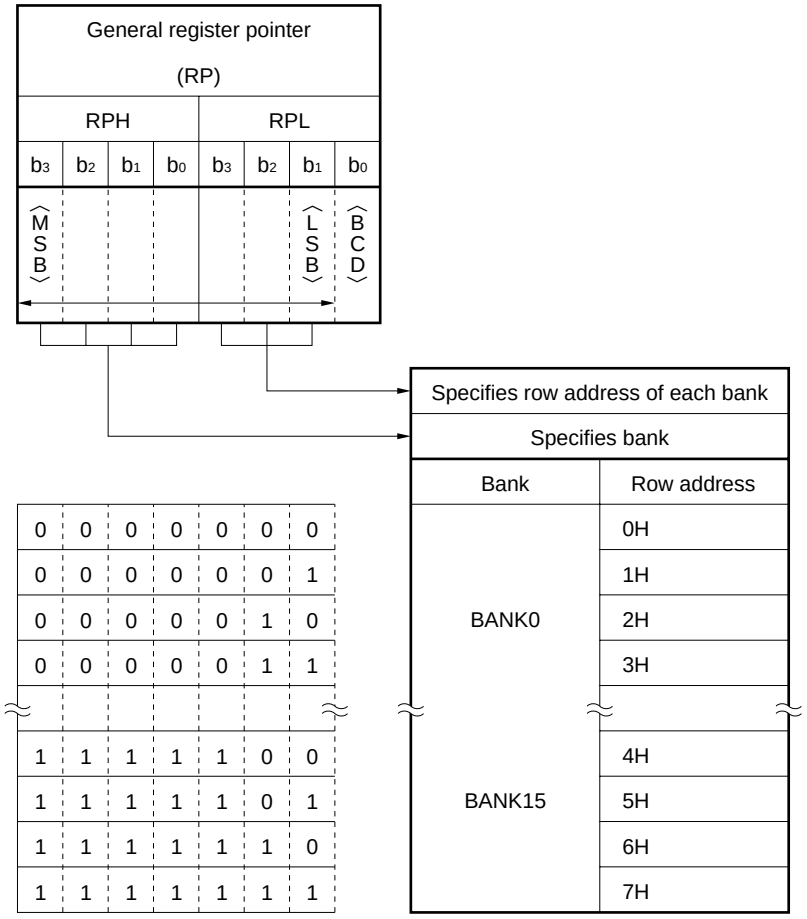
Figure 5-8 shows the addresses of the general registers specified by the general register pointer.

As shown in the figure, a bank is specified by the high-order 4 bits (RPH: address 7DH) of the general register pointer, and a row address is specified by the low-order 3 bits (RPL: address 7EH).

Because the valid number of bits of the general register pointer is 7, all the row addresses (0H through 7FH) of all the banks can be specified as general registers.

For the details of the operation of the general register, refer to **6. GENERAL REGISTER (GR)**.

Figure 5-8. Address of General Register Specified by General Register Pointer



Remark The μPD17717 and 17718 do not have BANKs 10 through 14.

Caution The area to which the data memory is allocated differs depending on the model. For details, refer to Figure 4-2 Configuration of Data Memory.

5.7.3 Cautions on using general register pointer

The lowest bit of address 7EH (RPL) of the general register pointer is allocated as the BCD flag of the program status word.

When rewriting RPL, therefore, pay attention to the value of the BCD flag.

5.8 Program Status Word (PSWORD)

5.8.1 Configuration of program status word

Figure 5-9 shows the configuration of the program status word.

As shown in the figure, th program status word consists of a total of 5 bits including the lowest bit of system register address 7EH (RPL) and 4 bits of address 7FH (PSW).

Each bit of the program status word has its own function. The 5 bits of the program status word are BCD flag (BCD), compare flag (CMP), carry flag (CY), zero flag (Z), and index enable flag (IXE).

Figure 5-9. Configuration of Program Status Word

Address		7EH				7FH			
Name					Program status word (PSWORD)				
Symbol		RPL				PSW			
Bit		b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data					BCD	CMP	CY	Z	IXE
At reset	Power-ON reset	0				0			
	WDT&SP reset	0				0			
	CE reset	0				0			
Clock stop		Retained				Retained			

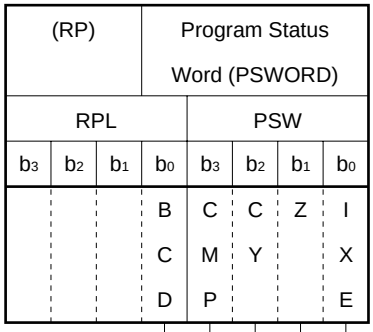
5.8.2 Function of program status word

The program status word is a register that sets the conditions under which the ALU (Arithmetic Logic Unit) executes an operation or data transfer, or indicates the result of an operation.

Table 5-2 outlines the function of each flag of the program status word.

For details, refer to **7. ALU (Arithmetic Logic Unit) BLOCK**.

Table 5-2. Outline of Function of Each Flag of Program Status Word



Flag Name	Function
Index enable flag (IXE)	Modifies address of data memory when data memory manipulation instruction is exeuted. 0 : Does not modify 1 : Modifies
Zero flag (Z)	Indicates result of arithmetic operation is zero. Status of this flag differs depending on contents of compare flag.
Carry flag (CY)	Indicates occurrence of carry or borrow as result of execution of addition or subtraction instruction. This flag is reset to 0 if no carry or borrow occurs. It is set to 1 if carry or borrow occurs. This flag is also used as shift bit of “RORC r” instruction.
Compare flag (CMP)	Indicates whether result of arithmetic operation is stored to data memory or general register. 0 : Stores result. 1 : Does not store result.
BCD flag (BCD)	Indicates whether arithmetic operation is performed in decimal or binary. 0 : Binary operation 1 : Decimla operation

5.8.3 Cautions on using program status word

When an arithmetic operation (addition or subtraction) is executed to the program status word, the “result” of the arithmetic operation is stored.

For example, even if an operation that generates a carry is executed, if the result of the operation is 0000B, 0000B is stored to the PSW.

6. GENERAL REGISTER (GR)

6.1 Outline of General Register

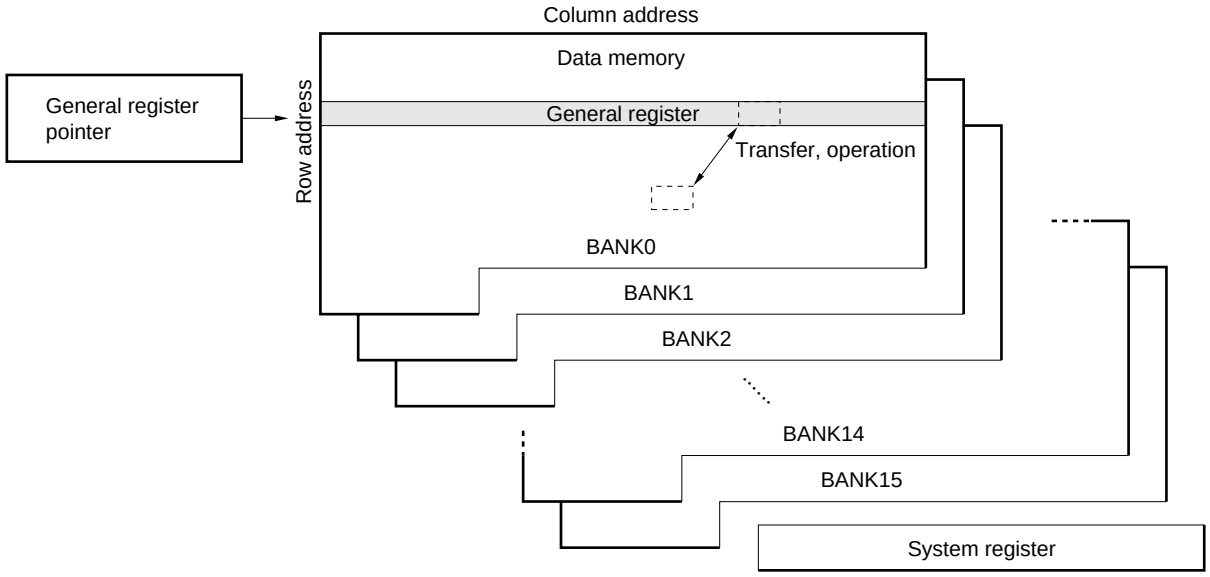
Figure 6-1 outlines the general register.

As shown in the figure, the general register is specified in the data memory by the general register pointer.

The bank and row address of the general register are specified by the general register pointer.

The general register is used to transfer or operate data between data memory addresses.

Figure 6-1. Outline of General Register



Remark The μPD17717 and 17718 do not have BANKs 10 through 14.

6.2 General Register

The general register consists of 16 nibbles (16 × 4 bis) of the same row address on the data memory.

For the range of the banks and row addresses that can be specified by the general register pointer as a general register, refer to **5.7 General Register Pointer (RP)**.

The 16 nibbles of the same row address specified as a general register operate or transfer data with the data memory by a single instruction.

In other words, operation or data transfer between data memory addresses can be executed by a single instruction.

The general register can be controlled by the data memory manipulation instruction, like the other data memory areas.

6.3 Generating Address of General Register by Each Instruction

The following sections 6.3.1 and 6.3.2 explain how the address of the general register is generated when each instruction is executed.

For the details of the operation of each instruction, refer to **7. ALU (Arithmetic Logic Unit) BLOCK**.

6.3.1 Add (“ADD r, m”, “ADDC r, m”) , subtract (“SUB r, m”, “SUBC r, m”) , logical operation (“AND r, m”, “OR r, m”, “XOR r, m”), direct transfer (“LD r, m”, “ST m, r”), and rotation (“RORC r”) instructions

Table 6-1 shows the address of the general register specified by operand “r” of an instruction. Operand “r” of an instruction specifies only a column address.

Table 6-1. Generating Address of General Register

	Bank				Row address			Column address			
	b_3	b_2	b_1	b_0	b_2	b_1	b_0	b_3	b_2	b_1	b_0
General register address	Contents of general register pointer							r			

6.3.2 Indirect transfer (“MOV @r, m”, “MOV m, @r”) instruction

Table 6-2 shows a general register address specified by instruction operand “r” and an indirect transfer address specified by “@r”.

Table 6-2. Generating Address of General Register

	Bank				Row address			Column address			
	b_3	b_2	b_1	b_0	b_2	b_1	b_0	b_3	b_2	b_1	b_0
General register address	Contents of general register pointer							r			
Indirect transfer address	Same as data memory							Contents of “r”			

6.4 Cautions on Using General Register

6.4.1 Row address of general register

Because the row address of the general register is specified by the general register pointer, the currently specified bank may differ from the bank of the general register.

6.4.2 Operation between general register and immediate data

No instruction is available that executes an operation between the general register and immediate data.

To execute an operation between the general register and immediate data, the general register must be treated as a data memory area.

7. ALU (Arithmetic Logic Unit) BLOCK

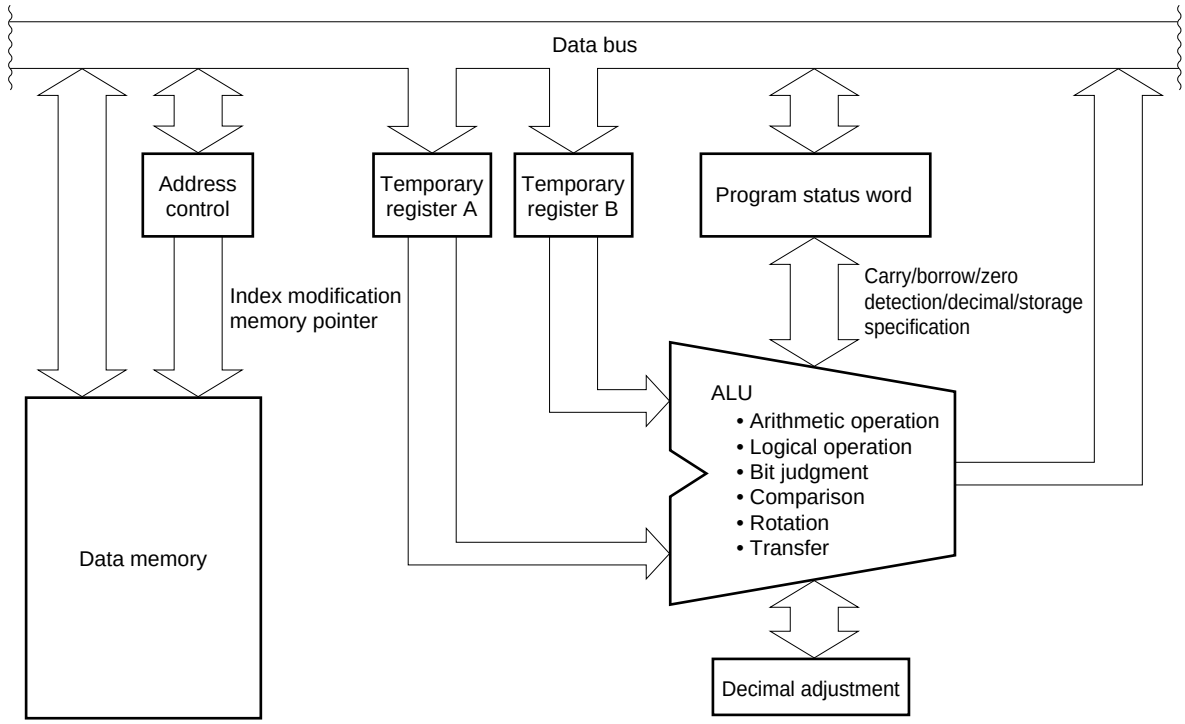
7.1 Outline of ALU Block

Figure 7-1 outlines the ALU block.

As shown in the figure, the ALU block consists of an ALU, temporary registers A and B, program status word, decimal adjustment circuit, and memory address control circuit.

The ALU operates on, judges, compares, rotates, and transfers 4-bit data in the data memory.

Figure 7-1. Outline of ALU Block



7.2 Configuration and Function of Each Block

7.2.1 ALU

The ALU performs arithmetic operation, logical operation, bit judgment, comparison, rotation, and transfer of 4-bit data according to instructions specified by the program.

7.2.2 Temporay registers A and B

Temporary registers A and B temporarily store 4-bit data.

These registers are automatically used when an instruction is executed, and cannot be controlled by program.

7.2.3 Program status word

The program status word controls the operation of and stores the status of the ALU.

For further information on the program status word, also refer to **5.8 Program Status Word (PSWORD)**.

7.2.4 Decimal adjustment circuit

The decimal adjustment circuit converts the result of an arithmetic operation into a decimal number if the BCD flag of the program status word is set to “1” during arthmetic operations.

7.2.5 Address control circuit

The address control circuit specifies an address of the data memory.

At this time, address modification by the index register and data memory row address pointer is also controlled.

7.3 ALU Processing Instruction List

Table 7-1 lists the ALU operations when each instruction is executed.

Table 7-2 shows how data memory addresses are modified by the index register and data memory row address pointer.

Table 7-3 shows decimal adjustment data when a decimal operation is performed.

Table 7-1. List of ALU Processing Instruction Operations

ALU Function	Instruction		Difference in Operation Depending on Program Status Word (PSWORD)					Address Modification	
			Value of BCD flag	Value of CMP flag	Operation	Operation of CY flag	Operation of Z flag	Index	Memory pointer
Add	ADD	r, m	0	0	Stores result of binary operation	Set if carry or borrow occurs; otherwise, reset	Set if result of operation is 0000B; otherwise, reset	Modifies	Does not modify
		m, #n4							
	ADDC	r, m	0	1	Does not store result of binary operation		Retains status if result of operation is 0000B; otherwise, reset		
		m, #n4							
Subtract	SUB	r, m	1	0	Stores result of decimal operation		Set if result of operation is 0000B; otherwise, reset		
		m, #n4							
	SUBC	r, m	1	1	Does not store result of decimal operation		Retains status if result of operation is 0000B; otherwise, reset		
		m, #n4							
Logical operation	OR	r, m	Don't care (retained)	Don't care (retained)	Not affected	Retains previous status	Retains previous status	Modifies	Does not modify
		m, #n4							
	AND	r, m							
		m, #n4							
	XOR	r, m							
		m, #n4							
Judge	SKT	m, #n	Don't care (retained)	Don't care (reset)	Not affected	Retains previous status	Retains previous status	Modifies	Does not modify
	SKF	m, #n							
Compare	SKE	m, #n4	Don't care (retained)	Don't care (retained)	Not affected	Retains previous status	Retains previous status	Modifies	Does not modify
	SKNE	m, #n4							
	SKGE	m, #n4							
	SKLT	m, #n4							
Transfer	LD	r, m	Don't care (retained)	Don't care (retained)	Not affected	Retains previous status	Retains previous status	Modifies	Does not modify
	ST	m, r							
	MOV	m, #n4							
		@r, m							Modifies
		m, @r							
Rotate	RORC	r	Don't care (retained)	Don't care (retained)	Not affected	Value of b_0 of general register	Retains previous status	Does not modify	Does not modify

Table 7-2. Modification of Data Memory Address and Indirect Transfer Address by Index Register and Data Memory Row Address Pointer

IXE	MPE	General Register Address Specified by "r"			Data Memory Address Specified by "m"			Indirect Transfer Address Specified by "@r"		
		Bank	Row address	Column address	Bank	Row address	Column address	Bank	Row address	Column address
		b_3 b_2 b_1 b_0	b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_3 b_2 b_1 b_0	b_2 b_1 b_0	b_3 b_2 b_1 b_0
0	0	RP		r	BANK		m	BANK	m_R	(r)
0	1		ditto			ditto		MP		(r)
1	0		ditto		BANK		m	BANK	m_R	
					Logical IX		OR	Logical IXH, IXM		OR (r)
1	1		ditto			ditto		MP		(r)

BANK : bank register
IX : index register
IXE : index enable flag
IXH : bits 10 through 8 of index register
IXM : bits 7 through 4 of index register
IXL : bits 3 through 0 of index register
m : data memory address indicated by m_R, m_C
m_R : data memory row address (high-order)
m_C : data memory column address (low-order)
MP : data memory row address pointer
MPE : memory pointer enable flag
r : general register column address
RP : general register pointer
(X) : contents addressed by X
X: direct address such as "m" and "r"

Table 7-3. Decimal Adjustment Data

Operation Result	Hexadecimal Addition		Decimal Addition	
	CY	Operation result	CY	Operation result
0	0	0000B	0	0000B
1	0	0001B	0	0001B
2	0	0010B	0	0010B
3	0	0011B	0	0011B
4	0	0100B	0	0100B
5	0	0101B	0	0101B
6	0	0110B	0	0110B
7	0	0111B	0	0111B
8	0	1000B	0	1000B
9	0	1001B	0	1001B
10	0	1010B	1	0000B
11	0	1011B	1	0001B
12	0	1100B	1	0010B
13	0	1101B	1	0011B
14	0	1110B	1	0100B
15	0	1111B	1	0101B
16	1	0000B	1	0110B
17	1	0001B	1	0111B
18	1	0010B	1	1000B
19	1	0011B	1	1001B
20	1	0100B	1	1110B
21	1	0101B	1	1111B
22	1	0110B	1	1100B
23	1	0111B	1	1101B
24	1	1000B	1	1110B
25	1	1001B	1	1111B
26	1	1010B	1	1100B
27	1	1011B	1	1101B
28	1	1100B	1	1010B
29	1	1101B	1	1011B
30	1	1110B	1	1100B
31	1	1111B	1	1101B

Operation Result	Hexadecimal Addition		Decimal Addition	
	CY	Operation result	CY	Operation result
0	0	0000B	0	0000B
1	0	0001B	0	0001B
2	0	0010B	0	0010B
3	0	0011B	0	0011B
4	0	0100B	0	0100B
5	0	0101B	0	0101B
6	0	0110B	0	0110B
7	0	0111B	0	0111B
8	0	1000B	0	1000B
9	0	1001B	0	1001B
10	0	1010B	1	1100B
11	0	1011B	1	1101B
12	0	1100B	1	1110B
13	0	1101B	1	1111B
14	0	1110B	1	1100B
15	0	1111B	1	1101B
–16	1	0000B	1	1110B
–15	1	0001B	1	1111B
–14	1	0010B	1	1100B
–13	1	0011B	1	1101B
–12	1	0100B	1	1110B
–11	1	0101B	1	1111B
–10	1	0110B	1	0000B
–9	1	0111B	1	0001B
–8	1	1000B	1	0010B
–7	1	1001B	1	0011B
–6	1	1010B	1	0100B
–5	1	1011B	1	0101B
–4	1	1100B	1	0110B
–3	1	1101B	1	0111B
–2	1	1110B	1	1000B
–1	1	1111B	1	1001B

Remark Decimal adjustment is not correctly carried out in the shaded area in the above table.

7.4 Cautions on Using ALU

7.4.1 Cautions on execution operation to program status word

If an arithmetic operation is executed to the program status word, the result of the operation is stored to the program status word.

The CY and Z flags in the program status word are usually set or reset by the result of the arithmetic operation. If an arithmetic operation is executed to the program status word itself, the result of the operation is stored to the program status word, and consequently, it cannot be judged whether a carry or borrow occurs or whether the result of the operation is zero.

If the CMP flag is set, however, the result of the operation is not stored to the program status word. Therefore, the CY and Z flags are set or reset normally.

7.4.2 Cautions on executing decimal operation

The decimal operation can be executed only when the result of the operation falls within the following ranges:

(1) Result of addition : 0 to 19 in decimal
(2) Result of subtraction: 0 to 9 or –10 to –1 in decimal

If a decimal operation is executed exceeding or falling below the above ranges, the result is a value greater than 1010B (0AH).

8. REGISTER FILE (RF)

8.1 Outline of Register File

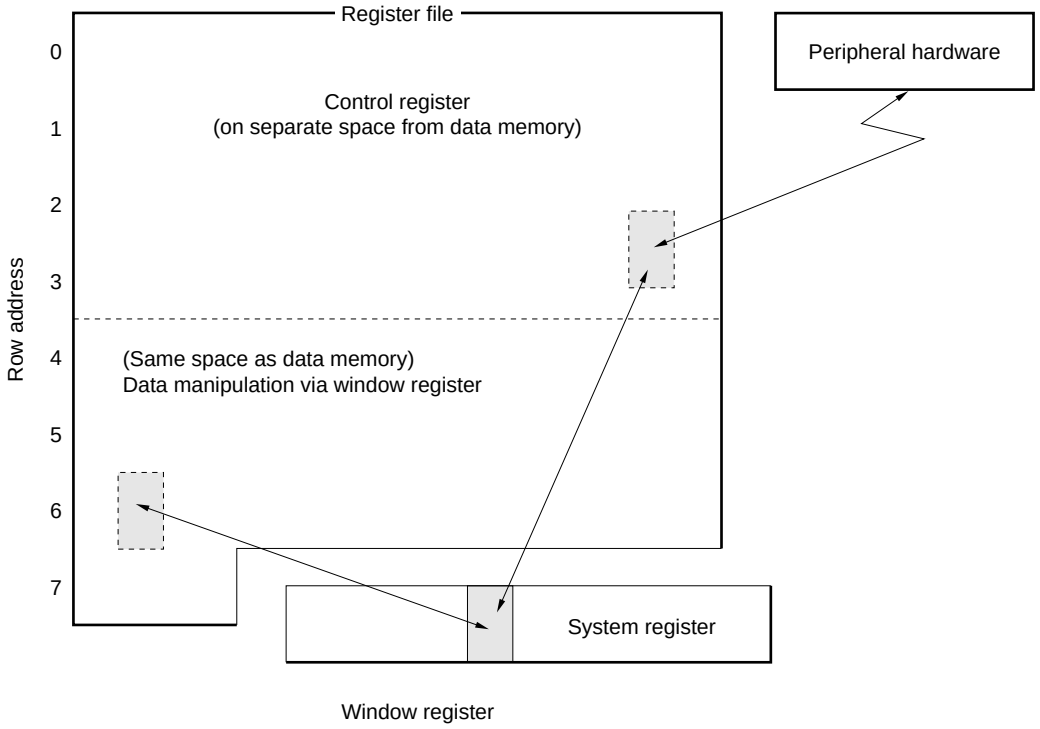
Figure 8-1 outlines the register file.

As shown in the figure, the rgister file consists of the control registers existing on a space different from the data memory, and a portion overlapping the data memory.

The control registers set conditions of the peripheral hardware units.

The data on the register file can be read or written via window register.

Figure 8-1. Outline of Register File



8.2 Configuration and Function of Register File

Figure 8-2 shows the configuration of the register file and the relationships between the register file and data memory.

The register file is assigned addresses in 4-bit units, like the data memory, and consists of a total of 128 nibbles with row addresses 0H through 7FH and column addresses 0H through 0FH.

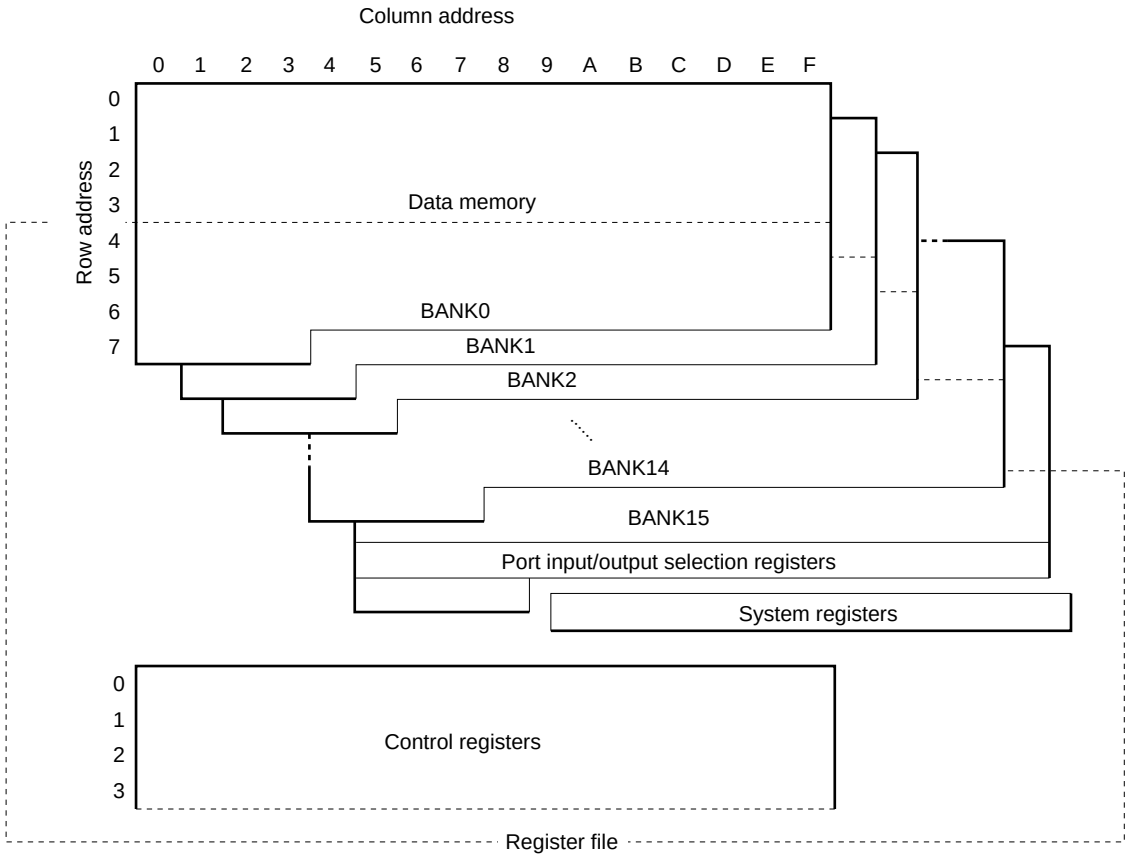
Addresses 00H through 3FH are control registers that sets the conditions of the peripheral hardware units.

Addresses 40H through 7FH overlap the data memory.

In other words, addresses 40H through 7FH of the register file are addresses 40H through 7FH of the currently-selected bank of data memory.

Because addresses 40H through 7FH of the register file overlap the same addresses of the data memory, these addresses of the register file can be manipulated in the same manner as the data memory, except that the addresses of the register file can also be manipulated by using register file manipulation instructions (“PEEK WR, rf” and “POKE rf, WR”). Note, however, that addresses 60H through 6FH of BANK15 are assigned port input/output selection registers (for details refer to **8.4 Port Input/Output Selection Registers**).

Figure 8-2. Configuration of Register File and Relationship with Data Memory



Remark The μPD17717 and 17718 do not have BANKs 10 through 14.

8.2.1 Register file manipulation instructions (“PEEK WR, rf”, “POKE rf, WR”)

Data is read from or written to the register file via the window register of the system registers, by using the following instructions.

(1) “PEEK WR, rf”

Reads data of the register file addressed by “rf” to the window register.

(2) “POKE rf, WR”

Writes the data of the window register to the register file addressed by “rf”.

8.3 Control Registers

Figure 8-3 shows the configuration of the control registers.

As shown in the figure, the control registers consist of a total of 64 nibbles (64 × 4 bits) of addresses 00H through 3FH of the register file.

Of these 64 nibbles, however, only 53 nibbles are actually used. The remaining 11 nibbles are unused registers and prohibited from being written or read.

Each control register has an attribute of 1 nibble that identifies four types of registers: read/write (R/W), read-only (R), write-only (W), and read-and-reset (R&Reset) registers.

Nothing is changed even if data is written to a read-only (R and R&Reset) register.

An “undefined” value is read if a write-only (W) register is read.

Among the 4-bit data in 1 nibble, the bit fixed to “0” is always “0” when it is read, and is also “0” when it is written.

The 11 nibbles of unused registers are undefined when their contents are read, and nothing changes even when they are written.

Table 8-1 lists the peripheral hardware control functions of the control registers.

[MEMO]

Figure 8-3. Configuration of Control Registers (1/2)

Column Address / Row Address	Item	0	1	2	3	4	5	6	7
0 (8)[Note]	Name		Stack pointer	Watchdog timer clock selection	Watchdog timer counter reset	Data buffer stack pointer	Stack overflow/ underflow reset selection	CE reset timer carry counter	MOVT bit selection
	Symbol		(SP3) (SP2) (SP1) (SP0)	0 0 WDTCK1 WDTCK0	WDTRES 0 0 0	0 0 (DBFSP1) (DBFSP0)	0 0 ISPRES ASPRES	CECNT3 CECNT2 CECNT1 CECNT0	0 0 MOVTSEL1 MOVTSEL0
	Read/ Write		R/W	R/W	W & Reset	R	R/W	R/W	R/W
1 (9)[Note]	Name	PLL mode selection	PLL reference frequency selection	PLL unlock FF	BEEP/general -purpose port pin function selection	BEEP clock selection		Watchdog timer/stack pointer reset status detection	Basic timer 0 carry
	Symbol	PLLSCNF 0 PLLMD1 PLLMD0	PLLRFCK3 PLLRFCK2 PLLRFCK1 PLLRFCK0	0 0 0 PLLUL	0 0 BEEP1SEL BEEP0SEL	BEEP1CK1 BEEP1CK0 BEEP0CK1 BEEP0CK0		0 0 0 WDTCY	0 0 0 BTM0CY
	Read/ Write	R/W	R/W	R&Reset	R/W	R/W		R&Reset	R&Reset
2 (A)[Note]	Name	FCG channel selection	IF counter gate status detection	IF counter mode selection	IF counter control	A/D converter channel selection	A/D converter mode selection	PWM clock selection	PWM/general-purpose port pin function selection
	Symbol	0 0 FCGCH1 FCGCH0	0 0 0 IFCGOSTT	IFCMD1 IFCMD0 IFCCK1 IFCCK0	0 0 IFCSTRT IFCRES	0 ADCCH2 ADCCH1 ADCCH0	0 ADCMD ADCSTT ADCCMP	0 PWMBIT 0 PWMCK	0 PWM2SEL PWM1SEL PWM0SEL
	Read/ Write	R/W	R	R/W	W	R/W	R/W R	R/W	R/W
3 (B)[Note]	Name					Serial interface 3 interrupt request	Serial interface 2 interrupt request	Timer 3 interrupt request	Timer 2 interrupt request
	Symbol					0 0 0 IRQSIO3	0 0 0 IRQSIO2	0 0 0 IRQTM3	0 0 0 IRQTM2
	Read/ Write					R/W	R/W	R/W	R/W

Note ( ) indicates an address that is used when the assembler is used.

Figure 8-3. Configuration of Control Register (2/2)

8	9	A	B	C	D	E	F
System register interrupt stack pointer		Serial I/O2 interrupt timing specification register 1	Serial I/O2 interrupt timing specification register 2	Serial I/O2 SBI register 1	Serial I/O2 SBI register 0	Serial I/O2 operation mode register 1	Serial I/O2 operation mode register 0
0 (SYSRSP2) (SYSRSP1) (SYSRSP0)		SIO2CLC SIO2WREL SIO2WAT1 SIO2WAT0	0 SIO2CLD SIO2SIC SIO2SVAM	SIO2CMDD SIO2RELD SIO2CMDT SIO2RELT	SIO2BSYE SIO2ACKD SIO2ACKE SIO2ACKT	SIO2WUP SIO2MD2 SIO2MD1 SIO2MD0	SIO2CSIE SIO2COI SIO2TCL1 SIO2TCL0
R		R/W	R R/W	R R/W	R/W R R/W	R/W	R/W R R/W
Basic timer 0 clock selection		Serial I/O3 operation mode register	Serial I/O3 asynchronous status register	Serial I/O3 asynchronous mode register 1	Serial I/O3 asynchronous mode register 0	Interrupt edge selection 1	Interrupt edge selection 2
0 0 BTM0CK1 BTM0CK0		SIO3CSIE SIO3HIZ SIO3TCL1 SIO3TCL0	0 SIO3PE SIO3FE SIO3OVE	SIO3PS1 SIO3PS0 SIO3CL SIO3SL	SIO3TXE SIO3RXE SIO3ISRM 0	IEG4 INT4SEL IEG3 INT3SEL	0 IEG2 IEG1 IEG0
R/W		R/W	R	R/W	R/W	R/W	R/W
Timer 3 control	Timer 2 counter clock selection	Timer 1 counter clock selection	Timer 0 counter clock selection	Timer 0 mode selection	Interrupt enable 1	Interrupt enable 2	Interrupt enable 3
TM3SEL 0 TM3EN TM3RES	TM2EN TM2RES TM2CK1 TM2CK0	TM1EN TM1RES TM1CK1 TM1CK0	TM0EN TM0RES TM0CK1 TM0CK0	TM0OVF TM0GCEG TM0GOEG TM0MD	IPSIO1 IPSIO0 IPTM3 IPTM2	IPTM3 IPTM2 IP4 IP3	IP2 IP1 IP0 IPCE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Timer 1 interrupt request	Timer 0 interrupt request	INT4 pin interrupt request	INT3 pin interrupt request	INT2 pin interrupt request	INT1 pin interrupt request	INT0 pin interrupt request	CE pin interrupt request
0 0 0 IRQTM1	0 0 0 IRQTM0	INT4 0 0 IRQ4	INT3 0 0 IRQ3	INT2 0 0 IRQ2	INT1 0 0 IRQ1	INT0 0 0 IRQ0	CE 0 CECNTSTT IRQCE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R R/W

Table 8-1. Peripheral Hardware Control Functions of Control Registers (1/8)

Peripheral Hardware	Control Register				Peripheral Hardware Control Function			At Reset			Clock Stop
	Name	Address	Read/ Write	b_3 b_2 b_1 b_0 Symbol	Function	Set value 0	Set value 1	Power-ON reset	WDT & SP reset	CE reset	
Stack	Stack pointer	01H	R/W	(SP3) (SP2) (SP1) (SP0)				F	F	F	Retained
	Interrupt stack pointer of system register	08H	R	0 (SYSRSP2) (SYSRSP1) (SYSRSP0)	Fixed to "0"			5	5	5	Retained
	Data buffer stack pointer	04H	R	0 0 (DBFSP1) (DBFSP0)	Fixed to "0" Detects nesting level of data buffer stack		0 0 Level 0 0 1 Level 1 1 0 Level 2 1 1 Level 3	0	0	0	Retained
	Stack overflow/ underflow reset selection	05H	R/W	0 0 ISPRES ASPRES	Fixed to "0" Selects interrupt stack overflow/underflow reset (can be set only once following power application) Selects address stack overflow/underflow reset (can be set only once following power application)	Reset prohibited	Reset valid	3	Retained	Retained	Retained
Watchdog timer	Watchdog timer clock selection	02H	R/W	0 0 WDTCK1 WDTCK0	Fixed to "0" Selects clock of watchdog timer (can be set only once following power application)		0 0 Not used 0 1 65536 instruction 1 0 Setting prohibited 1 1 131072 instruction	3	Retained	Retained	Retained
	Watchdog timer counter reset	03H	W & Reset	WDTRES 0 0 0	Resets watchdog timer counter Fixed to "0"	Invalid	Reset if written	Undefined	Undefined	Undefined	Undefined
	WDT&SP reset status detection	16H	R & Reset	0 0 0 WDTCY	Fixed to "0" Detects resetting of watchdog timer/stack pointer	No reset request	Reset request	0	1	Retained	Retained

Table 8-1. Peripheral Hardware Control Functions of Control Registers (2/8)

Peripheral Hardware	Control Register				Peripheral Hardware Control Function			At Reset			Clock Stop
	Name	Address	Read/ Write	b_3 b_2 b_1 b_0 Symbol	Function	Set value		Power-ON reset	WDT & SP reset	CE reset	
						0	1				
CE	CE reset timer carry counter	06H	R/W	CECNT3	Sets number of CE reset timer carry counts	0: Setting prohibited 1: 1 count 2: 2 counts 3: 3 counts 4: 4 counts 5: 5 counts 6: 6 counts 7: 7 counts 8: 8 counts 9: 9 counts A: 10 counts B: 11 counts C: 12 counts D: 13 counts E: 14 counts F: 15 coounts		1	Retained	Retained	1
				CECNT2							
				CECNT1							
				CECNT0							
	MOVT bit selection	07H	R/W	0	Fixed to "0"			0	0	0	Retained
				0							
				MOVTSEL1	Sets bit transferred by MOVT (transferred to DBF1, 0 during 8-bit transfer)	00 0 1 16-bit High-order Low-order transfer 8-bit transfer 8-bit transfer 01 1 0					
				MOVTSEL0							
Serial interface	Serial I/O2 interrupt timing specification register 1	0AH	R/W	SIO2CLC	Controls P0A2/SCL pin level (I^2C bus mode)	Not affected	High impedance	0	0	0	0
				SIO2WREL	Releases wait	Wait released status	Releases wait				
				SIO2WAT1	Controls wait and interrupt request issuance (10 and 11 are set in I^2C bus mode)	0, 1: Issues at rising of 8th clock 2: Issues at rising of 8th clock and waits 3: Issues at rising of 9th clock and waits					
				SIO2WAT0							
	Serial I/O2 interrupt timing specification register 0	0BH	R	0	Fixed to "0"			0	0	0	0
				SIO2CLD	Detects P0A2/SCL pin level	Low level	High level				
			R/W	SIO2SIC	Selects interrupt source	Only on completion of transmission	On completion of transmission or on detection of bus release signal				
				SIO2SVAM	Selects bit of SIO2SVA used	Bits 0-7	Bits 1-7				
	Serial I/O2 SBI register 1	0CH	R	SIO2CMDD	Detects command signal	Does not detect	Detects	0	0	0	0
				SIO2RELD	Detects bus release signal	Does not detect	Detects				
			R/W	SIO2CMDT	Controls trigger output of command signal	Automatically cleared after setting flag	Clears SO2 latch after clearing flag				
				SIO2RELT	Controls trigger output of bus release signal		Sets SO2 latch after setting flag				
	Serial I/O2 SBI register 0	0DH	R/W	SIO2BSYE	Controls sync busy signal output	Disables output	Enables output	0	0	0	0
			R	SIO2ACKD	Detects acknowledge signal	Does not detect	Detects				
			R/W	SIO2ACKE	Controls acknowledge signal output	Disables automatic output	Enables automatic output				
				SIO2ACKT	Controls trigger output of acknowledge signal	Does not output acknowledge	Output immediately after set				

Table 8-1. Peripheral Hardware Control Functions of Control Registers (3/8)

Peripheral Hardware	Control Register				Peripheral Hardware Control Function			At Reset			Clock
	Name	Address	Read/ Write	b_3 b_2 b_1 b_0 Symbol	Function	Set value		Power-ON reset	WDT & SP reset	CE reset	Stop
						0	1				
Serial interface	Serial I/O2 operation mode register 1	0EH	R/W	SIO2WUP	Controls wake-up function	Disables	Enables	0	0	0	0
				SIO2MD2 SIO2MD1	Sets operation mode of serial interface 2	0: 3-wire serial I/O 1: SBI (SB1 pin) 2: SBI (SB0 pin) 3: 2-wire serial I/O or I^2C bus					
				SIO2MD0	Sets direction of shift clock	Slave (external clock)	Master (internal clock)				
	Serial I/O2 operation mode register 0	0FH	R/W	SIO2CSIE	Controls operation of serial interface 2	Stops operation	Enables operation	0	0	0	0
			R	SIO2COI	Detects coincidence signal from address comparator	Does not coincide	Coincide				
			R/W	SIO2TCL1 SIO2TCL0	Sets frequency of internal shift clock	0 0: 93.7 kHz 0 1: 375 kHz	1 0: 281.25 kHz 1 1: 46.875 kHz				
	Serial I/O3 operation mode register	1AH	R/W	SIO3CSIE	Controls operation of serial interface 3	Stops operation	Enables operation	0	0	0	0
				SIO3HIZ	Sets status of SO3/P0B1 pin	General-purpose I/O port	Serial data output				
				SIO3TCL1 SIO3TCL0	Selects I/O clock of 3-wire serial I/O	0 0: External clock 0 1: 187.5 kHz	1 0: 375 kHz 1 1: 46.875 kHz				
	Serial I/O3 asynchronous status register	1BH	R	0	Fixed to "0"			0	0	0	0
				SIO3PE	Contents of parity error	Error does not occur	Parity does not coincide				
				SIO3FE	Contents of framing error	Error does not occur	Stop bit not detected				
				SIO3OVE	Contents of overrun error	Error does not occur	Data duplication				
	Serial I/O3 asynchronous mode register 1	1CH	R/W	SIO3PS1 SIO3PS0	Sets parity bit of UART	0: No parity 1: Appends parity during transmission, no parity error during reception 2: Odd parity 3: Even parity		0	0	0	0
				SIO3CL	Sets character length of UART	7 bits	8 bits				
				SIO3SL	Sets number of stop bits for UART transmission data	1	2				
	Serial I/O3 asynchronous mode register 0	1DH	R/W	SIO3TXE SIO3RXE	Sets operation of UART	0 0: Stops operation 0 1: Reception 1 0: Transmission 1 1: Transmission/ reception		0	0	0	0
				SIO3ISRM	Sets reception completion interrupt on occurrence of error	Enables interrupt	Disables interrupt				
				0	Fixed to "0"						

Table 8-1. Peripheral Hardware Control Functions of Control Registers (4/8)

Peripheral Hardware	Control Register				Peripheral Hardware Control Function			At Reset			Clock Stop
	Name	Address	Read/ Write	b_3 b_2 b_1 b_0 Symbol	Function	Set value 0	Set value 1	Power-ON reset	WDT & SP reset	CE reset	
PLL frequency synthesizer	PLL mode selection	10H	R/W	PLLSCNF	Sets low-order bits of swallow counter	Lowest bit is 0	Lowest bit is 1	U	U	R	R
				0	Fixed to "0"			0	0	0	0
				PLLMD1 PLLMD0	Sets division mode of PLL	0 0 (Disabled)	0 1 (MF) 1 0 (VHF) 1 1 (HF)				
	PLL reference frequency selection	11H	R/W	PLLRFCK3 PLLRFCK2 PLLRFCK1 PLLRFCK0	Sets reference frequency of PLL	0: 1.25 kHz 3: 10 kHz 6: 25 kHz 9: 9 kHz B: Setting prohibited C: 1 kHz E: Setting prohibited F: PLL disabled	1: 2.5 kHz 4: 6.25 kHz 7: 50 kHz A: 18 kHz D: 20 kHz	2: 5 kHz 5: 12.5 kHz 8: 3 kHz	F	F	F	F
	PLL unlcok FF	12H	R & Reset	0 0 0	Fixed to "0"			Undefined	Undefined	Retained	Retained
				PLLUL	Detects status of unlock FF	Locked	Unclocked				
BEEP	BEEP/general-purpose port pin function selection	13H	R/W	0 0	Fixed to "0"			0	0	0	0
				BEEP1SEL	Selects function of P1D1/BEEP1 pin	General-purpose	BEEP				
				BEEP0SEL	Selects function of P1D0/BEEP0 pin	I/O port					
	BEEP clock selection	14H	R/W	BEEP1CK1 BEEP1CK0	Sets output frequency of BEEP1	0 0 (4 kHz) 0 1 (3 kHz)	1 0 (200 Hz) 1 1 (67 Hz)		0	0	0	0
				BEEP0CK1 BEEP0CK0	Sets output frequency of BEEP0	0 0 (1 kHz) 0 1 (3 kHz)	1 0 (4 kHz) 1 1 (6.7 kHz)				
Timer	Basic timer 0 carry	17H	R & Reset	0 0 0	Fixed to "0"			0	Retained	1	Retained
				BTM0CY	Detects basic timer 0 carry FF	FF reset	FF set				
	Basic timer 0 clock selection	18H	R/W	0 0	Fixed to "0"			0	0	Retained	Retained
				BTM0CK1 BTM0CK0	Selects clock of basic timer 0	0 0 (10 Hz) 0 1 (20 Hz)	1 0 (50 Hz) 1 1 (100 Hz)				
	Timer 3 control	28H	R/W	TM3SEL	Selects timer 3 and D/A converter	D/A converter	Timer 3	0	0	Retained	0
				0	Fixed to "0"						
				TM3EN	Starts or stops timer 3 counter	Stops	Starts				
				TM3RES	Resets timer 3 counter	Not affected	Reset				

U: Undefined R: Retained

Table 8-1. Peripheral Hardware Control Functions of Control Registers (5/8)

Peripheral Hardware	Control Register				Peripheral Hardware Control Function			At Reset			Clock Stop
	Name	Address	Read/ Write	b_3 b_2 b_1 b_0 Symbol	Function	Set value 0	Set value 1	Power-ON reset	WDT & SP reset	CE reset	
Timer	Timer 2 counter clock selection	29H	R/W	TM2EN	Starts or stops timer 2 counter	Stops	Starts	0	0	Retained	0
				TM2RES	Resets timer 2 counter	Not affected	Reset				
				TM2CK1	Sets basic clock of timer 2 counter	0 0 100 kHz 10 kHz	1 1 2 kHz 1 kHz				
				TM2CK0		0 1	0 1				
	Timer 1 counter clock selection	2AH	R/W	TM1EN	Starts or stops timer 1 counter	Stops	Starts	0	0	Retained	0
				TM1RES	Resets timer 1 counter	Not affected	Reset				
				TM1CK1	Sets basic clock of timer 1 counter	0 0 100 kHz 10 kHz	1 1 2 kHz 1 kHz				
				TM1CK0		0 1	0 1				
	Timer 0 counter clock selection	2BH	R/W	TM0EN	Starts or stops timer 0 counter	Stops	Starts	0	0	Retained	0
				TM0RES	Resets timer 0 counter	Not affected	Reset				
				TM0CK1	Sets basic clock of timer 0 counter	0 0 100 kHz 10 kHz	1 1 2 kHz 1 kHz				
				TM0CK0		0 1	0 1				
	Timer 0 mode selection	2CH	R/W	TM0OVF	Detects timer 0 overflow	No overflow	Overflow	0	0	Retained	0
				TM0GCEG	Sets edge of gate close input signal	Rising edge	Falling edge				
				TM0GOEG	Sets edge of gate open input signal						
				TM0MD	Selects modulo counter/gate counter of timer 0	Modulo counter	Gate counter				
Interrupt	Interrupt edge selection 1	1EH	R/W	IEG4	Sets interrupt issuance edge (INT4 pin)	Rising edge	Falling edge	0	0	Retained	Retained
				INT4SEL	Sets interrupt request flag of P1A3/INT4 pin	Enables setting of flag	Disables setting of flag				
				IEG3	Sets interrupt issuance edge (INT3 pin)	Rising edge	Falling edge				
				INT3SEL	Sets interrupt request flag of P1A2/INT3 pin	Enables setting of flag	Disables setting of flag				
	Interrupt edge selection 2	1FH	R/W	0	Fixed to “0”			0	0	Retained	Retained
				IEG2	Sets interrupt issuance edge (INT2 pin)	Rising edge	Falling edge				
				IEG1	Sets interrupt issuance edge (INT1 pin)						
				IEG0	Sets interrupt issuance edge (INT0 pin)						

Table 8-1. Peripheral Hardware Control Functions of Control Registers (6/8)

Peripheral Hardware	Control Register				Peripheral Hardware Control Function			At Reset			Clock Stop
	Name	Address	Read/ Write	b_3 b_2 b_1 b_0 Symbol	Function	Set value		Power-ON reset	WDT & SP reset	CE reset	
						0	1				
Interrupt	Interrupt enable 1	2DH	R/W	IPSIO3	Enables serial interface 3 interrupt	Disables interrupt	Enables interrupt	0	0	Retained	Retained
				IPSIO2	Enables serial interface 2 interrupt						
				IPTM3	Enables timer 3 interrupt						
				IPTM2	Enables timer 2 interrupt						
	Interrupt enable 2	2EH	R/W	IPTM1	Enables timer 1 interrupt	Disables interrupt	Enables interrupt	0	0	Retained	Retained
				IPTM0	Enables timer 0 interrupt						
				IP4	Enables INT4 pin interrupt						
				IP3	Enables INT3 pin interrupt						
	Interrupt enable 3	2FH	R/W	IP2	Enables INT2 pin interrupt	Disables interrupt	Enables interrupt	0	0	Retained	Retained
				IP1	Enables INT1 pin interrupt						
				IP0	Enables INT0 pin interrupt						
				IPCE	Enables CE pin interrupt						
	Serial interface 3 interrupt request	34H	R/W	0	Fixed to “0”			0	0	Retained	Retained
				0							
				0							
				IRQSIO3	Detects serial interface 3 interrupt request	No interrupt request	Interrupt request				
	Serial interface 2 interrupt request	35H	R/W	0	Fixed to “0”			0	0	Retained	Retained
				0							
				0							
				IRQSIO2	Detects serial interface 2 interrupt request	No interrupt request	Interrupt request				
	Timer 3 interrupt request	36H	R/W	0	Fixed to “0”			0	0	Retained	Retained
				0							
				0							
				IRQTM3	Detects timer 3 interrupt request	No interrupt request	Interrupt request				
	Timer 2 interrupt request	37H	R/W	0	Fixed to “0”			0	0	Retained	Retained
				0							
				0							
				IRQTM2	Detects timer 2 interrupt request	No interrupt request	Interrupt request				
	Timer 1 interrupt request	38H	R/W	0	Fixed to “0”			0	0	Retained	Retained
				0							
				0							
				IRQTM1	Detects timer 1 interrupt request	No interrupt request	Interrupt request				

Table 8-1. Peripheral Hardware Control Functions of Control Registers (7/8)

Peripheral Hardware	Control Register				Peripheral Hardware Control Function			At Reset			Clock
	Name	Address	Read/ Write	b_3 b_2 b_1 b_0 Symbol	Function	Set value 0	Set value 1	Power-ON reset	WDT & SP reset	CE reset	Stop
Interrupt	Timer 0 interrupt request	39H	R/W	0	Fixed to "0"			0	0	Retained	Retained
				0							
				0							
				IRQTM0	Detects timer 0 interrupt request	No interrupt request	Interrupt request				
	INT4 pin interrupt request	3AH	R/W	INT4	Detects INT4 pin status	Low level	High level	U	U	U	U
				0	Fixed to "0"			0	0	Retained	Retained
				0							
				IRQ4	Detects INT4 pin interrupt request	No interrupt request	Interrupt request				
	INT3 pin interrupt request	3BH	R/W	INT3	Detects INT3 pin status	Low level	High level	U	U	U	U
				0	Fixed to "0"			0	0	Retained	Retained
				0							
				IRQ3	Detects INT3 pin interrupt request	No interrupt request	Interrupt request				
	INT2 pin interrupt request	3CH	R/W	INT2	Detects INT2 pin status	Low level	High level	U	U	U	U
				0	Fixed to "0"			0	0	Retained	Retained
				0							
				IRQ2	Detects INT2 pin interrupt request	No interrupt request	Interrupt request				
	INT1 pin interrupt request	3DH	R/W	INT1	Detects INT1 pin status	Low level	High level	U	U	U	U
				0	Fixed to "0"			0	0	Retained	Retained
				0							
				IRQ1	Detects INT1 pin interrupt request	No interrupt request	Interrupt request				
	INT0 pin interrupt request	3EH	R/W	INT0	Detects INT0 pin status	Low level	High level	U	U	U	U
				0	Fixed to "0"			0	0	Retained	Retained
				0							
				IRQ0	Detects INT0 pin interrupt request	No interrupt request	Interrupt request				
	CE pin interrupt request	3FH	R	CE	Detects CE pin status	Low level	High level	U	U	U	U
				0	Fixed to "0"			0	0	0	0
				CECNTSTT	Detects CE reset counter status	Stops	Operates				
			R/W	IRQCE	Detects CE pin interrupt request	No interrupt request	Interrupt request	0	0	R	R
IF counter	FCG channel selection	20H	R/W	0	Fixed to "0"			0	0	0	0
				0							
				FCGCH1	Sets pin to be used as FCG	0 FCG not used	0 FCG0 pin	1 FCG1 pin	1 Setting prohibited		
				FCGCH0		0	1	0	1		
	IF counter gate status detection	21H	R	0	Fixed to "0"			0	0	0	0
				0							
				0							
				IFCGOSTT	Detects IF counter gate status	Closed	Open				

U: Undefined

Table 8-1. Peripheral Hardware Control Functions of Control Registers (8/8)

Peripheral Hardware	Control Register				Peripheral Hardware Control Function			At Reset			Clock Stop
	Name	Address	Read/ Write	b_3 b_2 b_1 b_0 Symbol	Function	Set value		Power-ON reset	WDT & SP reset	CE reset	
						0	1				
IF counter	IF counter mode selection	22H	R/W	IFCMD1	Sets IF counter mode	0 0 1 1 FCG AMIFC FMIFC AMIFC2		0	0	0	0
				IFCMD0		0 1 0 1					
				IFCCK1	Sets IF counter gate time and FCG count frequency	0 0 1 1 1ms, 4 ms, 8 ms, Open, 1 kHz 100 kHz 900 kHz Setting prohibited					
				IFCCK0		0 1 0 1					
	IF counter control	23H	W	0	Fixed to “0”			0	0	0	0
				0							
				IFCSTRT	Starts or stops IF counter	Nothing affected	Starts counter				
				IFCRES	Resets IF counter data	Nothing affected	Starts counter				
A/D converter	A/D converter channel selection	24H	R/W	0	Fixed to “0”			0	0	Retained	Retained
				ADCCH2	Selects pin used for A/D converter	0: A/D converter not used 1: P0D0/AD0 pin 2: P0D1/AD1pin 3: P0D2/AD2 pin 4: P0D3/AD3 pin 5: P1C2/AD4 pin 6: P1C3/AD5 pin 7: Setting prohibited					
				ADCCH1							
				ADCCH0							
	A/D converter mode selection	25H	R/W	0	Fixed to “0”			0	0	0	0
				ADCMD	Selects comparison mode of A/D converter	Software mode	Hardware mode			Retained	Retained
			R	ADCSTT	Detects operating status of A/D converter	Conversion ends	Converting			0	0
				ADCCMP	Detects comparison result of A/D converter	$V_{ADCREF} > V_{ADCIN}$	$V_{ADCREF} < V_{ADCIN}$			0	Retained
D/A converter	PWM clock selection	26H	R/W	0	Fixed to “0”			0	0	Retained	0
				PWMBIT	Selects number of bits of PWM counter	8 bits	9 bits				
				0	Fixed to “0”						
				PWMCK	Selects output clock of timer 3	4.4 kHz (8)/ 2.2 kHz (9)	440 Hz (8)/ 220 Hz (9)				
	PWM/general-purpose port pin function selection	27H	R/W	0	Fixed to “0”			0	0	Retained	0
				PWM2SEL	Selects function of P1B2/PWM2 pin	General-purpose output port	D/A converter				
				PWM1SEL	Selects function of P1B1/PWM1 pin						
				PWM0SEL	Selects function of P1B0/PWM0 pin						

8.4 Port Input/Output Selection Registers

Figure 8-4 shows the configuration of the port input/output selection registers.

As shown in this figure, the port input/output select registers consist of a total of 16 nibbles (16 × 4 bits) at addresses 60H through 6FH of BANK 15 of the data memory.

Table 8-2 lists the control functions of the port input/output selection registers.

[MEMO]

Figure 8-4. Configuration of Port Input/Output Selection Registers (1/2)

(BANK15) Column Address Row Address Item		0	1	2	3	4	5	6	7
6	Name							Port 0D pull-down resistor selection	Group I/O selection
	Symbol							P0DPLD3 P0DPLD2 P0DPLD1 P0DPLD0	P3DGIO P3CGIO P3BGIO P3AGIO
	Read/ Write							R/W	R/W

Figure 8-4. Configuration of Port Input/Output Selection Registers (2/2)

8					9					A					B					C					D					E					F				
Port 2D bit I/O selection					Port 2C bit I/O selection					Port 2B bit I/O selection					Port 2A bit I/O selection					Port 1D bit I/O selection					Port 0C bit I/O selection					Port 0B bit I/O selection					Port 0A bit I/O selection				
0	P2DBIO2	P2DBIO1	P2DBIO0		P2CBIO3	P2CBIO2	P2CBIO1	P2CBIO0		P2BBIO3	P2BBIO2	P2BBIO1	P2BBIO0		0	P2ABIO2	P2ABIO1	P2ABIO0		P1DBIO3	P1DBIO2	P1DBIO1	P1DBIO0		P0CBIO3	P0CBIO2	P0CBIO1	P0CBIO0		P0BBIO3	P0BBIO2	P0BBIO1	P0BBIO0		P0ABIO3	P0ABIO2	P0ABIO1	P0ABIO0	
R/W					R/W					R/W					R/W					R/W					R/W					R/W					R/W				

Table 8-2. Control Functions of Port Input/Output Selection Registers (1/2)

Peripheral Hardware	Port Input/Output Selection Register				Control Function			At Reset			Clock Stop
	Name	Address (BANK15)	Read/ Write	b_3 b_2 b_1 b_0 Symbol	Function	Set value 0	Set value 1	Power-ON reset	WDT & SP reset	CE reset	
Input/ output port	Port 0D pull-down resistor selection	66H	R/W	P0DPLD3	Selects pull-down resistor of P0D3 pin	Pull-down resistor used	Pull-down resistor not used	0	0	Retained	Retained
				P0DPLD2	Selects pull-down resistor of P0D2 pin						
				P0DPLD1	Selects pull-down resistor of P0D1 pin						
				P0DPLD0	Selects pull-down resistor of P0D0 pin						
	Group I/O selection	67H	R/W	P3DGIO	Selects input/output of port 3D	Input	Output	0	0	Retained	Retained
				P3CGIO	Selects input/output of port 3C						
				P3BGIO	Selects input/output of port 3B						
				P3AGIO	Selects input/output of port 3A						
	Port 2D bit I/O selection	68H	R/W	0	Fixed to "0"			0	0	Retained	Retained
				P2DBIO2	Selects input/output of port P2D2	Input	Output				
				P2DBIO1	Selects input/output of port P2D1						
				P2DBIO0	Selects input/output of port P2D0						
	Port 2C bit I/O selection	69H	R/W	P2CBIO3	Selects input/output of port P2C3	Input	Output	0	0	Retained	Retained
				P2CBIO2	Selects input/output of port P2C2						
				P2CBIO1	Selects input/output of port P2C1						
				P2CBIO0	Selects input/output of port P2C0						
	Port 2B bit I/O selection	6AH	R/W	P2BBIO3	Selects input/output of port P2B3	Input	Output	0	0	Retained	Retained
				P2BBIO2	Selects input/output of port P2B2						
				P2BBIO1	Selects input/output of port P2B1						
				P2BBIO0	Selects input/output of port P2B0						
	Port 2A bit I/O selection	6BH	R/W	0	Fixed to "0"			0	0	Retained	Retained
				P2ABIO2	Selects input/output of port P2A2	Input	Output				
				P2ABIO1	Selects input/output of port P2A1						
				P2ABIO0	Selects input/output of port P2A0						
	Port 1D bit I/O selection	6CH	R/W	P1DBIO3	Selects input/output of port P1D3	Input	Output	0	0	Retained	Retained
				P1DBIO2	Selects input/output of port P1D2						
				P1DBIO1	Selects input/output of port P1D1						
				P1DBIO0	Selects input/output of port P1D0						
	Port 0C bit I/O selection	6DH	R/W	P0CBIO3	Selects input/output of port P0C3	Input	Output	0	0	Retained	Retained
				P0CBIO2	Selects input/output of port P0C2						
				P0CBIO1	Selects input/output of port P0C1						
				P0CBIO0	Selects input/output of port P0C0						
	Port 0B bit I/O selection	6EH	R/W	P0BBIO3	Selects input/output of port P0B3	Input	Output	0	0	Retained	Retained
				P0BBIO2	Selects input/output of port P0B2						
				P0BBIO1	Selects input/output of port P0B1						
				P0BBIO0	Selects input/output of port P0B0						

Table 8-2. Control Functions of Port Input/Output Selection Registers (2/2)

Peripheral Hardware	Port Input/Output Selection Register				Control Function			At Reset			Clock Stop
	Name	Address (BANK15)	Read/ Write	b_3 b_2 b_1 b_0 Symbol	Function	Set value		Power-ON reset	WDT & SP reset	CE reset	
						0	1				
Input/ output port	Port 0A bit I/O selection	6FH	R/W	P0ABIO3	Selects input/output of port P0A3	Input	Output	0	0	Retained	Retained
				P0ABIO2	Selects input/output of port P0A2						
				P0ABIO1	Selects input/output of port P0A1						
				P0ABIO0	Selects input/output of port P0A0						

8.5 Cautions on Using Register File

Keep in mind the following points (1) through (3) when using the write-only (W), read-only (R), and unused registers of the control registers (addresses 00H through 3FH of the register file).

(1) An “undefined value” is read if a write-only register is read.
(2) Nothing is affected even if a read-only register is written.
(3) An “undefined value” is read if an unused register is read. Nor is anything affected if this register is written.

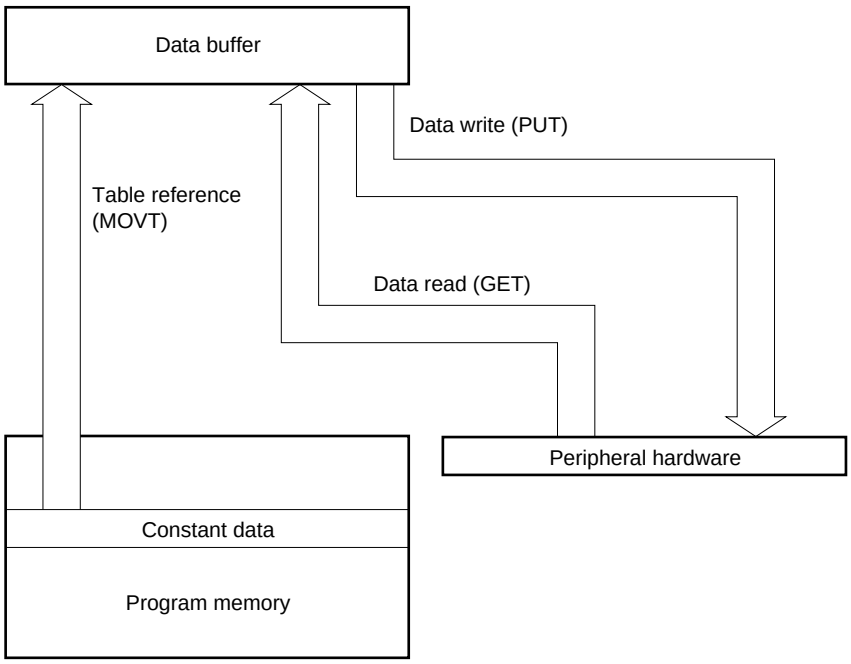
9. DATA BUFFER (DBF)

9.1 Outline of Data Buffer

Figure 9-1 outlines the data buffer.
The data buffer is located on the data memory and has the following two functions.

- Reads constant data on the program memory (table reference)
- Transfers data with the peripheral hardware units

Figure 9-1. Outline of Data Buffr



9.2 Data Buffer

9.2.1 Configuration of data buffer

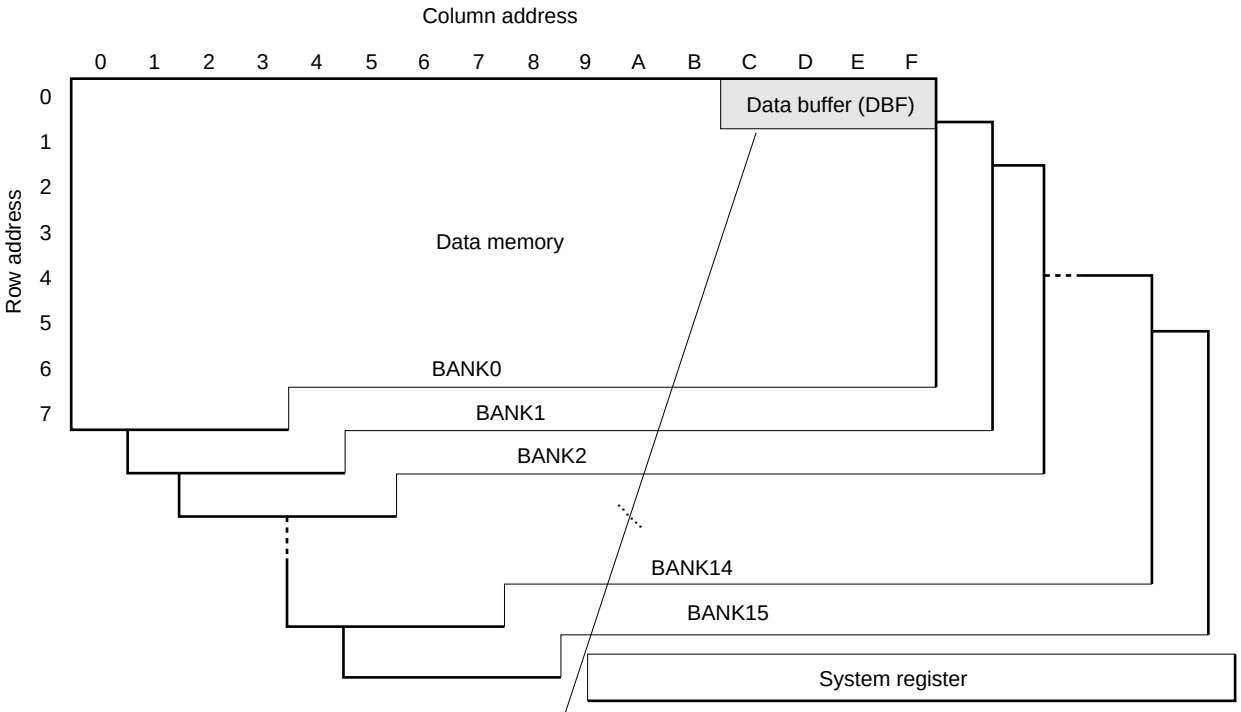
Figure 9-2 shows the configuration of the data buffer.

As shown in the figure, the data buffer consists of a total of 16 bits of addresses 0CH through 0FH of BANK 0 on the data memory.

The 16-bit data is configured with bit 3 of address 0CH as the MSB and bit 0 of address 0FH as the LSB.

Because the data buffer is located on the data memory, it can be manipulated by all data memory manipulation instructions.

Figure 9-2. Configuration of Data Buffer



Remark The μPD17717 and 17718 do not have BANKs 10 through 14.

Data memory	Address	0CH				0DH				0EH				0FH			
	Bit	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0	b_3	b_2	b_1	b_0
Data buffer	Bit	b_{15}	b_{14}	b_{13}	b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
	Signal	DBF3				DBF2				DBF1				DBF0			
	Data	MSB				Data								LSB			

9.2.2 Table reference instruction (“MOVT DBF, @AR”)

This instruction moves the contents of the program memory addressed by the contents of the address register to the data buffer.

The number of bits transferred by the table reference instruction can be specified by MOVT selection register (address 07H) of the control registers.

When 8-bit data is transferred, it is read to DBF1 and 0.

When the table reference instruction is used, one stack level is used.

All the addresses of the program memory can be referenced by the table reference instruction.

9.2.3 Peripheral hardware control instructions (“PUT” and “GET”)

The operations of the “PUT” and “GET” instructions are as follows:

(1) GET DBF, p

Reads the data of a peripheral register addressed by “p” to the data buffer.

(2) PUT p, DBF

Sets the data of the data buffer to a peripheral register addressed by “p”.

9.3 Relationships between Peripheral Hardware and Data Buffer

Table 9-1 shows the relationships between the peripheral hardware and the data buffer.

Table 9-1. Relationships between Peripheral Hardware and Data Buffer (1/2)

Peripheral Hardware		Peripheral Register Transferring Data with Data Buffer					
		Name	Symbol	Peripheral address	Execution of PUT/GET instruction	I/O bit	Actual bit
A/D converter		A/D converter reference voltage setting register	ADCR	02H	PUT/GET	8	8
Serial interface	Serial interface 2	Presettable shift register 2	SIO2SFR	03H	PUT/GET	8	8
		Serial I/O2 slave address register	SIO2SVA	04H	PUT/GET	8	8
	Serial interface 3	Serial I/O3 transmission register	SIO3TXS	05H	PUT	8	8
		Serial I/O3 reception buffer register	SIO3RXB		GET	8	8
Timer 0		Timer 0 modulo register	TM0M	1AH	PUT/GET	8	8
		Timer 0 counter	TM0C	1BH	GET	8	8
Timer 1		Timer 1 modulo register	TM1M	1CH	PUT/GET	8	8
		Timer 1 counter	TM1C	1DH	GET	8	8
Timer 2		Timer 2 modulo register	TM2M	1EH	PUT/GET	8	8
		Timer 2 counter	TM2C	1FH	GET	8	8
Address register		Address register	AR	40H	PUT/GET	16	16
Data buffer stack		DBF stack	DBFSTK	41H	PUT/GET	16	16
PLL frequency synthesizer[Note]		PLL data register	PLLR	42H	PUT/GET	16	16
Frequency counter		IFC data register	IFC	43H	GET	16	16
D/A converter (PWM output)	P1B0/PWM0 pin	PWM data register 0	PWMR0	44H	PUT/GET	16	9
	P1B1/PWM1 pin	PWM data register 1	PWMR1	45H			
	P1B2/PWM2 pin	PWM data register 2	PWMR2	46H	PUT/GET	16	9
Timer 3		Timer 3 modulo register	TM3M				8

Note The programmable counter of the PLL frequency synthesizer is configured of 17 bits, of which the high-order 16 bits indicate the PLL data register (PLLR) and the low-order bits are allocated to the PLLSCNF flag (the third bit of address 10H).

For details, refer to **17. PLL FREQUENCY SYNTHESIZER**.

Table 9-1. Relationships between Peripheral Hardware and Data Buffer (2/2)

At Reset			Clock Stop	Function
Power-ON reset	WDT&SP reset	CE reset		
0	0	0[Note]	0[Note]	Sets compare voltage V_{ADCREF} of A/D converter
Undefined	Undefined	Undefined	Undefined	Sets serial-out data and reads serial-in data
Undefined	Undefined	Undefined	Undefined	Sets slave address value of slave device
FF	FF	FF	FF	Sets transmission data in 3-wire serial I/O and UART modes
FF	FF	FF	FF	Stores receive data in 3-wire serial I/O and UART modes
FF	FF	Retained	FF	Sets modulo register value of timer 0
0	0	Retained	0	Reads count value of timer 0 counter
FF	FF	Retained	FF	Sets modulo register value of timer 1
0	0	Retained	0	Reads count value of timer 1 counter
FF	FF	Retained	FF	Sets modulo register value of timer 2
0	0	Retained	0	Reads count value of timer 2 counter
0	0	0	Retained	Transfers data with address register
Undefined	Undefined	Retained	Retained	Saves data of data buffer
Undefined	Undefined	Retained	Retained	Sets division value (N value) of PLL
0	0	0	0	Reads count value of frequency counter
1FF	1FF	Retained	1FF	Sets duty of output signal of D/A converter
				Sets duty of output signal of D/A converter (multiplexed with modulo register of timer 3)
				Sets modulo register value of timer 3

Note Value in hardare mode. “Retained” in software mode.

9.4 Cautions on Using Data Buffer

Keep the following points in mind concerning the unused peripheral addresses, write-only peripheral register (PUT only), and read-only peripheral register (GET only) when transferring data with the peripheral hardware via data buffer.

- An “undefined value” is read if a write-only register is read.
- Nothing is affected even if a read-only register is written.
- An “undefined value” is read if an unused address is read. Nor is anything affected if this address is written.

10. DATA BUFFER STACK

10.1 Outline of Data Buffer Stack

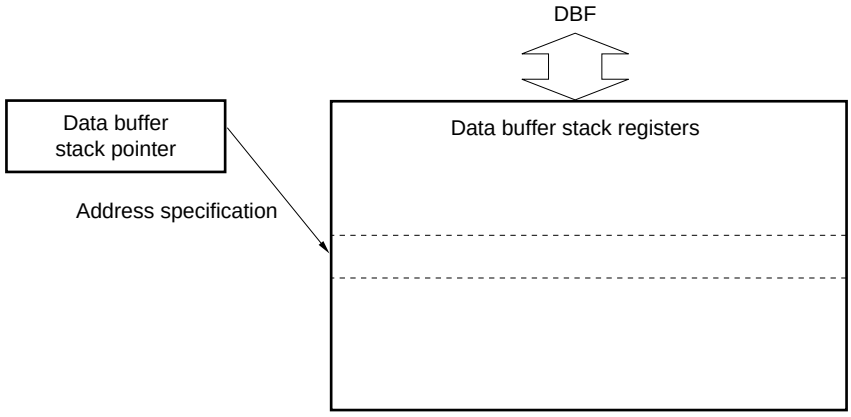
Figure 10-1 outlines the data buffer stack.

As shown in the figure, the data buffer stack consists of a data buffer stack pointer and data buffer stack registers.

The data buffer stack saves or restores the contents of the data buffer when the “PUT” or “GET” instruction is executed.

Therefore, the contents of the data buffer can be saved by one instruction when an interrupt is accepted.

Figure 10-1. Outline of Data Buffer Stack



10.2 Data Buffer Stack Register

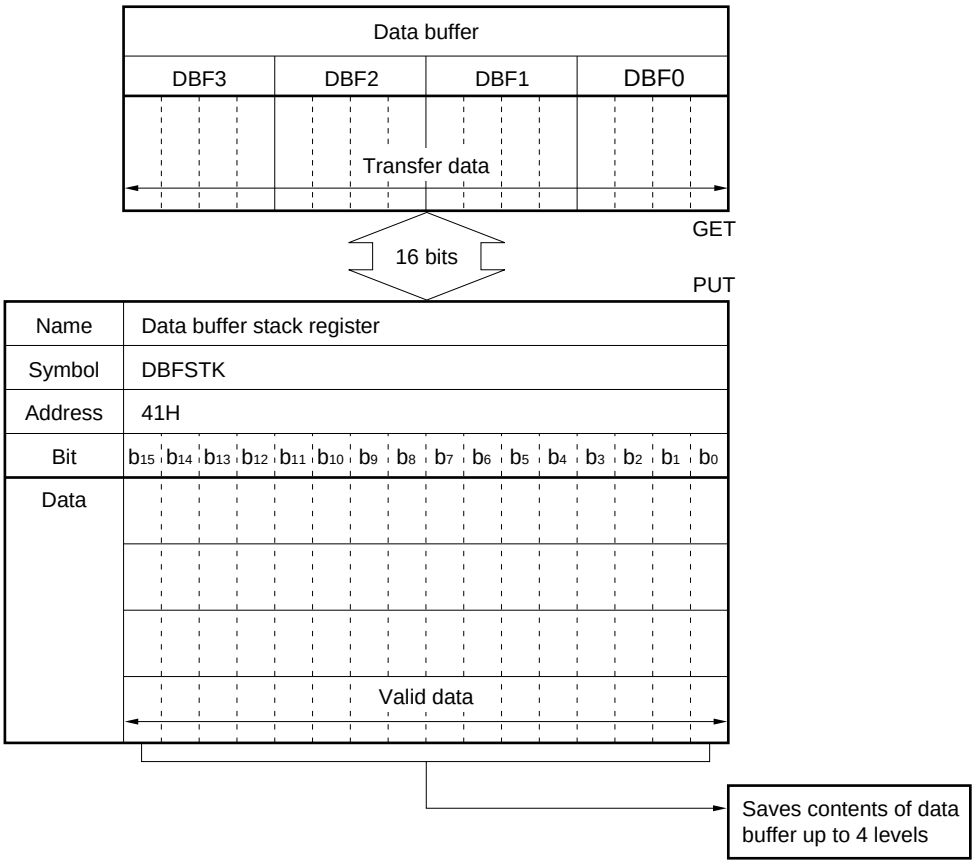
Figure 10-2 shows the configuration of the data buffer stack registers.

As shown in the figure, the data buffer stack registers consist of four 16-bit registers.

The contents of the data buffer are saved by executing the “PUT” instruction, and the saved data is restored by executing the “GET” instruction.

The data buffer contents can be successively saved up to 4 levels.

Figure 10-2. Configuration of Data Buffer Stack Register



10.3 Data Buffer Stack Pointer

The data buffer stack pointer detects the multiplexing level of the data buffer stack registers.

When the “PUT” instruction is executed to the data buffer stack, the value of the data buffer stack pointer is incremented by one; when the “GET” instruction is executed, the value of the pointer is decremented by one.

The data buffer stack pointer can be only read and cannot be written.

The configuration and function of the data buffer stack pointer are illustrated below.

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Data buffer stack pointer	0	0	DBFSP1	DBFSP0	04H	R

Detects multiplexing level of data buffer stack	
0 0	Level 0
0 1	Level 1
1 0	Level 2
1 1	Level 3

Fixed to “0”

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset			0	0
	CE reset			0	0
Clock stop				Retained	

10.4 Operation of Data Buffer Stack

Figure 10-3 shows the operation of the data buffer stack.

As shown in the figure, when the PUT instruction is executed, the contents of the data buffer are transferred to a data buffer stack register specified by the stack pointer, and the stack pointer is incremented by one.

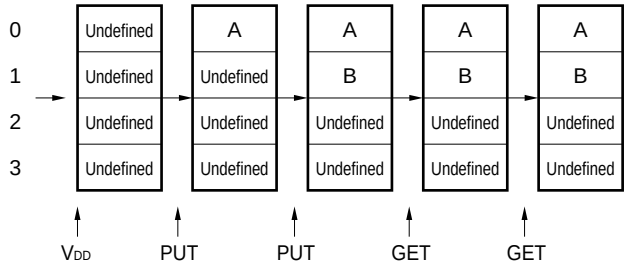
When the GET instruction is executed, the contents of a data buffer stack register specified by the stack pointer are transferred to the data buffer, and the stack pointer is decremented by one.

Therefore, note that the value of the stack pointer is set to 1 if data has been written once because its initial value is 0, and that the stack pointer is set to 0 when data has been written four times.

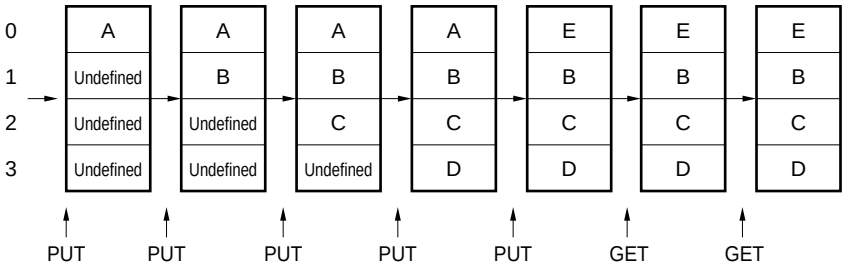
Note that when writing (PUT) exceeding four levels, the first data are discarded.

Figure 10-3. Operation of Data Buffer Stack

(a) If writing does not exceed level 4



(b) If writing exceeds level 4



10.5 Using Data Buffer Stack

A program example is shown below.

Example To save the contents of the data buffer and address register by using INT0 interrupt routine (the contents of the data buffer and address register are not automatically saved when an interrupt occurs).

```
START:
        BR    INITIAL        ; Reset address
        ; Interrupt vector address
        NOP                  ; SI01
        NOP                  ; SI00
        NOP                  ; TM3
        NOP                  ; TM2
        NOP                  ; TM1
        NOP                  ; TM0
        NOP                  ; INT4
        NOP                  ; INT3
        NOP                  ; INT2
        NOP                  ; INT1
        BR    INTINT0        ; INT0
        NOP                  ; Down edge of CE

INTINT0:
        PUT   DBFSTK, DBF    ; Saves contents of DBF to first level of data buffer
                             ; stack (DBFSTK)
        GET   DBF, AR        ; Transfers contents of address register (AR) to DBF
        PUT   DBFSTK, DBF    ; Saves contents of AR to second level of data buffer
                             ; stack

        Processing B         ; INT0 interrupt processing

        GET   DBF, DBFSTK    ; Restores second level of data buffer stack to data buffer,
        PUT   AR, DBF        ; and restores contents of data buffer to address register
        GET   DBF, DBFSTK    ; Restores first level of data buffer stack to data buffer
        EI
        RETI
INITIAL:
        SET1  IP0
        EI
LOOP:

        Processing A

        BR    LOOP
END
```

10.6 Cautions on Using Data Buffer Stack

The contents of the data buffer stack are not automatically saved when an interrupt is accepted, and therefore, must be saved by software.

Even when a bank of the data memory other than BANK0 is specified, the contents of the data buffer (existing in BANK0) can be saved or restored by using the “PUT” and “GET” instructions.

11. GENERAL-PURPOSE PORT

The general-purpose ports output high-level, low-level, or floating signals to external circuits, and read high-level or low-level signals from external circuits.

11.1 Outline of General-purpose Port

Table 11-1 shows the relationships between each port and port register.

The general-prupose ports are classified into I/O, input, and output ports.

The I/O ports are further subclassified into bit I/O ports that can be set in the input or output mode in 1-bit (1-pin) units, and group I/O ports that can be set in the input or output mode in 4-bit (4-pin) units. The inut or output mode of each I/O port is specified by the port input/output selection registers (addresses 60H through 6FH) of BANK15.

Table 11-1. Relationships between Port (Pin) and Port Register (1/3)

Port	Pin			Data Setting Method				
	No.	Symbol	I/O	Port register (data memory)				
				Bank	Address	Symbol	Bit symbol (reserved word)	
Port 0A	63	P0A3	I/O (bit I/O)	BANK0	70H	P0A	b_3	P0A3
	64	P0A2					b_2	P0A2
	65	P0A1					b_1	P0A1
	66	P0A0					b_0	P0A0
Port 0B	67	P0B3	I/O (bit I/O)		71H	P0B	b_3	P0B3
	68	P0B2					b_2	P0B2
	69	P0B1					b_1	P0B1
	70	P0B0					b_0	P0B0
Port 0C	59	P0C3	I/O (bit I/O)		72H	P0C	b_3	P0C3
	60	P0C2					b_2	P0C2
	61	P0C1					b_1	P0C1
	62	P0C0					b_0	P0C0
Port 0D	22	P0D3	Input		73H	P0D	b_3	P0D3
	23	P0D2					b_2	P0D2
	24	P0D1					b_1	P0D1
	25	P0D0					b_0	P0D0

Table 11-1. Relationships between Port (Pin) and Port Register (2/3)

Port	Pin			Data Setting Method				
	No.	Symbol	I/O	Port register (data memory)				
				Bank	Address	Symbol	Bit symbol (reserved word)	
Port 1A	2	P1A3	Input	BANK1	70H	P1A	b_3	P1A3
	3	P1A2					b_2	P1A2
	4	P1A1					b_1	P1A1
	5	P1A0					b_0	P1A0
Port 1B	17	P1B3	Output		71H	P1B	b_3	P1B3
	18	P1B2					b_2	P1B2
	19	P1B1					b_1	P1B1
	20	P1B0					b_0	P1B0
Port 1C	26	P1C3	Input		72H	P1C	b_3	P1C3
	27	P1C2					b_2	P1C2
	28	P1C1					b_1	P1C1
	29	P1C0					b_0	P1C0
Port 1D	37	P1D3	I/O (bit I/O)		73H	P1D	b_3	P1D3
	38	P1D2					b_2	P1D2
	39	P1D1					b_1	P1D1
	40	P1D0					b_0	P1D0
Port 2A	No pin		I/O (bit I/O)	BANK2	70H	P2A	b_3	–
	14	P2A2					b_2	P2A2
	15	P2A1					b_1	P2A1
	16	P2A0					b_0	P2A0
Port 2B	43	P2B3	I/O (bit I/O)		71H	P2B	b_3	P2B3
	44	P2B2					b_2	P2B2
	45	P2B1					b_1	P2B1
	46	P2B0					b_0	P2B0
Port 2C	55	P2C3	I/O (bit I/O)		72H	P2C	b_3	P2C3
	56	P2C2					b_2	P2C2
	57	P2C1					b_1	P2C1
	58	P2C0					b_0	P2C0
Port 2D	No pin		I/O (bit I/O)		73H	P2D	b_3	–
	71	P2D2					b_2	P2D2
	72	P2D1					b_1	P2D1
	73	P2D0					b_0	P2D0

Table 11-1. Relationships between Port (Pin) and Port Register (3/3)

Port	Pin			Data Setting Method				
	No.	Symbol	I/O	Port register (data memory)				
				Bank	Address	Symbol	Bit symbol (reserved word)	
Port 3A	6	P3A3	I/O (group I/O)	BANK3	70H	P3A	b_3	P3A3
	7	P3A2					b_2	P3A2
	8	P3A1					b_1	P3A1
	9	P3A0					b_0	P3A0
Port 3B	10	P3B3	I/O (group I/O)		71H	P3B	b_3	P3B3
	11	P3B2					b_2	P3B2
	12	P3B1					b_1	P3B1
	13	P3B0					b_0	P3B0
Port 3C	47	P3C3	I/O (group I/O)		72H	P3C	b_3	P3C3
	48	P3C2					b_2	P3C2
	49	P3C1					b_1	P3C1
	50	P3C0					b_0	P3C0
Port 3D	51	P3D3	I/O (group I/O)		73H	P3D	b_3	P3D3
	52	P3D2					b_2	P3D2
	53	P3D1					b_1	P3D1
	54	P3D0					b_0	P3D0
–	No pin		–	BANK4 | BANK15[Note]	70H-73H	–	Fixed to “0”	

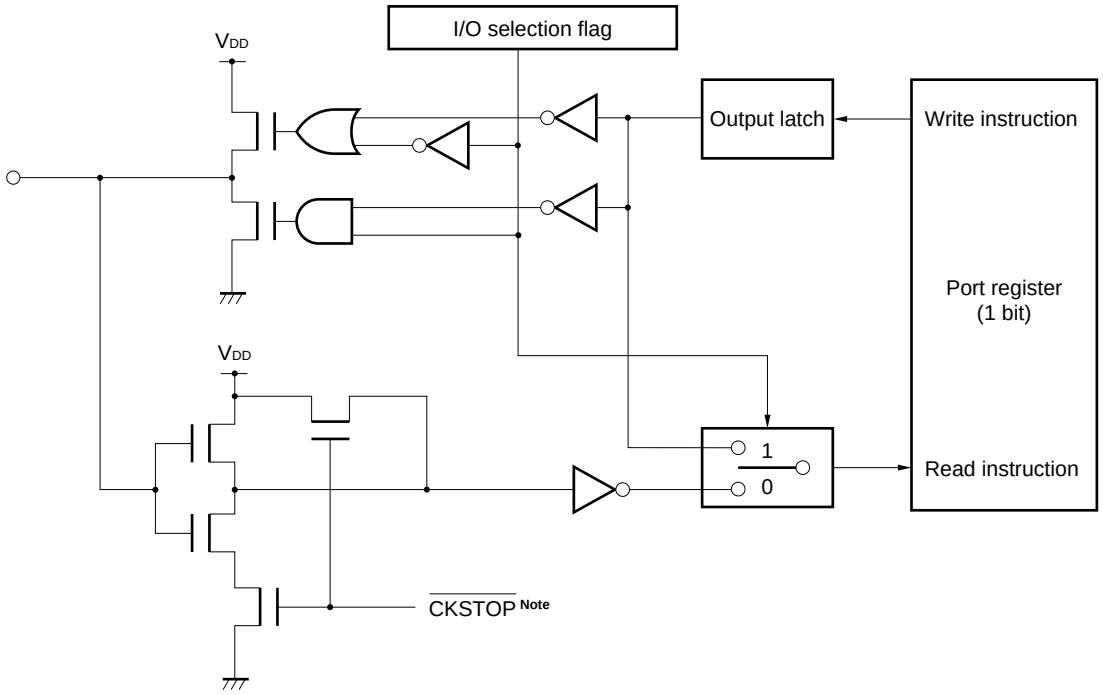
Note μPD17717 and 17718 do not have BANKs 10 through 14.

11.2 General-Purpose I/O Port (P0A, P0B, P0C, P1D, P2A, P2B, P2C, P2D, P3A, P3B, P3C, P3D)

11.2.1 Configuration of I/O port

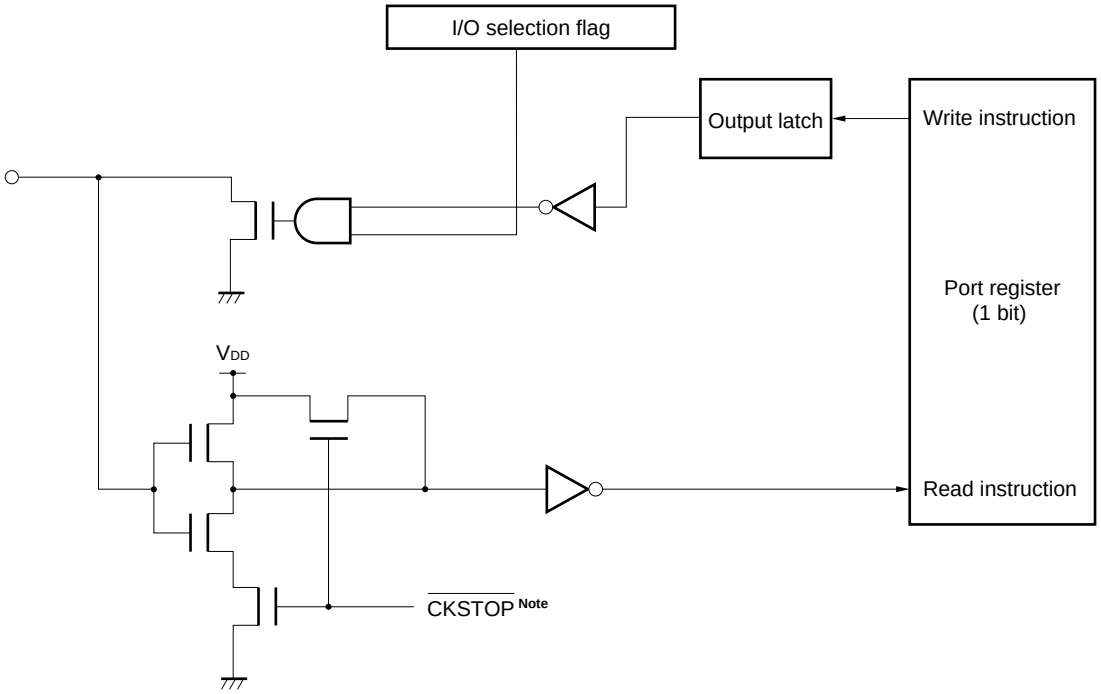
The following paragraphs (1) and (2) show the configuration of the I/O ports.

(1) P0A (P0A1, P0A0)
P0B (P0B3, P0B2, P0B1, P0B0)
P0C (P0C3, P0C2, P0C1, P0C0)
P1D (P1D3, P1D2, P1D1, P1D0)
P2A (P2A2, P2A1, P2A0)
P2B (P2B3, P2B2, P2B1, P2B0)
P2C (P2C3, P2C2, P2C1, P2C0)
P2D (P2D2)
P3A (P3A3, P3A2, P3A1, P3A0)
P3B (P3B3, P3B2, P3B1, P3B0)
P3C (P3C3, P3C2, P3C1, P3C0)
P3D (P3D3, P3D2, P3D1, P3D0)



Note This is an internal signal that is output when the clock stop instruction is executed, and this circuit is designed not to increase the current consumption due to noise even if it is floated.

(2) P0A (P0A3, P0A2)
P2D (P2D1, P2D0)



Note This is an internal signal that is output when the clock stop instruction is executed, and this circuit is designed not to increase the current consumption due to noise even if it is floated.

11.2.2 Using I/O port

The input or output mode of the I/O ports is set by I/O selection register P0A, P0B, P0C, P1D, P2A, P2B, P2C, P2D, P3A, P3B, P3C, or P3D of the control registers.

Because P0A, P0B, P0C, P1D, P2A, P2B, P2C, and P2D are bit I/O ports, they can be set in the input or output mode in 1-bit units.

P3A, P3B, P3C, and P3D are group I/O ports, and therefore they are set in the input or output mode in 4-bit units.

Setting the output data of or reading the input data of a port is carried out by executing an instruction that writes data to or reads data from the port.

11.2.3 shows the configuration of the I/O selection register of each port.

11.2.4 and 11.2.5 describe how each port is used as an input or output port.

11.2.6 describes the points to be noted when using the I/O ports.

11.2.3 I/O port I/O selection register

The following I/O selection registers of the I/O ports are available.

- Port 0A bit I/O selection register
- Port 0B bit I/O selection register
- Port 0C bit I/O selection register
- Port 1D bit I/O selection register
- Port 2A bit I/O selection register
- Port 2B bit I/O selection register
- Port 2C bit I/O selection register
- Port 2D bit I/O selection register
- Group I/O selection registers (port 3A, port 3B, port 3C, port 3D)

Each I/O selection register sets the input or output mode of the corresponding port pin.

The following paragraphs (1) through (9) descibe the configuration and functions of the above I/O selection registers.

(1) Port 0A bit I/O selection register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Port 0A bit I/O selection	P0AB IO3	P0AB IO2	P0AB IO1	P0AB IO0	(BANK15) 6FH	R/W

Bit	Sets input/output mode of port
b_0	
0	Sets P0A0 pin in input mode
1	Sets P0A0 pin in output mode

Bit	Sets input/output mode of port
b_1	
0	Sets P0A1 pin in input mode
1	Sets P0A1 pin in output mode

Bit	Sets input/output mode of port
b_2	
0	Sets P0A2 pin in input mode
1	Sets P0A2 pin in output mode

Bit	Sets input/output mode of port
b_3	
0	Sets P0A3 pin in input mode
1	Sets P0A3 pin in output mode

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset	0	0	0	0
	CE reset	Retained			
Clock stop		Retained			

(2) Port 0B bit I/O selection register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Port 0B bit I/O selection	P0BBIO3	P0BBIO2	P0BBIO1	P0BBIO0	(BANK15) 6EH	R/W

b_3 (P0BBIO3)	Sets input/output mode of port
0	Sets P0B3 pin in input mode
1	Sets P0B3 pin in output mode

b_2 (P0BBIO2)	Sets input/output mode of port
0	Sets P0B2 pin in input mode
1	Sets P0B2 pin in output mode

b_1 (P0BBIO1)	Sets input/output mode of port
0	Sets P0B1 pin in input mode
1	Sets P0B1 pin in output mode

b_0 (P0BBIO0)	Sets input/output mode of port
0	Sets P0B0 pin in input mode
1	Sets P0B0 pin in output mode

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset	0	0	0	0
	CE reset	Retained			
Clock stop		Retained			

(3) Port 0C bit I/O selection register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Port 0C bit I/O selection	P0CBIO3	P0CBIO2	P0CBIO1	P0CBIO0	(BANK15) 6DH	R/W

P0CBIO0	Sets input/output mode of port
0	Sets P0C0 pin in input mode
1	Sets P0C0 pin in output mode

P0CBIO1	Sets input/output mode of port
0	Sets P0C1 pin in input mode
1	Sets P0C1 pin in output mode

P0CBIO2	Sets input/output mode of port
0	Sets P0C2 pin in input mode
1	Sets P0C2 pin in output mode

P0CBIO3	Sets input/output mode of port
0	Sets P0C3 pin in input mode
1	Sets P0C3 pin in output mode

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset	0	0	0	0
	CE reset	Retained			
Clock stop		Retained			

(4) Port 1D bit I/O selection register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Port 1D bit I/O selection	P1DBIO3	P1DBIO2	P1DBIO1	P1DBIO0	(BANK15) 6CH	R/W

Bit	Sets input/output mode of port
0	Sets P1D0 pin in input mode
1	Sets P1D0 pin in output mode

Bit	Sets input/output mode of port
0	Sets P1D1 pin in input mode
1	Sets P1D1 pin in output mode

Bit	Sets input/output mode of port
0	Sets P1D2 pin in input mode
1	Sets P1D2 pin in output mode

Bit	Sets input/output mode of port
0	Sets P1D3 pin in input mode
1	Sets P1D3 pin in output mode

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset	0	0	0	0
	CE reset	Retained			
Clock stop		Retained			

(5) Port 2A bit I/O selection register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Port 2A bit I/O selection	0	P2AIO2	P2AIO1	P2AIO0	(BANK15) 6BH	R/W

P2AIO0	Sets input/output mode of port
0	Sets P2A0 pin in input mode
1	Sets P2A0 pin in output mode

P2AIO1	Sets input/output mode of port
0	Sets P2A1 pin in input mode
1	Sets P2A1 pin in output mode

P2AIO2	Sets input/output mode of port
0	Sets P2A2 pin in input mode
1	Sets P2A2 pin in output mode

Fixed to “0”

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset		0	0	0
	CE reset		Retained		
Clock stop			Retained		

(6) Port 2B bit I/O selection register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Port 2B bit I/O selection	P2BIO3	P2BIO2	P2BIO1	P2BIO0	(BANK15) 6AH	R/W

b_3 (P2BIO3)	Sets input/output mode of port
0	Sets P2B3 pin in input mode
1	Sets P2B3 pin in output mode

b_2 (P2BIO2)	Sets input/output mode of port
0	Sets P2B2 pin in input mode
1	Sets P2B2 pin in output mode

b_1 (P2BIO1)	Sets input/output mode of port
0	Sets P2B1 pin in input mode
1	Sets P2B1 pin in output mode

b_0 (P2BIO0)	Sets input/output mode of port
0	Sets P2B0 pin in input mode
1	Sets P2B0 pin in output mode

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset	0	0	0	0
	CE reset	Retained			
Clock stop		Retained			

(7) Port 2C bit I/O selection register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Port 2C bit I/O selection	P2CBIO3	P2CBIO2	P2CBIO1	P2CBIO0	(BANK15) 69H	R/W

P2CBIO3	Sets input/output mode of port
0	Sets P2C3 pin in input mode
1	Sets P2C3 pin in output mode

P2CBIO2	Sets input/output mode of port
0	Sets P2C2 pin in input mode
1	Sets P2C2 pin in output mode

P2CBIO1	Sets input/output mode of port
0	Sets P2C1 pin in input mode
1	Sets P2C1 pin in output mode

P2CBIO0	Sets input/output mode of port
0	Sets P2C0 pin in input mode
1	Sets P2C0 pin in output mode

At reset		b_3	b_2	b_1	b_0
	Power-ON reset	0	0	0	0
	WDT&SP reset	0	0	0	0
	CE reset	Retained			
Clock stop		Retained			

(8) Port 2D bit I/O selection register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Port 2D bit I/O selection	0	P2DBIO2	P2DBIO1	P2DBIO0	(BANK15) 68H	R/W

P2DBIO0	Sets input/output mode of port
0	Sets P2D0 pin in input mode
1	Sets P2D0 pin in output mode

P2DBIO1	Sets input/output mode of port
0	Sets P2D1 pin in input mode
1	Sets P2D1 pin in output mode

P2DBIO2	Sets input/output mode of port
0	Sets P2D2 pin in input mode
1	Sets P2D2 pin in output mode

Fixed to “0”

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset		0	0	0
	CE reset		Retained		
Clock stop			Retained		

(9) Group I/O selection register (ports 3A, 3B, 3C, 3D)

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Group I/O selection	P3DGIO	P3CGIO	P3BGIO	P3AGIO	(BANK15) 67H	R/W

Sets input/output mode of port	
0	Sets P3A0 through P3A3 pins in input mode
1	Sets P3A0 through P3A3 pins in output mode

Sets input/output mode of port	
0	Sets P3B0 through P3B3 pins in input mode
1	Sets P3B0 through P3B3 pins in output mode

Sets input/output mode of port	
0	Sets P3C0 through P3C3 pins in input mode
1	Sets P3C0 through P3C3 pins in output mode

Sets input/output mode of port	
0	Sets P3D0 through P3D3 pins in input mode
1	Sets P3D0 through P3D3 pins in output mode

At reset	Power-ON reset	0	0	0	0	
	WDT&SP reset	0	0	0	0	
	CE reset	Retained				
Clock stop		Retained				

11.2.4 When using I/O port as input port

The port pin to be set in the input mode is selected by the I/O selection register corresponding to the port.

Ports P0A, P0B, P0C, P1D, P2A, P2B, P2C, and P2D can be set in the input or output mode in 1-bit units. P3A, P3B, P3C, and P3D can be set in the input or output mode in 4-bit units.

The pin set in the input mode is floated (Hi-Z) and waits for input of an external signal.

The input data is read by executing a read instruction (such as SKT) to the port register corresponding to the port pin.

"1" is read from the port register when a high level is input to the corresponding port pin; when a low level is input to the port pin, "0" is read from the register.

When a write instruction (such as MOV) is executed to the port register corresponding to the pin set in the input mode, the contents of the output latch are rewritten.

11.2.5 When using I/O port as output port

The port pin to be set in the output mode is selected by the I/O selection register corresponding to the port.

Ports P0A, P0B, P0C, P1D, P2A, P2B, P2C, and P2D can be set in the input or output mode in 1-bit units. P3A, P3B, P3C, and P3D can be set in the input or output mode in 4-bit units.

The pin set in the output mode outputs the contents of the output latch.

The output data is set by executing a write instruction (such as MOV) to the port register corresponding to the port pin.

Write "1" to the port register to output a high level to the port pin; write "0" to output a low level. The port pin can be also floated (Hi-Z) if it is set in the input mode.

If a read instruction (such as SKT) is executed to the port register corresponding to a port pin set in the output mode, the contents of the output latch are read.

Note, however, that the contents of the output latch of the P0A3 and P0A2 pins may differ from the read contents because the status of these pins are read as are (refer to **11.2.6**).

11.2.6 Cautions on using I/O port (P0A3 and P0A2 pins)

When using the P0A3 and P0A2 pins in the output mode, the contents of the output latch may be rewritten as shown in the example below.

Example To set the P0A3 and P0A2 pins in the output mode

```
 BANK15
 INITFLG P0ABI03, P0ABI02, NOT P0ABI01, NOT P0ABI00  ; Sets P0A3 and P0A2 pins in
                                                       output mode
 INITFLG P0A3, P0A2, NOT P0A1, NOT P0A0              ; Outputs high level to P0A3 and
                                                       P0A2 pins
;<1>
 CLR1    P0A3                                        ; Outputs low level to P0A3 pin
 MACRO EXTEND
 AND     .MF.P0A3 SHR 4, #.DF.(NOT P0A3 AND 0FH)
```

If the P0A2 pin is externally made low when the instruction in the above example <1> is executed, the contents of the output latch of the P0A2 pin are rewritten to "0" by the CLR1 instruction.
In other words, if an instruction that reads the contents of port register P0A is executed while the P0A3 or P0A2 pin is set in the output mode, the contents of the output latch are rewritten to the pin level at that time, regardless of the previous status.

11.2.7 Status of I/O port at reset

(1) At power-ON reset

All the I/O ports are set in the input mode.
The contents of the output latch are reset to “0”.

(2) At WDT&SP reset

All the I/O ports are set in the input mode.
The contents of the output latch are reset to “0”.

(3) At CE reset

The setting of the input or output mode is retained.
The contents of the output latch are also retained.

(4) On execution of clock stop instruction

The setting of the input or output mode is retained.
The contents of the output latch are also retained.

(5) In halt status

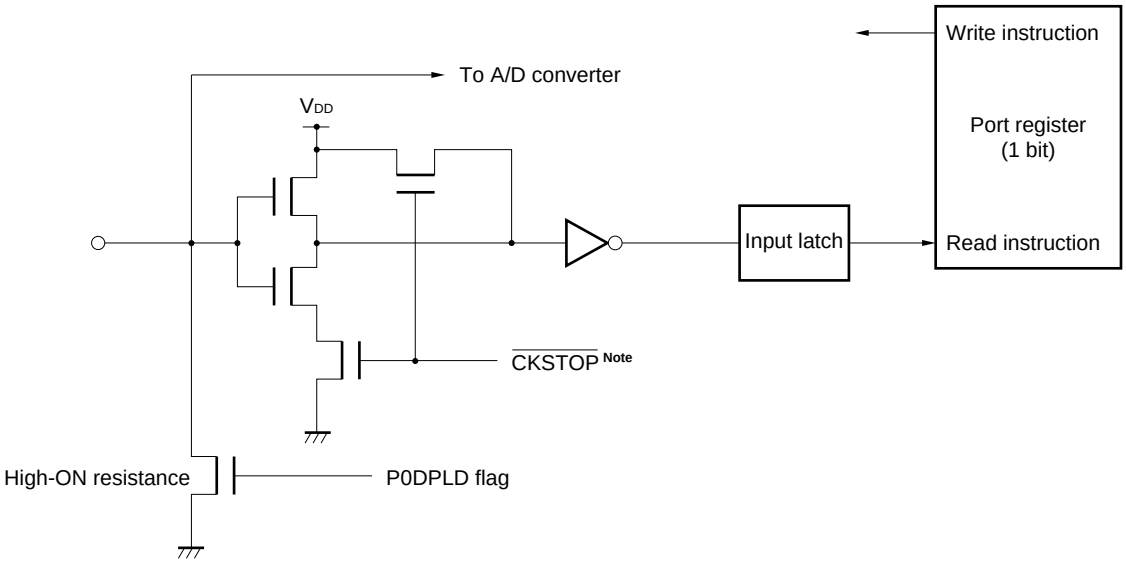
The previous status is retained.

11.3 General-Purpose Input Port (P0D, P1A, P1C)

11.3.1 Configuration of input port

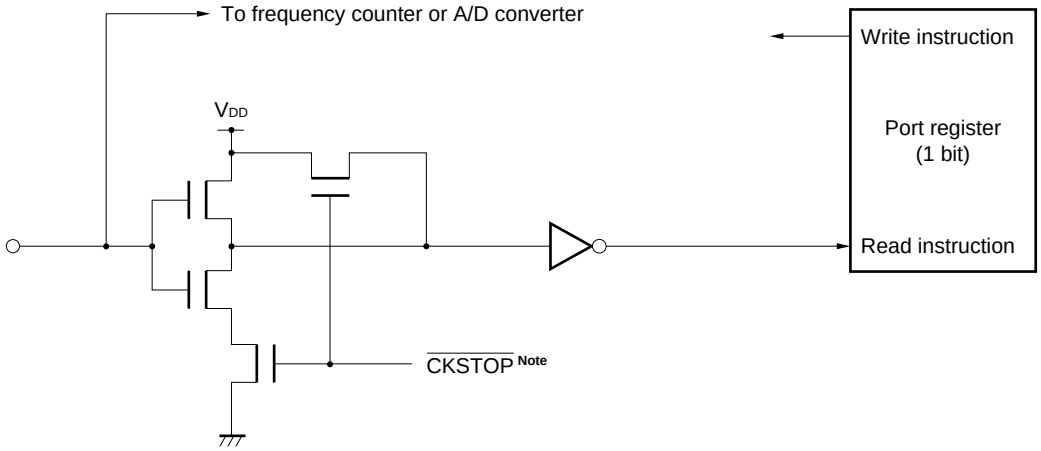
The following paragraphs (1) and (2) show the configuration of the input port.

(1) P0D (P0D3, P0D2, P0D1, P0D0)



Note This is an internal signal output on execution of the clock stop instruction, and its circuit is designed not to increase the current consumption due to noise even if the pin is floated.

(2) P1A (P1A3, P1A2, P1A1, P1A0)
P1C (P1C3, P1C2, P1C1, P1C0)



Note This is an internal signal output on execution of the clock stop instruction, and its circuit is designed not to increase the current consumption due to noise even if the pin is floated (except P1A3, P1A2, and P1A0).

11.3.2 Using input port

The input data is read by executing a read instruction (such as SKT) to the port register corresponding to the port pin.

"1" is read from the port register when a high level is input to the corresponding port pin; when a low level is input to the port pin, "0" is read from the register.

Nothing is affected even if a write instruction (such as MOV) is executed to the port register.

P0D has a pull-down resistor that can be connected or disconnected by software in 1-bit units. The pull-down resistor is connected when "0" is written to the corresponding bit of the port 0D pull-down resistor selection register. When "1" is written to the corresponding bit of this register, the pull-down resistor is disconnected.

11.3.3 Port 0D pull-down resistor selection register

The port 0D pull-down resistor selection register specifies whether a pull-down resistor is connected to P0D3 through P0D0 pins. The configuration and function of this register are illustrated below.

- Port 0D pull-down resistor selection register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Port 0D pull-down resistor selection	P0DPLD3	P0DPLD2	P0DPLD1	P0DPLD0	(BANK15) 66H	R/W

P0DPLD0	Selects pull-down resistor of P0D0 pin
0	Connects pull-down resistor to P0D0 pin
1	Disconnects pull-down resistor from P0D0 pin

P0DPLD1	Selects pull-down resistor of P0D1 pin
0	Connects pull-down resistor to P0D1 pin
1	Disconnects pull-down resistor from P0D1 pin

P0DPLD2	Selects pull-down resistor of P0D2 pin
0	Connects pull-down resistor to P0D2 pin
1	Disconnects pull-down resistor from P0D2 pin

P0DPLD3	Selects pull-down resistor of P0D3 pin
0	Connects pull-down resistor to P0D3 pin
1	Disconnects pull-down resistor from P0D3 pin

		P0DPLD3	P0DPLD2	P0DPLD1	P0DPLD0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset	0	0	0	0
	CE reset	Retained			
Clock stop		Retained			

11.3.4 Status of input port at reset

(1) At power-ON reset

All the input ports are set in the input mode.
All the pull-down resistors of P0D are connected.

(2) At WDT&SP reset

All the input ports are set in the input mode.
All the pull-down resistors of P0D are connected.

(3) At CE reset

The input ports are set in the input mode.
The pull-down resistors of P0D retain the previous status.

(4) On execution of clock stop instruction

The input ports are set in the input mode.
The pull-down resistors of P0D retain the previous status.

(5) In halt status

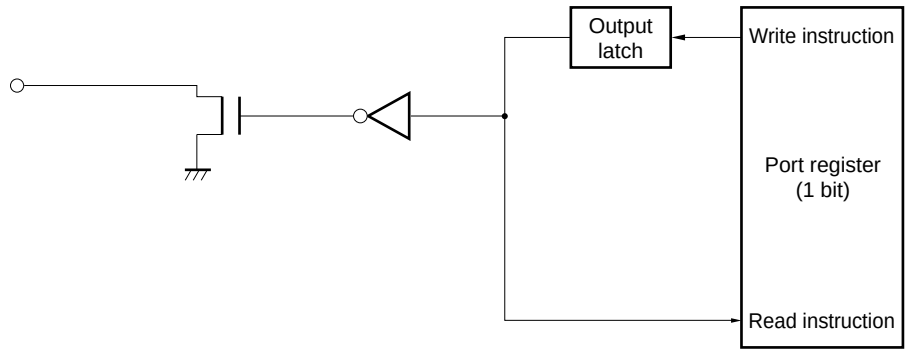
The previous status is retained.

11.4 General-Purpose Output Port (P1B)

11.4.1 Configuration of output port

The configuration of the output port is shown below.

(1) P1B (P1B3, P1B2, P1B1, P1B0)



11.4.2 Using output port

The output port outputs the contents of the output latch to each pin.

The output data is set by executing a write instruction (such as MOV) to the port register corresponding to the port pin.

Write “1” to the port register to output a high level to the port pin; write “0” to output a low level.

However, because P1B is an N-ch open-drain output port, it is floated when it outputs a high level. Therefore, an external pull-up resistor must be connected to this port.

If a read instruction (such as SKT) is executed to the port register, the contents of the output latch are read.

11.4.3 Status of output port at reset

(1) At power-ON reset

The contents of the output latch are output.
The contents of the output latch are reset to “0”.

(2) At WDT&SP reset

The contents of the output latch are output.
The contents of the output latch are reset to “0”.

(3) At CE reset

The contents of the output latch are output.
The contents of the output latch are retained.

(4) On execution of clock stop instruction

The contents of the output latch are output.
The contents of the output latch are retained.

(5) In halt status

The contents of the output latch are output.
The contents of the output latch are retained.

12. INTERRUPT

12.1 Outline of Interrupt Block

Figure 12-1 outlines the interrupt block.

As shown in the figure, the interrupt block temporarily stops the currently executed program and branches execution to a vector address in response to an interrupt request output by a peripheral hardware unit.

The interrupt block consists of an “interrupt request servicing block” corresponding to each peripheral hardware unit, “interrupt enable flip-flop” that enables all interrupts, “stack pointer” that is controlled when an interrupt is accepted, “address stack registers”, “program counter”, and “interrupt stack”.

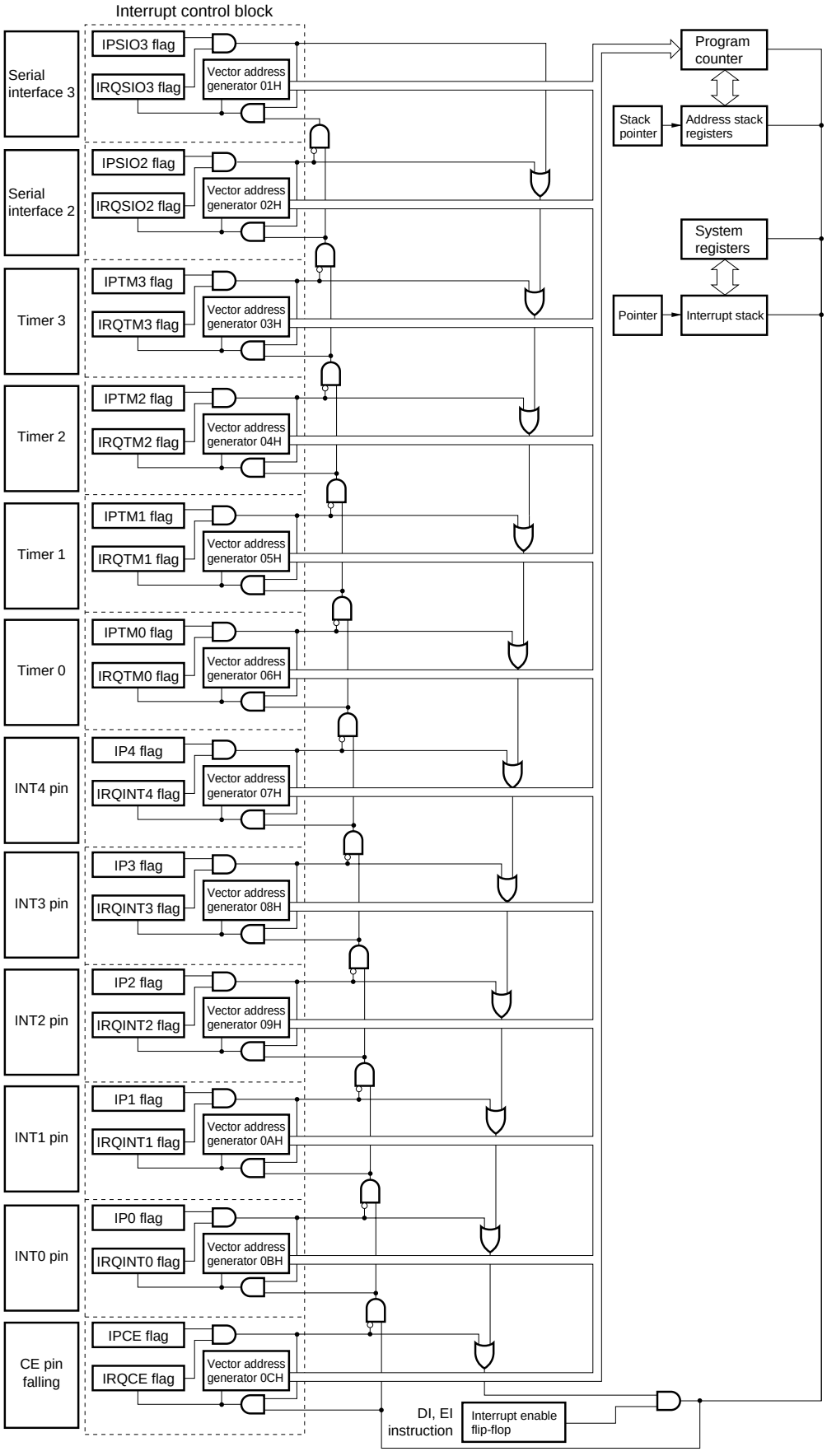
The “interrupt control block” of each peripheral hardware unit consists of an “interrupt request flag (IRQ×××)” that detects the corresponding interrupt request, “interrupt enable flag (IP×××)” that enables the interrupt, and “vector address generator (VAG)” that specifies a vector address when the interrupt is accepted.

The μPD17719 has the following 12 types of maskable interrupts.

- CE pin falling edge interrupt
- INT0 through INT4 interrupts
- Timer 0 through timer 3 interrupts
- Serial interface 2 and serial interface 3 interrupts

When an interrupt is accepted, execution branches to a predetermined address, and the interrupt is serviced.

Figure 12-1. Outline of Interrupt Block



12.2 Interrupt Control Block

An interrupt control block is provided for each peripheral hardware unit. This block detects issuance of an interrupt request, enables the interrupt, and generates a vector address when the interrupt is accepted.

12.2.1 Configuration and function of interrupt request flag (IRQ×××)

Each interrupt request flag is set to 1 when an interrupt request is issued by the corresponding peripheral hardware unit, and is reset to 0 when the interrupt is accepted.

Writing the interrupt request flag to “1” via a window register is equivalent to issuance of the interrupt request.

By detecting the interrupt request flag when an interrupt is not enabled, issuance status of each interrupt request can be detected.

Once the interrupt request flag has been set, it is not reset until the corresponding interrupt is accepted, or until “0” is written to the flag via a window register.

Even if two or more interrupt requests are issued at the same time, the interrupt request flag corresponding to the interrupt that has not been accepted is not reset.

Figures 12-2 through 12-13 show the configuration and function of the respective interrupt request registers.

Figure 12-2. Configuration of Serial Interface 3 Interrupt Request Register

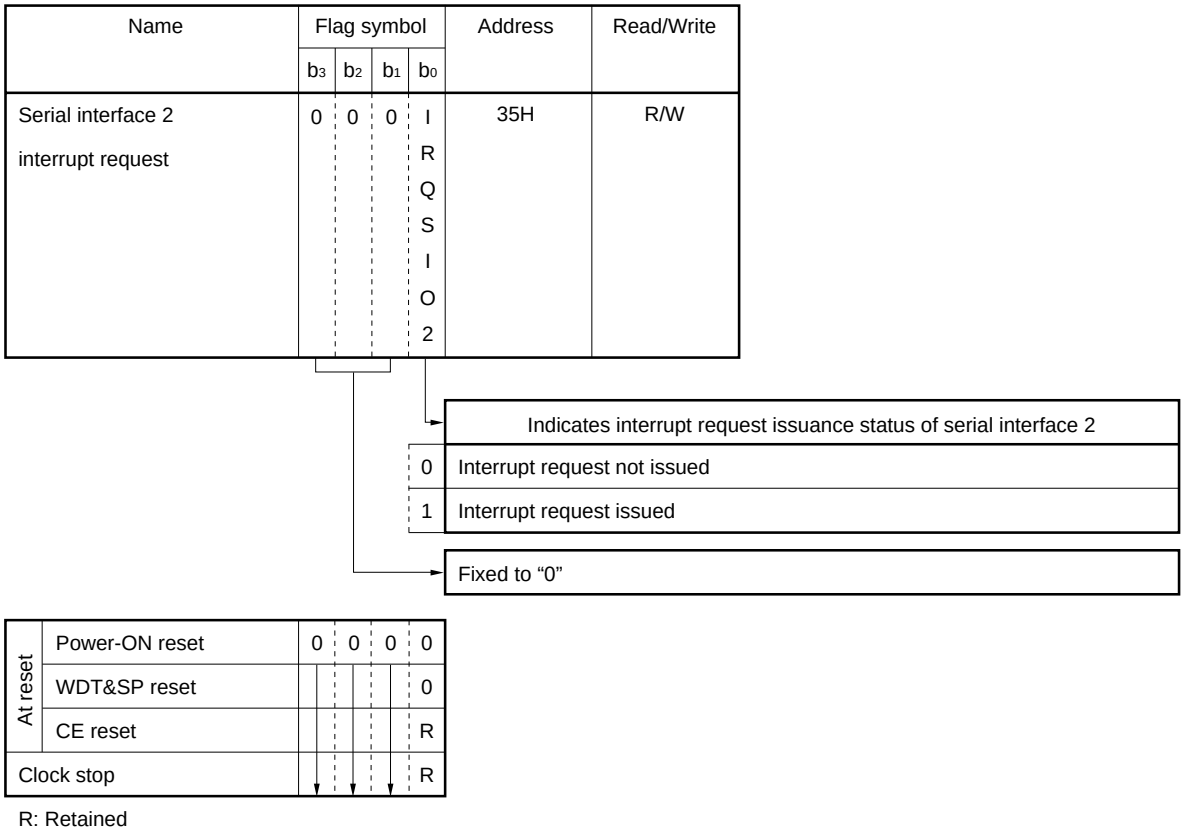
Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Serial interface 3 interrupt request	0	0	0	IRQSIO3	34H	R/W

Indicates interrupt request issuance status of serial interface 3	
0	Interrupt request not issued
1	Interrupt request issued
Fixed to “0”	

At reset	Power-ON reset	0	0	0	0
	WDT&SP reset				0
	CE reset				R
Clock stop					R

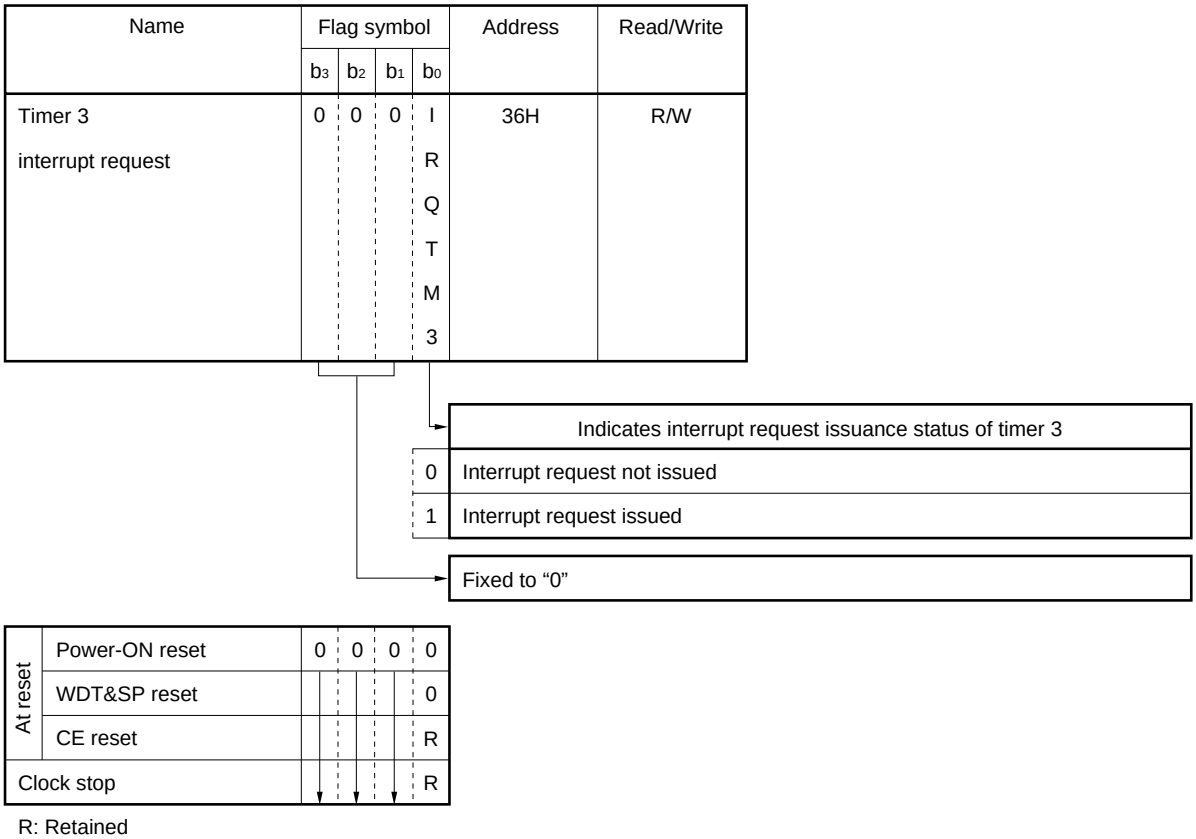
R: Retained

Figure 12-3. Configuration of Serial Interface 2 Interrupt Request Register



R: Retained

Figure 12-4. Configuration of Timer 3 Interrupt Request Register



R: Retained

Figure 12-5. Configuration of Timer 2 Interrupt Request Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Timer 2 interrupt request	0	0	0	IRQTM2	37H	R/W

	Indicates interrupt request issuance status of timer 2
0	Interrupt request not issued
1	Interrupt request issued

Fixed to "0"

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset				0
	CE reset				R
Clock stop					R

R: Retained

Figure 12-6. Configuration of Timer 1 Interrupt Request Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Timer 1 interrupt request	0	0	0	IRQTM1	38H	R/W

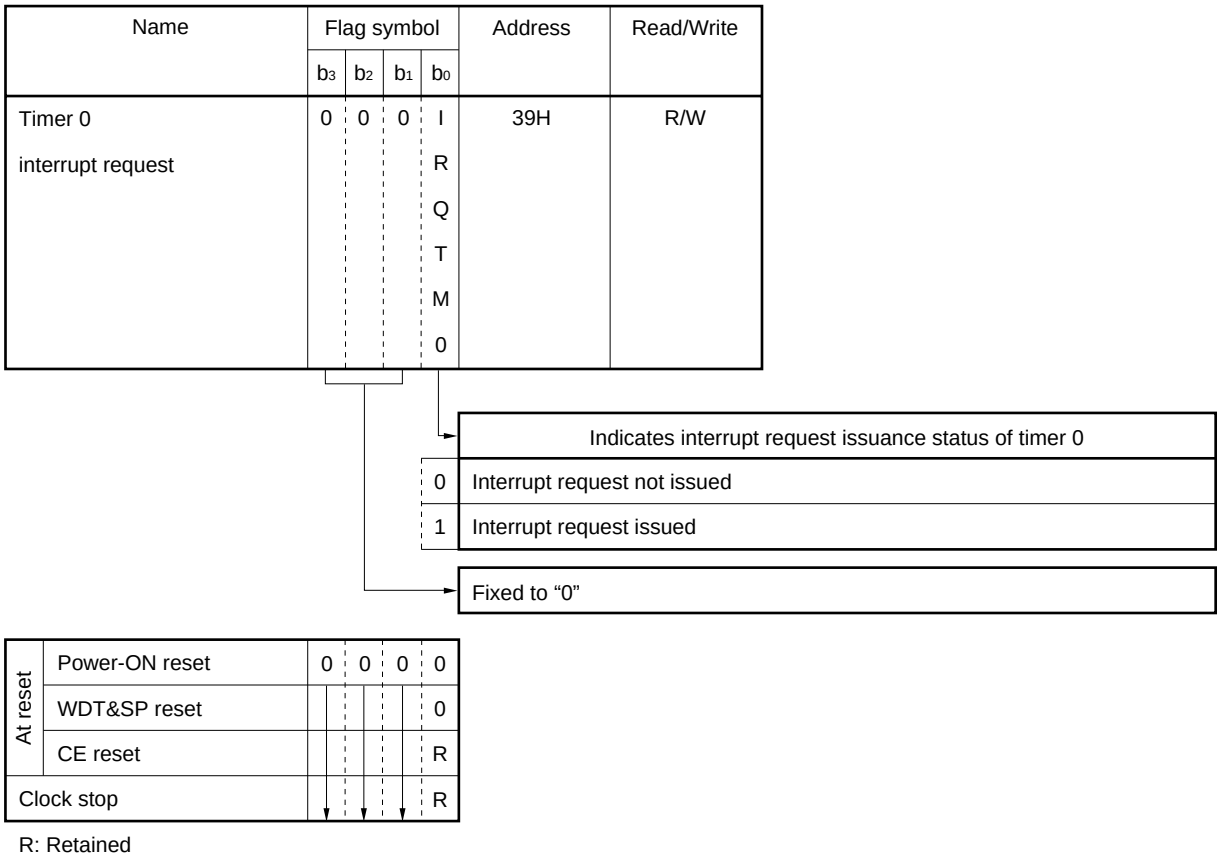
	Indicates interrupt request issuance status of timer 1
0	Interrupt request not issued
1	Interrupt request issued

Fixed to "0"

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset				0
	CE reset				R
Clock stop					R

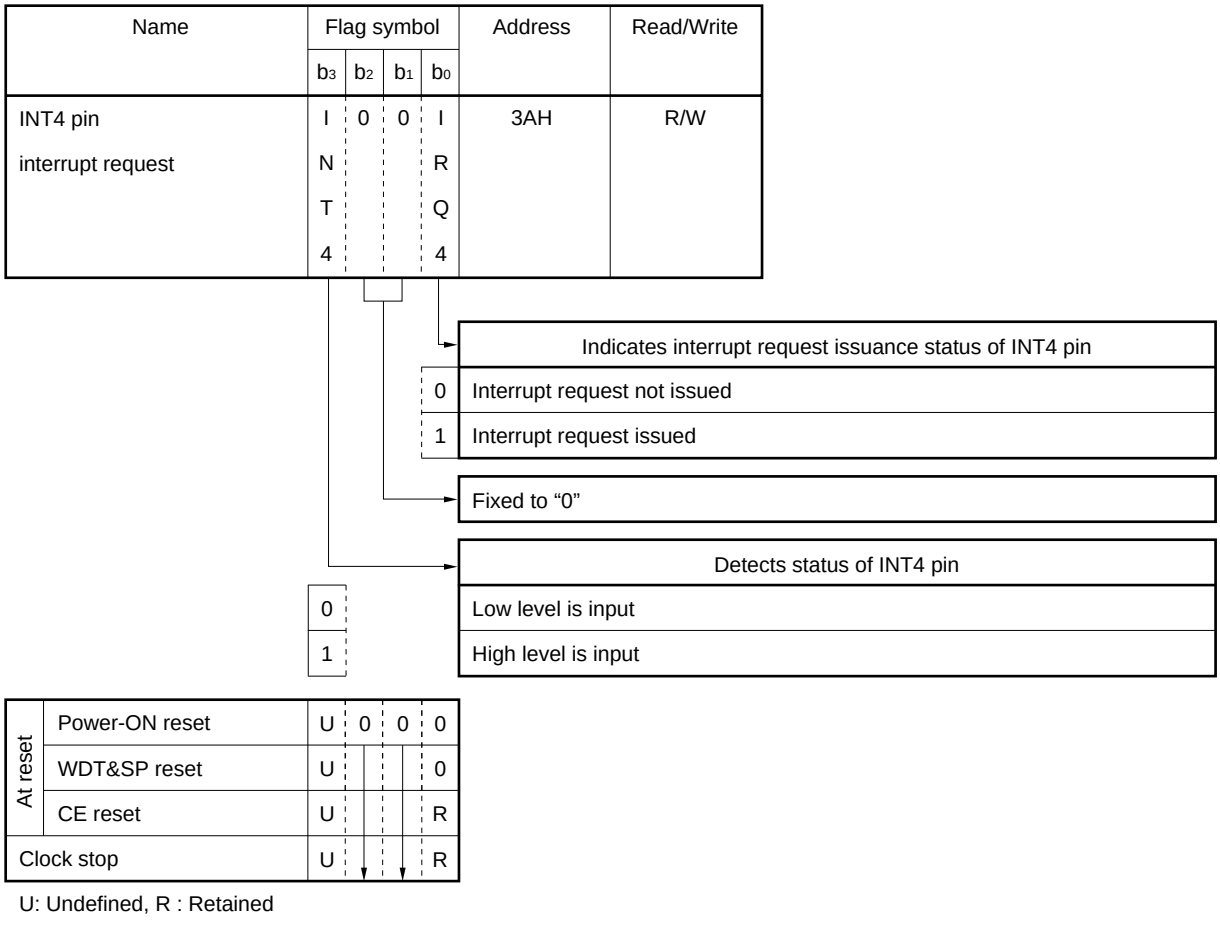
R: Retained

Figure 12-7. Configuration of Timer 0 Interrupt Request Register



R: Retained

Figure 12-8. Configuration of INT4 Pin Interrupt Request Register



U: Undefined, R : Retained

Figure 12-9. Configuration of INT3 Pin Interrupt Request Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
INT3 pin interrupt request	INT3	0	0	IRQ3	3BH	R/W

Bit	Indicates interrupt request issuance status of INT3 pin
0	Interrupt request not issued
1	Interrupt request issued

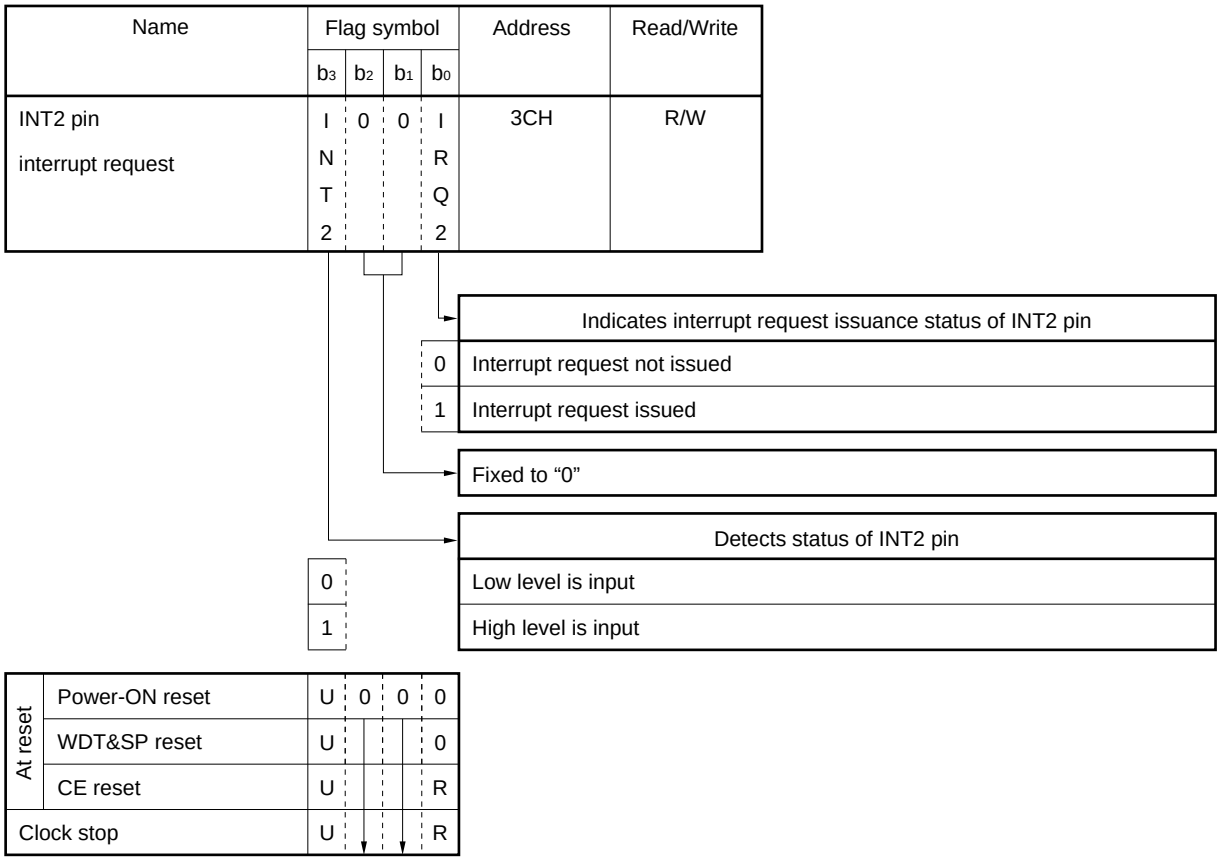
Bit	Fixed to "0"

Bit	Detects status of INT3 pin
0	Low level is input
1	High level is input

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	U	0	0	0
	WDT&SP reset	U			0
	CE reset	U			R
Clock stop		U			R

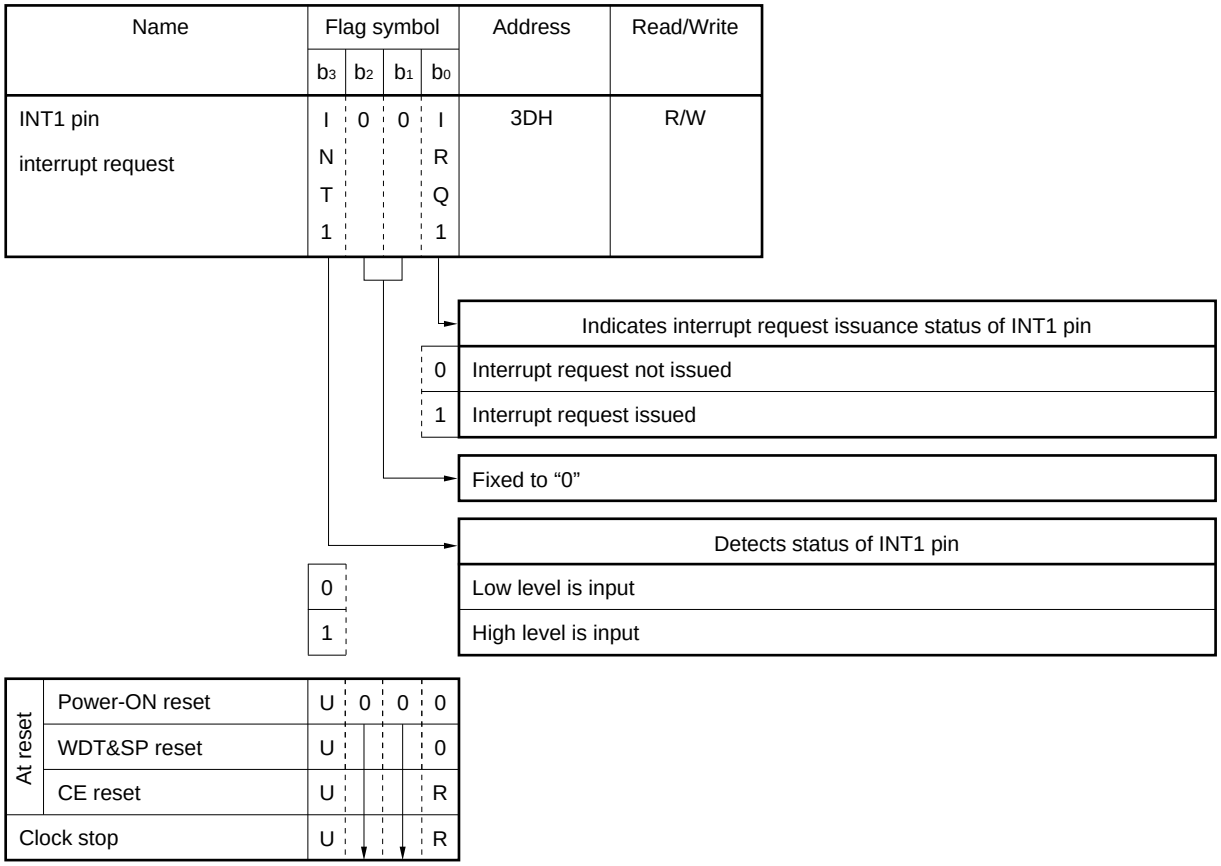
U: Undefined, R : Retained

Figure 12-10. Configuration of INT2 Pin Interrupt Request Register



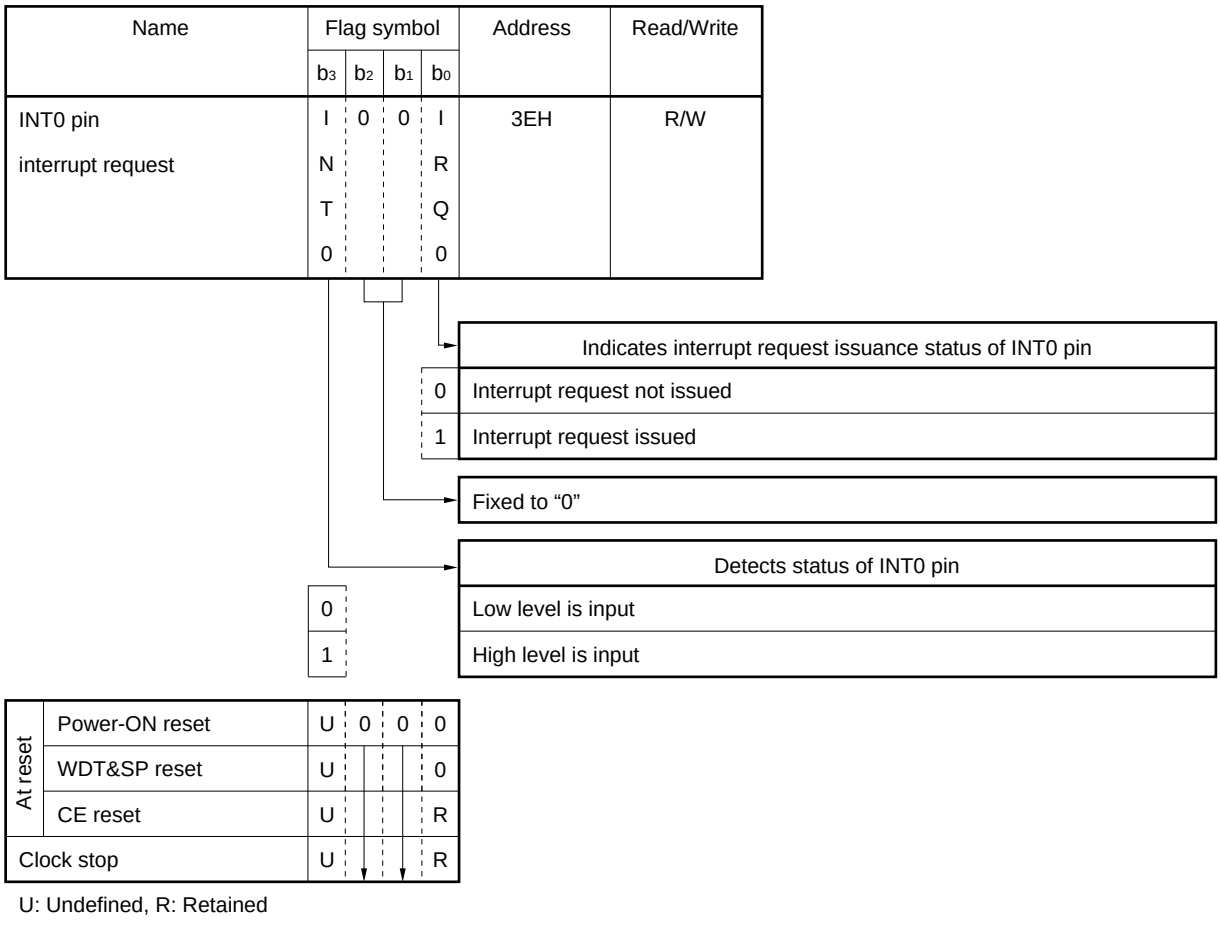
U: Undefined, R : Retained

Figure 12-11. Configuration of INT1 Pin Interrupt Request Register



U: Undefined, R: Retained

Figure 12-12. Configuration of INT0 Pin Interrupt Request Register



U: Undefined, R: Retained

Figure 12-13. Configuration of CE Pin Interrupt Request Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
CE pin interrupt request	CE	0	CECNTSTT	IRQCE	3FH	R/W

IRQCE — Indicates interrupt request issuance status of CE pin

0	Interrupt request not issued
1	Interrupt request issued

CECNTSTT — Detects status of CE reset counter

0	Stops
1	Operates

b_2 — Fixed to “0”

CE — Detects status of CE pin

0	Low level is input
1	High level is input

		CE	b_2	CECNTSTT	IRQCE
At reset	Power-ON reset	U	0	0	0
	WDT&SP reset	U		0	0
	CE reset	U		0	R
Clock stop		U		0	R

U : Undefined, R : Retained

12.2.2 Function and configuration of interrupt request flag (IP×××)

Each interrupt request flag enables the interrupt of the corresponding peripheral hardware unit.
In order for an interrupt to be accepted, all the following conditions must be satisfied.

- The interrupt must be enabled by the corresponding interrupt request flag.
- The interrupt request must be issued by the corresponding interrupt request flag.
- The EI instruction (which enables all interrupts) must be executed.

The interrupt enable flags are located on the interrupt enable register on the register file.
Figures 12-14 through 12-16 show the configuration and function of each interrupt enable register.

Figure 12-14. Configuration of Interrupt Enable Register 1

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Interrupt enable 1	IPSIO3	IPSIO2	IPTM3	IPTM2	2DH	R/W

IPTM2 — Enables or disables timer 2 interrupt

0	Disables
1	Enables

IPTM3 — Enables or disables timer 3 interrupt

0	Disables
1	Enables

IPSIO2 — Enables or disables serial interface 2 interrupt

0	Disables
1	Enables

IPSIO3 — Enables or disables serial interface 3 interrupt

0	Disables
1	Enables

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset	0	0	0	0
	CE reset	Retained			
Clock stop		Retained			

Figure 12-15. Configuration of Interrupt Enable Register 2

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Interrupt enable 2	IPTM1	IPTM0	IP4	IP3	2EH	R/W

IPTM1 — Enables or disables timer 1 interrupt

0	Disables
1	Enables

IPTM0 — Enables or disables timer 0 interrupt

0	Disables
1	Enables

IP4 — Enables or disables INT4 pin interrupt

0	Disables
1	Enables

IP3 — Enables or disables INT3 pin interrupt

0	Disables
1	Enables

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset	0	0	0	0
	CE reset	Retained			
Clock stop		Retained			

Figure 12-16. Configuration of Interrupt Enable Register 3

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Interrupt enable 3	IP2	IP1	IP0	IPCE	2FH	R/W

IPCE — Enables or disables CE pin interrupt

0	Disables
1	Enables

IP0 — Enables or disables INT0 pin interrupt

0	Disables
1	Enables

IP1 — Enables or disables INT1 pin interrupt

0	Disables
1	Enables

IP2 — Enables or disables INT2 pin interrupt

0	Disables
1	Enables

		IP2	IP1	IP0	IPCE
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset	0	0	0	0
	CE reset	Retained			
Clock stop		Retained			

12.2.3 Vector address generator (VAG)

The vector address generator generates a branch address (vector address) of the program memory corresponding to an interrupt source that has been accepted from the corresponding peripheral hardware.

Table 12-1 shows the vector addresses of the respective interrupt sources.

Table 12-1. Interrupt Sources and Vector Addresses

Interrupt Source	Vector Address
Falling edge of CE pin	00CH
INT0 pin	00BH
INT1 pin	00AH
INT2 pin	009H
INT3 pin	008H
INT4 pin	007H
Timer 0	006H
Timer 1	005H
Timer 2	004H
Timer 3	003H
Serial interface 2	002H
Serial interface 3	001H

12.3 Interrupt Stack Register

12.3.1 Configuration and function of interrupt stack register

Figure 12-17 shows the configuration of the interrupt stack register.

The interrupt stack register saves the contents of the following system registers (except the address register (AR)) when an interrupt is accepted.

- Window register (WR)
- Bank register (BANK)
- Index register (IX)
- General pointer (RP)
- Program status word (PSWORD)

When an interrupt is accepted and the contents of the above system registers are saved to the interrupt stack, the contents of the above system registers, except the window register, are reset to “0”.

The interrupt stack can save the contents of the above system registers at up to four levels.

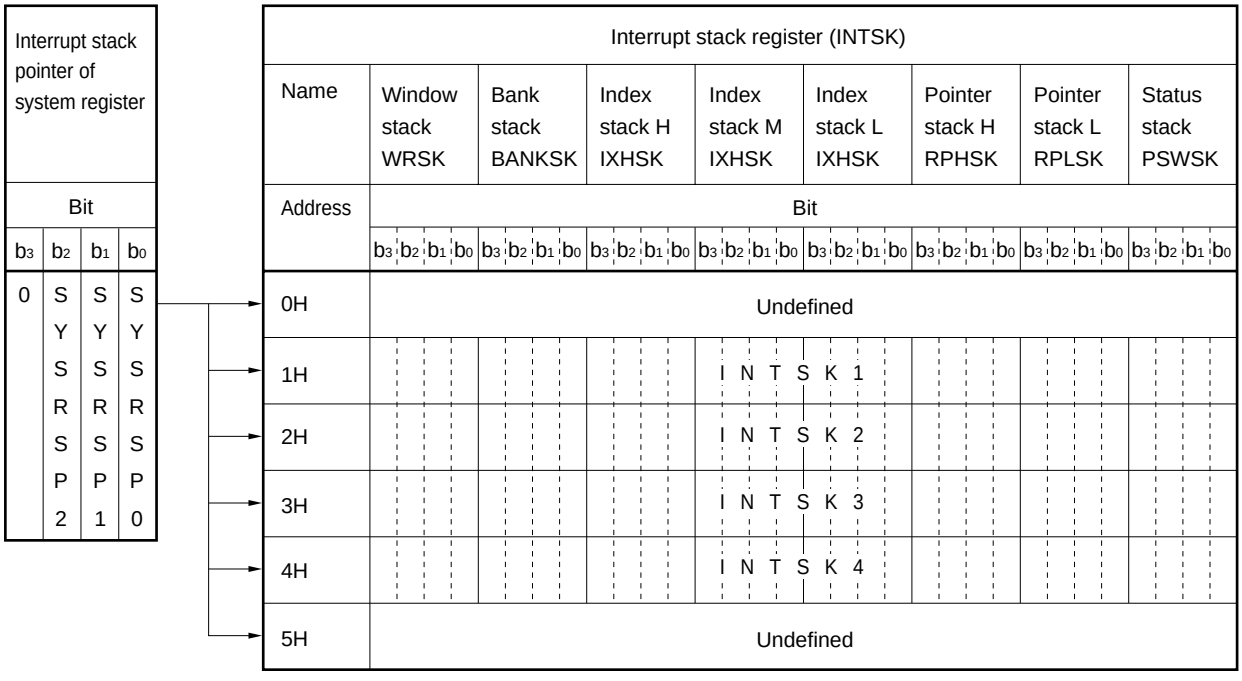
Therefore, interrupts can be nested up to four levels.

The contents of the interrupt stack register are restored to the system registers when the interrupt return (RETI) instruction is executed.

The contents of the interrupt stack register are undefined at power-ON reset.

The previous contents are retained at CE reset and on execution of the clock stop instruction.

Figure 12-17. Configuration of Interrupt Stack Register



12.3.2 Interrupt stack pointer of system register

The interrupt stack pointer of the system register detects the nesting level of interrupts. The interrupt stack pointer can be only read and cannot be written.

The configuration and function of the interrupt stack pointer are illustrated below.

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Interrupt stack pointer of system registers	0	SYSRSP2	SYSRSP1	SYSRSP0	08H	R

Detects level of interrupt stack of system registers			
0	0	0	Use prohibited
0	0	1	4 levels (INTSK1)
0	1	0	3 levels (INTSK2)
0	1	1	2 levels (INTSK3)
1	0	0	1 level (INTSK4)
1	0	1	0 level

Fixed to “0”

At reset	Power-ON reset	0	1	0	1
	WDT&SP reset		1	0	1
	CE reset		1	0	1
Clock stop		Retained			

12.3.3 Interrupt stack operation

Figure 12-8 shows the operation of the interrupt stack.

When nested interrupts exceeding four levels are accepted, since the contents saved first are discarded they therefore must be saved by the program.

Figure 12-18. Operation of Interrupt Stack (1/2)

(a) Where interrupt nesting level is 4 or less

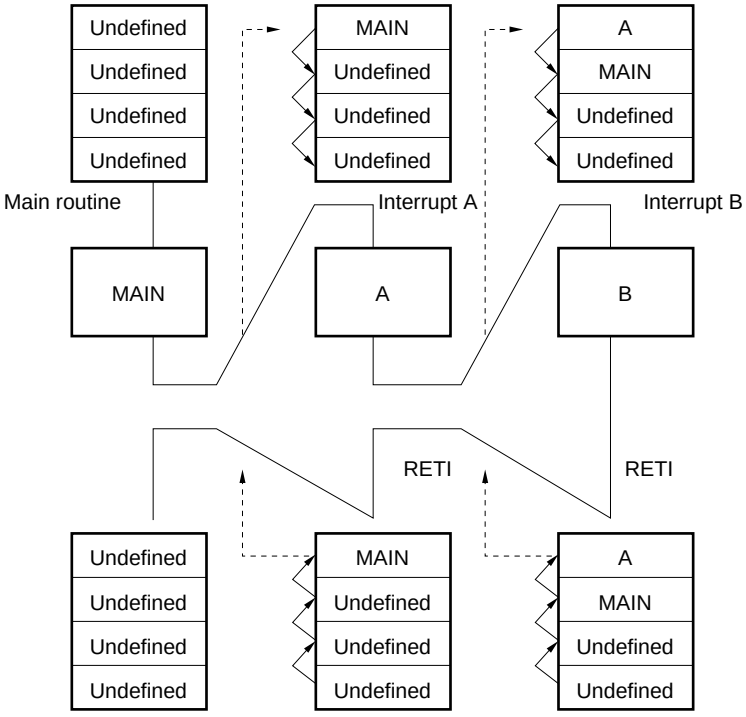
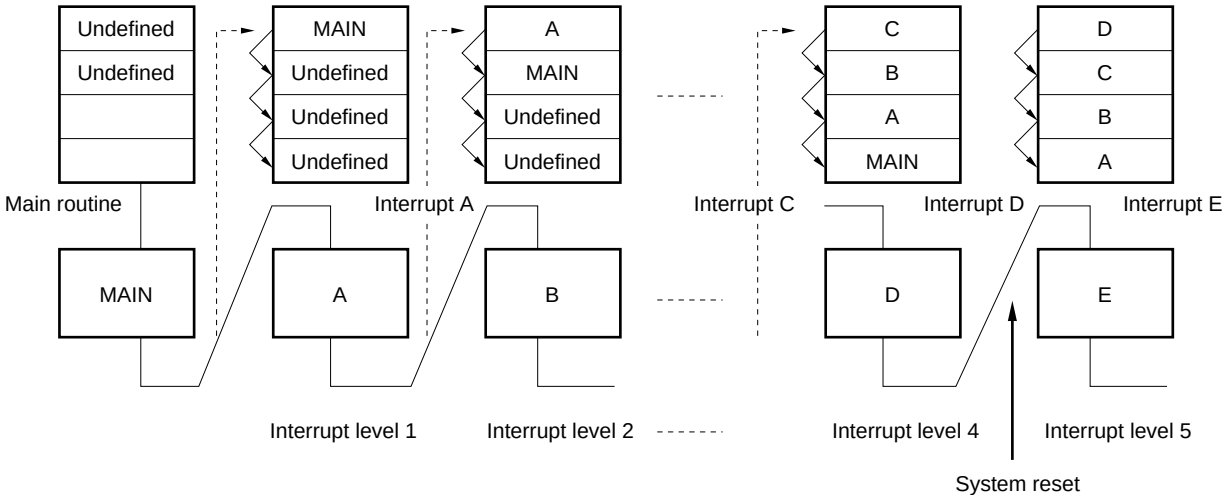


Figure 12-18. Operation of Interrupt Stack (2/2)

(b) Where interrupt nesting level is 5 or more



Caution The system is reset when an interrupt of level 5 is accepted.
However, the ISPRES flag, which resets the non-maskable interrupt if the interrupt stack overflows or underflows, must be set to "1". This flag is "1" after system reset, and can then be written only once.

12.4 Stack Pointer, Address Stack Registers, and Program Counter

The address stack registers save a return address when execution returns from an interrupt routine.

The stack pointer specifies the address of an address stack register.

When an interrupt is accepted, the value of the stack pointer is decremented by one, and the value of the program counter at that time is saved to an address stack register specified by the stack pointer.

Next, the interrupt routine is executed. When the interrupt return (RETI) instruction is executed after that, the contents of an address stack register specified by the stack pointer are restored to the program counter, and the value of the stack pointer is incremented by one.

For further information, also refer to **3. ADDRESS STACK (ASK)**.

12.5 Interrupt Enable Flip-Flop (INTE)

The interrupt enable flip-flop enables or disables the 12 types of maskable interrupts.

When this flip-flop is set, all the interrupts are enabled. When it is reset, all the interrupts are disabled.

This flip-flop is set or reset by dedicated instructions EI (to set) and DI (to reset).

The EI instruction sets this flip-flop when the instruction next to EI is executed, and the DI instruction resets the flip-flop while it is being executed.

When an interrupt is accepted, this flip-flop is automatically reset.

This flip-flop is also reset at power-ON reset, at a reset by the $\overline{\text{RESET}}$ pin, at a watchdog timer, overflow or underflow of the stack, and at CE reset. The flip-flop retains the previous status on execution of the clock stop instruction.

12.6 Accepting Interrupt

12.6.1 Accepting interrupt and priority

The following operations are performed before an interrupt is accepted.

(1) Each peripheral hardware unit outputs an interrupt request signal to the corresponding interrupt request block if a given interrupt condition (for example, input of the falling signal to the INT0 pin) is satisfied.
(2) When each interrupt request block accepts an interrupt request signal from the corresponding peripheral hardware unit, it sets the corresponding interrupt request flag (for example, IRQ0 flag if it is the INT0 pin that has issued the interrupt request) to “1”.
(3) The interrupt enable flag corresponding to each interrupt request flag (for example, IP0 flag if the interrupt request flag is IRQ0) is set to “1” when each interrupt request flag is set to “1”, and each interrupt request block outputs “1”.
(4) The signal output by the interrupt request block is ORed with the output of the interrupt enable flip-flop, and an interrupt accept signal is output.
This interrupt enable flip-flop is set to “1” by the EI instruction, and reset to “0” by the DI instruction.
If “1” is output by each interrupt request processing block while the interrupt enable flip-flop is set to “1”, the interrupt is accepted.

As shown in Figure 12-1, the output of the interrupt enable flip-flop is input to each interrupt request block via an AND circuit when an interrupt is accepted.

The signal input to each interrupt request block causes the interrupt request flag corresponding to each interrupt request flag to be reset to “0” and the vector address corresponding to each interrupt to be output.

If the interrupt request block outputs “1” at this time, the interrupt accept signal is not transferred to the next stage. If two or more interrupt requests are issued at the same time, therefore, the interrupts are accepted according to the priority shown in Table 12-2.

Unless the interrupt request enable flag is set to “1”, the corresponding interrupt is not accepted.

Therefore, by resetting the interrupt enable flag to “0”, the interrupt with a high hardware priority can be disabled.

Table 12-2. Interrupt Priority

Interrupt Source	Priority
Falling edge of CE pin	1
INT0 pin	2
INT1 pin	3
INT2 pin	4
INT3 pin	5
INT4 pin	6
Timer 0	7
Timer 1	8
Timer 2	9
Timer 3	10
Serial interface 2	11
Serial interface 3	12

12.6.2 Timing chart when interrupt is accepted

The timing charts in Figure 12-19 illustrate the operations performed when an interrupt or interrupts are accepted.

Figure 12-19 (1) is the timing chart when one interrupt is accepted.

(a) in (1) is the timing chart where the interrupt request flag is set to “1” after all the others, and (b) is the timing chart where the interrupt enable flag is set to “1” after all the others.

In either case, the interrupt is accepted when the interrupt request flag, interrupt enable-flip flop, and interrupt enable flag all have been set to “1”.

If the flag or flip-flop that has been set last is set in the first instruction cycle of the “MOVT DBF, @AR” instruction or by an instruction that satisfies a given skip condition, the interrupt is accepted in the second instruction cycle of the “MOVT DBF, @AR” instruction or after the instruction that is skipped (this instruction is treated as NOP) has been executed.

The interrupt enable flip-flop is set in the instruction cycle next to that in which the EI instruction is executed.

Therefore, the interrupt is accepted after the instruction next to the EI instruction has been executed even when the interrupt request flag is set in the execution cycle of the EI instruction.

(2) in Figure 12-19 is the timing chart where two or more interrupts are used.

When two or more interrupts are used, the interrupts are accepted according to the hardware priority if all the interrupt enable flags are set. However, the hardware priority can be changed by setting the interrupt enable flags by the program.

“Instruction cycle” shown in Figure 12-19 is a special cycle in which the interrupt request flag is reset, a vector address is specified, and the contents of the program counter are saved after an interrupt has been accepted. It takes 1.78 μs, which is equivalent to one instruction execution time, to be completed.

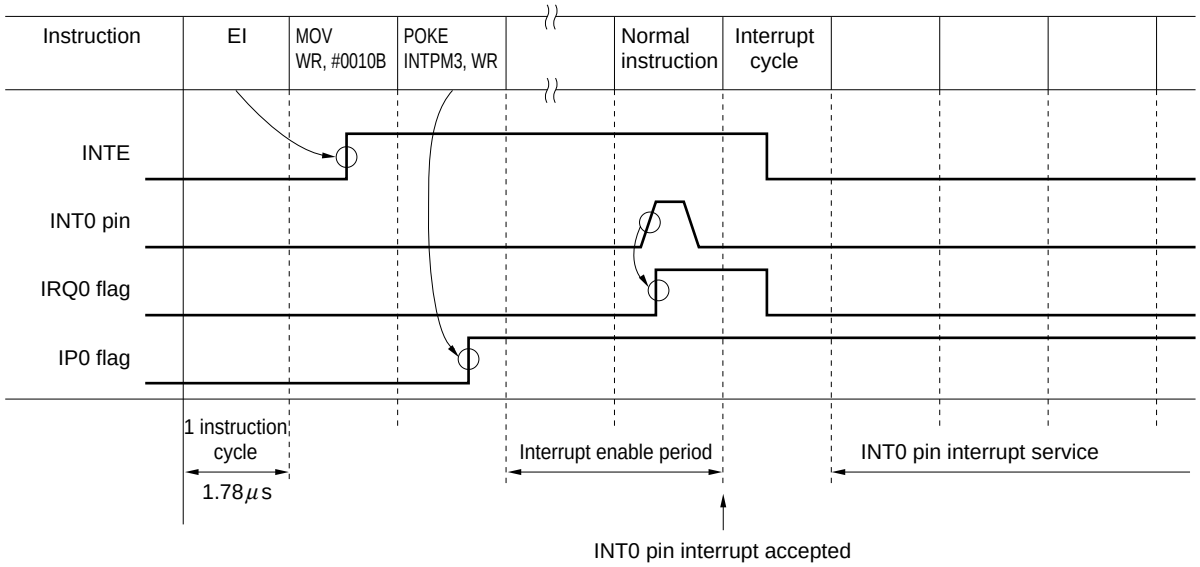
For details, refer to **12.7 Operation after Interrupt Has Been Accepted**.

Figure 12-19. Timing Charts When Interrupt Is Accepted (1/3)

(1) When one interrupt (e.g., rising of INT0 pin) is used

(a) If there is no interrupt mask time by the interrupt flag (IP×××)

<1> If a normal instruction which is not "MOVT" or an instruction that satisfies a skip condition is executed when interrupt is accepted



<2> If "MOVT" or an instruction that satisfies a skip condition is executed when interrupt is accepted

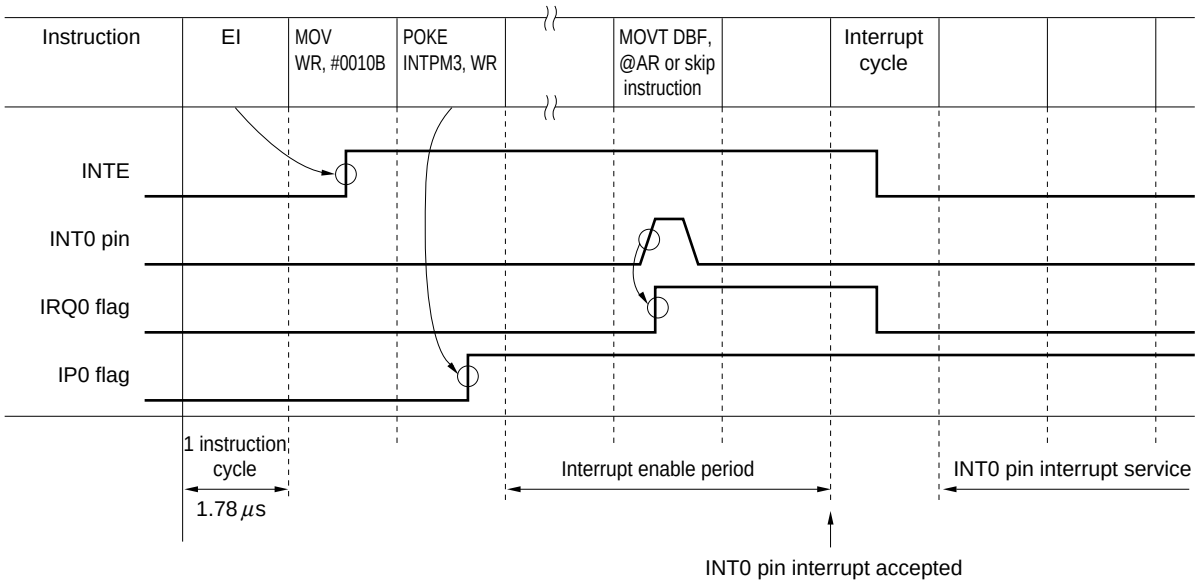
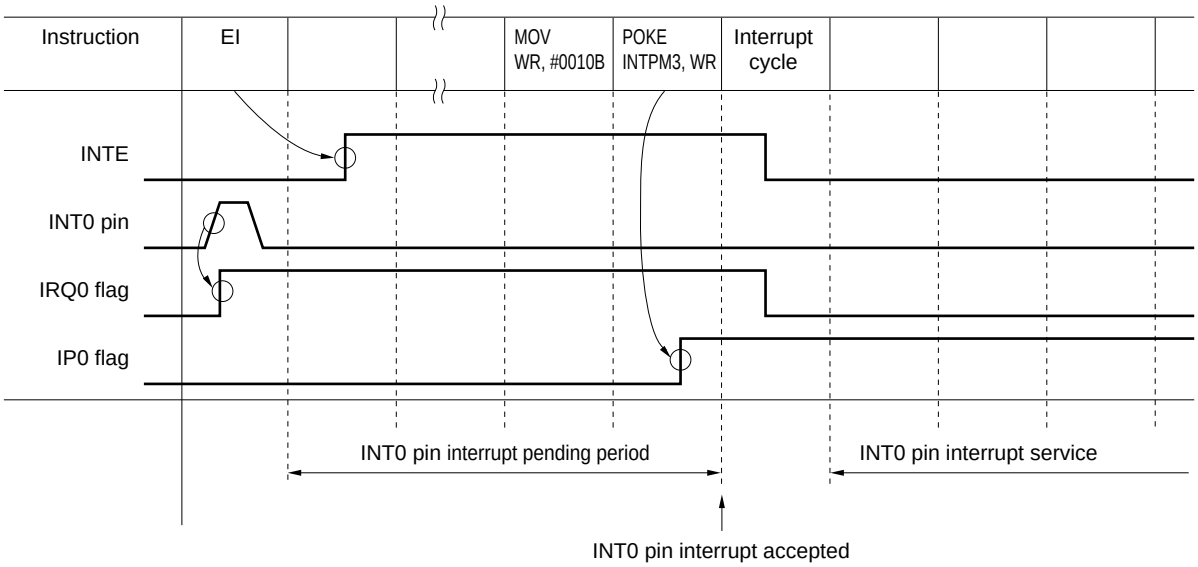


Figure 12-19. Timing Charts When Interrupt Is Accepted (2/3)

(b) If interrupt is kept pending by the interrupt enable flag



(2) If two or more interrupts (e.g., INT0 pin and INT1 pin) are used

(a) Hardware priority

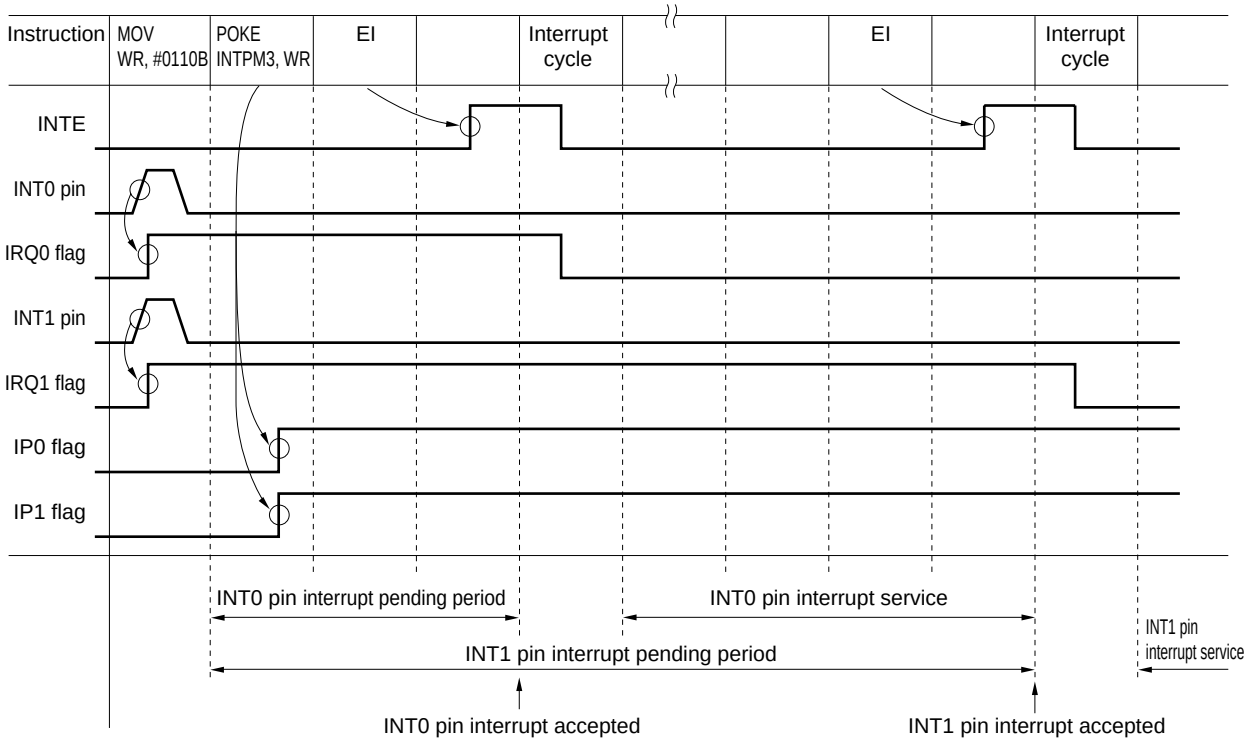
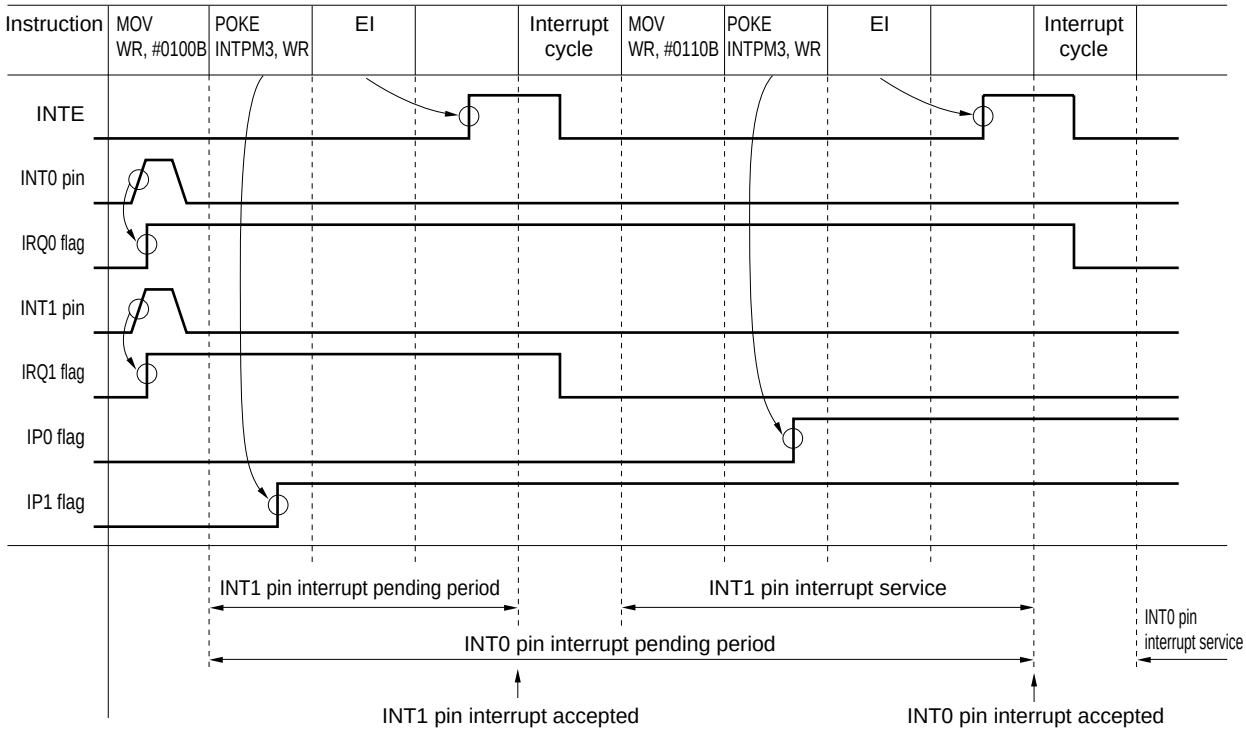


Figure 12-19. Timing Charts When Interrupt Is Accepted (3/3)

(b) Software priority



12.7 Operations after Interrupt Has Been Accepted

When an interrupt is accepted, the following operations are sequentially performed automatically.

(1) The interrupt enable flip-flop and the interrupt request flag corresponding to the accepted interrupt request are reset to "0". As a result, the other interrupts are disabled.
(2) The contents of the stack pointer are decremented by one.
(3) The contents of the program counter are saved to an address stack register specified by the stack pointer. At this time, the contents of the program counter are the program memory address after the address at which the interrupt has been accepted.
For example, if a branch instruction is executed when the interrupt has been accepted, the contents of the program counter are the branch destination address. If a subroutine call instruction is executed, the contents of the program counter are the call destination address. If the skip condition of a skip instruction is satisfied, the next instruction is executed as NOP and then the interrupt is accepted. Consequently, the contents of the program counter are the address after that of the instruction that is skipped.
(4) The contents of the system registers (except the address register) are saved to the interrupt stack.
(5) The contents of the vector address generator corresponding to the interrupt that has been accepted are transferred to the program counter. In other words, execution branches to the interrupt routine.

The operations (1) through (5) above require the time of one special instruction cycle (1.78 μs) in which normal instruction execution is not performed.

This instruction cycle is called an "interrupt cycle".

In other words, the time of one instruction cycle (1.78 μs) is required after an interrupt has been accepted until execution branches to the corresponding vector address.

12.8 Returning from Interrupt Routine

The interrupt return (RETI) instruction is used to return from an interrupt routine to the processing during which an interrupt was accepted.

When the RETI instruction is executed, the following operations are sequentially performed automatically.

(1) The contents of an address stack register specified by the stack pointer are restored to the program counter.
(2) The contents of the interrupt stack are restored to the system registers.
(3) The contents of the stack pointer are incremented by one.

The operations (1) through (3) above require one instruction cycle (1.78 μs) in which the RETI instruction is executed.

The only difference between the RETI instruction and the RET and RETSK instructions, which are subroutine return instructions, is the restoration of the bank register and index register in step (2) above.

12.9 External Interrupts (CE and INT0 through INT4 pins)

12.9.1 Outline of external interrupts

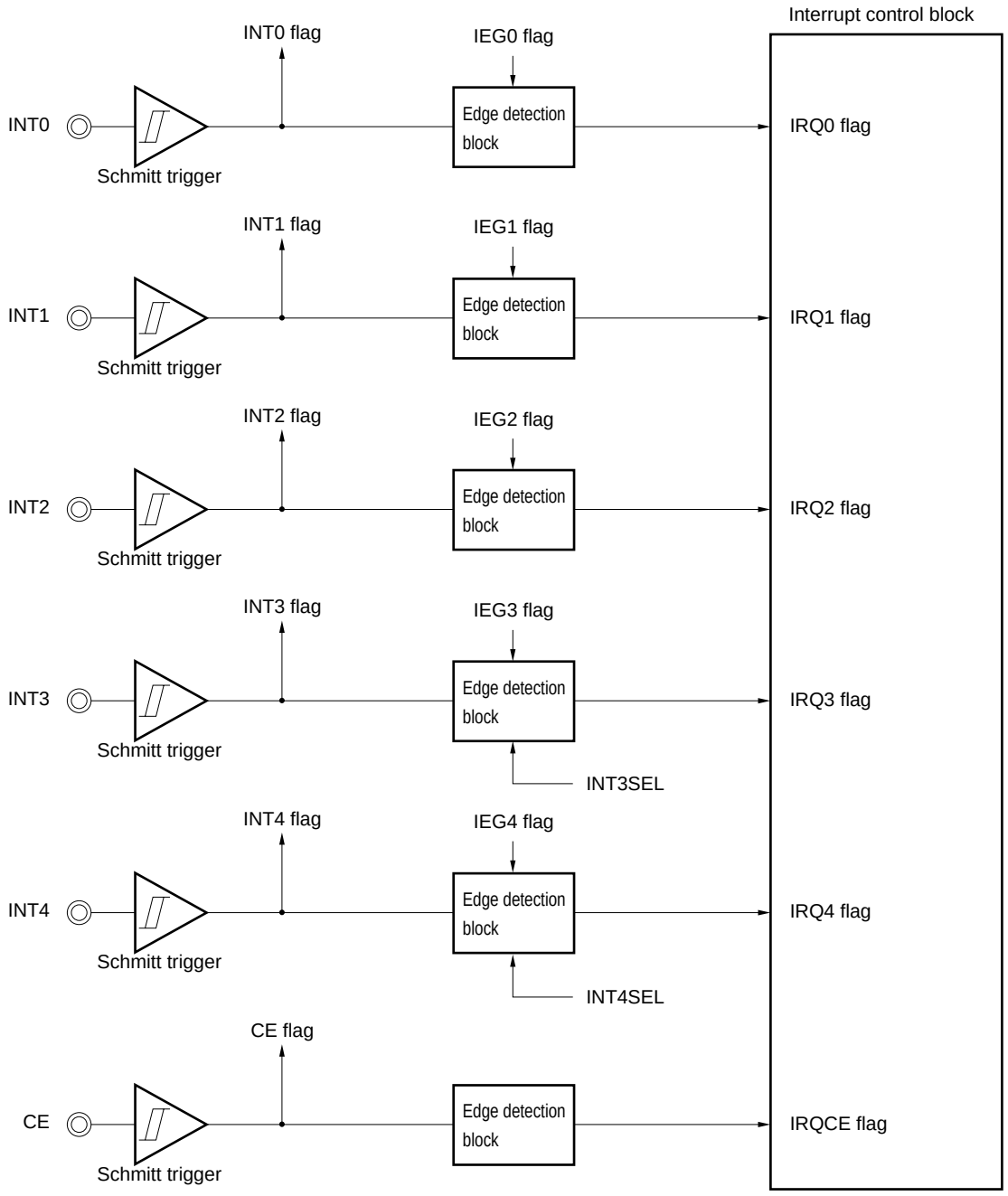
Figure 19-20 outlines the external interrupts.

As shown in the figure, external interrupt requests are issued at the rising or falling edges of signals input to the INT0 through INT4 pins, and at the falling edge of the CE pin.

Whether an interrupt request is issued at the rising or falling edge of an INT pin is independently specified by the program.

The INT0 through INT4 and CE pins are Schmitt trigger input pins to prevent malfunctioning due to noise. These pins do not accept a pulse input of less than 100 ns.

Figure 12-20. Outline of External Interrupts



12.9.2 Edge detection block

The edge detection block specifies the valid edge (rising or falling edge) of an input signal that issues the interrupt request of INT0 to INT4 pins, by using an interrupt edge selection register.

Figure 12-21 shows the configuration and function of the interrupt edge selection register.

Figure 12-21. Configuration of Interrupt Edge Selection Register (1/2)

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Interrupt edge selection 1	IEG4	INT4SEL	IEG3	INT3SEL	1EH	R/W

INT3SEL — Selects function of P1A2/INT3 pin

0	Interrupt pin (edge detection cricuit operates)
1	General-purpose port pin (edge detection cricuit stops)

IEG3 — Selects input edge to issue interrupt request (INT3 pin)

0	Rising edge
1	Falling edge

INT4SEL — Selects function of P1A3/INT4 pin

0	Interrupt pin (edge detection cricuit operates)
1	General-purpose port pin (edge detection cricuit stops)

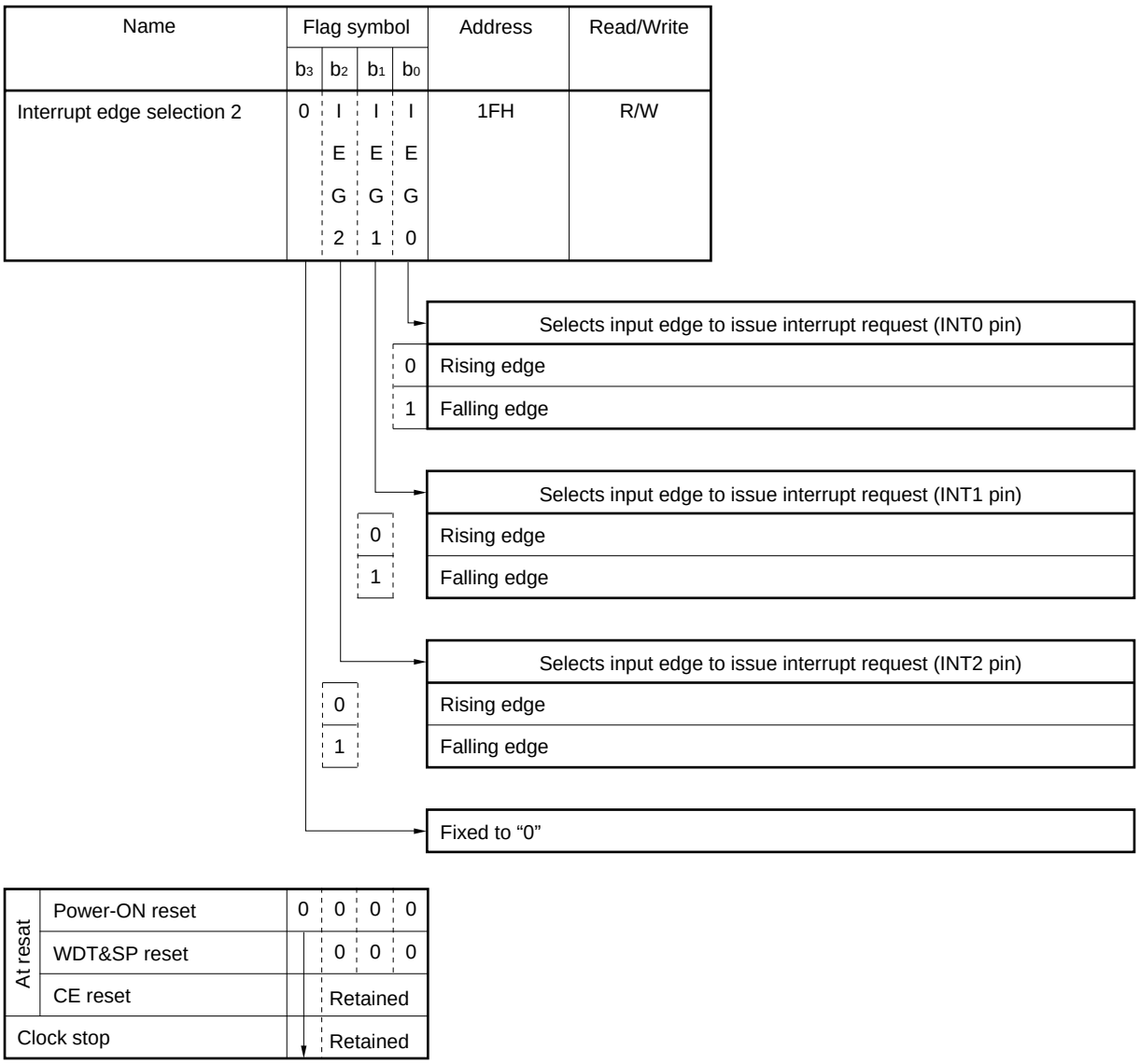
IEG4 — Selects input edge to issue interrupt request (INT4 pin)

0	Rising edge
1	Falling edge

		b_3	b_2	b_1	b_0
At resat	Power-ON reset	0	0	0	0
	WDT&SP reset	0	0	0	0
	CE reset	Retained			
Clock stop		Retained			

Caution The external input delays about 100 ns.

Figure 12-21. Configuration of Interrupt Edge Selection Register (2/2)



Caution The external input is delayed about 100 ns.

Note that an interrupt request signal may be issued at the time when the interrupt request issuance edge is switched by the interrupt edge selection flags (IEG0 through IEG4).

As indicated in the table 12-3, for example, if the IEG0 flag is set to “1” (falling edge), the high level is input from the INT0 pin and the IEG0 flag is reset to “0”, the edge detection circuit judges that the rising edge is input and an interrupt request is issued.

Table 12-3. Issuance of Interrupt Request by Changing IEG Flag

Changes in IEG0 through IEG4 Flags	Status of INT0 through INT4 Pins	Issuance of Interrupt Request	Status of Interrupt Request Flag
1 → 0 (Falling)	Low level	Not issued	Retains previous status
	High level	Issued	Set to "1"
0 → 1 (Rising)	Low level	Issued	Set to "1"
	High level	Not issued	Retains previous status

12.9.3 Interrupt control block

The signal levels that are input to the INT0 through INT4 pins can be detected by using the INT0 through INT4 flags.

Because these flags are reset independently of interrupts, when the interrupt function is not used the INT0 through INT2 pins can be used as a 3-bit input port, and P1A2/INT3 and P1A3/INT4 pins can be used as a 2-bit general-purpose input port.

If the interrupts are not enabled, these ports can be used as general-purpose port pins whose rising or falling edge can be detected by reading the corresponding interrupt request flags.

At this time, however, the interrupt request flags are not automatically reset and must be reset by the program.

For further information, also refer to **12.2.1 Configuration and function of interrupt request flag (IRQ×××)**.

12.10 Internal Interrupts

The following six internal interrupts are available.

- Timer 0
- Timer 1
- Timer 2
- Timer 3
- Serial interface 2
- Serial interface 3

12.10.1 Timer 0, timer 1, timer 2, and timer 3 interrupts

Interrupt requests are issued at fixed intervals.

For details, refer to **13. TIMER**.

12.10.2 Serial interface 2 and serial interface 3 interrupts

Interrupt requests can be issued at the end of a serial output or serial input operation.

For details, refer to **16. SERIAL INTERFACE**.

13. TIMERS

Timers are used to manage the program execution time.

13.1 Outline of Timers

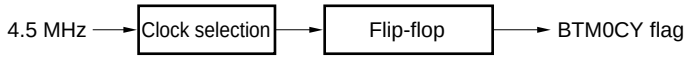
Figure 13-1 outlines the timers.
The following five timers are available.

- Basic timer 0
- Timer 0
- Timer 1
- Timer 2
- Timer 3

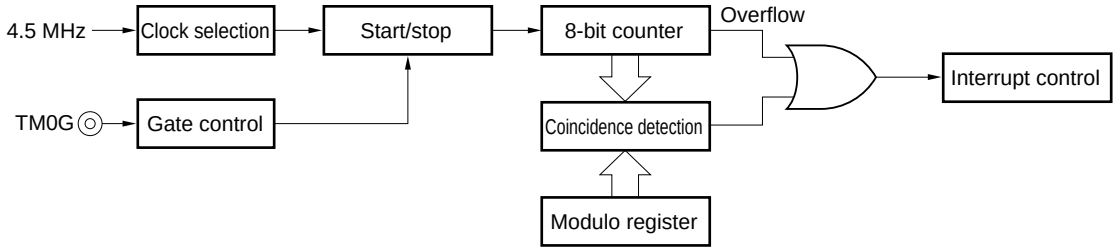
Basic timer 0 detects the status of a flip-flop that is set at fixed time intervals in software.
Timers 0 through 3 are modulo timers and can use interrupts.
Basic timer 0 can also be used to detect a power failure. Timer 3 is multiplexed with the D/A converter.
The clock of each timer is created by dividing the system clock (4.5 MHz).

Figure 13-1. Outline of Timers (1/2)

(1) Basic timer 0



(2) Timer 0



(3) Timer 1

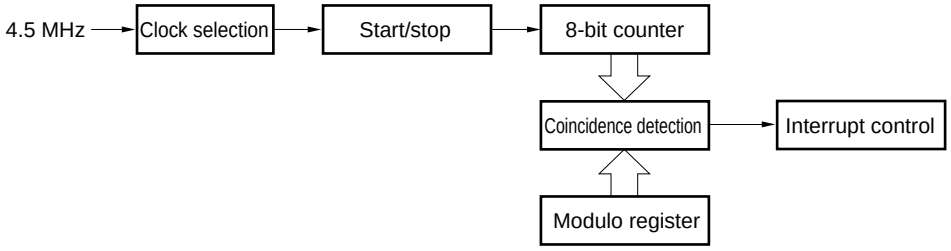
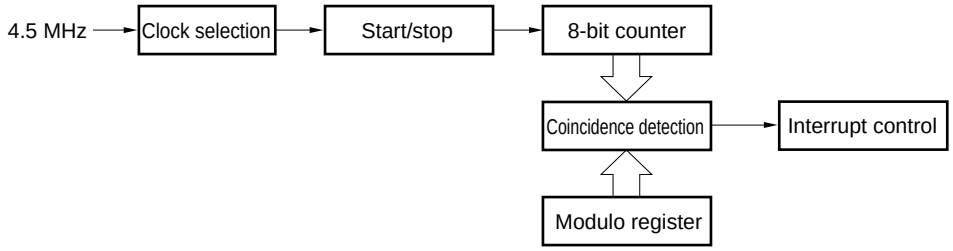
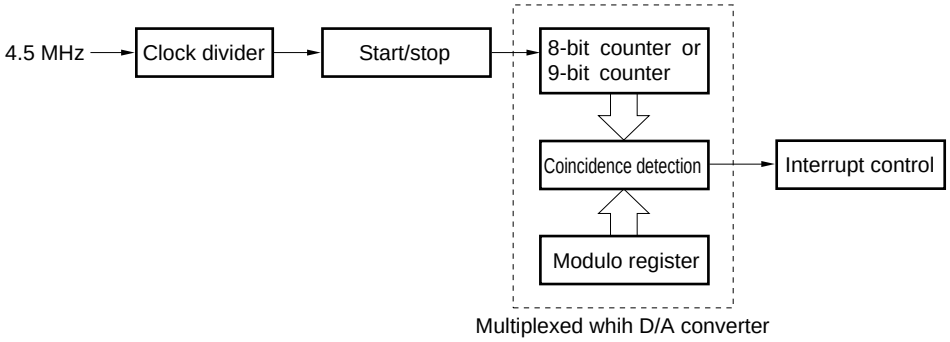


Figure 13-1. Outline of Timers (2/2)

(4) Timer 2



(5) Timer 3



13.2 Basic Timer 0

13.2.1 Outline of basic timer 0

Figure 13-2 outlines basic timer 0.

Basic timer 0 is used as a timer by detecting in software the BTM0CY flag that is set at fixed intervals (100, 50, 20, or 10 ms).

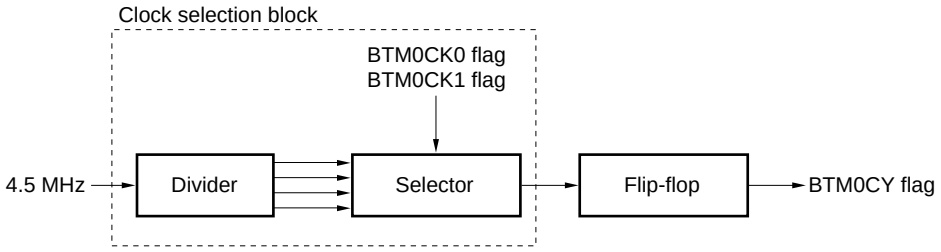
If the BTM0CY flag is read first after power-ON reset, “0” is always read. After that, the flag is set to “1” at fixed intervals.

If the CE pin goes high, CE reset is effected in synchronization with the timing at which the BTM0CY flag is set next.

Therefore, a power failure can be detected by reading the content of the BTM0CY flag at system reset (power-ON reset or CE reset).

For the details of power failure detection, refer to **21. RESET**.

Figure 13-2. Outline of Basic Timer 0



Remarks 1. BTM0CK1 and BTM0CK0 (bits 1 and 0 of basic timer 0 clock selection register: refer to **Figure 13-3**) set the time intervals at which the BTM0CY flag is set.

2. BTM0CY (bit 0 of basic timer 0 carry register: refer to **Figure 13-4**) detects the status of the flip-flop.

13.2.2 Clock selection block

The clock selection block divides the system clock (4.5 MHz) and sets the time interval at which the BTM0CY flag is to be set, by using the BTM0CK0 and BTM0CK1 flags.

Figure 13-3 shows the configuration of the basic timer 0 clock selection register.

Figure 13-3. Configuration of Basic Timer 0 Clock Selection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Basic timer 0 clock selection	0	0	BTM0CK1	BTM0CK0	18H	R/W

b_1	b_0	Sets time interval at which BTM0CY flag is set
0	0	10 Hz (100 ms)
0	1	20 Hz (50 ms)
1	0	50 Hz (20 ms)
1	1	100 Hz (10 ms)

Fixed to “0”

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset			0	0
	CE reset			Retained	
Clock stop				Retained	

13.2.3 Flip-flop and BTM0CY flag

The flip-flop is set at fixed intervals and its status is detected by the BTM0CY flag of the basic timer 0 carry register.

When the BTM0CY flag is read, it is reset to “0” (Read & Reset).

The BTM0CY flag is “0” at power-ON reset, and is “1” at CE reset and on execution of the clock stop instruction. Therefore, this flag can be used to detect a power failure.

The BTM0CY flag is not set after power application until an instruction that reads it is executed. Once the read instruction has been executed, the flag is set at fixed intervals.

Figure 13-4 shows the configuration of the basic timer 0 carry register.

Figure 13-4. Configuration of Basic Timer 0 Carry Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Basic timer 0 carry	0	0	0	BTM0CY	17H	R & Reset

Detects status of flip-flop	
0	Flip-flop is not set
1	Flip-flop is set

Fixed to “0”

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset				R
	CE reset				1
Clock stop					R

R: Retained

13.2.4 Example of using basic timer 0

An example of a program using basic timer 0 is shown below.

This program executes processing A every 1 second.

Example

```
        CLR2   BTM0CK1, BTM0CK0   ; Sets BTM0CY flag setting pulse to 10 Hz (100 ms)
        MOV    M1, #0
LOOP:
        SKT1   BTM0CY             ; Branches to NEXT if BTM0CY flag is “0”
        BR     NEXT
        ADD    M1, #1             ; Adds 1 to M1
        SKE    M1, #0AH           ; Executes processing A if M1 is “10” (1 second has elapsed)
        BR     NEXT
        MOV    M1, #0

        Processing A

NEXT:

        Processing B              ; Executes processing B and branches to LOOP

        BR     LOOP
```

13.2.5 Errors of basic timer 0

Errors of basic timer 0 include an error due to the detection time of the BTM0CY flag, and an error that occurs when the time interval at which the BTM0CY flag is to be set is changed.
The following paragraphs (1) and (2) describe each error.

(1) Error due to detection time of BTM0CY flag

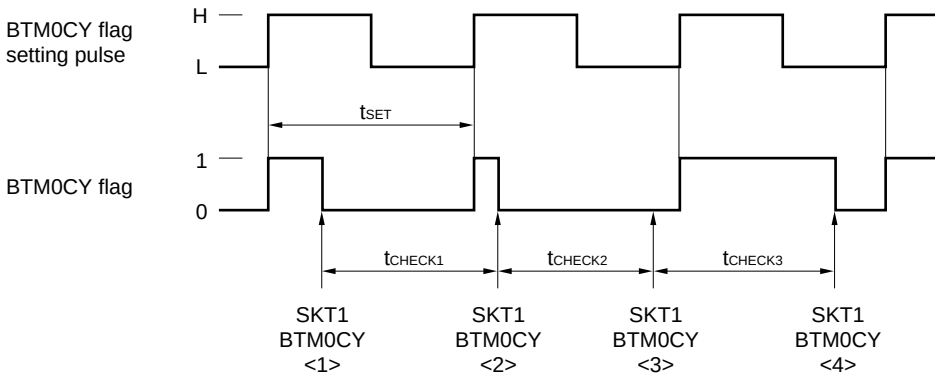
The time to detect the BTM0CY flag must be shorter than the time at which the BTM0CY flag is set (refer to **13.2.6 Notes on using basic timer 0**).
Where the time interval at which the BTM0CY flag is detected is t_{CHECK} and the time interval at which the flag is set is t_{SET} (100, 50, 20, or 10 ms), t_{CHECK} and t_{SET} must relate as follows.

$$t_{CHECK} < t_{SET}$$

At this time, the error of the timer when the BTM0CY flag is detected is as follows, as shown in Figure 13-5.

$$0 < \text{Error} < t_{SET}$$

Figure 13-5. Error of Basic Timer 0 due to Detection Time of BTM0CY Flag



As shown in Figure 13-5, the timer is updated because BTM0CY flag is “1” when it is detected in step <2>.
When the flag is detected next in step <3>, it is “0”. Therefore, the timer is not updated until the flag is detected again in <4>.
This means that the timer is extended by the time of t_{CHECK3}.

(2) Error when time interval to set BTM0CY flag is changed

The BTM0CK1 and BTM0CK0 flags set the time of the BTM0CY flag.

As described in 13.2.2, four types of timer time-setting pulses can be selected: 10 Hz, 20 Hz, 50 Hz, and 100 Hz.

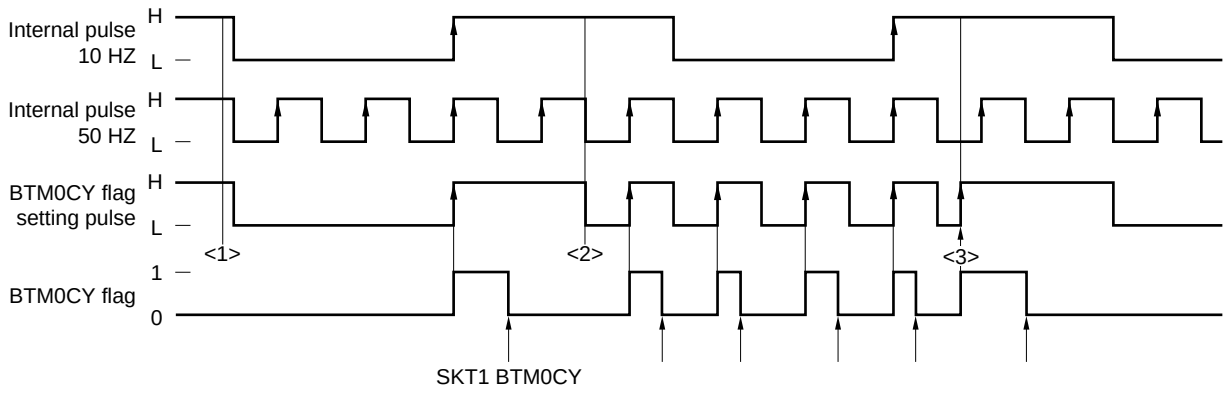
At this time, these four pulses operate independently. If the timer time-setting pulse is changed by using the BTM0CK1 and BTM0CK0 flags, an error occurs as described in the example below.

Example

```
; <1>
  INTIFLG NOT BTM0CK1, NOT BTM0CK0
                                    ; Sets BTM0CY flag setting pulse to 10 Hz (100 ms)

  Processing A

; <2>
  INITFLG BTM0CK1, NOT BTM0CK0  ; Sets BTM0CY flag setting pulse to 50 Hz (20 ms)

  Processing A

; <3>
  INITFLG NOT BTM0CK1, NOT BTM0CK0
                                    ; Sets BTM0CY flag setting pulse to 10 Hz (100 ms)
```

At this time, the BTM0CY flag setting pulse is changed as shown in Figure 13-6.

Figure 13-6. Changing BTM0CY Flag Setting Pulse



As shown in Figure 13-6, if the BTM0CY flag setting time is changed and the new pulse falls, the BTM0CY flag retains the previous status (<2> in the figure). If the new pulse rises, however, the BTM0CY flag is set to “1” (<3> in the figure).

Although changing the pulse setting between 10 Hz (100 ms) and 50 Hz (20 ms) is described in this example, the same applies to changing the pulse in respect to 20 Hz (50 ms) and 100 Hz (10 ms).

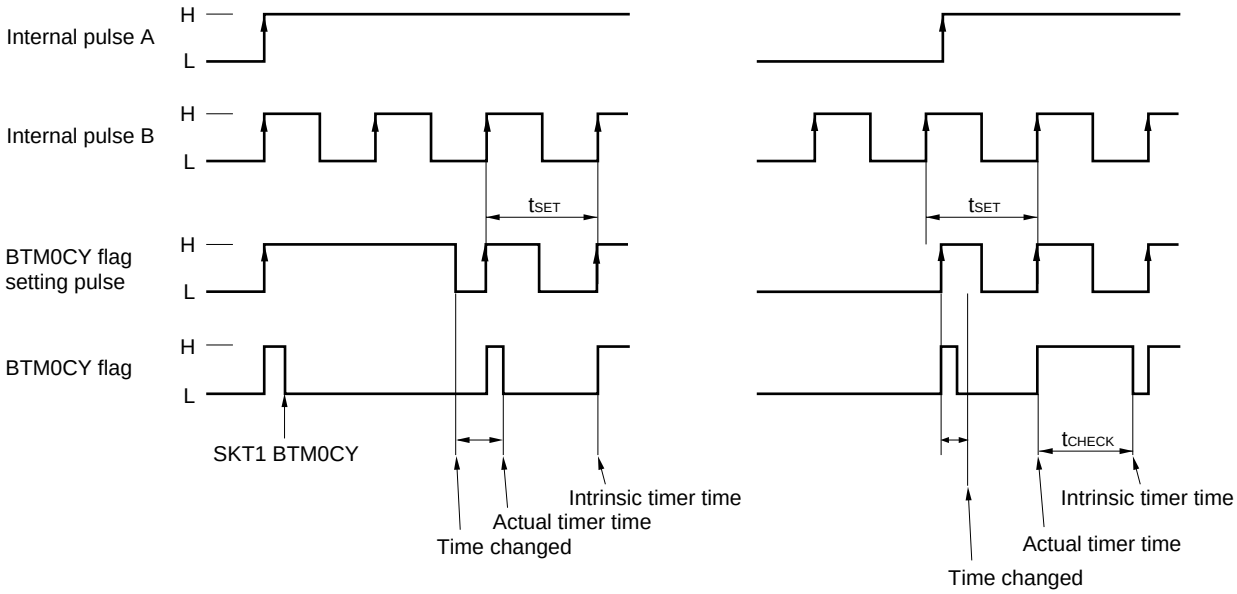
Therefore, as shown in Figure 13-7, the error of the time until the BTM0CY flag is first set after the BTM0CY flag setting time has been changed is as follows:

$-t_{SET} < \text{Error} < t_{CHECK}$

t_{SET} : new setting time of BTM0CY flag
t_{CHECK}: time to detect BTM0CY flag

Phase differences are provided among the internal pules of 10, 20, 50, and 100 Hz. Because these phase differences are shorter than the newly set pulse time, they are included in the above error.

Figure 13-7. Timer Error When BTM0CY Flag Setting Time Is Changed from A to B



An error of $-t_{SET}$ occurs if BTM0CY flag is detected immediately after the timer time has been changed because the flag then becomes "1".

An error of t_{CHECK} occurs if the timer time is changed immediately after BTM0CY flag has been detected because the flag is then reset once.

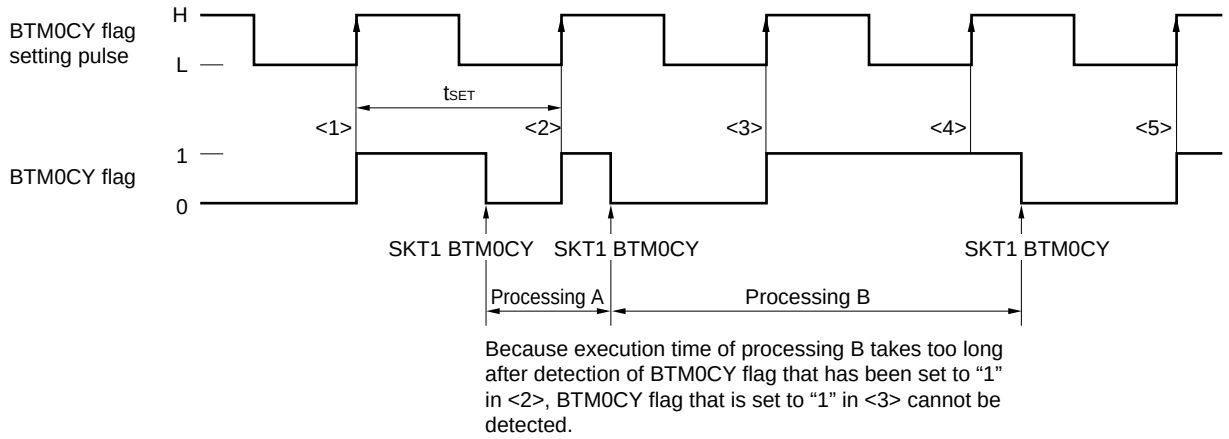
13.2.6 Cautions on using basic timer 0

(1) BTM0CY flag detection time interval

Keep the time to detect the BTM0CY flag shorter than the time at which the BTM0CY flag is set.

This is because, if the time of processing B is longer than the time interval at which the BTM0CY flag is set as shown in Figure 13-8, setting of the BTM0CY flag is overlooked.

Figure 13-8. BTM0CY Flag Detection and BTM0CY Flag



(2) Timer updating processing time and BTM0CY flag detection time interval

As described in (1) above, time interval t_{SET} at which the BTM0CY flag is detected must be shorter than the time for which to set the BTM0CY flag.

At this time, even if the time interval at which the BTM0CY flag is detected is short, if the updating processing time of the timer is long the processing of the timer may not be executed normally at CE reset. Therefore, the following condition must be satisfied.

$$t_{CHECK} + t_{TIMER} < t_{SET}$$

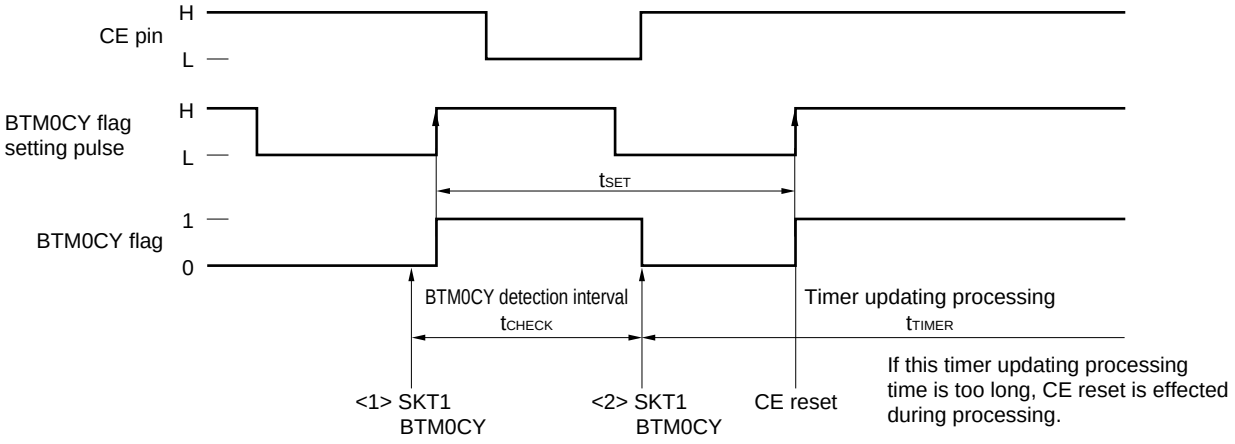
t_{CHECK}: time to detect BTM0CY flag
t_{TIMER} : timer updating processing time
t_{SET} : time to set BTM0CY flag

An example is given below.

Example Example of timer updating processing and BTM0CY flag detection time interval

```
START:
      CLR2   BTM0CK1, BTM0CK0   ; Sets BTM0CY flag setting pulse to 10 Hz (100 ms)
BTIMER:
      ; <1>
      SKT1   BTM0CY             ; Updates timer if BTM0CY flag is "1"
      BR     AAA

      [ Timer updating ]

      BR     BTIMER
AAA:

      [ Processing A ]

      BR     BTIMER
```

The timing chart of the above program is shown below.



(3) Compensating basic timer 0 carry at CE reset

Next, an example of compensating the timer at CE reset is described below.

As shown in the example below, the timer must be compensated at CE reset "if the BTM0CY flag is used for power failure detection and if the BTM0CY flag is used for a watch timer".

The BTM0CY flag is reset (to 0) first on power application (power-ON reset), and is disabled from being set until it is read once by the "PEEK" instruction.

If the CE pin goes high, CE reset is effected in synchronization with the rising edge of the BTM0CY flag setting pulse. At this time, the BTM0CY flag is set (to 1) and the timer is started.

By detecting the status of the BTM0CY flag at system reset (power-ON reset or CE reset), therefore, it can be identified whether a power-ON reset or CE reset has been effected (power failure detection). That is, power-ON reset has been effected if the flag is "0", and CE reset has been effected if it is "1".

At this time, the watch timer must continue operating even if CE reset has been effected.

However, because the BTM0CY flag is reset to 0 when it is read to detect a power failure, the set status (1) of the BTM0CY flag is overlooked once. If the delay function of CE reset is used, the value set to the CE reset timer carry counter (control register address 06H) is overlooked.

Consequently, the watch timer must be updated if CE reset is identified by means of power failure detection.

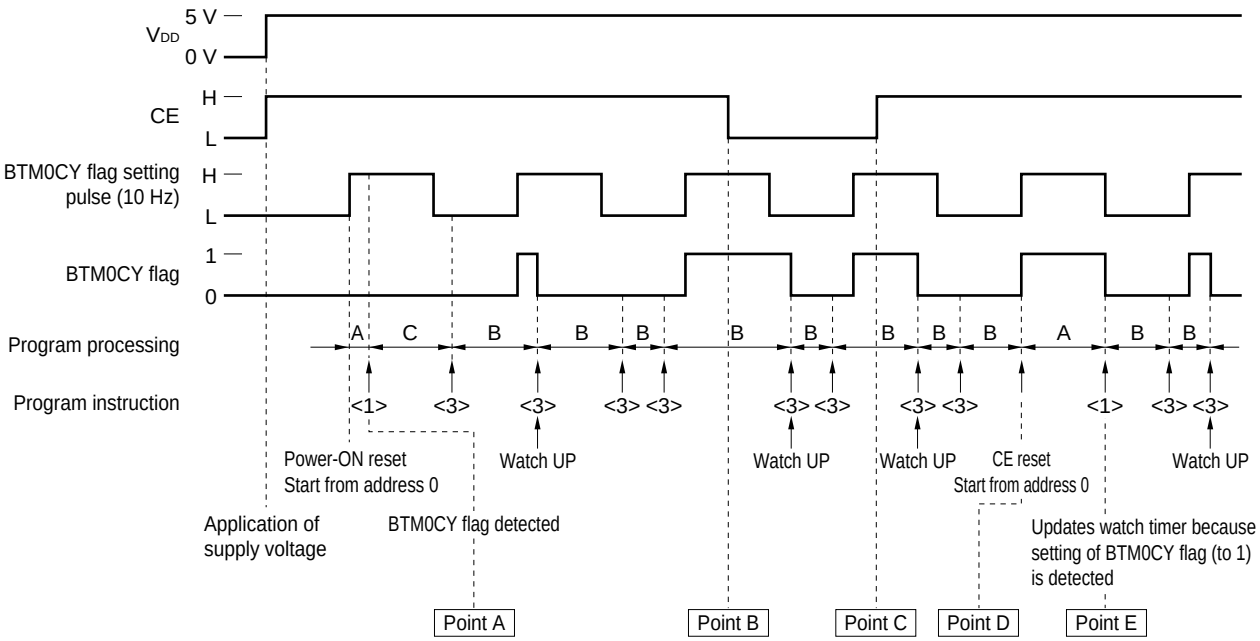
For the details of power failure detection, refer to **21. RESET**.

Example Example of compensating timer at CE reset (to detect power failure and update watch timer using BTM0CY flag)

```
START:                              ; Program address 0000H

        Processing A

      ; <1>
        SKT1   BTM0CY               ; Embedded macro
                                    ; Tests BTM0CY flag
        BR     INITIAL              ; if “0”, branches to INITIAL (power failure detection)
BACKUP:
      ; <2>

        100-ms watch updating       ; Compensates watch timer because of backup (CE reset)
                                    ; Initial value “1” is stored as CE reset timer carry
                                      ;      counter value

LOOP:
      ; <3>

        Processing B                : While performing processing B,

        SKF1   BTM0CY               ; tests BTM0CY flag and updates watch timer
        BR     BACKUP
        BR     LOOP
INITIAL:
        CLR2   BTM0CK1, BTM0CK0
                                    ; Embedded macro
                                    ; Because power failure (power-ON reset) occurs,
                                    ; sets setting time of BTM0CY flag to 100 ms, and
                                    ; executes processing C

        Processing C

        BR     LOOP
```

Figure 13-9 shows the timing chart of the above program.

Figure 13-9. Timing Chart



As shown in Figure 13-9, the program is started from address 0000H because the internal 10-Hz pulse rises when supply voltage V_{DD} is first applied.

When the BTM0CY flag is detected at point A, it is judged that the BTM0CY flag is reset (to 0) and that a power failure (power-ON reset) has occurred because the power has just been applied.

Therefore, “processing C” is executed, and the BTM0CY flag setting pulse is set to 100 ms.

Because the content of the BTM0CY flag is read once at point A, the BTM0CY flag will be set to 1 every 100 ms.

Next, even if the CE pin goes low at point B and high at point C, the program counts up the watch timer while executing “processing B”, unless the clock stop instruction is executed.

At point C, because the CE pin goes high, CE reset is effected at point D at which the BTM0CY flag setting pulse rises next time, and the program is started from address 0000H.

When the BTM0CY flag is detected at point E at this time, it is set to 1. Therefore, this is judged to be a back up (CE reset).

As is evident from the above figure, unless the watch is updated by 100 ms at point E, the watch is delayed by 100 ms each time CE reset is effected.

If processing A takes longer than 100 ms when a power failure is detected at point E, the setting of the BTM0CY flag is overlooked two times. Therefore, processing A must be completed within 100 ms.

The above description also applies when the BTM0CY flag setting pulse is set to 50, 20, or 10 ms.

Therefore, the BTM0CY flag must be detected for power failure detection within the BTM0CY flag setting time after the program has been started from address 0000H.

(4) If BTM0CY flag is detected at the same time as CE reset

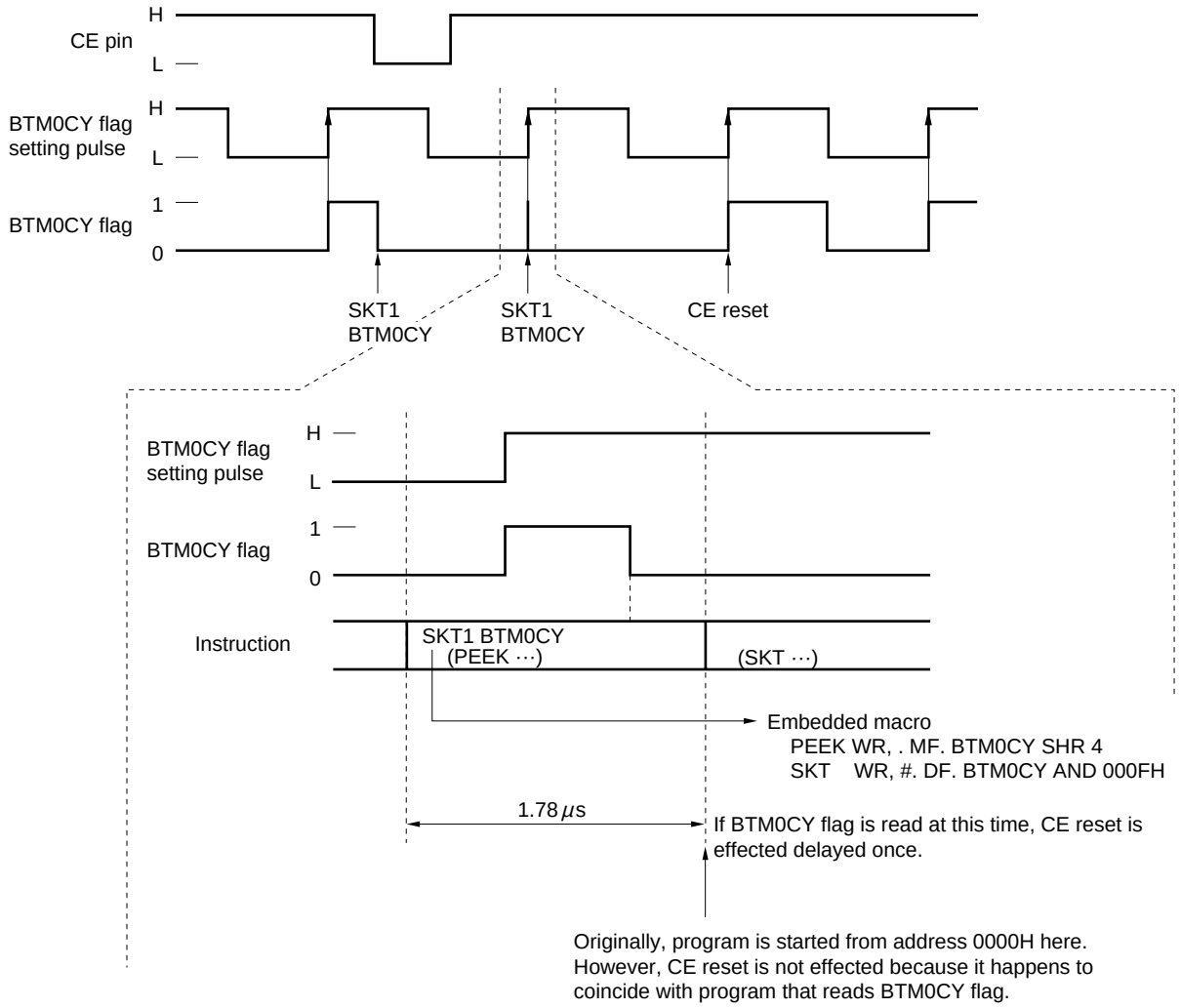
As described in (3) above, CE reset is effected as soon as the BTM0CY flag is set to 1.

If the instruction that reads the BTM0CY flag happens to be executed at the same time as CE reset at this time, the BTM0CY flag reading instruction takes precedence.

Therefore, if the next setting the BTM0CY flag (rising of BTM0CY flag setting pulse) after the CE pin has gone high coincides with execution of the BTM0CY flag reading instruction, CE reset is effected at "the next timing at which the BTM0CY flag is set".

This operation is illustrated in Figure 13-10.

Figure 13-10. Operation When CE Reset Coincides with BTM0CY Flag Reading Instruction



Consequently, if the BTM0CY flag detection time interval coincides with the BTM0CY flag setting time in a program that cyclically detects the BTM0CY flag, CE reset is never effected.

Therefore, the following point must be noted.

Because one instruction cycle is 1.78 μs (1/562.5 kHz), a program that detects the BTM0CY flag once, say, every 1125 instructions, reads the BTM0CY flag every 1.78 μs × 1125 = 2 ms.

Because the timer time setting pulse is 100 ms at this time, if setting and detection of the BTM0CY flag coincide once, CE reset is never effected.

Therefore, do not create a cyclic program that satisfies the following condition.

$$\frac{t_{SET} \times 1125}{X} = n \text{ (n: natural number)}$$

t_{SET} : B TM0CY flag setting time

X : Cycle X step of instruction that reads BTM0CY flag

An example of a program that satisfies the above condition is shown below. Do not create such a program.

Example

```
        Processing A

        CLR2   BTM0CK1, BTM0CK0   ; Embedded macro
                                  ; Sets BTM0CY flag setting pulse to 100 ms
LOOP:
      ; <1>
        SKT1   BTM0CY             ; Embedded macro
        BR     BBB
AAA:

        1121 steps

        BR     LOOP
BBB:

        1121 steps

        BR     LOOP
```

Because the BTM0CY flag reading instruction in <1> is repeatedly executed every 1125 instruction in this example, CE reset is not effected if the BTM0CY flag happens to be set at the timing of instruction in <1>.

13.3 Timer 0

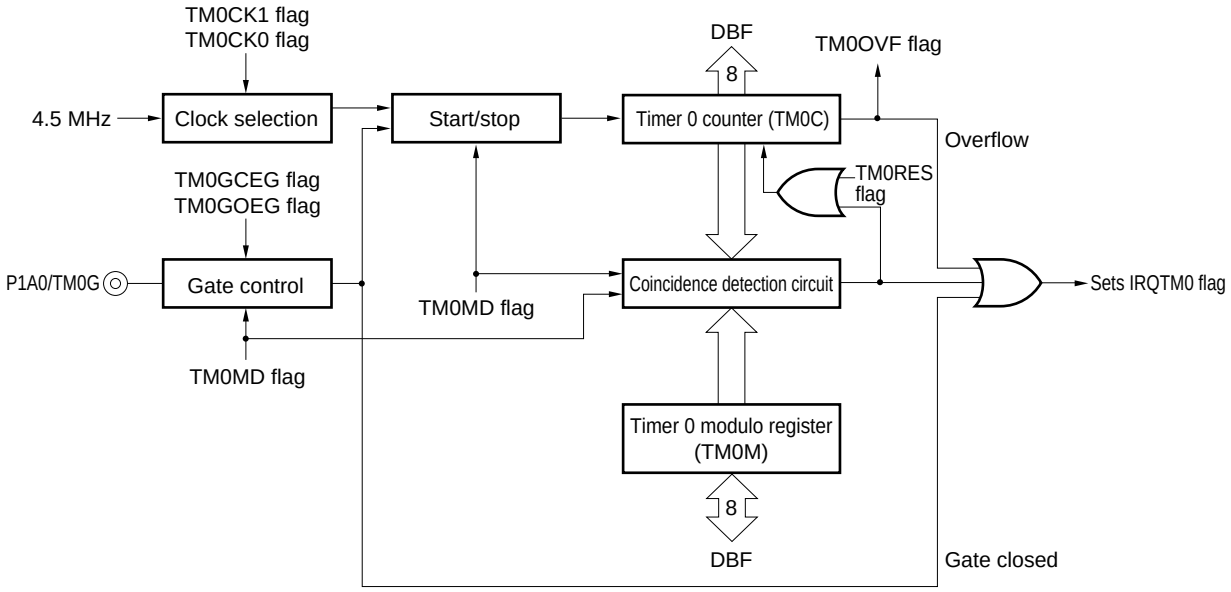
13.3.1 Outline of timer 0

Figure 13-11 shows the outline of timer 0.

The timer 0 is used as timer (modulo mode) by comparing the count value with the previously set value after the basic clock (100 kHz, 10 kHz, 2 kHz, and 1 kHz) has counted by the 8-bit counter.

The pulse width of the signal input from the TM0G pin can be measured (external gate counter).

Figure 13-11. Outlines Timer 0



Remarks 1. TM0CK1 and TM0CK0 (bits 1 and 0 of timer 0 counter clock selection register: refer to **Figure 13-13**) set a basic clock frequency.

2. TM0MD (bit 0 of timer 0 mode selection register: refer to **Figure 13-14**) selects the modulo counter and gate counter.

3. TM0GOEG (bit 1 of timer 0 mode selection register: refer to **Figure 13-14**) sets the open edge of an external gate.

4. TM0GCEG (bit 2 of timer 0 mode selection register: refer to **Figure 13-14**) sets the close edge of an external gate.

5. TM0OVF (bit 3 of timer 0 mode selection register: refer to **Figure 13-14**) detects an overflow of timer 0 counter.

6. TM0RES (bit 2 of timer 0 counter clock selection register: refer to **Figure 13-13**) resets timer 0 counter.

13.3.2 Clock selection, start/stop control, and gate control blocks

Figure 13-12 shows the configuration of these blocks.

The clock selection block selects a basic clock to operate timer 0 counter.

Four types of basic clocks can be selected by using the TM0CK1 and TM0CK0 flags.

Figure 13-13 shows the configuration and function of each flag.

The start/stop block controls the TM0MD flag and open/close signal from the gate control block, and starts or stops the basic clock to be input to timer 0 counter by the TM0EN flag.

The gate control block sets the opening or closing conditions of the gate.

It sets whether the gate is opened or closed by a rising or falling of the input signal, by using the TM0GOEG and TM0GCEG flags. This block also issues an interrupt request when the closing condition of the gate is detected.

Figure 13-14 shows the configuration and function of each flag.

Figure 13-12. Configuration of Clock Selection, Start/Stop Control, and Gate Control Blocks

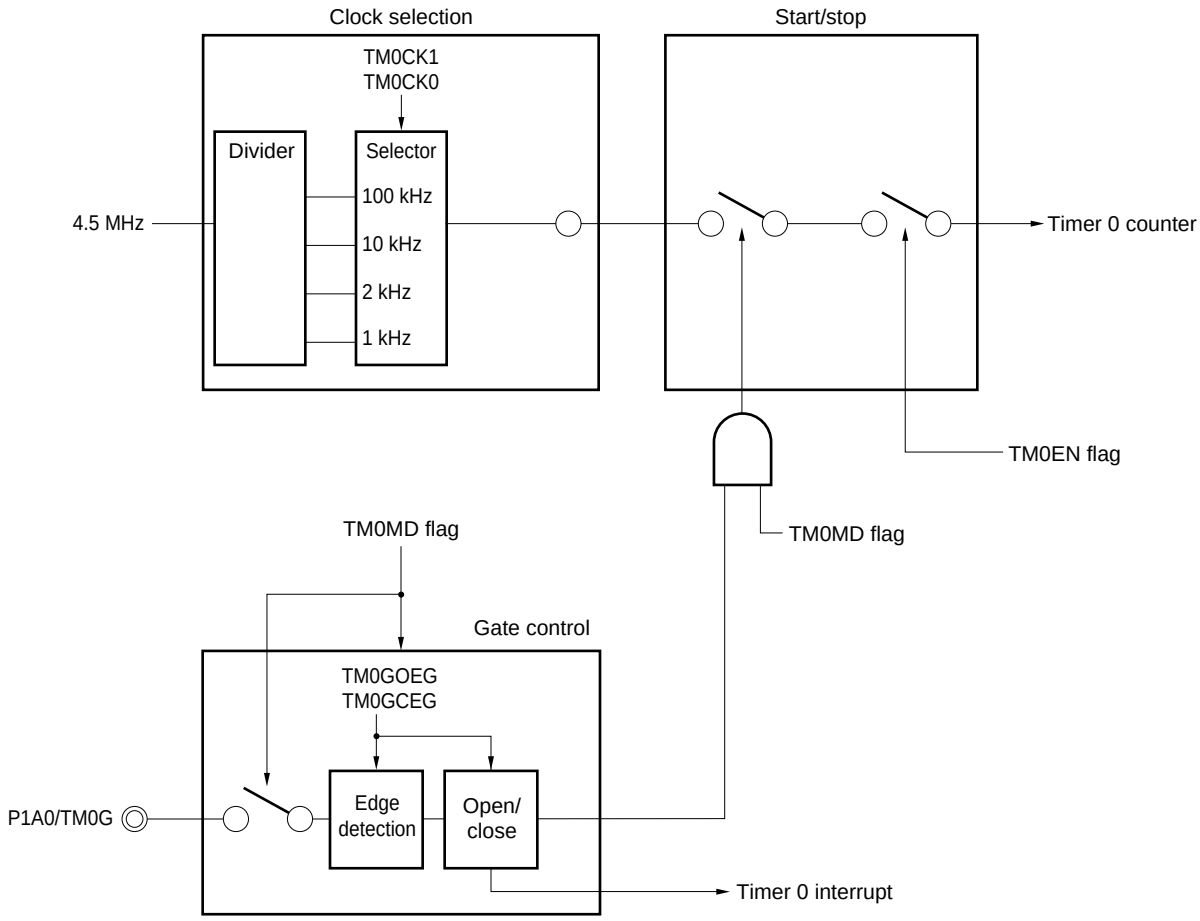


Figure 13-13. Configuration of Timer 0 Counter Clock Selection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Timer 0 counter clock selection	TM0EN	TM0RES	TM0CK1	TM0CK0	2BH	R/W

TM0CK1	TM0CK0	Sets basic clock of timer 0 counter
0	0	100 kHz (10 μs)
0	1	10 kHz (100 μs)
1	0	2 kHz (500 μs)
1	1	1 kHz (1 ms)

TM0RES	Resets timer 0 counter
0	Does not change
1	Resets counter

TM0EN	Starts or stops timer 0
0	Stops
1	Starts

		TM0EN	TM0RES	TM0CK1	TM0CK0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset	0	0	0	0
	CE reset	Retained			
Clock stop		0	0	0	0

Caution When the TM0RES flag is read, 0 is always read.

13.3.3 Count block

The count block counts the basic clock with an 8-bit timer 0 counter, reads the count value, and issues an interrupt request if the value of the timer 0 modulo register coincides with its value.

Timer 0 counter can be reset by the TM0RES flag.

The TM0OVF flag can detect an overflow of the counter. When an overflow occurs, an interrupt request can be issued.

The value of the timer 0 counter can be read via data buffer.

The value of the timer 0 modulo register can be written or read via data buffer.

Figure 13-14 shows the configuration of the timer 0 mode selection register.

Figure 13-15 shows the configuration of the timer 0 counter.

Figure 13-16 shows the configuration of the timer 0 modulo register.

Figure 13-14. Configuration of Timer 0 Mode Selection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Timer 0 mode selection	T M 0 O V F	T M 0 G C E G	T M 0 G O E G	T M 0 M D	2CH	R/W

Selects modulo counter or gate counter of timer 0	
0	Modulo counter
1	Gate counter

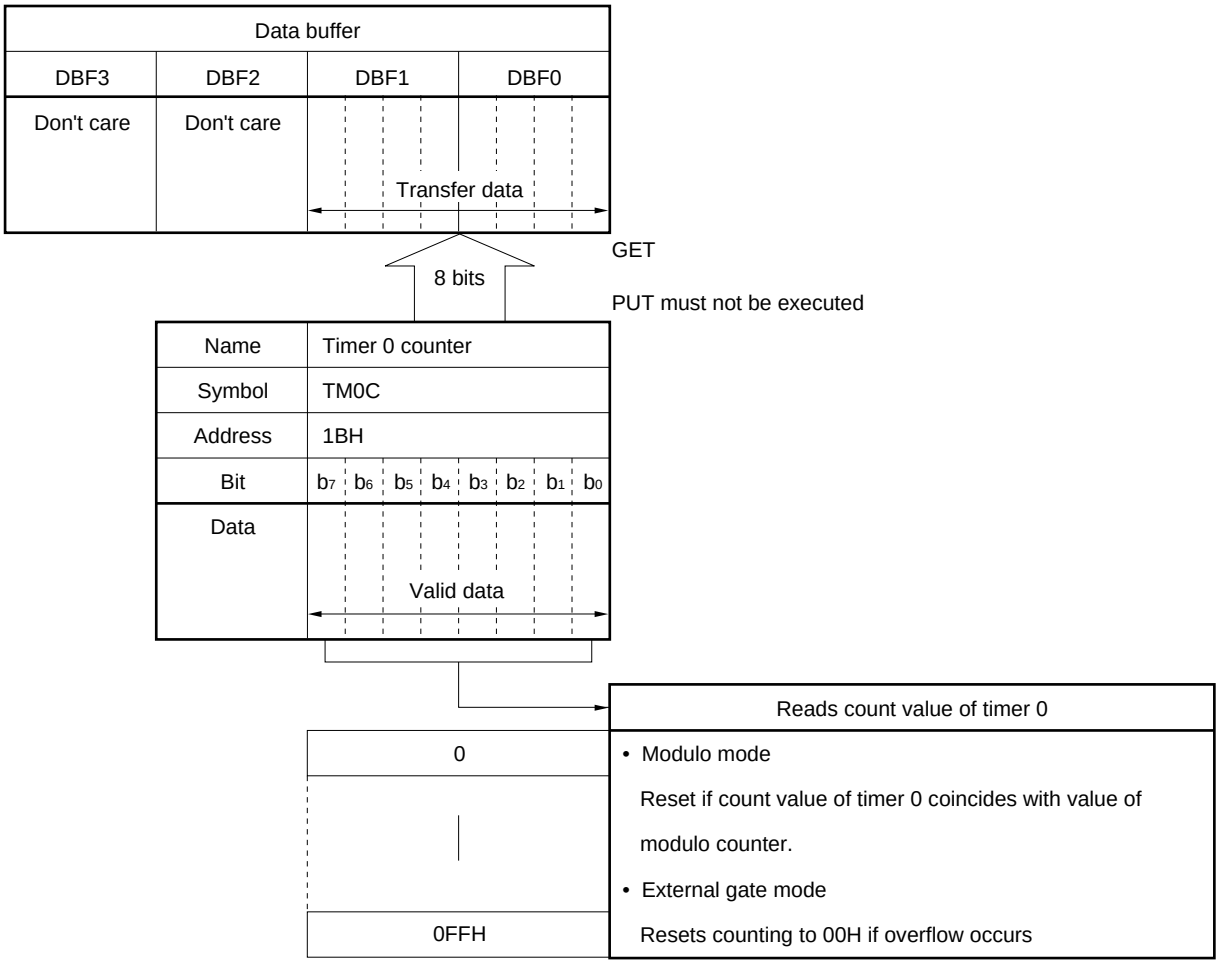
Specifies edge of gate open input signal	
0	Rising edge
1	Falling edge

Specifies edge of gate close input signal	
0	Rising edge
1	Falling edge

Detects timer 0 overflow	
0	No overflow
1	Overflow

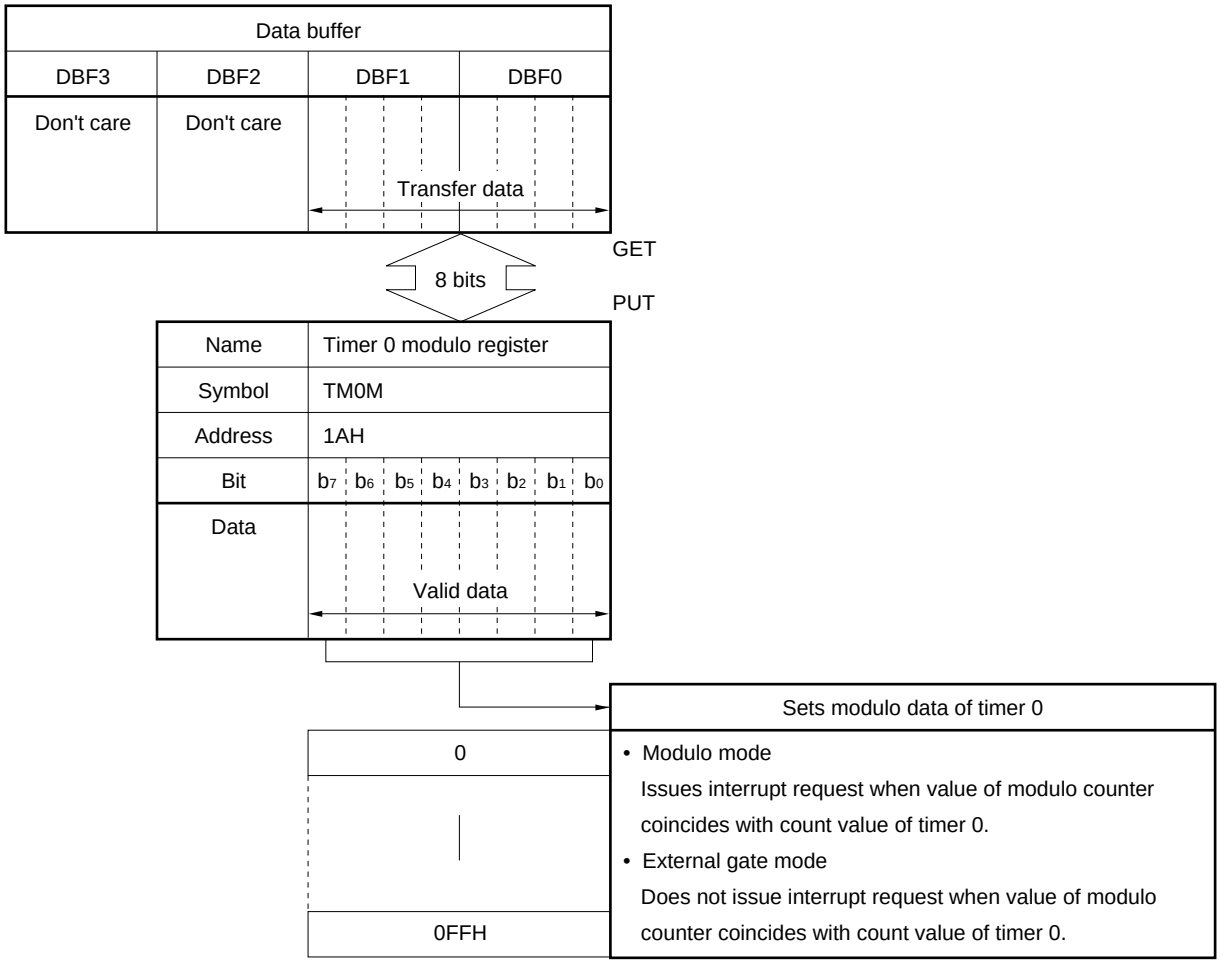
At reset	Power-ON reset	0	0	0	0	
	WDT&SP reset	0	0	0	0	
	CE reset	Retained				
Clock stop		0	0	0	0	

Figure 13-15. Configuration of Timer 0 Counter



At reset	Power-ON reset	0	0	0	0	0	0	0	0
	WDT&SP reset	0	0	0	0	0	0	0	0
	CE reset	Retained							
Clock stop		0	0	0	0	0	0	0	0

Figure 13-16. Configuration of Timer 0 Modulo Register



At reset	Power-ON reset	1	1	1	1	1	1	1	1
	WDT&SP reset	1	1	1	1	1	1	1	1
	CE reset	Retained							
Clock stop		1	1	1	1	1	1	1	1

13.3.4 Example of using timer 0

(1) Modulo counter mode

The modulo counter mode is used for time management by generating timer 0 interrupt at fixed intervals.

An example of a program is shown below.

This program executes processing B every 500 μs.

```
TM0DATA   DAT     0032H           ; MODULO DATA = 50

START:
          BR      INITIAL         ; Reset address
          ; Interrupt vector address
          NOP                     ; SIO3
          NOP                     ; SIO2
          NOP                     ; TM3
          NOP                     ; TM2
          NOP                     ; TM1
          BR      INT_TM0         ; TM0
          NOP                     ; INT4
          NOP                     ; INT3
          NOP                     ; INT2
          NOP                     ; INT1
          NOP                     ; INT0
          NOP                     ; Down edge of CE

INITIAL:
          INITFLG NOT TM0EN, TM0RES, NOT TM0CK1, NOT TM0CK0
          ;          (Stop)   , (Reset) ,     (Basic clock = 10 μs)
          CLR1    TM0MD           ; Modulo mode
          MOV     DBF0, #(TM0MDATA SHR 0) AND 0FH
          MOV     DBF1, #(TM0MDATA SHR 4) AND 0FH
          PUT     TM0M, DBF       ; Sets count data
          SET1    IPTM0           ; Enables timer 0 interrupt
          EI
          SET1    TM0EN           ; Starts timer 0

LOOP:
          Processing A

          BR      LOOP

INT_TM0:
          Processing B            ; Timer 0 interrupt service

          EI
          RETI                    ; Return
```

(2) Gate counter mode

The gate counter mode is used to count the width of a pulse input to the TM0G pin.

An example of a program is shown below.

In this program example, the width of the pulse input to the TM0G pin is counted from the falling edge to the falling edge.

If the pulse width is 800 to 1200 μs, processing C is executed; otherwise, processing B is executed.

If the pulse width is 2560 μs or more, processing D is executed.

```
TM0800      DAT     0050H           ; Count data = 80
TM01200     DAT     0078H           ; Count data = 120

START:
            BR      INITIAL         ; Reset address
            ; Interrupt vector address
            NOP                     ; SIO3
            NOP                     ; SIO2
            NOP                     ; TM3
            NOP                     ; TM2
            NOP                     ; TM1
            BR      INT_TM0         ; TM0
            NOP                     ; INT4
            NOP                     ; INT3
            NOP                     ; INT2
            NOP                     ; INT1
            NOP                     ; INT0
            NOP                     ; Down edge of CE

INITIAL:
            INITFLG NOT TM0EN, TM0RES, NOT TM0CK1, NOT TM0CK0
            ;           (Stop)   , (Reset) ,     (Basic clock = 10 μs)
            INITFLG   TM0GCEG  ,  TM0GOEG  ,    TM0MD
            ;         (Falling close), (Falling open), (Gate counter)
            SET1    TM0EN           ; START
            SET1    IPTM0           ; Enables timer 0 interrupt
            EI

LOOP:
            Processing A

            BR      LOOP

INT_TM0:
            PUT     DBFSTK, DBF     ; Saves data buffer
            GET     DBF, TM0C
            INITFLG TM0EN, TM0RES
            SKT1    TM0OVF          ; Detects overflow status (2560 μs or more?)
            BR      AAA

            Processing D

            BR      EI_RETI

AAA:
            SUB     DBF0, #TM0800 AND 0FH
            SUBC    DBF1, #TM0800 SHR4 AND 0FH
            SKF1    CY              ; 800 μs or more?
            BR      BBB
            SUB     DBF0, #TM01200 AND 0FH
```

```
                SUBC    DBF1, #TM01200 SHR4 AND 0FH
                SKT1    CY                  ; 1200 μs or more?
                BR      BBB

                Processing C

                BR      EI_RETI

BBB:

                Processing B

EI_RETI:
                GET     DBF, DBFSTK         ; Restores data buffer
                EI
                RETI                        ; Return

END
```

13.3.5 Error of timer 0

Timer 0 has an error of up to 1 basic clock in the following cases.

(1) On starting/stopping counter

The counter is started or stopped by ANDing the open/close condition of the gate and TM0EN flag setting condition.

Therefore, an error of 0 to +1 clocks occurs when the gate is opened or the TM0EN flag is set, and an error of –1 to 0 clocks occurs when the gate is closed or the flag is reset.

In all, an error of ±1 count occurs.

(2) On resetting counter operation

An error of 0 to +1 clocks occurs when the counter is reset.

(3) On selecting basic clock during counter operation

An error of 0 to +1 clocks of the newly selected clock occurs.

13.3.6 Cautions on using timer 0

Timer 0 interrupt may occur simultaneously with the other timer interrupts and CE reset. If it is necessary to update the timer at CE reset, do not use timer 0, use basic timer 0 instead.

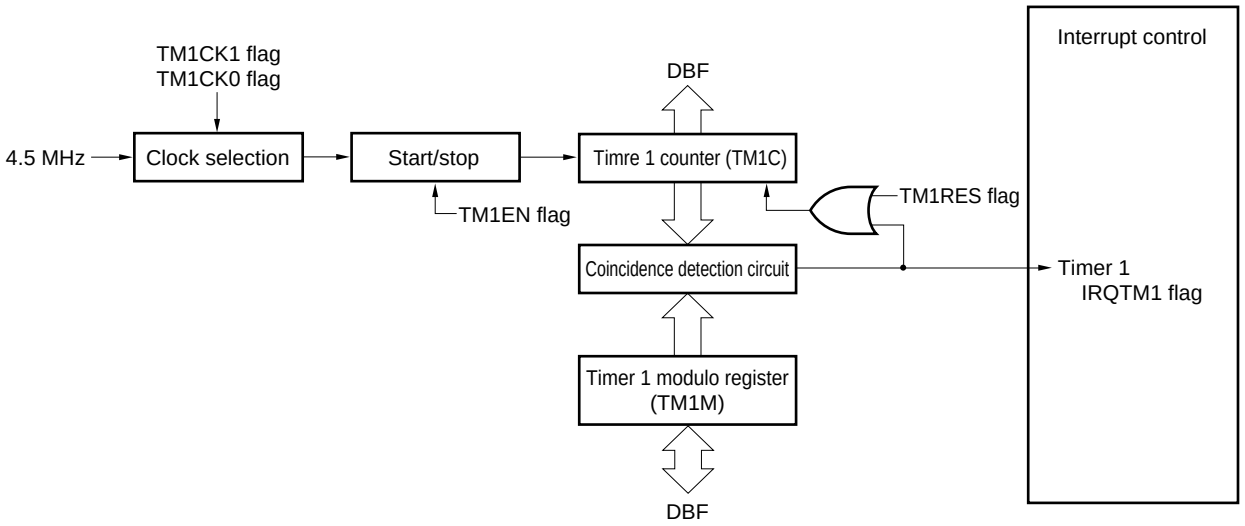
13.4 Timer 1

13.4.1 Outline of timer 1

Figure 13-17 outlines timer 1.

Timer 1 counts the basic clock (100, 10, 2, or 1 kHz) with an 8-bit counter, and compares the count value with a value set in advance.

Figure 13-17. Outline of Timer 1



Remarks 1. TM1CK1 and TM1CK0 (bits 1 and 0 of timer 1 counter clock selection register: refer to **Figure 13-18**) set the basic clock frequency.

2. TM1EN (bit 3 of timer 1 counter clock selection register: refer to **Figure 13-18**) starts or stops timer 1.

3. TM1RES (bit 2 of timer 1 counter clock selection register: refer to **Figure 13-18**) resets timer 1 counter.

13.4.2 Clock selection and start/stop control blocks

The clock selection block selects a basic clock to operate timer 1 counter.

Four types of basic clocks can be selected by using the TM1CK1 and TM1CK0 flags.

The start/stop block starts or stops the basic clock input to timer 1 by using the TM1EN flag.

Figure 13-18 shows the configuration and function of each flag.

13.4.3 Count block

The count block counts the basic clock with timer 1 counter, reads the count value, and issues an interrupt request when its count value coincides with the value of the timer 1 modulo register.

The timer 1 counter can be reset by the TM1RES flag.

The timer 1 counter is automatically reset when its value coincides with the value of the timer 1 modulo register.

The value of the timer 1 counter can be read via data buffer.

Data can be written to the value of the timer 1 modulo register via data buffer.

Figure 13-18 shows the configuration of timer 1 counter clock selection register.

Figure 13-19 shows the configuration of the timer 1 counter.

Figure 13-20 shows the configuration of the timer 1 modulo register.

Figure 13-18. Configuration of Timer 1 Counter Clock Selection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Timer 1 counter clock selection	TM1EN	TM1RES	TM1CK1	TM1CK0	2AH	R/W

Sets basic clock of timer 1 counter	
0	0	100 kHz (10 μs)
0	1	10 kHz (100 μs)
1	0	2 kHz (500 μs)
1	1	1 kHz (1 ms)

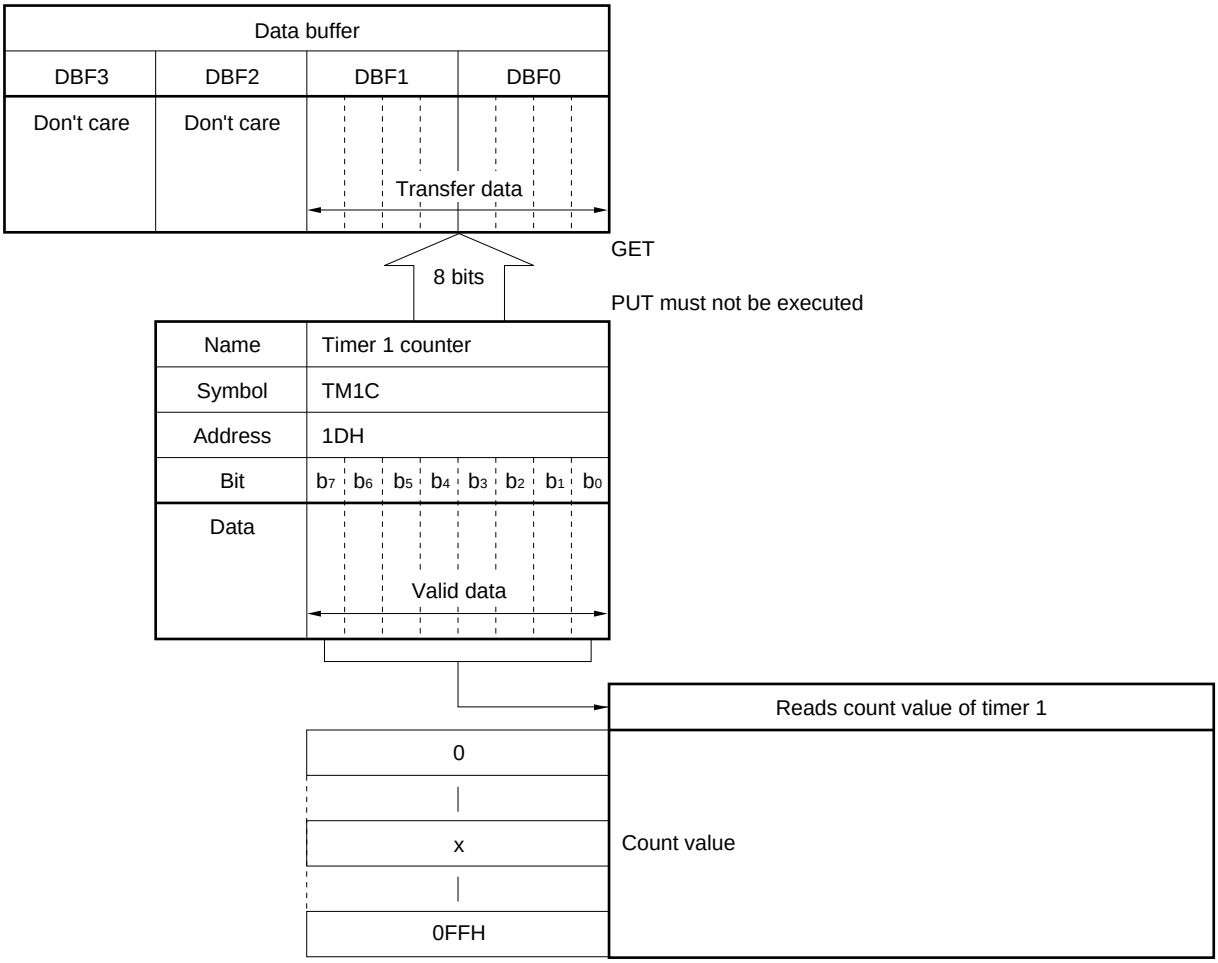
Resets timer 1 counter (valid on writing)	
0	Does not change
1	Resets counter

Starts or stops timer 1	
0	Stops
1	Starts

		TM1EN	TM1RES	TM1CK1	TM1CK0
At reset	Power-On reset	0	0	0	0
	WDT&SP reset	0	0	0	0
	CE reset	Retained			
Clock stop		0	0	0	0

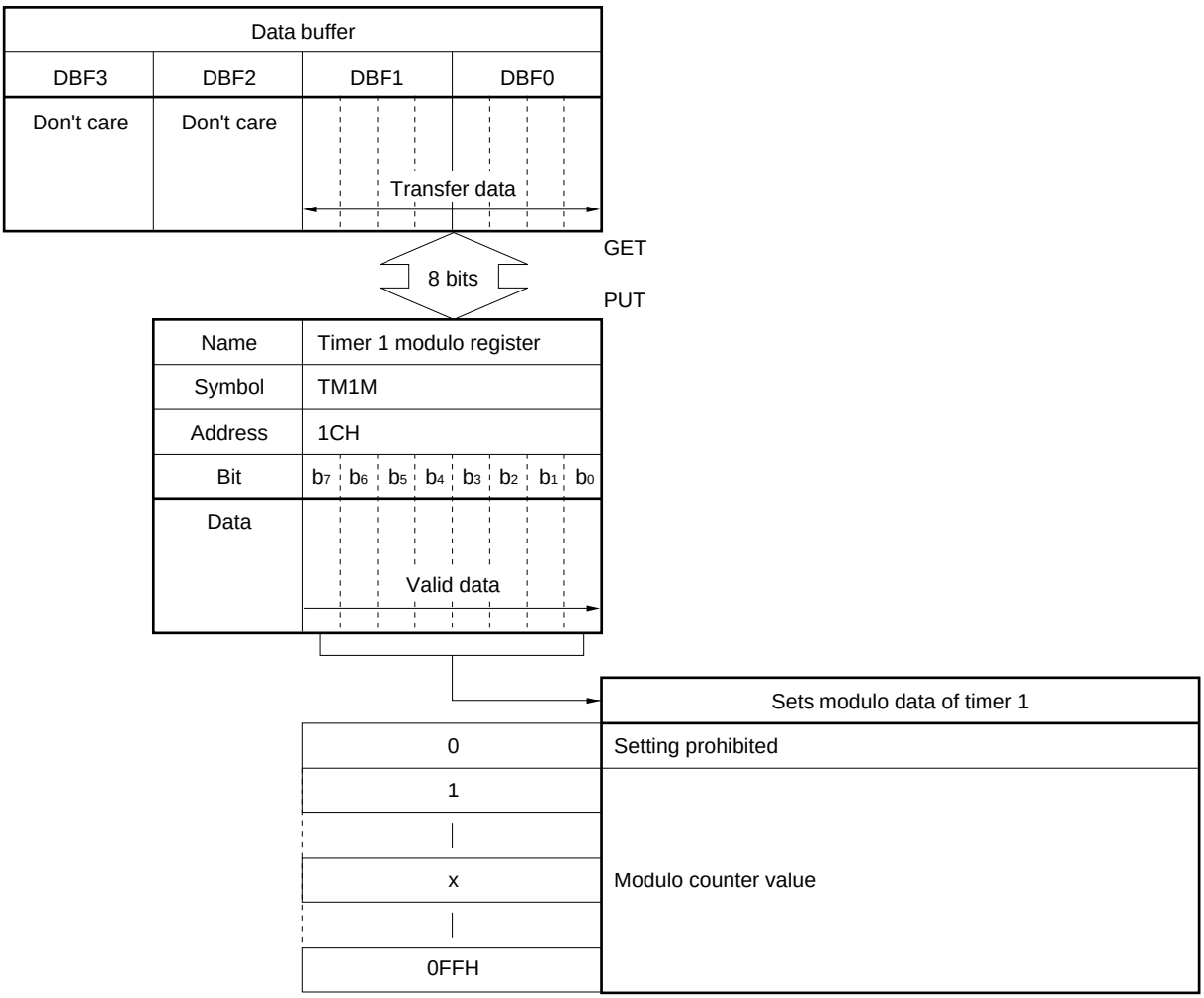
Caution When the TM1RES flag is read, 0 is always read.

Figure 13-19. Configuration of Timer 1 Counter



At reset	Power-ON reset	0	0	0	0	0	0	0	0
	WDT&SP reset	0	0	0	0	0	0	0	0
	CE reset	Retained							
Clock stop		0	0	0	0	0	0	0	0

Figure 13-20. Configuration of Timer 1 Modulo Register



At reset	Power-ON reset	1	1	1	1	1	1	1	1
	WDT&SP reset	1	1	1	1	1	1	1	1
	CE reset	Retained							
Clock stop		1	1	1	1	1	1	1	1

13.4.4 Example of using timer 1

(1) Modulo timer

The modulo timer is used for time management by generating timer 1 interrupt at fixed intervals.
An example of a program is shown below.
This program executes processing B every 500 μs.

```
TM1DATA   DAT      0032H              ; Count data = 50

START:
          BR       INITIAL            ; Reset address
          ; Interrupt vector address
          NOP                         ; SIO3
          NOP                         ; SIO2
          NOP                         ; TM3
          NOP                         ; TM2
          BR       INT_TM1            ; TM1
          NOP                         ; TM0
          NOP                         ; INT4
          NOP                         ; INT3
          NOP                         ; INT2
          NOP                         ; INT1
          NOP                         ; INT0
          NOP                         ; Down edge of CE

INITIAL:
          INITFLG  NOT TM1EN, TM1RES, NOT TM1CK1, NOT TM1CK0
          ;            (Stop)    , (Reset) ,      (Basic clock = 10 μs)
          MOV      DBF0, #TM1DATA
          MOV      DBF1, #TM1DATA SHR4 AND 0FH
          PUT      TM1, DBF
          SET1     TM1EN              ; START
          SET1     IPTM1              ; Enables timer 1 interrupt
          EI

LOOP:
          Processing A

          BR       LOOP

INT_TM1:
          PUT      DBFSTK, DBF        ; Saves data buffer

          Processing B

          GET      DBF, DBFSTK
          EI
          RETI                        ; Return

END
```

13.4.5 Error of timer 1

Timer 1 has an error of up to 1 basic clock in the following cases.

(1) On starting/stopping counter

The counter is started or stopped by setting the TM1EN flag.

Therefore, an error of 0 to +1 clocks occurs when the TM1EN flag is set, and an error of –1 to 0 clocks occurs when the flag is reset.

In all, an error of ±1 count occurs.

(2) On resetting counter operation

An error of 0 to +1 clocks occurs when the counter is reset.

(3) On selecting basic clock during counter operation

An error of 0 to +1 clocks of the newly selected clock occurs.

13.4.6 Cautions on using timer 1

Timer 1 interrupt may occur simultaneously with the other timer interrupts and CE reset. If it is necessary to update the timer at CE reset, do not use timer 1, use basic timer 0 instead.

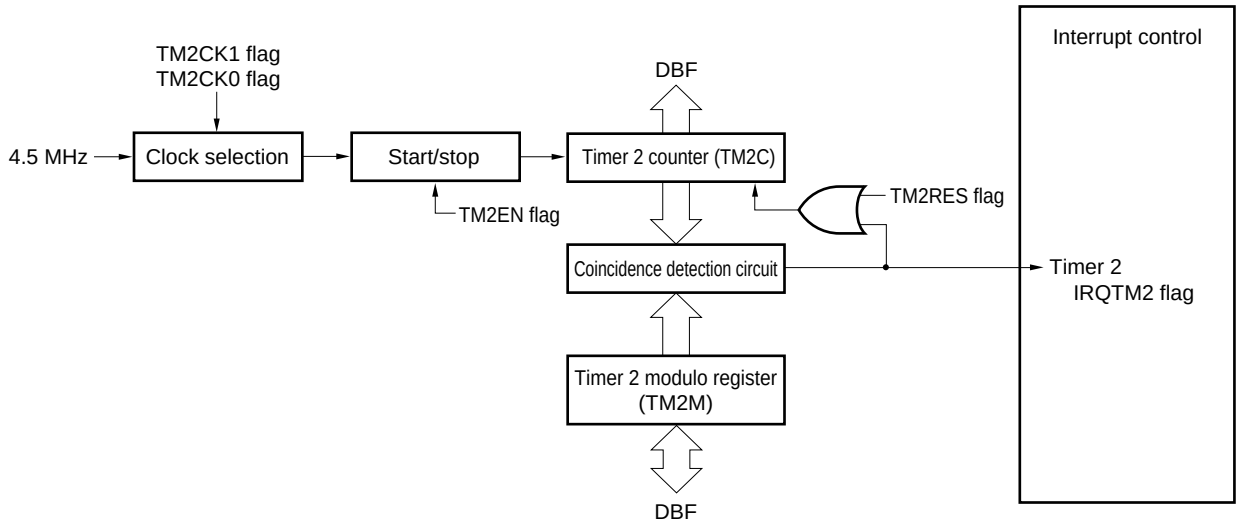
13.5 Timer 2

13.5.1 Outline of timer 2

Figure 13-21 outlines timer 2.

Timer 2 counts the basic clock (100, 10, 2, or 1 kHz) with an 8-bit counter, and compares the count value with a value set in advance.

Figure 13-21. Outline of Timer 2



Remarks 1. TM2CK1 and TM2CK0 (bits 1 and 0 of timer 2 counter clock selection register: refer to **Figure 13-22**) set the basic clock frequency.

2. TM2EN (bit 3 of timer 2 counter clock selection register: refer to **Figure 13-22**) starts or stops timer 2.

3. TM2RES (bit 2 of timer 2 counter clock selection register: refer to **Figure 13-22**) resets timer 2 counter.

13.5.2 Clock selection and start/stop control blocks

The clock selection block selects a basic clock to operate timer 2 counter.

Four types of basic clocks can be selected by using the TM2CK1 and TM2CK0 flags.

The start/stop block starts or stops the basic clock input to timer 2 by using the TM2EN flag.

Figure 13-22 shows the configuration and function of each flag.

13.5.3 Count block

The count block counts the basic clock with timer 2 counter, reads the count value, and issues an interrupt request when its count value coincides with the value of the timer 2 modulo register.

The timer 2 counter can be reset by the TM2RES flag.

The timer 2 counter is automatically reset when its value coincides with the value of the timer 2 modulo register.

The value of the timer 2 counter can be read via data buffer.

Data can be written to the value of the timer 2 modulo register via data buffer.

Figure 13-22 shows the configuration of timer 2 counter clock selection register.

Figure 13-23 shows the configuration of the timer 2 counter.

Figure 13-24 shows the configuration of the timer 2 modulo register.

Figure 13-22. Configuration of Timer 2 Counter Clock Selection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Timer 2 counter clock selection	TM2EN	TM2RES	TM2CK1	TM2CK0	29H	R/W

TM2CK1, TM2CK0 — Sets basic clock of timer 2 counter

TM2CK1	TM2CK0	
0	0	100 kHz (10 μs)
0	1	10 kHz (100 μs)
1	0	2 kHz (500 μs)
1	1	1 kHz (1 ms)

TM2RES — Resets timer 2 counter (valid on writing)

TM2RES	
0	Does not change
1	Resets counter

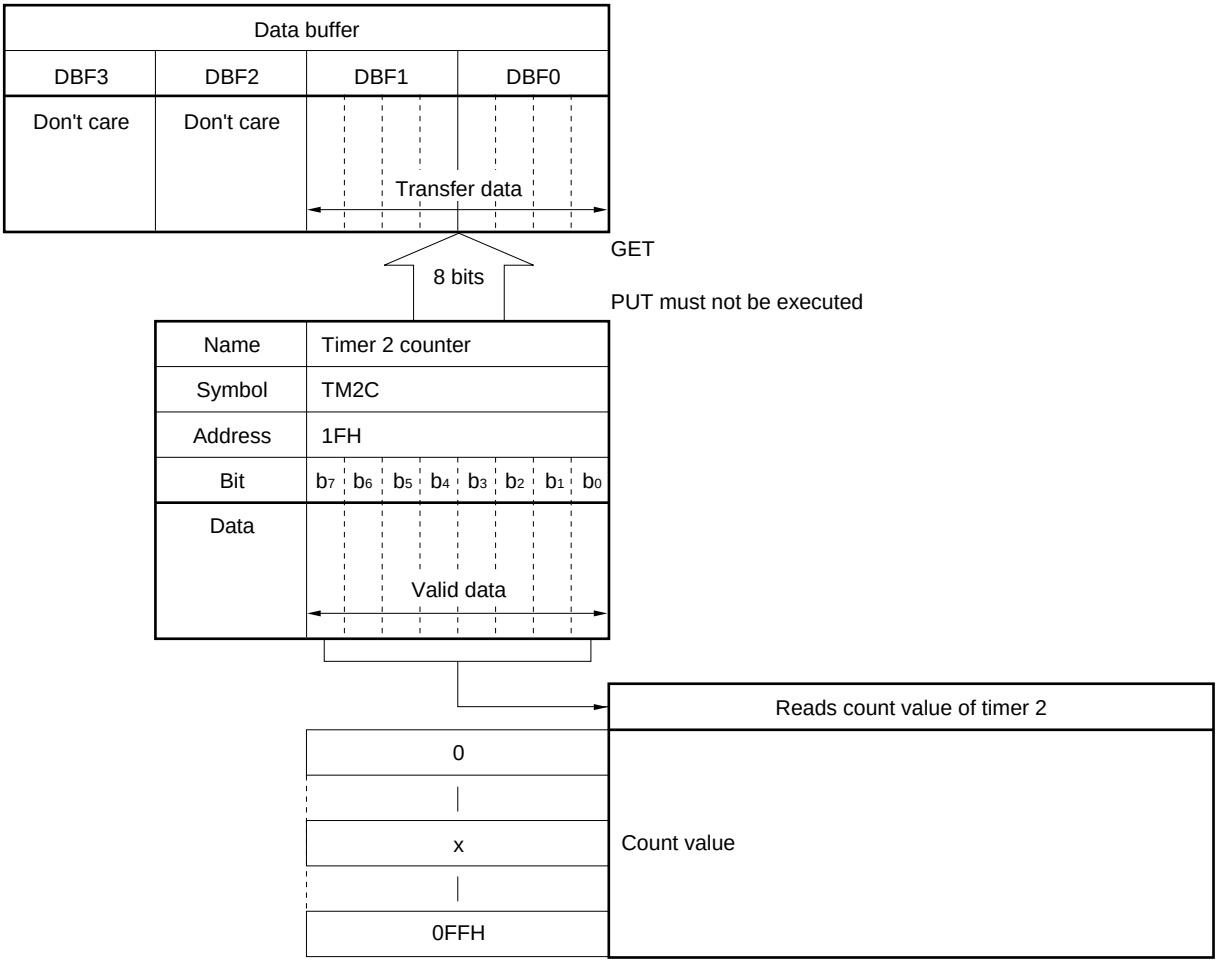
TM2EN — Starts or stops timer 2

TM2EN	
0	Stops
1	Starts

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset	0	0	0	0
	CE reset	Retained			
Clock stop		0	0	0	0

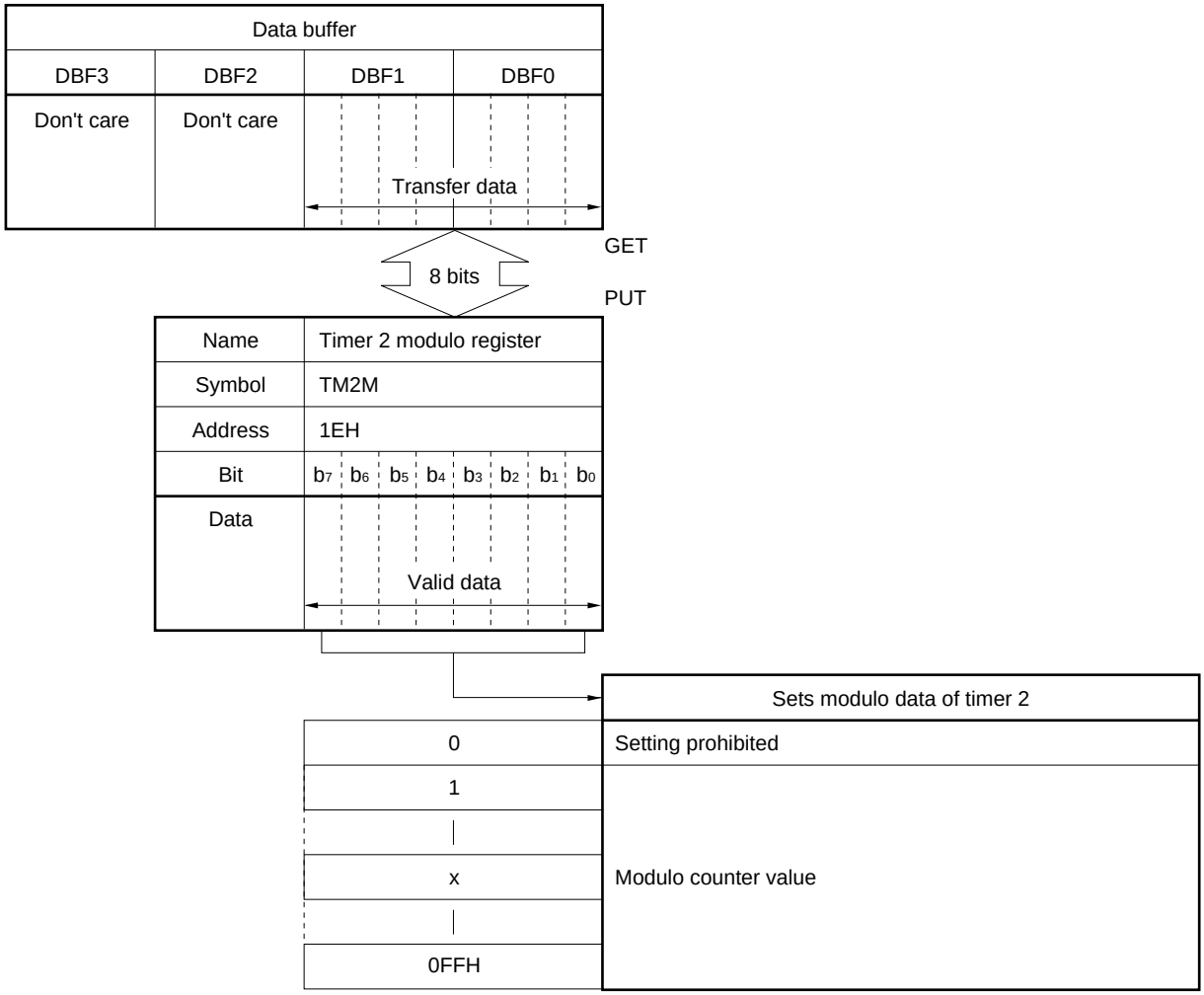
Caution When the TM2RES flag is read, 0 is always read.

Figure 13-23. Configuration of Timer 2 Counter



		b7	b6	b5	b4	b3	b2	b1	b0
At reset	Power-ON reset	0	0	0	0	0	0	0	0
	WDT&SP reset	0	0	0	0	0	0	0	0
	CE reset	Retained							
Clock stop		0	0	0	0	0	0	0	0

Figure 13-24. Configuration of Timer 2 Modulo Register



At reset	Power-On reset	1	1	1	1	1	1	1	1
	WDT&SP reset	1	1	1	1	1	1	1	1
	CE reset	Retained							
Clock stop		1	1	1	1	1	1	1	1

13.5.4 Example of using timer 2

(1) Modulo timer

The modulo timer is used for time management by generating a timer 2 interrupt at fixed intervals.
An example of a program is shown below.
This program executes processing B every 500 μs.

```
TM2DATA   DAT      0032H              ; Count data = 50

START:
          BR       INITIAL            ; Reset address
          ; Interrupt vector address
          NOP                         ; SIO3
          NOP                         ; SIO2
          NOP                         ; TM3
          BR       INT_TM2            ; TM2
          NOP                         ; TM1
          NOP                         ; TM0
          NOP                         ; INT4
          NOP                         ; INT3
          NOP                         ; INT2
          NOP                         ; INT1
          NOP                         ; INT0
          NOP                         ; Down edge of CE

INITIAL:
          INITFLG  NOT TM2EN, TM2RES, NOT TM2CK1, NOT TM2CK0
          ;             (Stop)    , (Reset) ,      (Basic clock = 10 μs)
          MOV      DBF0, #TM2DATA
          MOV      DBF1, #TM2DATA SHR4 AND 0FH
          PUT      TM2, DBF
          SET1     TM2EN              ; START
          SET1     IPTM2              ; Enables timer 2 interrupt
          EI

LOOP:
          Processing A

          BR       LOOP

INT_TM2:
          PUT      DBFSTK, DBF        ; Saves data buffer
          INITFLG  TM2EN, TM2RES      ; Resets and starts

          Processing B

          GET      DBF, DBFSTK
          EI
          RETI                        ; Return

END
```

13.5.5 Error of timer 2

Timer 2 has an error of up to 1 basic clock in the following cases.

(1) On starting/stopping counter

The counter is started or stopped by setting the TM2EN flag.

Therefore, an error of 0 to +1 clocks occurs when the TM2EN flag is set, and an error of –1 to 0 clocks occurs when the flag is reset.

In all, an error of ±1 count occurs.

(2) On resetting counter operation

An error of 0 to +1 clocks occurs when the counter is reset.

(3) On selecting basic clock during counter operation

An error of 0 to +1 clocks of the newly selected clock occurs.

13.5.6 Cautions on using timer 2

Timer 2 interrupt may occur simultaneously with the other timer interrupts and CE reset. If it is necessary to update the timer at CE reset, do not use timer 2, use basic timer 0 instead.

13.6 Timer 3

13.6.1 Outline of timer 3

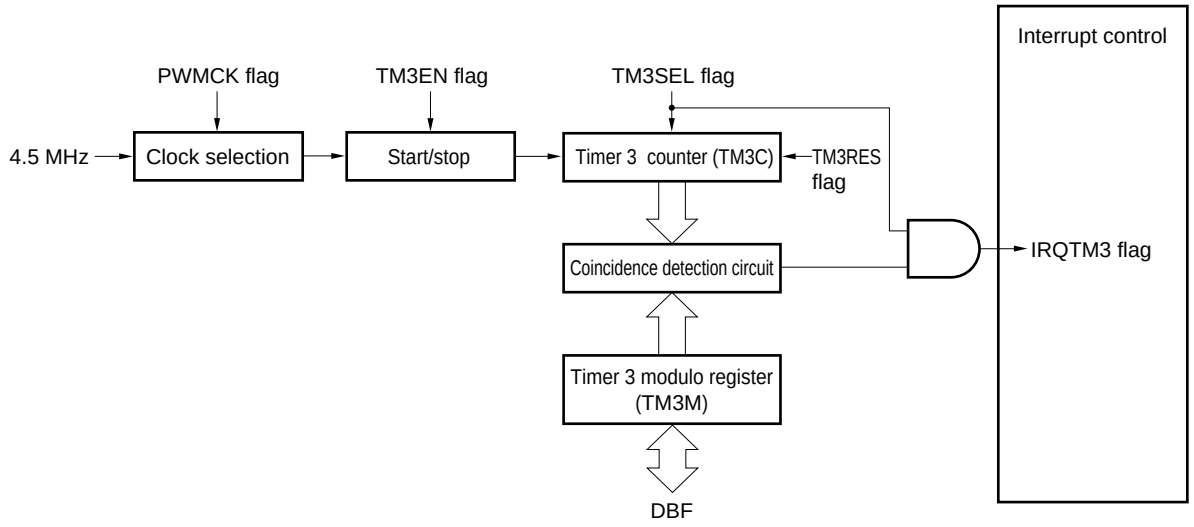
Figure 13-25 outlines timer 3.

Timer 3 counts the basic clock (1.125 MHz or 112.5 kHz selectable) with an 8-bit counter[Note], and compares the count value with a value set in advance.

Because timer 3 is multiplexed with a D/A converter, all the three D/A converter pins are automatically set in the general-purpose port mode when timer 3 is used.

Note A 9-bit or 8-bit counter can be selected for the D/A converter, but the 8-bit counter is automatically selected when the timer function is selected.

Figure 13-25. Outline of Timer 3



Remarks 1. PWMCK (bit 0 of PWM clock selection register: refer to **Figure 13-26**) selects the output frequency of timer 3.

2. TM3SEL (bit 3 of timer 3 control register: refer to **Figure 13-27**) selects timer 3 or D/A converter.

3. TM3EN (bit 1 of timer 3 control register: refer to **Figure 13-27**) starts or stops counting by timer 3.

4. TM3RES (bit 0 of timer 3 control register: refer to **Figure 13-27**) controls resetting of timer 3 counter.

13.6.2 Clock selection block

The clock of timer 3 is selected by the PWMCK flag of the PWM clock selection register.

Figure 13-26 shows the configuration of the flag.

Figure 13-26. Configuration of PWM Clock Selection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
PWM clock selection	0	PWMBIT	0	PWMCK	26H	R/W

PWMCK	Selects output frequency of timer 3
0	4.4 kHz (8 bits)/2.2 kHz (9 bits)
1	440 Hz (8 bits)/220 Hz (9 bits)

Fixed to “0”

PWMBIT	Selects number of bits of PWM counter
0	8 bits
1	9 bits

Fixed to “0”

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset		0		0
	CE reset		R		R
Clock stop			0		0

R:Retained

13.6.3 Start/stop control block

The start/stop block starts or stops the basic clock to be input to timer 3 counter by using the TM3EN flag.

To control timer 3, timer 3 must be selected by the TM3SEL flag.

Figure 13-27 shows the configuration of each flag.

Figure 13-27. Configuration of Timer 3 Control Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Timer 3 control	TM3SEL	0	TM3EN	TM3RES	28H	R/W

Resets counter

0	Dose not change
1	Resets

Starts or stops counter

0	Stop
1	Starts

Fixed to “0”

Selects timer 3 or D/A converter

0	D/A converter (PWM output)
1	Timer 3

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset	0		0	0
	CE reset	R		Retained	
Clock stop		0		0	0

R:Retained

13.6.4 Count block

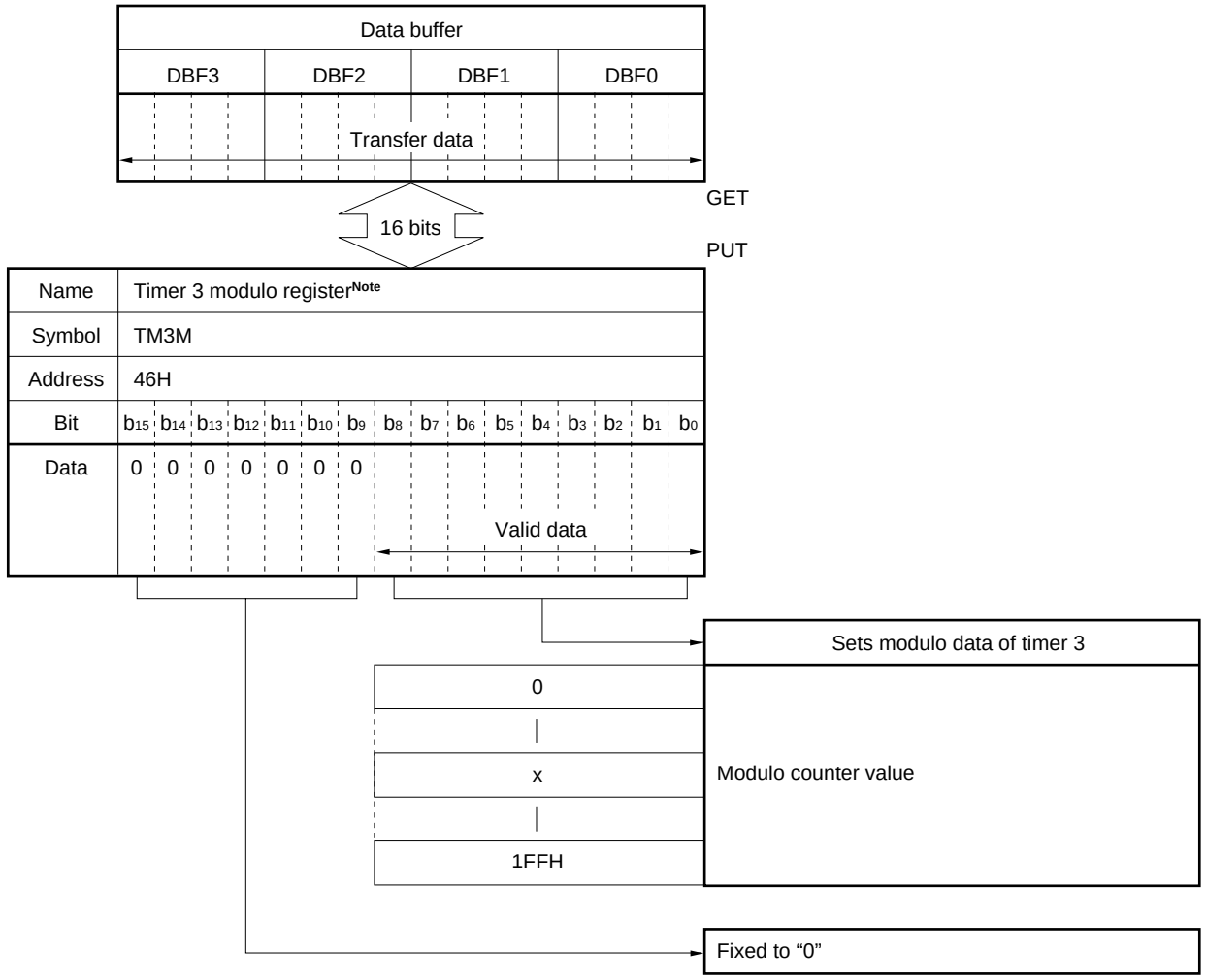
The count block counts the basic clock with timer 3 and issues an interrupt request when the count value of timer 3 coincides with the value of the timer 3 modulo register.

Timer 3 counter can be reset by the TM3RES flag.

Because the PWM data register 2 (PWMR2) and timer 3 modulo register (TM3M) are multiplexed, these registers cannot be used at the same time.

When timer 3 is used, the PWM data register 1 (PWMR1) and PWM data register 0 (PWMR0) can be used as 9-bit data latches (refer to **15. D/A CONVERTER (PWM mode)**).

Figure 13-28. Configuration of Timer 3 Modulo Register



		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
At reset	Power-ON reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	WDT&SP reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	CE reset	Retained															
Clock stop		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note This register is multiplexed with the PWM data register 2.

13.6.5 Example of using timer 3

An example of a program using timer 3 (multiplexed with PWM) is given below.

This program executes processing B every 888 μs.

```
TM3DATA   DAT     0064H           ; Count data = 100

START:
          BR      INITIAL         ; Reset address
          ; Interrupt vector address
          NOP                     ; SIO3
          NOP                     ; SIO2
          BR      INT_TM3         ; TM3
          NOP                     ; TM2
          NOP                     ; TM1
          NOP                     ; TM0
          NOP                     ; INT4
          NOP                     ; INT3
          NOP                     ; INT2
          NOP                     ; INT1
          NOP                     ; INT0
          NOP                     ; Down edge of CE

INITIAL:
          INITFLG   NOT PWMSEL2 , NOT PWMSEL1 , NOT PWMSEL0
          ;         (General-purpose port), (General-purpose port), (General-purpose port)
          INITFLG   NOT PWMBIT, PWMCK
          ;         (   8BIT   ), (440 Hz)
          INITFLG     TM3SEL  , NOT TM3EN, TM3RES
          ;         (Timer 3 mode),   (Stop)  ,  (Reset)

          MOV       DBF0, #TM3DATA
          MOV       DBF1, #TM3DATA SHR4 AND 0FH
          PUT       TM3M, DBF
          SET1      TM3EN           ; START
          SET1      IPTM3           ; Enables timer 3 interrupt
          EI

LOOP:
          Processing A

          BR        LOOP

INT_TM3:
          PUT       DBFSTK, DBF     ; Saves data buffer

          Processing B

          GET       DBF, DBFSTK
          EI
          RETI                      ; Return

END
```

13.6.6 Error of timer 3

Timer 3 has an error of up to 1 basic clock in the following cases.

(1) On starting/stopping counter

The counter is started or stopped by setting the TM3EN flag.

Therefore, an error of 0 to +1 clocks occurs when the TM3EN flag is set, and an error of –1 to 0 clocks occurs when the flag is reset.

In all, an error of ±1 count occurs.

(2) On resetting counter operation

An error of 0 to +1 clocks occurs when the counter is reset.

(3) On selecting basic clock during counter operation

An error of 0 to +1 clocks of the newly selected clock occurs.

13.6.7 Cautions on using timer 3

Timer 3 interrupt may occur simultaneously with the other timer interrupts and CE reset. If it is necessary to update the timer at CE reset, do not use timer 3, use basic timer 0 instead.

When timer 3 is used, the three output port pins multiplexed with the D/A converter pins, P1B2/PWM2 through P1B0/PWM0, are automatically set in the general-purpose output port mode.

13.6.8 Status at reset

(1) At power-ON reset

The P1B2/PWM2 through P1B0/PWM0 pins are set in the general-purpose output port mode.
The output value is “low level”.
The value of each PWM data register (including the timer 3 modulo register) is “1FFH”.

(2) At WDT&SP reset

The P1B2/PWM2 through P1B0/PWM0 pins are set in the general-purpose output port mode.
The output value is “low level”.
The value of each PWM data register (including the timer 3 modulo register) is “1FFH”.

(3) On execution of clock stop instruction

The P1B2/PWM2 through P1B0/PWM0 pins are set in the general-purpose output port mode.
The output value is the “previous contents of the output latch”.
The value of each PWM data register (including the timer 3 modulo register) is “1FFH”.

(4) At CE reset

The previous status is retained.
That is, if the D/A converter is being used, the PWM output is retained as is. If timer 3 is being used, counting continues.
While timer 3 is being used, the DI status is set (in which all interrupts are disabled).

(5) In halt status

The previous status is retained.
That is, if the D/A converter is being used, the PWM output is retained as is. If timer 3 is being used, counting continues.

14. A/D CONVERTER

14.1 Outline of A/D Converter

Figure 14-1 outlines the A/D converter.

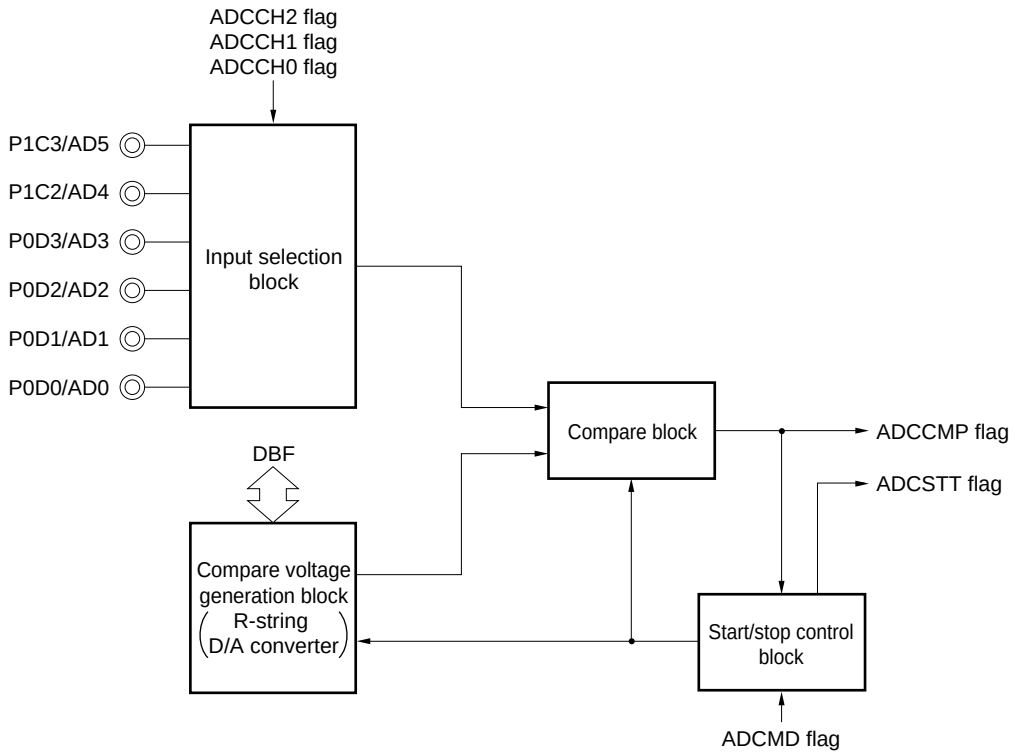
The A/D converter converts an analog voltage input to the AD5 to AD0 pins into an 8-bit digital signal.

Two modes can be selected by using the ADCMD flag: software mode and hardware mode.

In the software mode, a voltage input to a pin is compared with an internal reference voltage, and the result of the comparison is detected by the ADCCMP flag. By judging this result in software and by sequentially selecting reference voltages, the A/D converter can be used as a successive approximation A/D converter.

In the hardware mode, reference voltages are automatically selected, and the input voltage is directly detected as 8-bit digital data.

Figure 14-1. Outline of A/D Converter



Remarks 1. ADCCH2 through ADCCH0 (bits 2 through 0 of A/D converter channel selection register: refer to **Figure 14-3**) select pins used for the A/D converter.

2. ADCCMP (bit 0 of A/D converter mode selection register: refer to **Figure 14-5**) detects the result of comparison.

3. ADCSTT (bit 1 of A/D converter mode selection register: refer to **Figure 14-5**) detects the operating status.

4. ADCMD (bit 2 of A/D converter mode selection register: refer to **Figure 14-5**) selects software or hardware mode.

14.2 Input Selection Block

Figure 14-2 shows the configuration of the input selection block.

The input selection block selects a pin to be used by using the ADCCH2 through ADCCH0 flags. Only one pin can be used for the A/D converter. When one of the P0D0/AD0 through P0D3/AD3, P1C2/AD4, and P1C3/AD5 pins is selected, the other five pins are forcibly set in the input port mode.

The P0D0/AD0 through P0D3/AD3 pins can be connected to a pull-down resistor if so specified by the P0DPL0 through P0DPLD3 flags. To use the P0D0/AD0 through P0D3/AD3 pins for the A/D converter, therefore, disconnect their pull-down resistors to correctly detect an external input analog voltage.

Figure 14-3 shows the configuration of the A/D converter channel selection register.

Figure 14-2. Configuration of Input Selection Block

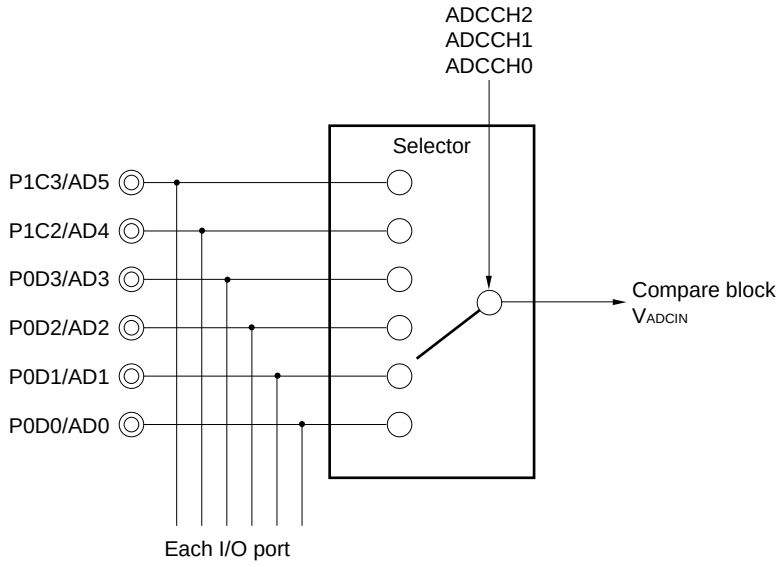


Figure 14-3. Configuration of A/D Converter Channel Selection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
A/D converter channel selection	0	ADCCCH2	ADCCCH1	ADCCCH0	24H	R/W

ADCCCH2	ADCCCH1	ADCCCH0	Selects pin used for A/D converter
0	0	0	A/D converter not used (general-purpose input port)
0	0	1	P0D0/AD0 pin
0	1	0	P0D1/AD1 pin
0	1	1	P0D2/AD2 pin
1	0	0	P0D3/AD3 pin
1	0	1	P1C2/AD4 pin
1	1	0	P1C3/AD5 pin
1	1	1	Setting prohibited

Fixed to “0”

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset		0	0	0
	CE reset		Retained		
Clock stop			Retained		

14.3 Compare Voltage Generation and Compare Blocks

Figure 14-4 shows the configuration of the compare voltage generation block and compare block.

The compare voltage generation block switches a tap decoder according to the 8-bit data set to the A/D converter reference voltage setting register and generates 256 different of compare voltages V_{ADCREF}.

In other words, this block is an R-string D/A converter.

The supply voltage to this R-string D/A converter is the same as the supply voltage V_{DD} of the device.

The compare block compares voltage V_{ADCIN} input from a pin with compare voltage V_{ADCREF}.

Comparison can be made in two modes, software mode and hardware mode, which can be selected by the ADCMD flag.

In the software mode, a compare voltage is set to the A/D converter reference voltage setting register by software, and one set compare voltage is compared with the input voltage, and the result of the comparison is detected by the ADCCMP flag.

In the hardware mode, once comparison has been started, the hardware automatically changes the value of the A/D converter reference voltage setting register. On completion of the comparison, the value of the A/D converter reference voltage setting register is read and is loaded as an 8-bit data.

Figures 14-5 and 14-6 show the configuration of each flag and A/D converter reference voltage setting register.

Figure 14-4. Configuration of Compare Voltage Generation and Compare Blocks

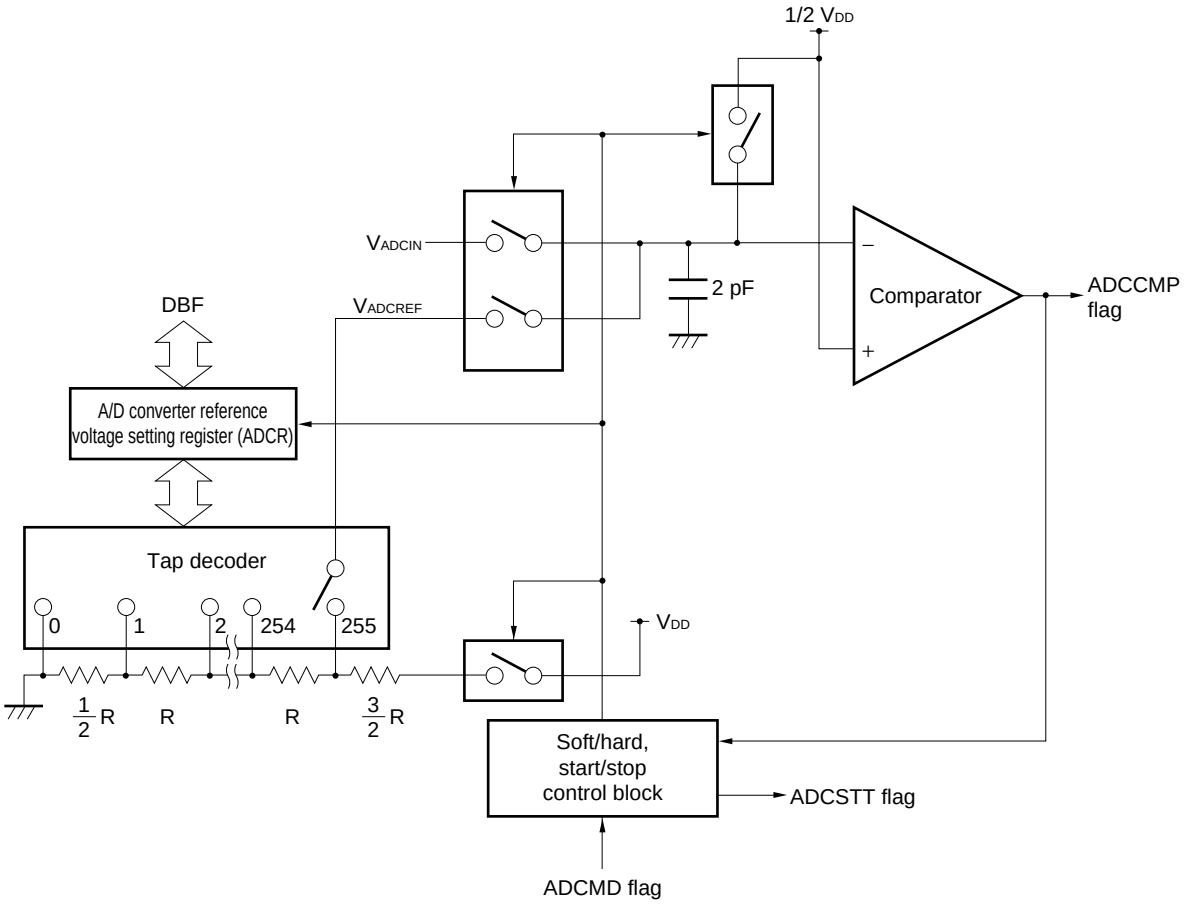


Figure 14-5. Configuration of A/D Converter Mode Selection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
A/D converter mode selection	0	ADCCMD	ADCSTT	ADCCMP	25H	R/W

ADCCMP — Detects result of comparison by A/D converter

0	$V_{ADCIN} < V_{ADCREF}$
1	$V_{ADCIN} > V_{ADCREF}$

ADCSTT — Detects operating status of A/D converter in hardware mode

0	End of conversion
1	Conversion in progress

ADCCMD — Selects compare mode of A/D converter and starts or stops A/D converter

0	Software mode[Note 1]
1	Hardware mode[Note 2]

Fixed to “0”

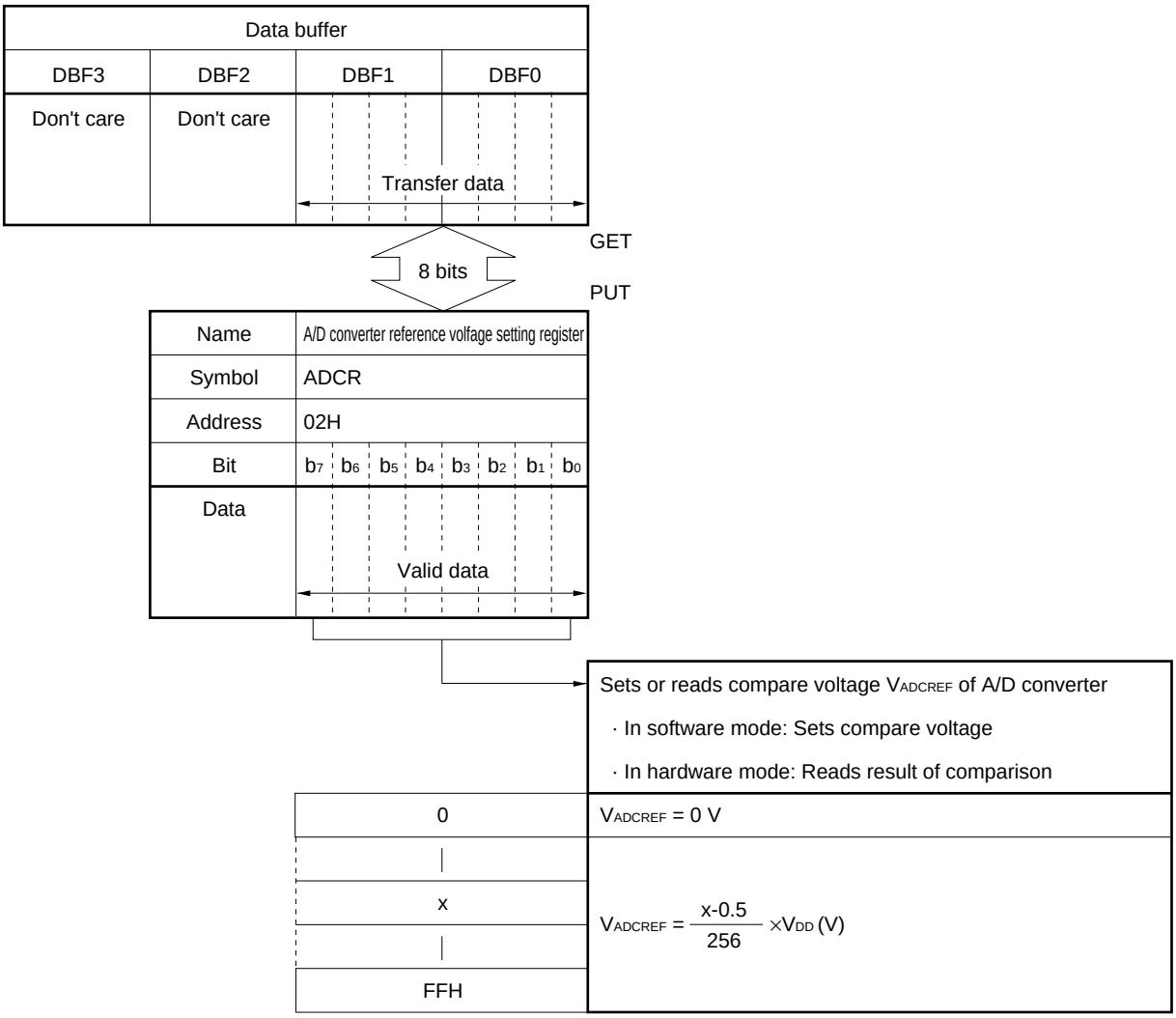
		b_3	b_2	b_1	b_0
At reset	Power-ON reser	0	0	0	0
	WDT&SP reser		0	0	0
	CE reser		R	0	0
Clock stop			R	0	R

R:Retained

Notes 1. A/D conversion under execution is stopped if “0” is written to this bit.

2. A/D operation is started in the hardware mode when “1” is written to this bit. In the software mode, operation is started as soon as data has been written (by the PUT instruction) to the A/D converter reference voltage setting register (ADCR).

Figure 14-6. Configuration of A/D Converter Reference Voltage Setting Register



At reset	Power-ON reset	0
	WDT&SP reset	0
	CE reset	Retained[Note]
Clock stop		Retained[Note]

Note "0" in the hardware mode.

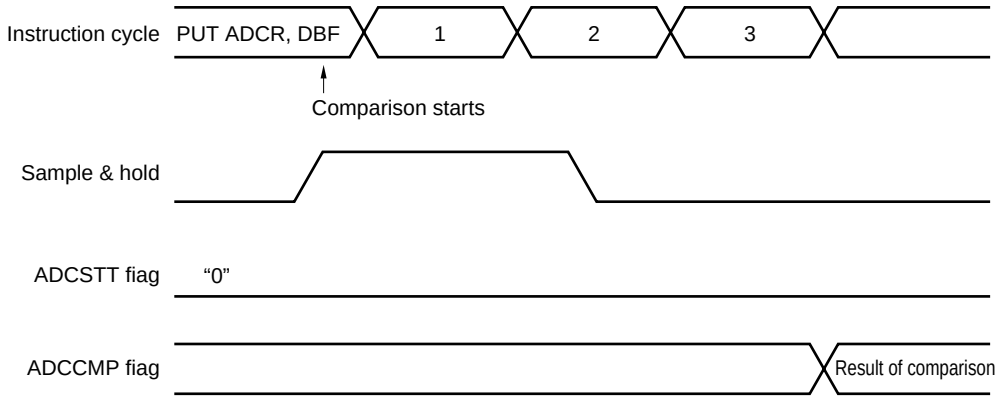
14.4 Comparison Timing Chart

14.4.1 In software mode

Comparison is completed three instructions after data has been set (by the PUT instruction) to the A/D converter reference voltage setting register (ADCR).

Figure 14-7 shows the timing chart.

Figure 14-7. Timing Chart of Comparison by A/D Converter

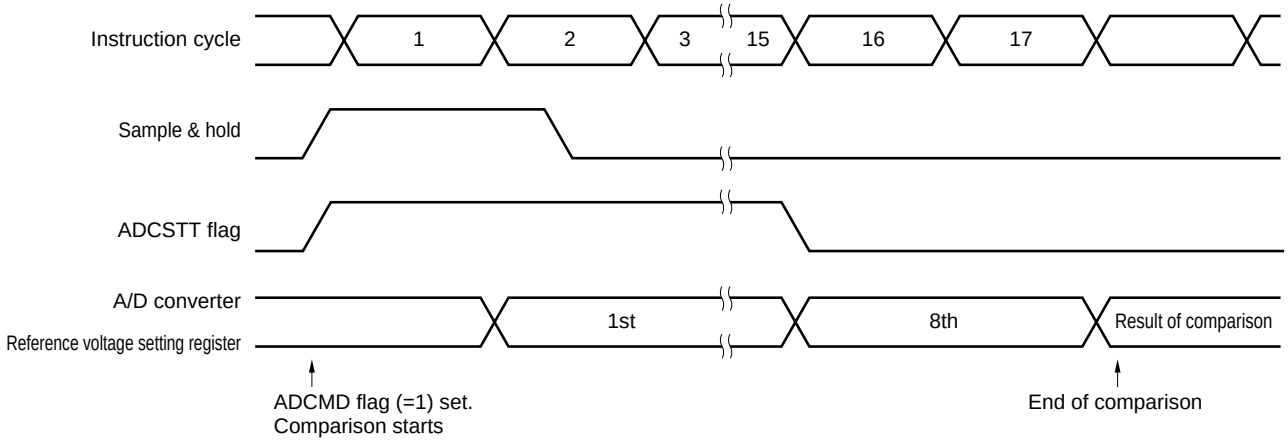


14.4.2 In hardware mode

When the ADCMD flag is set to “1”, A/D conversion is started. The ADCSTT flag is set to “1”, and comparison is completed after 17 instructions have been executed. At this time, the ADCSTT flag is reset to “0” after 15 instructions have been executed after the ADCMD flag was set to “1”. This is because execution time of two instructions is required to judge the status of the ADCSTT flag. For details, also refer to **14.5 Using A/D Converter**.

Figure 14-8 shows the timing chart.

Figure 14-8. Timing Chart of Comparison by A/D Converter



14.5 Using A/D Converter

14.5.1 Software mode

The software mode is convenient for comparing one compare voltage.

An example of a program in this mode is shown below.

Example To compare input voltage V_{ADCIN} of AD0 pin with compare voltage V_{ADCREF} (127.5/256 V_{DD}), and branch to AAA if $V_{ADCIN} < V_{ADCREF}$, or to BBB if $V_{ADCIN} > V_{ADCREF}$

```
ADCR7  FLG          0.0EH.3  ; Defines each bit of DBF as ADCR data setting flag
ADCR6  FLG          0.0EH.2
ADCR5  FLG          0.0EH.1
ADCR4  FLG          0.0EH.0
ADCR3  FLG          0.0FH.3
ADCR2  FLG          0.0FH.2
ADCR1  FLG          0.0FH.1
ADCR0  FLG          0.0FH.0

BANK15
INITFLG NOT P0DPLD3, NOT P0DPLD2, NOT P0DPLD1, P0DPLD0      ; Disconnects pull-down resistor of P0D0 pin
BANK0
INITFLG NOT ADCCH2, NOT ADCCH1, ADCCH0                      ; Selects AD0 pin for A/D converter
CLR1    ADCMD                                               ; Sets software mode
INITFLG ADCR7, NOT ADCR6, NOT ADCR5, NOT ADCR4              ;
INITFLG NOT ADCR3, NOT ADCR2, NOT ADCR1, NOT ADCR0          ;
PUT     ADCR, DBF                                           ; Sets compare voltage VADCREF
NOP                                                         ; Waits for duration of  three instructions
NOP                                                         ;
NOP                                                         ;
SKT1    ADCCMP                                              ; Judges result of comparison
BR      AAA
BR      BBB
```

14.5.2 Hardware mode

Here is a program example:

Example To detect the value of analog input roltage V_{ADCIN} of AD0 pin.

```
        BANK15
        INITFLG NOT P0DPLD3, NOT P0DPLD2, NOT P0DPLD1, P0DPLD0  ; Disconnects pull-down resistor of P0D0 pin
        BANK0
        INITFLG NOT ADCCH2, NOT ADCCH1, ADCCH0                  ; Selects AD0 pin for A/D converter
        SET1    ADCMD                                           ; Sets hardware mode and starts conversion
LOOP:
        SKT1    ADCSTT                                          ; Detects end of A/D conversion
                                                                ; Embedded macro instruction
        ;PEEK WR, .MF. ADCSTT SHR4 AND 0FH
        ;SKT1 WR,#.DF.ADCSTT AND 0FH
        BR      LOOP                                            ; Conversion in progress

        GET     DBF,ADCR                                        ; Stores result of conversion to DBF
```

14.6 Cautions on Using A/D Converter

14.6.1 Cautions on selecting A/D converter pin

When one of the P0D0/AD0 through P0D3/AD3, P1C2/AD4, and P1C3/AD5 pins is selected, the other five pins are forcibly set in the input port mode. The P0D0/AD0 through P0D3/AD3 pins can be connected to a pull-down resistor if so specified by the P0DPL0 through P0DPLD3 flags in bank 15. To use the P0D0/AD0 through P0D3/AD3 pins for the A/D converter, therefore, disconnect their pull-down resistors to correctly detect an external input analog voltage.

14.7 Status at Reset

14.7.1 At power-ON reset

All the P0D0/AD0 through P0D3/AD3, P1C2/AD4, and P1C3/AD5 pins are set in the general-purpose input port mode.

The P0D0 through P0D3 pins are connected with a pull-down resistor.

14.7.2 At WDT&SP reset

All the P0D0/AD0 through P0D3/AD3, P1C2/AD4, and P1C3/AD5 pins are set in the general-purpose input port mode.

The P0D0 through P0D3 pins are connected with a pull-down resistor.

14.7.3 At CE reset

The status of the pin selected for the A/D converter is retained as is.

The previous status of the pull-down resistor of the P0D0 through P0D3 pins is retained.

14.7.4 On execution of clock stop instruction

The status of the pin selected for the A/D converter is retained as is.

The previous status of the pull-down resistor of the P0D0 through P0D3 pins is retained.

14.7.5 In halt status

The status of the pin selected for the A/D converter is retained as is.

The previous status of the pull-down resistor of the P0D0 through P0D3 pins is retained.

15. D/A CONVERTER (PWM mode)

15.1 Outline of D/A Converter

Figure 15-1 outlines the D/A converter.

The D/A converter outputs a signal whose duty factor is varied by means of PWM (Pulse Width Modulation). By connecting an external lowpass filter to the D/A converter, a digital signal can be converted into an analog signal.

Each pin of the D/A converter can output a variable-duty signal independently of the others.

Whether an 8-bit counter or 9-bit counter is used for the D/A converter can be specified by software.

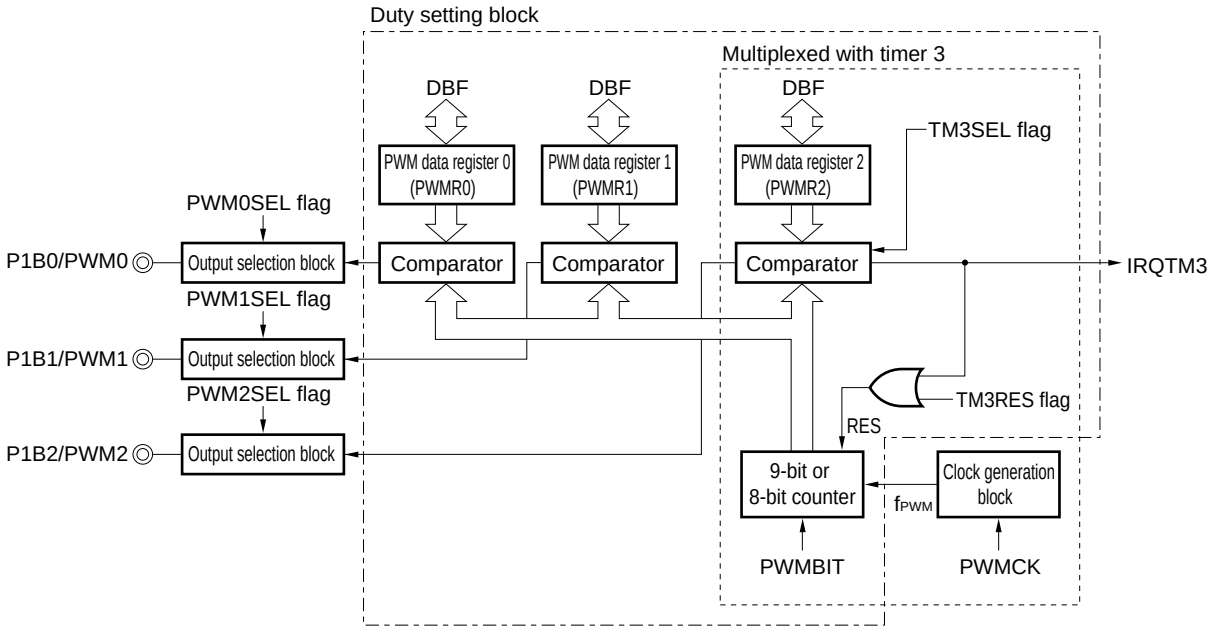
When the 8-bit counter is selected, two output frequencies, 4.4 kHz and 440 Hz can be selected, and the duty factor of the output signal can be varied in 256 steps.

When the 9-bit counter is selected, two output frequencies, 2.2 kHz and 220 Hz, can be selected, and the duty factor can be varied in 512 steps.

When the D/A converter is not used, it can be used as timer 3, which counts the basic clock (1.125 or 0.1125 MHz) with an 8-bit counter.

For the details of timer 3, refer to **13. TIMER 3**.

Figure 15-1. Outline of D/A Converter



Remarks 1. PWM2SEL through PWM0SEL (bits 2 through 0 of PWM/general-purpose port pin function selection register: refer to **Figure 15-4**) select a general-purpose output port of D/A converter.

2. PWMBIT (bit 2 of PWM clock selection register: refer to **Figure 15-2**) selects the number of bits (8 or 9 bits) of the PWM counter.

3. PWMCK (bit 0 of PWM clock selection register: refer to **Figure 15-2**) selects the output frequency of PWM timer.

4. TM3SEL (bit 3 of timer 3 control register: refer to **Figure 15-5**) selects timer 3 or D/A converter.

5. TM3RES (bit 0 of timer 3 control register: refer to **Figure 15-5**) controls resetting of timer 3 counter.

15.2 PWM Clock Selection Register

The PWM clock selection register specifies whether the PWM counter is used as an 8-bit counter or 9-bit counter when the D/A converter is used for PWM output.

Figure 15-2 shows the configuration of the PWM clock selection register.

Figure 15-2. Configuration of PWM Clock Selection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
PWM clock selection	0	PWMBIT	0	PWMCK	26H	R/W

Selects output frequency of timer 3	
0	4.4 kHz (8 bits) /2.2 kHz (9 bits)
1	440 Hz (8 bits) /220 Hz (9 bits)

Fixed to “0”

Selects number of bits of PWM counter	
0	8 bits
1	9 bits

Fixed to “0”

At reset	Power-ON reset	0	0	0	0
	WDT&SP reset		0		0
	CE reset		R		R
Clock stop			0		0

R: Retained

15.3 PWM Output Selection Block

The output selection block specifies whether each output pin of the D/A converter is used for the D/A converter or as a general-purpose output port, by using the PWM2SEL through PWM0SEL flags of the PWM/general-purpose port pin function selection register.

Figure 15-3 shows the configuration of the output selection block, and Figure 15-4 shows the configuration of the PWM/general-purpose port pin function selection register.

Each pin can be changed between the D/A converter mode and general-purpose output port mode independently of the others.

Because each output pin is an N-ch open-drain output pin, an external pull-up resistor is necessary.

When the D/A converter is used as timer 3, the P1B2/PWM2 through P1B0/PWM0 pins are automatically set in the general-purpose output port mode, regardless of the values set to the PWM2SEL through PWM0SEL flags.

Figure 15-3. Configuration of Output Selection Block

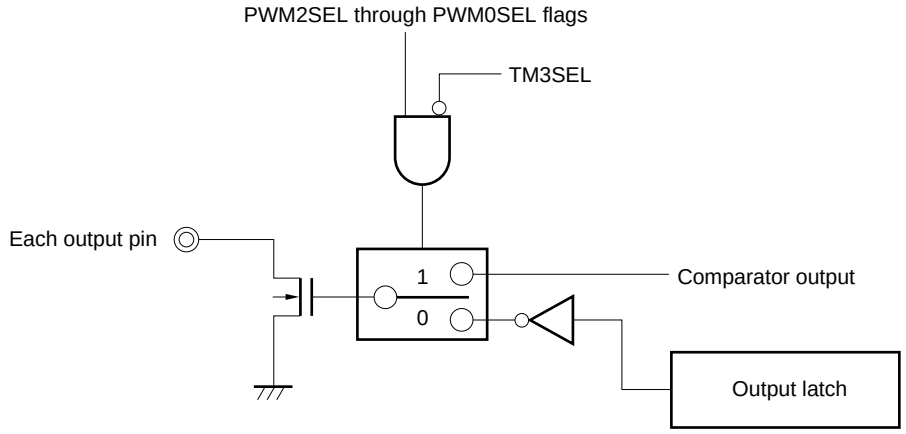


Figure 15-4. Configuration of PWM/General-Purpose Port Pin Function Selection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
PWM/general-purpose port pin function selection	0	PWM2SEL	PWM1SEL	PWM0SEL	27H	R/W

PWM2SEL — Selects function of P1B2/PWM2 pin
- 0: General-purpose output port
- 1: D/A converter

PWM1SEL — Selects function of P1B1/PWM1 pin
- 0: General-purpose output port
- 1: D/A converter

PWM0SEL — Selects function of P1B0/PWM0 pin
- 0: General-purpose output port
- 1: D/A converter

b_3 — Fixed to “0”

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset		0	0	0
	CE reset		Retained		
Clock stop			0	0	0

Figure 15-5. Configuration of Timer 3 Control Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Timer 3 control	TM3SEL	0	TM3EN	TM3RES	28H	R/W

TM3RES — Resets counter

0	Does not change
1	Resets

TM3EN — Starts or stops counter

0	Stops
1	Starts

Bit 2 — Fixed to “0”

TM3SEL — Selects timer 3 or D/A converter

0	D/A converter (PWM output)
1	Timer 3

		TM3SEL	0	TM3EN	TM3RES
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset	0		0	0
	CE reset	R		Retained	
Clock stop		0		0	0

R: Retained

15.4 Duty Setting Block

15.4.1 PWM duty with 8-bit counter selected

The duty setting block compares the value set to each PWM data register (PWMR2 to PWMR0) with the value of the basic clock counted by each 8-bit counter. If the value of the PWM data register is greater, the block outputs a high level. If the value of the PWM data register is less, it outputs a low level.

Where the value set to the PWM data register is "x", therefore, the duty factor can be calculated by the following expression.

$$\text{Duty: } D = \frac{x + 0.25}{256} \times 100\%$$

Remark 0.25 is an offset, and a high level is output even where x = 0.

Data is set to each PWM data register for each pin via data buffer (DBF). However, the same basic clock, PWM counter, and output frequency must be selected for each pin. This means that each pin cannot output a duty factor of a different cycle independently of the others.

Because the basic clock frequency is 1.125 or 0.1125 MHz, the frequency and cycle of the output signal can be calculated as follows.

(1) Where output frequency is 4.4 kHz and basic clock frequency is 1.125 MHz

$$\text{Frequency: } f = \frac{1.125\ \text{MHz}}{256} = 4.3945\ \text{kHz}$$

$$\text{Cycle: } T = \frac{256}{1.125\ \text{MHz}} = 227.56\ \mu\text{s}$$

(2) Where output frequency is 440 Hz and basic clock frequency is 0.1125 MHz

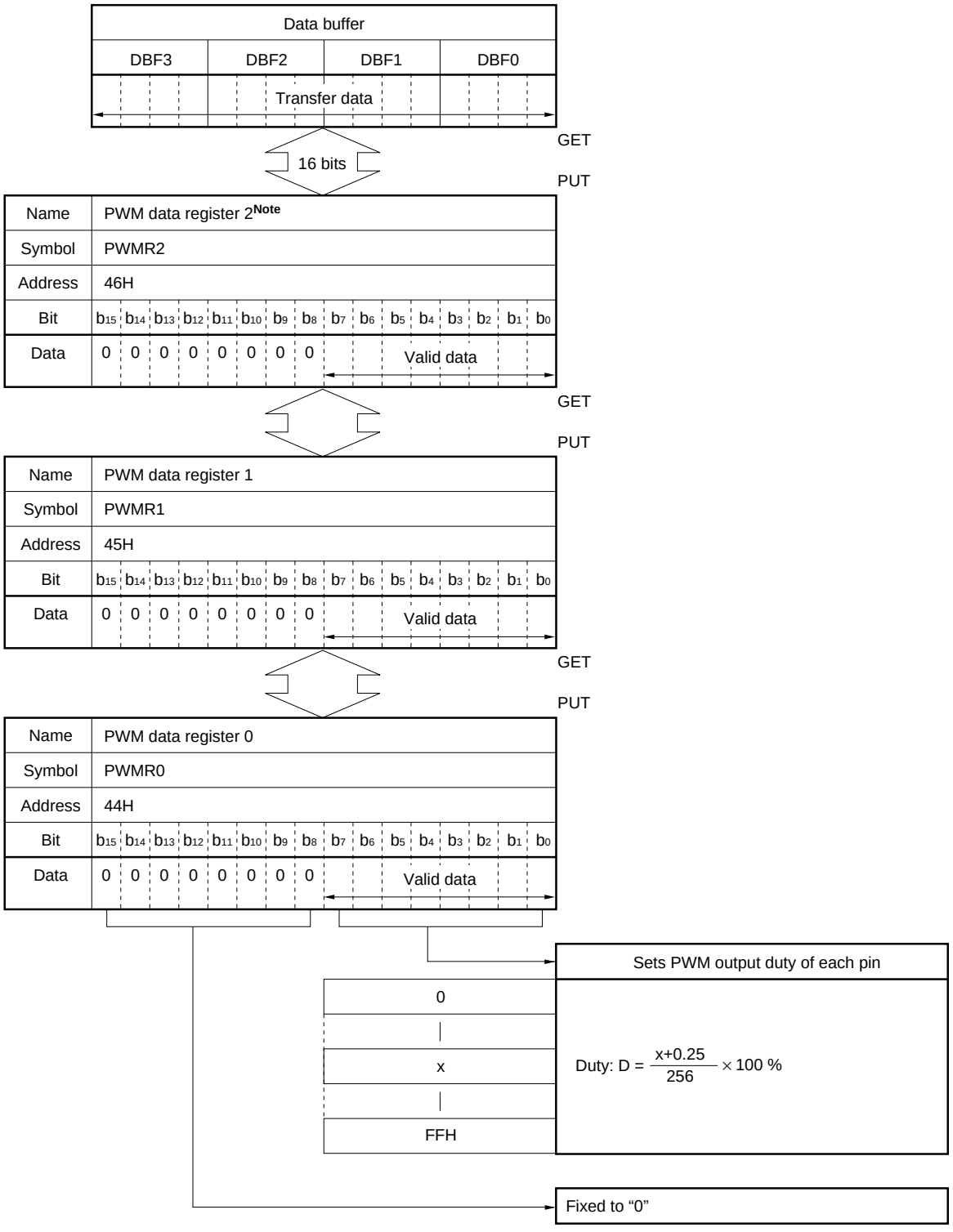
$$\text{Frequency: } f = \frac{0.1125\ \text{MHz}}{256} = 439.45\ \text{Hz}$$

$$\text{Cycle: } T = \frac{256}{0.1125\ \text{MHz}} = 2.2756\ \text{ms}$$

Because the duty setting register of the PWM data registers and timer 3 modulo register are the same register, they cannot be used at the same time.

When timer 3 is used, PWM data registers 1 and 0 can be used as 8-bit data latches.

Figure 15-6. Configuration of PWM Data Registers (with 8-bit counter selected)



Note This register is multiplexed with timer 3 modulo register.

15.4.2 PWM duty with 9-bit counter selected

The duty setting block compares the value set to each PWM data register (PWMR2 to PWMR0) with the value of the basic clock counted by each 9-bit counter. If the value of the PWM data register is greater, the block outputs a high level. If the value of the PWM data register is less, it outputs a low level.

Where the value set to the PWM data register is "x", therefore, the duty factor can be calculated by the following expression.

$$\text{Duty: } D = \frac{x + 0.25}{512} \times 100\%$$

Remark 0.25 is an offset, and a high level is output even where x = 0.

Data is set to each PWM data register for each pin via data buffer (DBF). However, the same basic clock, PWM counter, and output frequency must be selected for each pin. This means that each pin cannot output a duty factor of a different cycle independently of the others.

Because the basic clock frequency is 1.125 or 0.1125 MHz, the frequency and cycle of the output signal can be calculated as follows.

(1) Where output frequency is 2.2 kHz and basic clock frequency is 1.125 MHz

$$\text{Frequency: } f = \frac{1.125\ \text{MHz}}{512} = 2.197\ \text{kHz}$$

$$\text{Cycle: } T = \frac{512}{1.125\ \text{MHz}} = 455.11\ \mu\text{s}$$

(2) Where output frequency is 220 Hz and basic clock frequency is 0.1125 MHz

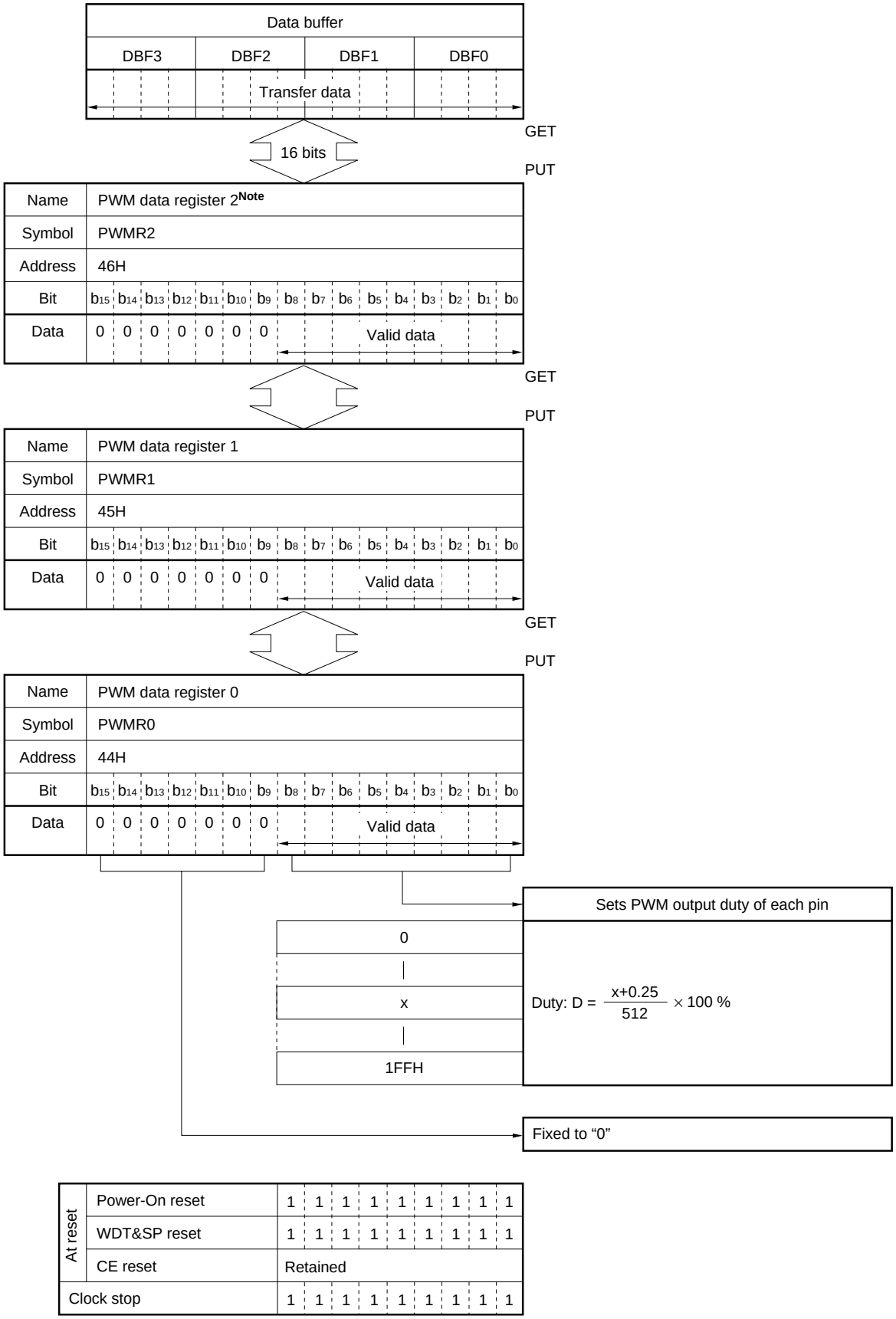
$$\text{Frequency: } f = \frac{0.1125\ \text{MHz}}{512} = 219.73\ \text{Hz}$$

$$\text{Cycle: } T = \frac{512}{0.1125\ \text{MHz}} = 4.5511\ \text{ms}$$

Because the duty setting register of the PWM data registers and timer 3 modulo register are the same register, they cannot be used at the same time.

When timer 3 is used, PWM data registers 1 and 0 can be used as 8-bit data latches.

Figure 15-7. Configuration of PWM Data Registers (with 9-bit counter selected)



Note This register is multiplexed with timer 3 modulo register.

15.5 Clock Generation Block

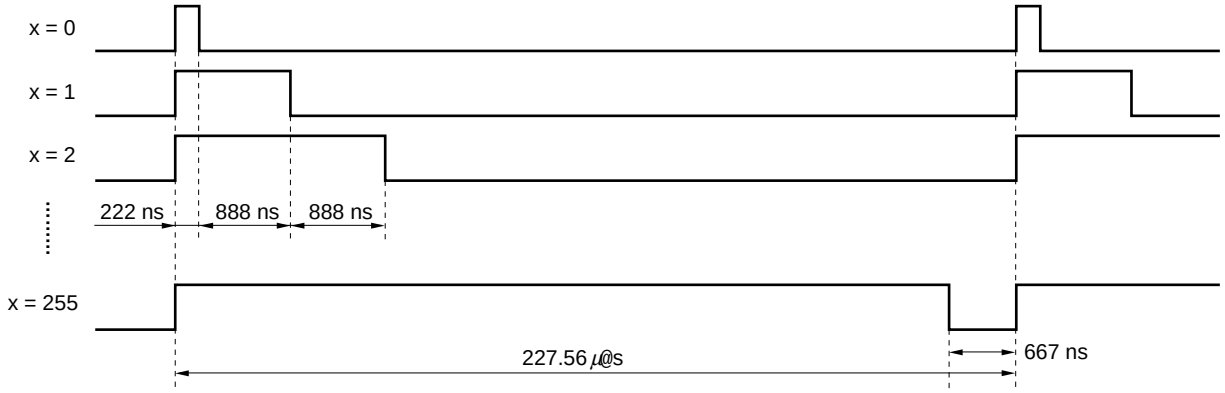
The clock generation block outputs a basic clock to set the duty factor of each output signal.
Two output frequencies, 1.125 MHz and 112.5 kHz, can be selected.

15.6 D/A Converter Output Wave

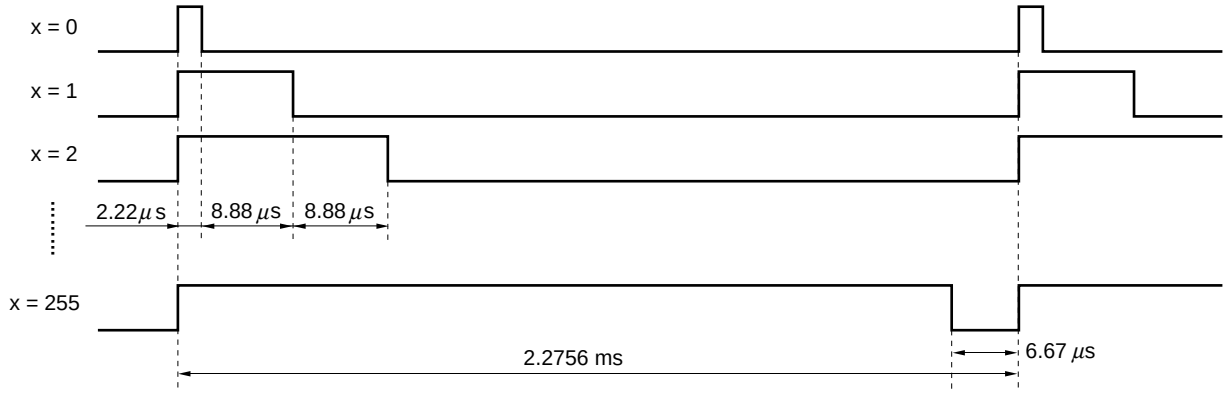
(1) shows the relationship between the duty factor and output wave.
(2) shows the output wave of each pin. Each output pin has a phase different from the others.

(1) Duty and output wave

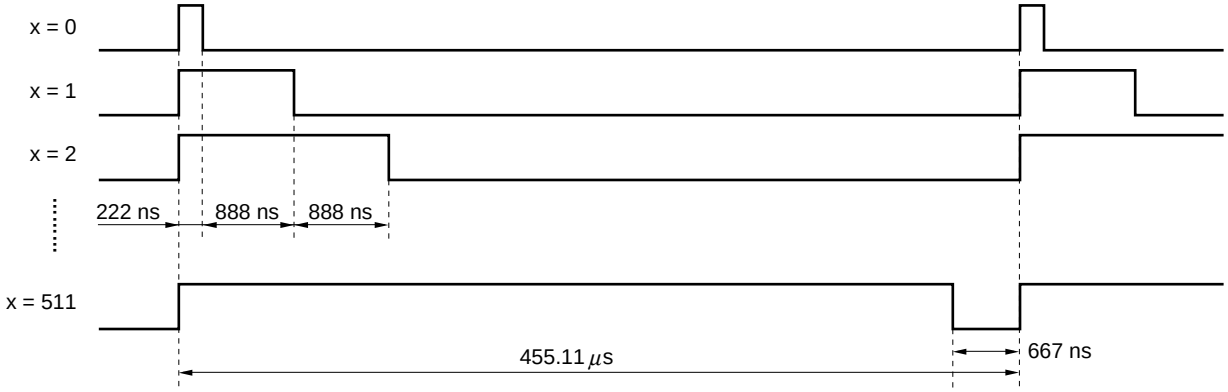
(a) With 8-bit counter and 4.4 kHz selected



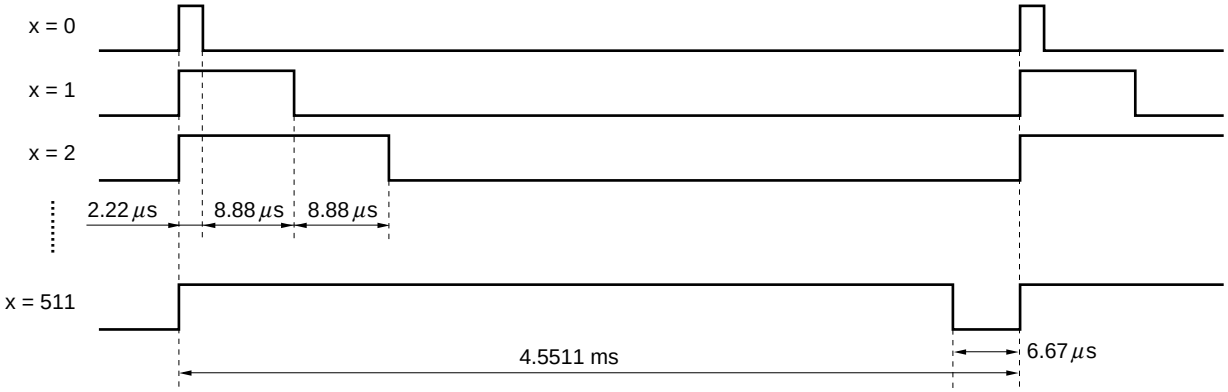
(b) With 8-bit counter and 440 Hz selected



(c) With 9-bit counter and 2.2 kHz selected

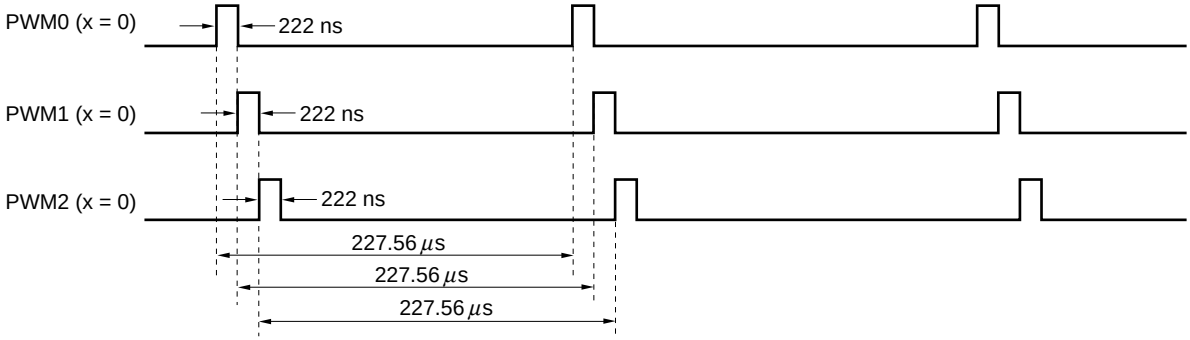


(d) With 9-bit counter and 220 Hz selected

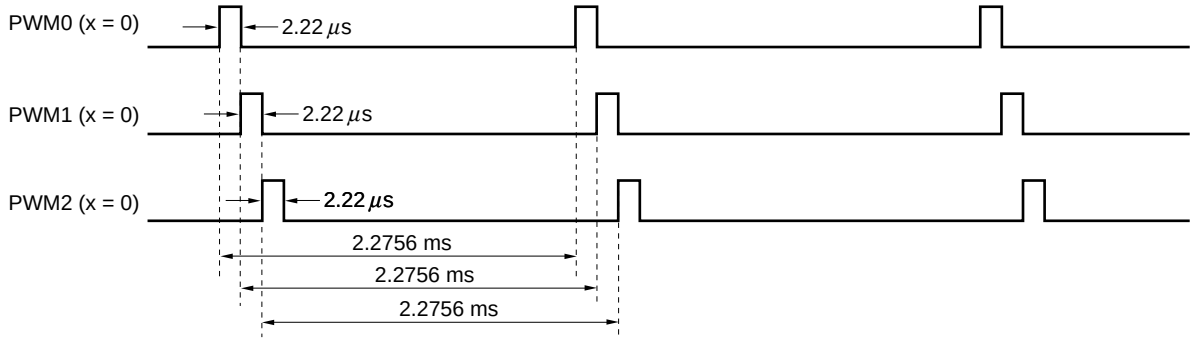


(2) Each pin and output wave

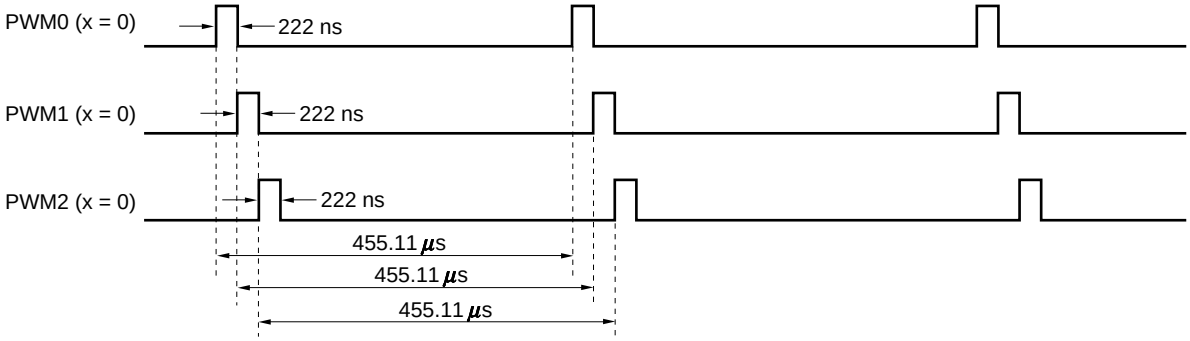
(a) With 8-bit counter and 4.4 kHz selected



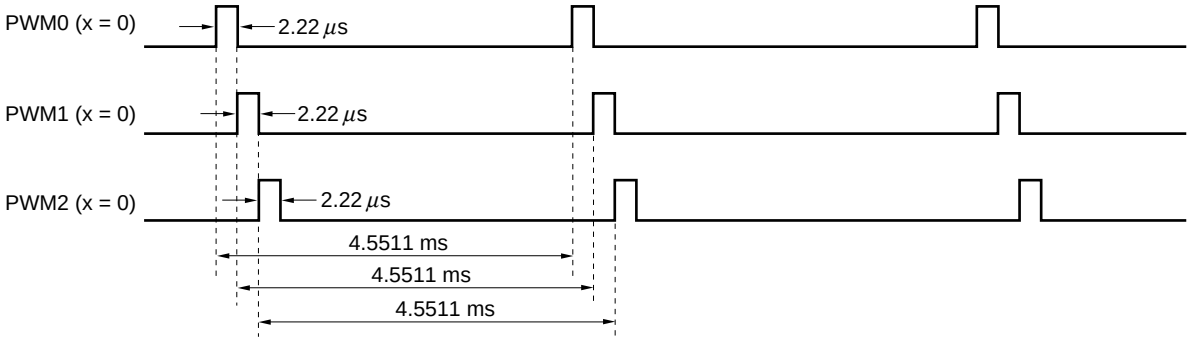
(b) With 8-bit counter and 440 Hz selected



(c) With 9-bit counter and 2.2 kHz selected



(d) With 9-bit counter and 220 Hz selected



15.7 Example of Using D/A Converter

An example of a program using the D/A converter is shown below.

Example This program increments the duty factor of the PWM1 pin every 1 second.

```
        PWM1DATA  DAT     0000H

    INITIAL:
        INITFLG   NOT PWM2SEL, NOT PWM1SEL, NOT PWM0SEL
        ;         (General-purpose port), (General-purpose port), (General-purpose port)
        INITFLG     PWMBIT  , NOT PWMCK
        ;         (9-bit counter), (1.125 MHz)

LOOP0:
        BANK1
        CLR1      P1B1
        BANK0
        CLR1      TM3SEL          ; Selects D/A converter

        MOV       DBF2, #PWM1DATA SHR 8 AND 0FH
        MOV       DBF1, #PWM1DATA SHR 4 AND 0FH
        MOV       DBF0, #PWM1DATA AND 0FH
        SET1      PWM1SEL         ; Sets PWM1/P1B1 pin in PWM output port mode

LOOP1:                            ; Duty: 0.25/512 to 511.25/512 (PWM output)
        PUT       PWM1R, DBF
        GET2      TM3RES, TM3EN   ; Resets and starts counter

        Waits for 1 second

        GET       DBF, PWM1R
        ADD       DBF0, #1
        ADDC      DBF1, #0
        ADDC      DBF2, #1
        SKGE      DBF2, #2
        BR        LOOP1

LOOP2:                            ; Port outputs high level
        BANK1
        SET1      P1B1
        BANK0
        CLR1      PWM1SEL         ; Sets PWM1/P1B1 pin in general-purpose output port mode

        Waits for 1 second

        BR        LOOP0
```

15.8 Status at Reset

15.8.1 At power-ON reset

The P1B0/PWM0 through P1B2/PWM2 pins are set in the general-purpose output port mode.

The output value is “low level”.

The value of each PWM data register (including the timer 3 modulo register) is “1FFH”.

15.8.2 At WDT&SP reset

The P1B0/PWM0 through P1B2/PWM2 pins are set in the general-purpose output port mode.

The output value is “low level”.

The value of each PWM data register (including the timer 3 modulo register) is “1FFH”.

15.8.3 At CE reset

The P1B0/PWM2 through P1B2/PWM2 pins retain the previous status.

That is, if the D/A converter is being used, the PWM output is retained as is. If timer 3 is being used, counting continues.

15.8.4 On execution of clock stop instruction

The P1B0/PWM0 through P1B2/PWM2 pins are set in the general-purpose output port mode.

The output value is the “previous contents of the output latch”.

The value of each PWM data register (including the timer 3 modulo register) is “1FFH”.

15.8.5 In halt status

The P1B0/PWM0 through P1B2/PWM2 pins retain the previous status.

That is, if the D/A converter is being used, the PWM output is retained as is. If timer 3 is being used, counting continues.

16. SERIAL INTERFACE

16.1 Outline of Serial Interface

Figure 16-1 shows the outline of the serial interface.

Table 16-1 shows classification and communication modes of the serial interface.

The serial interface consists of serial interface 2 (SIO2) and serial interface 3 (SIO3).

Figure 16-1. Outline of Serial Interface

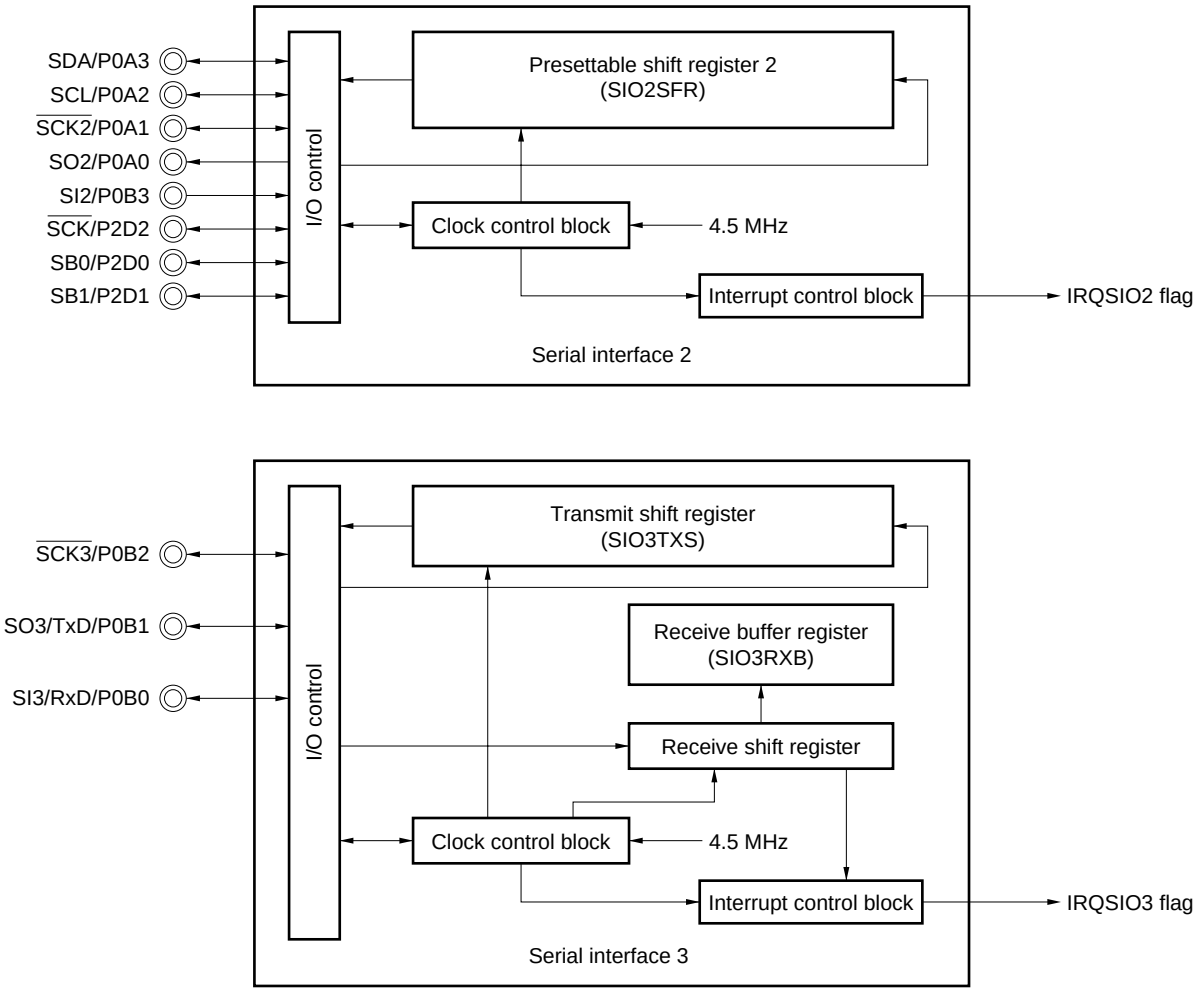


Table 16-1. Classification and Communication Modes of Serial Interface

Channel	Communication Mode	Pin Used
Serial interface 2	I^2C (Inter IC) bus mode	P0A2/SCL
	2-wire serial I/O mode	P0A3/SDA
	SBI (serial bus interface) mode	P2D0/SB0
		P2D1/SB1
		P2D2/$\overline{\text{SCK}}$
	3-wire serial I/O mode	P0A0/SO2
		P0A1/$\overline{\text{SCK2}}$
		P0B3/SI2
Serial interface 3	3-wire serial I/O mode	P0B0/SI3
		P0B1/SO3
		P0B2/$\overline{\text{SCK3}}$
	UART	P0B0/RxD
		P0B1/TxD

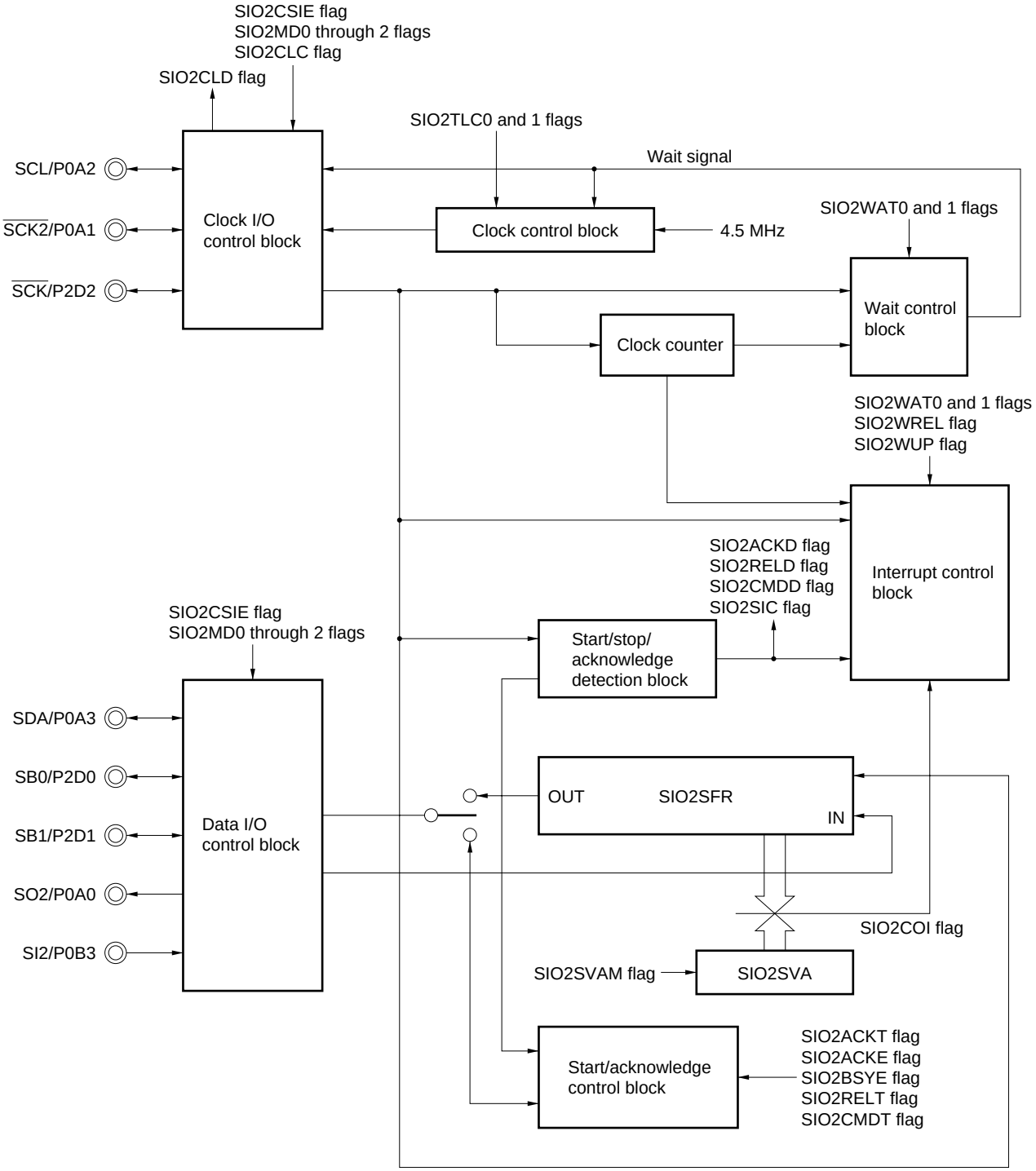
16.2 Serial Interface 2

16.2.1 Outline of serial interface 2

Figure 16-2 shows the outline of serial interface 2.

Serial interface 2 can be used in the I^2C bus, SBI, and 2-wire or 3-wire serial I/O modes.

Figure 16-2. Outline of Serial Interface 2



16.2.2 Control registers of serial interface 2

Serial interface 2 is controlled by the following six registers:

- Serial I/O2 operation mode register 0
- Serial I/O2 operation mode register 1
- Serial I/O2SBI register 0
- Serial I/O2SBI register 1
- Serial I/O2 interrupt timing specification register 0
- Serial I/O2 interrupt timing specification register 1

(1) Serial I/O2 operation mode register 0

Figure 16-3 shows the configuration of the serial I/O2 operation mode register 0.

This register controls the operation of serial interface 2, and select the clock to be used.

Figure 16-3. Configuration of Serial I/O2 Operation Mode Register 0

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Serial I/O2 operation mode register 0	SIO2CSIE	SIO2COI	SIO2TCL1	SIO2TCL0	0FH	R/W

Sets internal shift clock frequency	
SIO2TCL1	SIO2TCL0
0	0	93.75 kHz
0	1	375 kHz
1	0	281.25 kHz
1	1	46.875 kHz

Coincidence signal from address comparator	
SIO2COI	
0	Does not coincide
1	Coincides

Controls operation of serial I/O2	
SIO2CSIE	
0	Stops operation (wait status)
1	Enables operation

		SIO2CSIE	SIO2COI	SIO2TCL1	SIO2TCL0
At reset	Power-ON reset	0	0	0	0
	WDT & SP reset	0	0	0	0
	CE reset	0	0	0	0
Clock stop		0	0	0	0

Caution Read the SIO2COI flag after completion of transfer. This flag is undefined during transfer.

(2) Serial I/O2 operation mode register 1

Figure 16-4 shows the configuration of the serial I/O2 operation mode register 1.

This register controls the operation of serial interface 2 and selects the clock to be used.

Figure 16-4. Configuration of Serial I/O2 Operation Mode Register 1

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Serial I/O2 operation mode register 1	SIO2WUP	SIO2MD2	SIO2MD1	SIO2MD0	0EH	R/W

Sets direction of shift clock (SIO2MD0)

	SBI/I²C bus mode	Serial I/O mode
0	Slave operation (external clock input)	External clock input
1	Master operation (internal clock output)	Internal clock output

Sets operation mode of serial I/O2 (SIO2MD2, SIO2MD1)

SIO2MD2	SIO2MD1	
0	0	3-wire serial I/O mode
0	1	SBI mode (SB1)
1	0	SBI mode (SB0)
1	1	2-wire serial I/O mode or I²C bus mode

Controls wake-up function (SIO2WUP)

0	Stops wake up
1	Enables wake up (used in SBI and I²C bus modes)

		SIO2WUP	SIO2MD2	SIO2MD1	SIO2MD0
At reset	Power-ON reset	0	0	0	0
	WDT & SP reset	0	0	0	0
	CE reset	0	0	0	0
Clock stop		0	0	0	0

Caution Set the SIO2WUP flag before starting address reception.

(3) Serial I/O2SBI register 0

This register controls and detects the status of the serial bus interface.

Figure 16-5 shows the configuration of the serial I/O2SBI register 0.

Figure 16-5. Configuration of Serial I/O2SBI Register 0

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Serial I/O2 SBI register 0	SIO2BSYE	SIO2ACKD	SIO2ACKE	SIO2ACKT	0DH	R/W

SIO2ACKT — Controls trigger output of acknowledge signal

0	Automatically cleared after flag has been set
1	Acknowledge signal is output in synchronization with next clock if this bit is set after completion of transfer.

SIO2ACKE — Controls output of acknowledge signal

0	Stops automatic output of acknowledge signal (output by SIO2ACKT is possible).
1	Before completion of transfer: Acknowledge signal is output in synchronization with 9th clock. After completion of transfer : Acknowledge signal is output in synchronization with clock immediately after set instruction. Unlike SIO2ACKT, this bit is not cleared automatically after output of acknowledge signal.

SIO2ACKD — Detects acknowledge signal

0	Acknowledge signal is not detected.
1	Acknowledge signal is detected (in synchronization with rising of clock).

SIO2BSYE — Control of synchronous busy signal output

0	Disables output of busy signal in synchronization with falling of clock immediately after clear instruction.
1	Outputs busy signal in synchronization with falling of clock after acknowledge signal.

		SIO2BSYE	SIO2ACKD	SIO2ACKE	SIO2ACKT
At reset	Power-ON reset	0	0	0	0
	WDT & SP reset	0	0	0	0
	CE reset	0	0	0	0
Clock stop		0	0	0	0

Cautions 1. When using the SIO2ACKE flag, set the flag until the falling of the 9th counting of clock SCL during I^2C bus operation.

2. When using the SIO2ACKT flag, clear SIO2ACKE to "0". During I^2C bus operation, set the flag until the falling of the 9th counting of clock SCL.

3. Because the SIO2ACKT flag is automatically cleared after it has been set to "1", it is always "0" when read.

(4) Serial I/O2SBI register 1

This register controls and detects the status of the serial bus interface.

Figure 16-6 shows the configuration of the serial I/O2SBI register 1.

Figure 16-6. Configuration of Serial I/O2SBI Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Serial I/O2 SBI register 1	SIO2CMDD	SIO2RELD	SIO2CMDT	SIO2RELT	0CH	R/W

SIO2RELT — Controls trigger output of bus release signal

0	Automatically cleared after flag has been set.
1	Sets SO2 latch by setting flag. Used to output bus release signal.

SIO2CMDT — Controls trigger output of command signal

0	Automatically cleared after flag has been set.
1	Resets SO2 latch by setting flag. Used to output command signal.

SIO2RELD — Detects bus release signal

0	Bus release signal is not detected.
1	Bus release signal is detected.

SIO2CMDD — Detects command signal

0	Command signal is not detected.
1	Command signal is detected.

		SIO2CMDD	SIO2RELD	SIO2CMDT	SIO2RELT
At reset	Power-ON reset	0	0	0	0
	WDT & SP reset	0	0	0	0
	CE reset	0	0	0	0
Clock stop		0	0	0	0

Cautions 1. The SIO2CMDT flag is disabled from being set during serial transfer.

2. Because the SIO2CMDT flag is automatically cleared after it has been set to “1”, it is always “0” when read.

3. Because the SIO2RELT flag is automatically cleared after it has been set to “1”, it is always “0” when read.

(5) Serial I/O2 interrupt timing specification register 0

This register controls and detects the status of the serial bus interface.

Figure 16-7 shows the configuration of the serial I/O2 interrupt timing specification register 0.

Figure 16-7. Configuration of Serial I/O2 Interrupt Timing Specification Register 0

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Serial I/O2 interrupt timing specification register 0	0	SIO2CLD	SIO2SIC	SIO2SVAM	0BH	R/W

Bits of SIO2SVA register used as slave address	
0	Bits 0 through 7
1	Bits 1 through 7

Selects interrupt source of serial I/O2	
0	Interrupt occurs only on completion of serial transfer
1	Interrupt occurs on completion of serial transfer or on detection of bus release

Detects P0A2/SCL pin level	
0	P0A2/SCL pin is low
1	P0A2/SCL pin is high

Fixed to "0"

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT & SP reset	0	0	0	0
	CE reset	0	0	0	0
Clock stop		0	0	0	0

Caution Writing this register is inhibited during serial transfer.

(6) Serial I/O2 interrupt timing specification register 1

This register controls and detects the status of the serial bus interface.

Figure 16-8 shows the configuration of the serial I/O2 interrupt timing specification register 1.

Figure 16-8. Configuration of Serial I/O2 Interrupt Timing Specification Register 1

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Serial I/O2 interrupt timing specification register 1	SIO2CLC	SIO2WREL	SIO2WAT1	SIO2WAT0	0AH	R/W

Controls wait and interrupt

0 0	Issues interrupt request at rising of 8th clock
0 1	Issues interrupt request at rising of 8th clock
1 0	Used in I^2C bus mode (8-clock wait). Issues interrupt request at rising of 8th clock of SCL (master makes SCL output low and waits after 8 clocks have been output. Slave makes SCL pin low and requests for wait after it has input 8 clocks).
1 1	Used in I^2C bus mode (9 clock wait). Issues interrupt request at rising of 9th clock of SCL (master makes SCL output low and waits after 9 clocks have been output. Slave makes SCL pin low and requests for wait after it has input 9 clocks).

Wait release control (used in I^2C bus mode)

0	Wait released status
1	Releases wait status (Flag is automatically cleared after wait status has been released.)

Control of P0A2/SCL pin level (used in I^2C bus mode)

0	P0A2/SCL pin is not affected.
1	P0A2/SCL pin goes into high-impedance state.

		SIO2CLC	SIO2WREL	SIO2WAT1	SIO2WAT0
At reset	Power-ON reset	0	0	0	0
	WDT & SP reset	0	0	0	0
	CE reset	0	0	0	0
Clock stop		0	0	0	0

Cautions 1. The SIO2WREL flag can be manipulated only in the wait status.
Because this flag is automatically cleared after it has been set to "1", it is always "0" when read.

2. The SIO2CLC flag is set to "1" when a start/stop signal is created in the I^2C bus mode. It is usually cleared to "0".

3. The wait status set by SIO2WAT0 and SIO2WAT1 is released in the following sequence:

- **SIO2WREL = 1**
- **Data is written to SIO2SFR.**
- **SIO2CSIE = 0**

16.2.3 Presettable shift register 2 (SIO2SFR)

The presettable shift register 2 is an 8-bit register that is used to write serial out data and read serial in data.

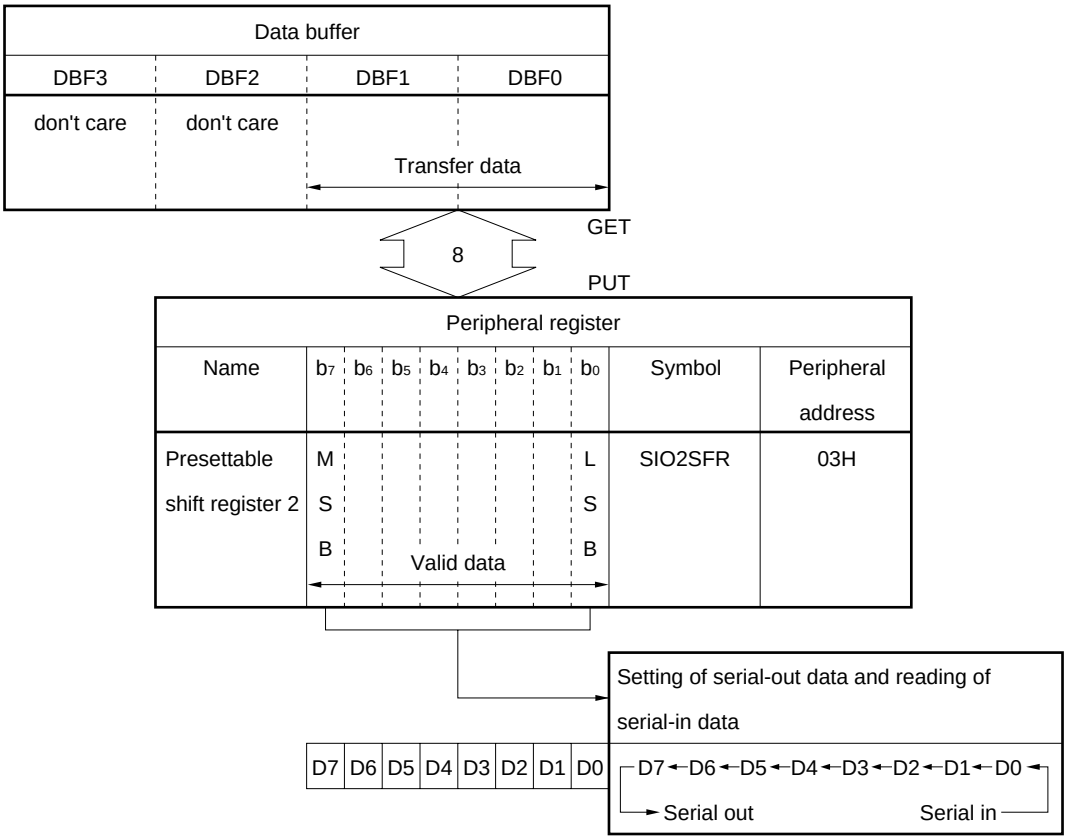
This register writes or reads data via data buffer.

Because the input pin and output pin are multiplexed to configure the bus in the 2-wire serial I/O, I^2C bus, and SBI modes, write FFH to SIO2SFR in the 2-wire serial I/O mode. In the I^2C bus mode, set 1 to SIO2BSYE, and write FFH to SIO2SFR. In the SBI mode, the device that is to receive data must write FFH to SIO2SFR (except when the device receives address with 1 set to SIO2WUP).

In the SBI mode, the busy signal can be released by writing data to SIO2SFR. In this case, SIO2BSYE is not cleared to 0.

Figure 16-9 shows the configuration of the presettable shift register 2.

Figure 16-9. Configuration of Presettable Shift Register 2



16.2.4 Serial I/O2 slave address register (SIO2SVA)

This is an 8-bit register that sets a slave address value when the microcontroller is connected to the serial bus as a slave device.

A slave address is output to the slave devices connected to the master to select a specific slave. The two values (slave address output by the master and the value of SIO2SVA) are compared by the address comparator, and if they coincide, the slave having that slave address is selected. At this time, the SIO2COI flag of the serial I/O2 operation mode register 0 is set to 1.

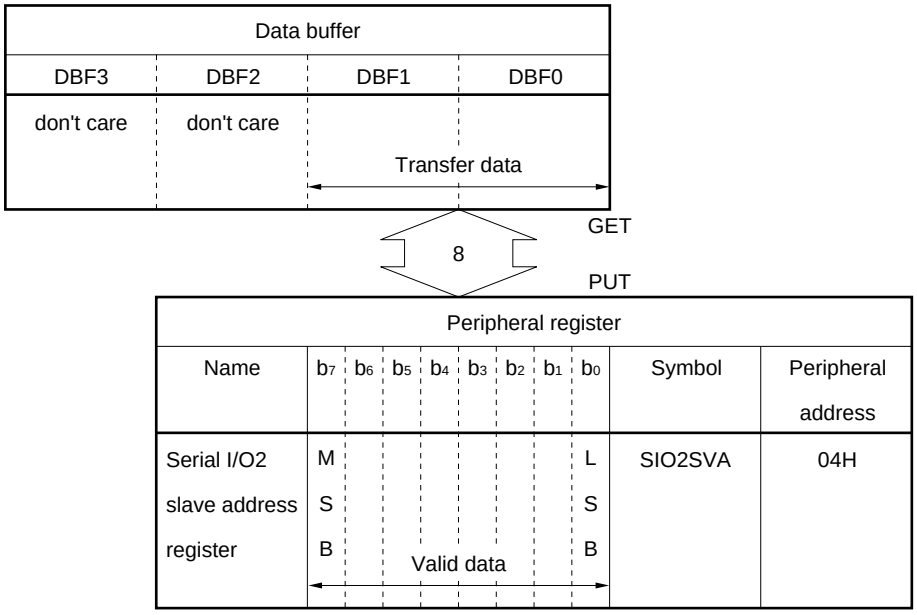
In addition, the data of the high-order 7 bits with the LSB masked can be compared as an address by using the SIO2SVAM flag of the serial I/O2 interrupt timing specification register.

If coincidence is not detected during address reception, SIO2RELD of serial I/O2SBI register 1 is cleared to 0. If the SIO2WUP flag of the serial I/O2 operation mode register is 1, an interrupt request is issued only when coincidence is detected. This interrupt indicates that the master has requested communication.

This register also detects an error when the device is used as the master or slave in the 2-wire serial I/O, I^2C bus, or SBI mode.

Figure 16-10 shows the configuration of the serial I/O2 slave address register.

Figure 16-10. Configuration of Serial I/O2 Slave Address Register



16.2.5 Operation of serial interface 2

The serial interface 2 has the following four operation modes:

- 3-wire serial I/O mode
- SBI mode
- 2-wire serial I/O mode
- I^2C bus mode

Table 16-2 shows the setting of each pin by a control flag in each operation mode.

Table 16-2. Setting Status of Each Pin by Each Control Flag (1/2)

Flag							Pin											
SIO2CSIE	SIO2MD2	SIO2MD1	SIO2WAT1	Communication mode	SIO2MD0	Clock direction	Pin name	P0ABIO3	P0A3	P0ABIO2	P0A2	P0ABIO1	P0A1	P0ABIO0	P0A0	P0BBIO3	P0B3	Setting status of pin
1	0	0	0	3-wire serial I/O	0	External (slave)	P0A1/$\overline{SCK2}$					0	×					External clock input
												1	×					General-purpose output port
					1	Internal (master)						0	×					External clock input
												1	1					Internal clock output
					×		P0A0/SO2							0	×			General-purpose input port
														1	0			Serial output
							P0B3/SI2									0	×	Serial input
																1	×	General-purpose output port
	1	1	0	2-wire serial I/O	0	External (slave)	P0A2/SCL			0	×							External clock input
										1	×							General-purpose output port
					1	Internal (master)				0	×							General-purpose input port
										1	1							Internal clock output
					×		P0A3/SDA	0	×									Serial input
								1	0									Serial output
	1	1	1	I^2C bus serial I/O	0	External (slave)	P0A2/SCL			0	×							External clock input
										1	×							General-purpose output port
					1	Internal (master)				0	×							General-purpose input port
										1	1							Internal clock output
					×		P0A3/SDA	0	×									Serial input
								1	0									Serial output

×: don't care

Table 16-2. Setting Status of Each Pin by Each Control Flag (2/2)

Flag							Pin							
SIO2CSIE	SIO2MD2	SIO2MD1	SIO2WAT1	Communication mode	SIO2MD0	Clock direction	Pin name	P2DBIO2	P2D2	P2DBIO1	P2D1	P2DBIO0	P2D0	Setting status of pin
1	1	0	0	SBI (when data is input to or output from SB0 pin)	0	External (slave)	P2D2/$\overline{SCK}$	0	×					External clock input
								1	×					General-purpose output port
					1	Internal (master)		0	×					General-purpose input port
								1	1					Internal clock output
					×		P2D0/SB0					0	×	Serial input
												1	0	Serial output
	0	1	0	SBI (when data is input to or output from SB1 pin)	0	External (slave)	P2D2/$\overline{SCK}$	0	×					External clock input
								1	×					General-purpose output port
					1	Internal (master)		0	×					General-purpose input port
								1	1					Internal clock output
					×		P2D1/SB1			0	×			Serial input
										1	0			Serial output

×: don't care

16.2.6 3-wire serial I/O mode

(1) Outline of 3-wire serial I/O mode

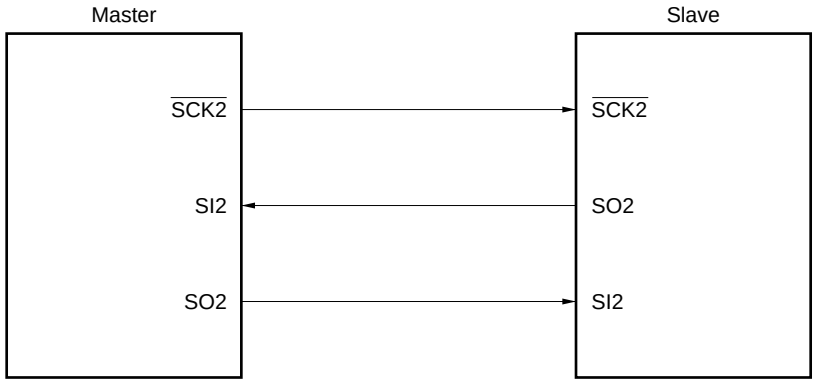
In the 3-wire serial I/O mode, communication is executed by using the $\overline{SCK2}$, SI2, and SO2 pins.

Table 16-3 outlines the 3-wire serial I/O mode.

Table 16-3. Outline of 3-Wire Serial I/O Mode

Pins used for communication	• $\overline{SCK2}$ pin (serial clock I/O pin) • SI2 pin (serial data input pin) • SO2 pin (serial data output pin)	
Transmission/reception operation	Transmit data	Sequentially output from MSB of shift register to data output pin in synchronization with fall of $\overline{SCK2}$ pin
	Receive data	Value of data input pin is sequentially input from LSB of shift register in synchronization with rising of $\overline{SCK2}$ pin.
Transmission/reception start	Master	Transmission/reception is started by setting transfer data to shift register after 3-wire serial I/O master mode has been set.
	Slave	Waits for clock from master with $\overline{SCK2}$ pin going into high-impedance state after 3-wire serial I/O slave mode has been set.
Interrupt	Issues interrupt request flag IRQSIO2 at rising edge of 8th count of clock.	
Clock pin	Master	Stops output of $\overline{SCK2}$ pin at rising edge of 8th count and keeps $\overline{SCK2}$ pin high until next transmission/reception operation is started.
	Slave	Goes into high-impedance state.

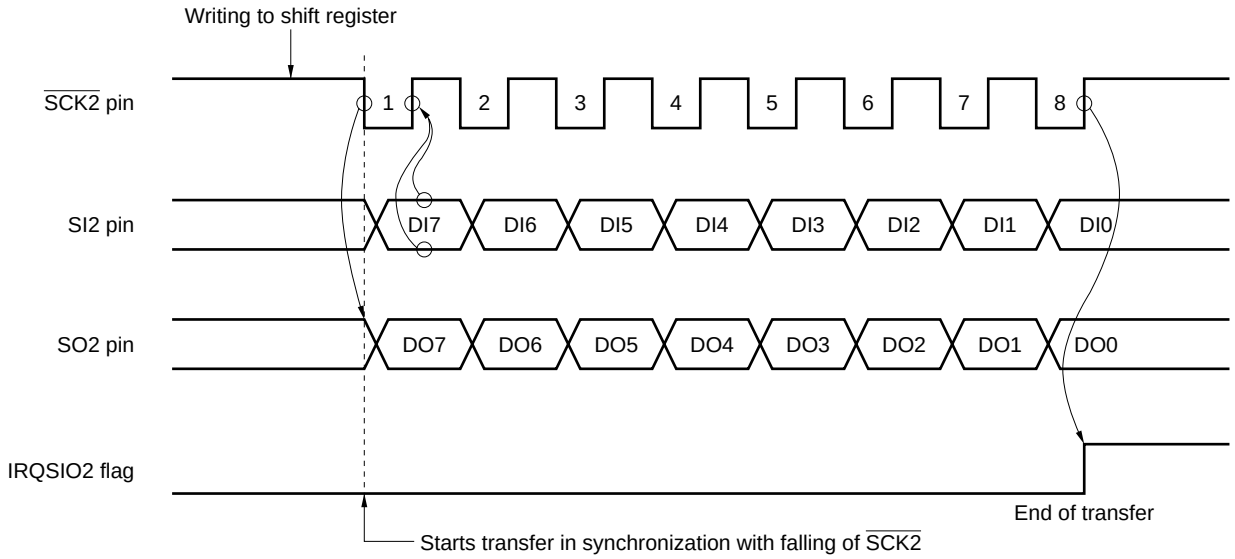
Figure 16-11. Example of Serial Bus Configuration in 3-Wire Serial I/O



(2) Timing chart

Figure 16-12 shows the timing chart in the 3-wire serial I/O mode.

Figure 16-12. Timing Chart in 3-Wire Serial I/O Mode



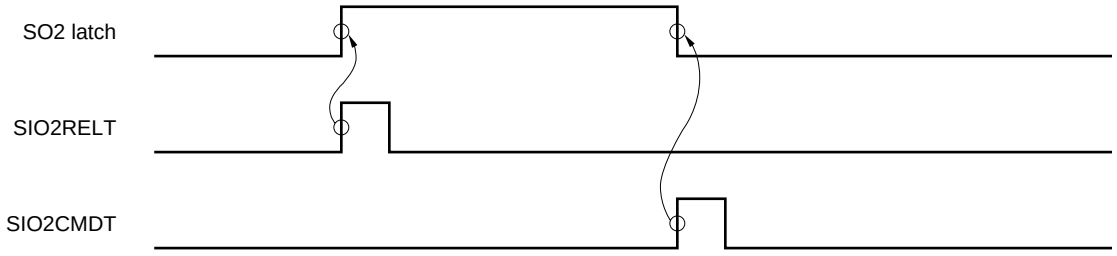
The SO2 pin serves as a CMOS output pin and outputs the status of the SO2 latch. Therefore, the SO2 pin output status can be manipulated by setting the SIO2RELT and SIO2CMDT flags.

However, do not perform this manipulation during serial transfer.

(3) Signals

Figure 16-13 shows the operations of SIO2RELT and SIO2CMDT.

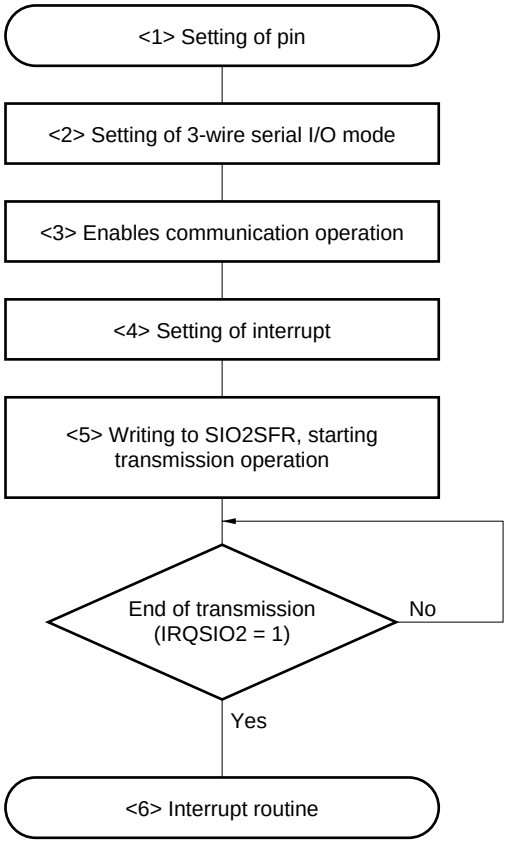
Figure 16-13. Operation of SIO2RELT and SIO2CMDT



(4) Program flowchart in 3-wire serial I/O mode

A program flowchart example in the 3-wire serial I/O transmission mode is shown below.

Figure 16-14. Example Flowchart in 3-Wire Serial I/O Transmission Mode



Remark To execute a 3-wire serial I/O operation with the same setting as before, start from step <5>.

<1> Setting of pin

(a) Setting of data pin in 3-wire serial I/O mode

Set the I/O control mode of the data pin to “1” (output), and the port latch of the data pin to “0”.

(b) Setting of shift clock in 3-wire serial I/O mode

Set the I/O control mode of the shift clock to “1” (output), and the port latch of the shift clock to “1”.

<2> Setting 3-wire serial I/O transmission mode as communication mode

SIO2MD2 = 0, SIO2MD1 = 0

<3> Enabling communication operation (SIO2CSIE = “1”)

(a) To output internal clock from shift clock (SIO2MD0 = “1”)

Output the internal clock.

(b) To input external clock as shift clock (SIO2MD0 = “0”)

Input the external clock.

<4> Setting of interrupt

Execute the “EI instruction” and set the IPSIO2 flag to “1”.

<5> Setting of transmit data to SIO2SFR (PUT SIO2SFR)

The 3-wire serial I/O transmission operation is started as soon as data has been set, and the 8-bit transmit data is output from the SO2 pin.

<6> Interrupt routine

When the 3-wire serial I/O transmission operation has been completed, the interrupt request flag IRQSIO2 is issued. When the interrupt is accepted, execution branches to the vector address.

Caution Transfer is not started even if the SIO2CSIE flag is set to “1” after data has been written to SIO2SFR.

16.2.7 SBI mode

In the SBI (serial bus interface) mode, an “address” to select a target device for serial communication, a “command” that gives the selected device an instruction, and actual “data” can be output to the serial data bus. Therefore, a signal line for handshaking, which is necessary for connecting two or more devices with the conventional clocked serial interface, is not necessary.

(1) Outline of SBI mode

In the SBI mode, communication is performed by using the $\overline{SCK}$ and SB0 (or SB1) pins.

Table 16-4 shows the outline of the SBI mode.

Table 16-4. Outline of SBI Mode

Pin used for communication	• $\overline{SCK}$ pin (serial clock I/O pin) • SB0 (SB1) pin (serial data I/O pin)	
Transmission/reception operation	Transmit data	Sequentially output from MSB of shift register to data I/O pin in synchronization with falling of $\overline{SCK}$ pin
	Receive data	Value of data I/O pin is sequentially input from LSB of shift register in synchronization with rising of $\overline{SCK}$ pin.
Transmission/reception start	Master	Transmission/reception is started by setting transfer data to shift register after SBI mode has been set.
	Slave	Waits for clock from master with $\overline{SCK}$ pin going into high-impedance state after SBI mode has been set.
Interrupt	Issues interrupt request IRQSIO2 at rising edge of 9th count of clock.	
Clock pin	Master	Outputs more than 10 counts and uses 9th count and those that follow for acknowledge. Used to control busy after acknowledge has been detected. Clock line goes high after releasing of busy has been detected.
	Slave	Goes into high-impedance state.

16.2.8 SBI mode operation

SBI is a high-speed serial interface in compliance with the NEC serial bus format.

SBI uses a single master device and employs the clocked serial I/O format with the addition of a bus configuration function. This function enables devices to communicate using only two lines. Thus, when making up a serial bus with two or more microcontrollers and peripheral ICs, the number of ports to be used and the number of wires on the board can be decreased.

The master device outputs three kinds of data to slave devices on the serial data bus: “addresses” to select a device to be communicated with, “commands” to instruct the selected device, and “data” which is actually required.

The slave device can identify the received data into “address”, “command”, or “data”, by hardware. This function enables the application program serial interface 2 control portions to be simplified.

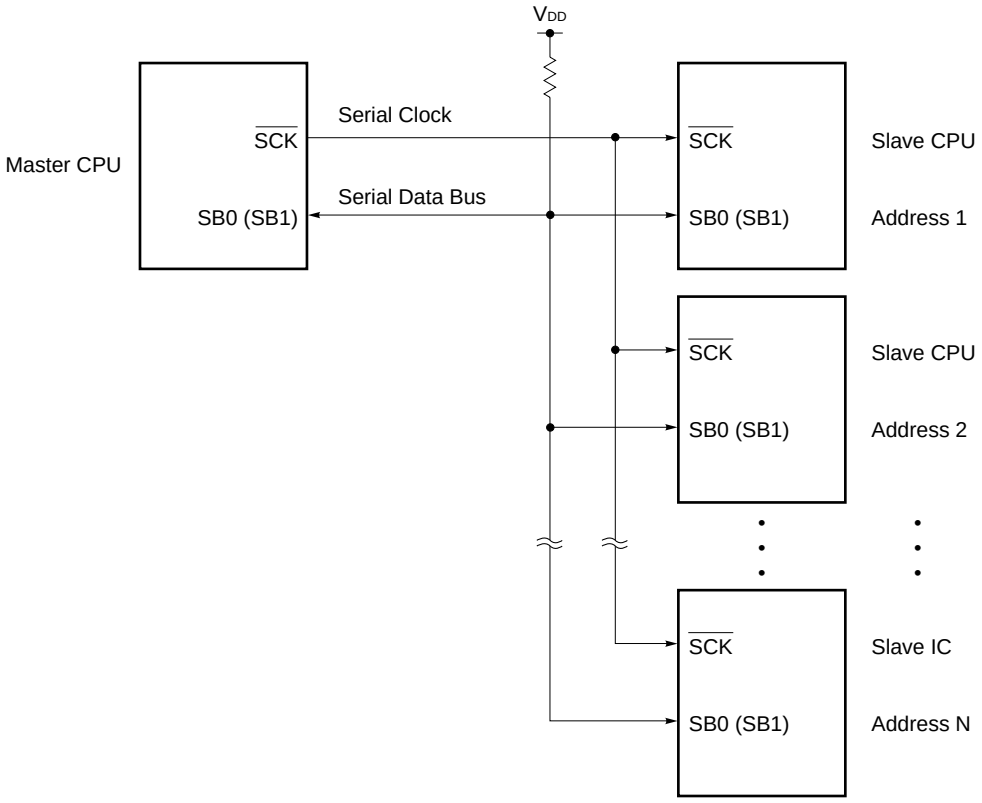
The SBI function is incorporated into various devices including 75X/XL-series devices and 78K-series and 17K-series of 8-bit and 16-bit single-chip microcontrollers.

Figure 16-15 shows a serial bus configuration example when a CPU having a serial interface compliant with SBI and peripheral ICs are used.

In SBI, the SB0 (SB1) serial data I/O pin is an open-drain output pin and therefore the serial data bus line behaves in the same way as the wired-OR configuration. In addition, a pull-up resistor must be connected to the serial data bus line.

When the SBI mode is used, refer to **(9) SBI mode precautions (d)** described later.

Figure 16-15. Example of Serial Bus Configuration with SBI



Caution When exchanging the master CPU/slave CPU, a pull-up resistor is necessary for the serial clock line ($\overline{SCK}$) as well because serial clock line ($\overline{SCK}$) input/output switching is carried out asynchronously between the master and slave CPUs.

(1) SBI functions

In the conventional serial I/O format, when a serial bus is configured by connecting two or more devices, many ports and wiring are necessary, to provide chip select signal to identify command and data, and to judge the busy state, because only the data transfer function is available. If these operations are to be controlled by software, the software must be heavily loaded.

In SBI, a serial bus can be configured with two signal lines of $\overline{SCK}$ and SB0 (SB1). Thus, use of SBI leads to reduction in the number of microcontroller ports and that of wirings and routings on the board.

The SBI functions are described below.

(a) Address/command/data identify function

Serial data is distinguished into addresses, commands, and data.

(b) Chip select function by address transmission

The master executes slave chip selection by address transmission.

(c) Wake-up function

The slave can easily judge address reception (chip select judgment) with the wake-up function (which can be set/reset by software).

When the wake-up function is set, the interrupt request signal (IRQSIO2) is generated upon reception of a match address.

Thus, when communication is executed with two or more devices, the CPU except the selected slave devices can operate regardless of underway serial communications.

(d) Acknowledge signal ($\overline{ACK}$) control function

The acknowledge signal to check serial data reception is controlled.

(e) Busy signal ($\overline{BUSY}$) control function

The busy signal to report the slave busy state is controlled.

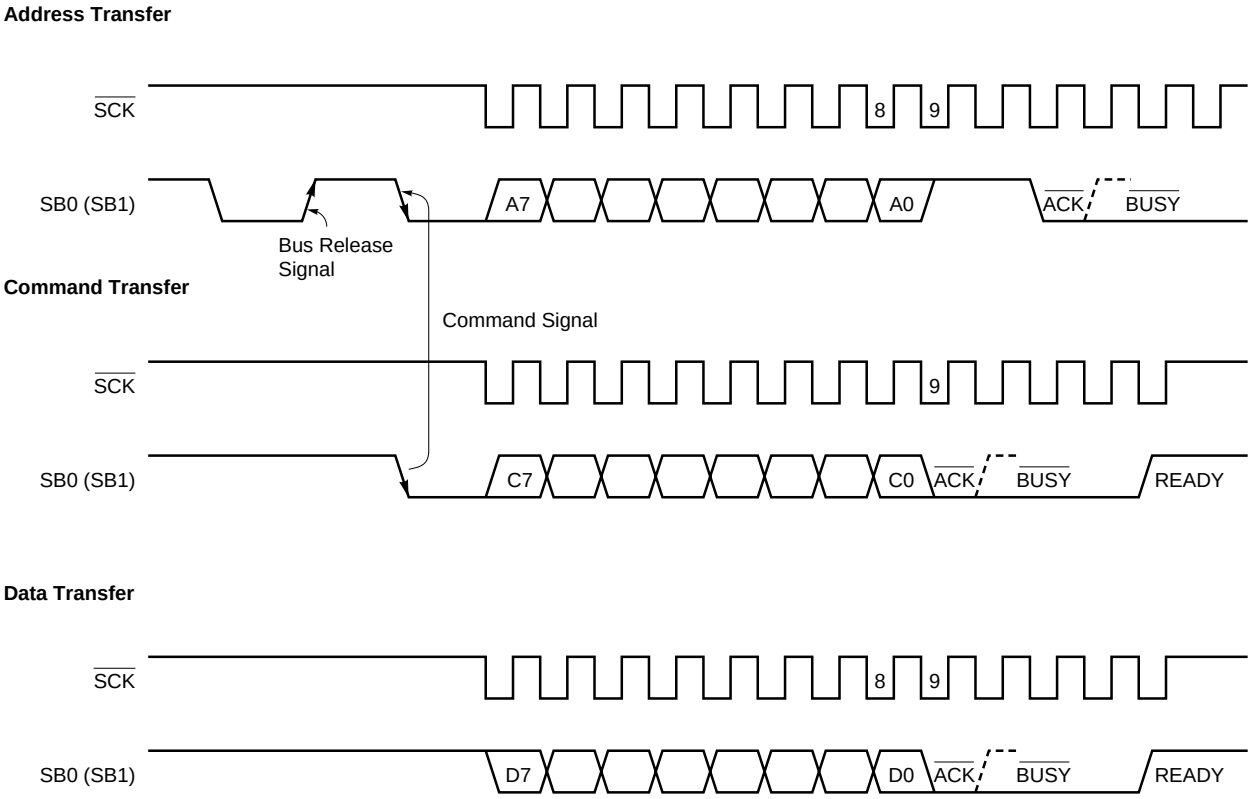
(2) SBI definition

The SBI serial data format and the signals to be used are defined as follows.

Serial data to be transferred with SBI consists of three kinds of data: “address”, “command”, and “data”.

Figure 16-16 shows the address, command, and data transfer timings.

Figure 16-16. SBI Transfer Timings



Remark The dotted line indicates READY status.

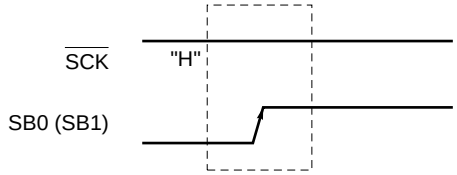
The bus release signal and the command signal are output by the master device. $\overline{\text{BUSY}}$ is output by the slave signal. $\overline{\text{ACK}}$ can be output by either the master or slave device (normally, the 8-bit data receiver outputs). Serial clocks continue to be output by the master device from 8-bit data transfer start to $\overline{\text{BUSY}}$ reset.

(a) Bus release signal (REL)

The bus release signal is a signal with the SB0 (SB1) line which has changed from the low level to the high level when the $\overline{SCK}$ line is at the high level (without serial clock output).

This signal is output by the master device.

Figure 16-17. Bus Release Signal

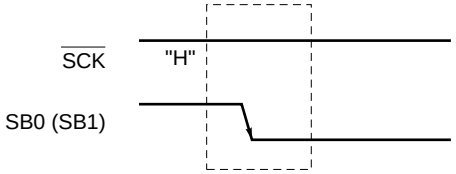


The bus release signal indicates that the master device is going to transmit an address to the slave device. The slave device incorporates hardware to detect the bus release signal.

(b) Command signal (CMD)

The command signal is a signal with the SB0 (SB1) line which has changed from the high level to the low level when the $\overline{SCK}$ line is at the high level (without serial clock output). This signal is output by the master device.

Figure 16-18. Command Signal

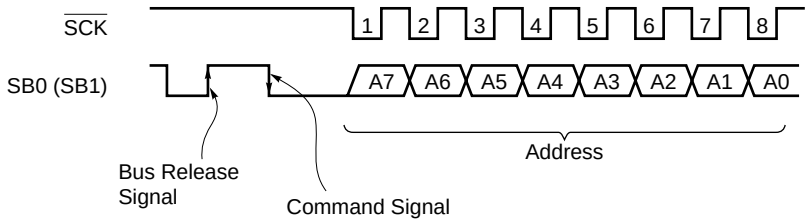


The slave device incorporates hardware to detect the command signal.

(c) Address

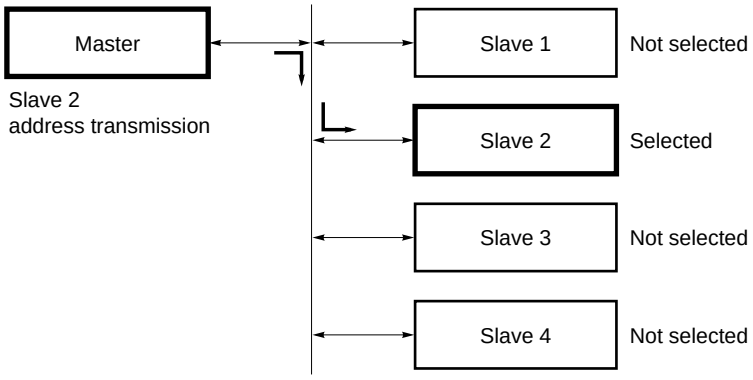
An address is 8-bit data which the master device outputs to the slave device connected to the bus line in order to select a particular slave device.

Figure 16-19. Addresses



8-bit data following bus release and command signals is defined as an “address”. In the slave device, this condition is detected by hardware and whether or not 8-bit data matches the own specification number (slave address) is checked by hardware. If the 8-bit data matches the slave address, the slave device has been selected. After that, communication with the master device continues until a release instruction is received from the master device.

Figure 16-20. Slave Selection with Address



(d) Command and data

The master device transmits commands to, and transmits/receives data to/from the slave device selected by address transmission.

Figure 16-21. Commands

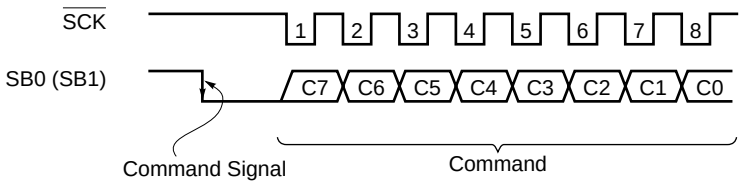
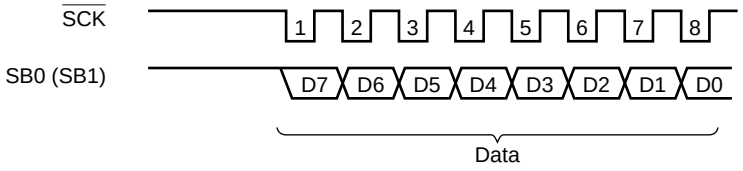


Figure 16-22. Data



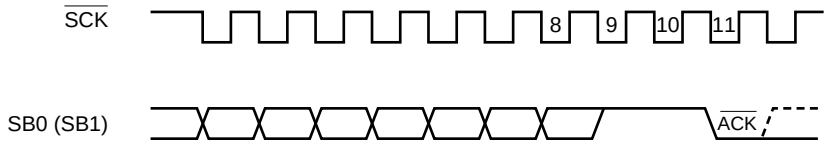
8-bit data following a command signal is defined as “command” data. 8-bit data without command signal is defined as “data”. Command and data operation procedures are allowed to determine by user according to communications specifications.

(e) Acknowledge signal ($\overline{ACK}$)

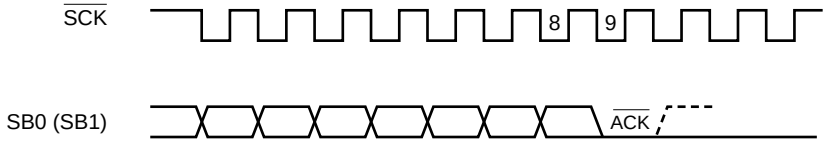
The acknowledge signal is used to check serial data reception between transmitter and receiver.

Figure 16-23. Acknowledge Signal

[When output in synchronization with 11th clock $\overline{SCK}$]



[When output in synchronization with 9th clock $\overline{SCK}$]



Remark The dotted line indicates READY status.

The acknowledge signal is one-shot pulse to be generated at the falling edge of $\overline{SCK}$ after 8-bit data transfer. It can be positioned anywhere and can be synchronized with any clock $\overline{SCK}$.

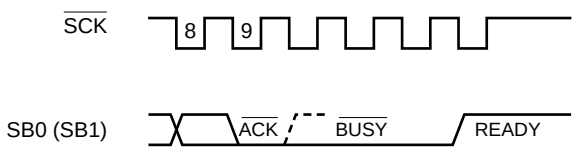
After 8-bit data transmission, the transmitter checks whether the receiver has returned the acknowledge signal. If the acknowledge signal is not returned for the preset period of time after data transmission, it can be judged that data reception has not been carried out correctly.

(f) Busy signal ($\overline{\text{BUSY}}$) and ready signal (READY)

The $\overline{\text{BUSY}}$ signal is intended to report to the master device that the slave device is preparing for data transmission/reception.

The READY signal is intended to report to the master device that the slave device is ready for data transmission/reception.

Figure 16-24. $\overline{\text{BUSY}}$ and READY Signals



In SBI, the slave device notifies the master device of the busy state by setting SB0 (SB1) line to the low level.

The $\overline{\text{BUSY}}$ signal output follows the acknowledge signal output from the master or slave device. It is set/reset at the falling edge of $\overline{\text{SCK}}$. When the $\overline{\text{BUSY}}$ signal is reset, the master device automatically terminates the output of $\overline{\text{SCK}}$ serial clock.

When the $\overline{\text{BUSY}}$ signal is reset and the READY signal is set, the master device can start the next transfer.

(3) Various signals in SBI mode

Figures 16-25 to 16-30 show various signals and each flag operation of serial I/O2SBI registers 0 and 1 in SBI.

Table 16-5 lists various signals in SBI.

Figure 16-25. SIO2RELT, SIO2CMDT, SIO2RELD, and SIO2CMDD Operations (Master)

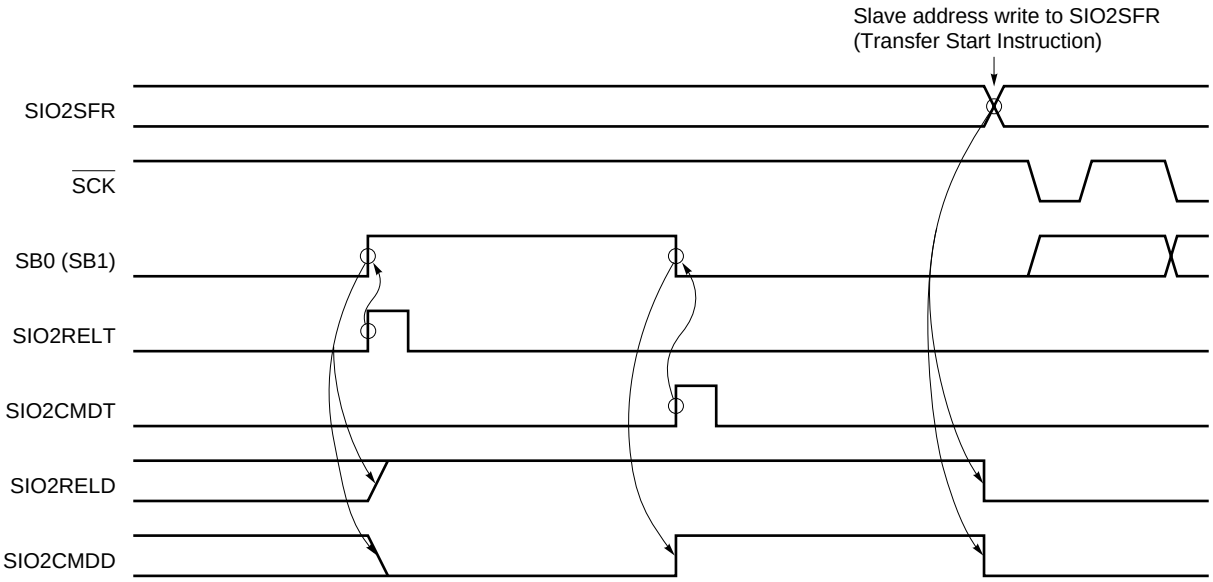


Figure 16-26. SIO2RELT and SIO2CMDD Operations (Slave)

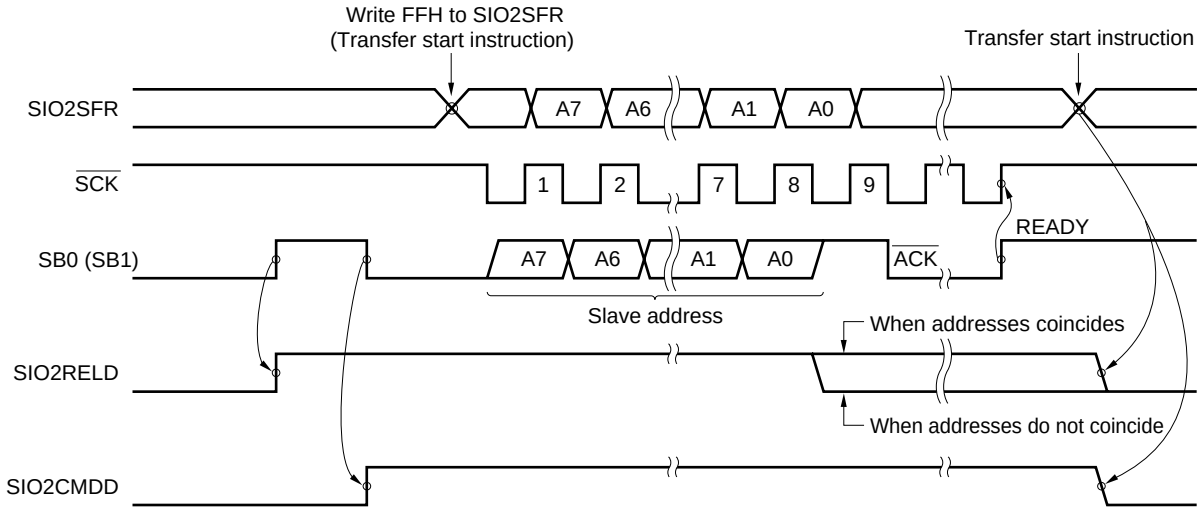
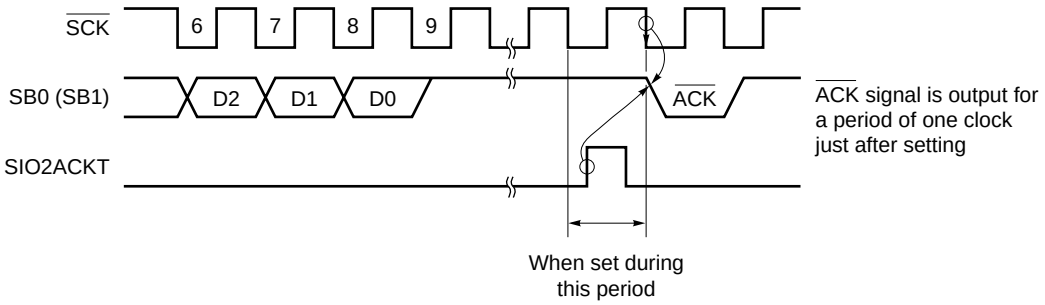


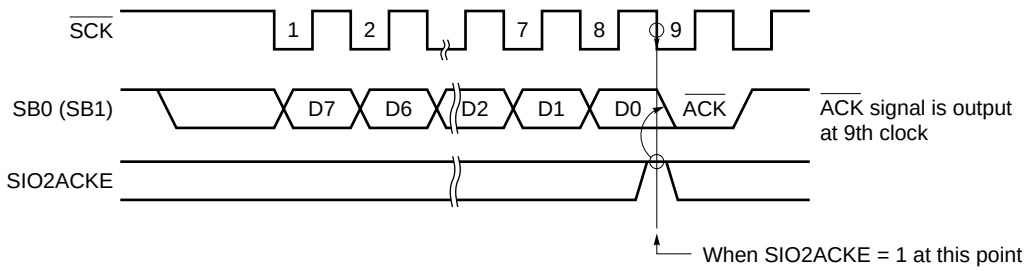
Figure 16-27. SIO2ACKT Operation



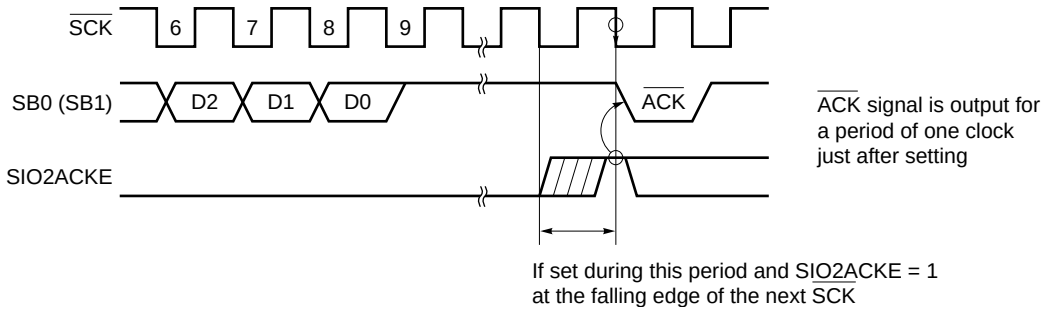
Caution Do not set SIO2ACKT before completion of transfer.

Figure 16-28. SIO2ACKE Operations

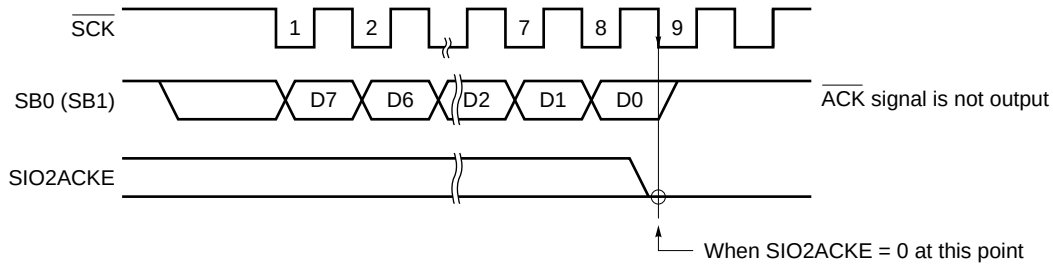
(a) When SIO2ACKE = 1 upon completion of transfer



(b) When set after completion of transfer



(c) When SIO2ACKE = 0 upon completion of transfer



(d) When "SIO2ACKE = 1" period is short

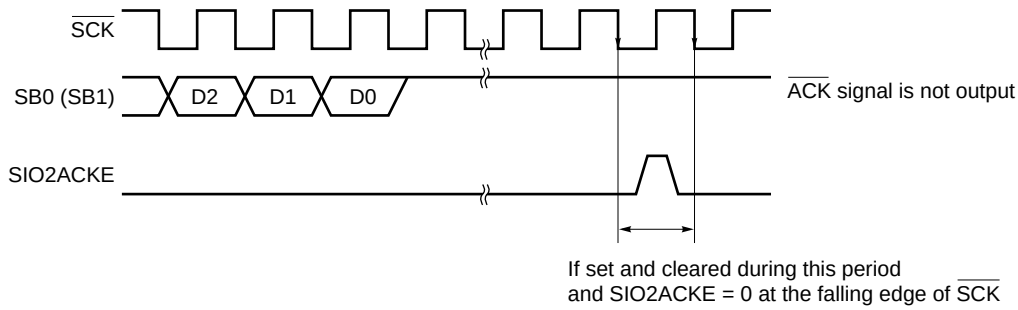
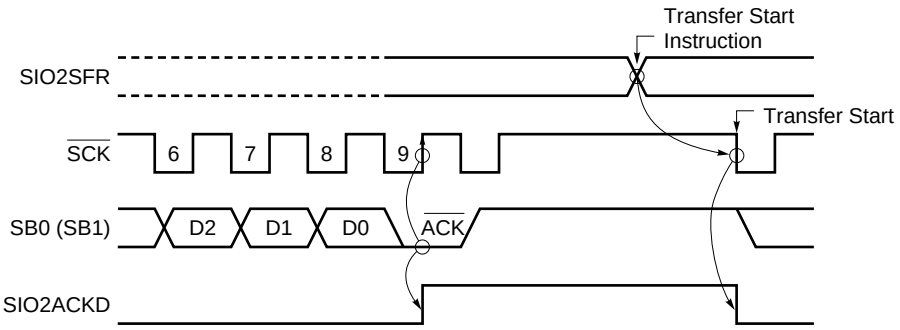
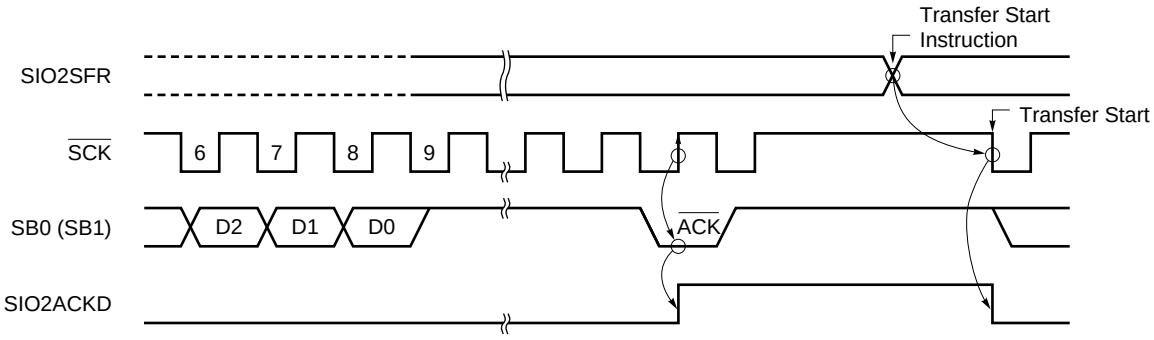


Figure 16-29. SIO2ACKD Operations

(a) When $\overline{ACK}$ signal is output at 9th clock of $\overline{SCK}$



(b) When $\overline{ACK}$ signal is output after 9th clock of $\overline{SCK}$



(c) Clear timing when transfer start is instructed in BUSY

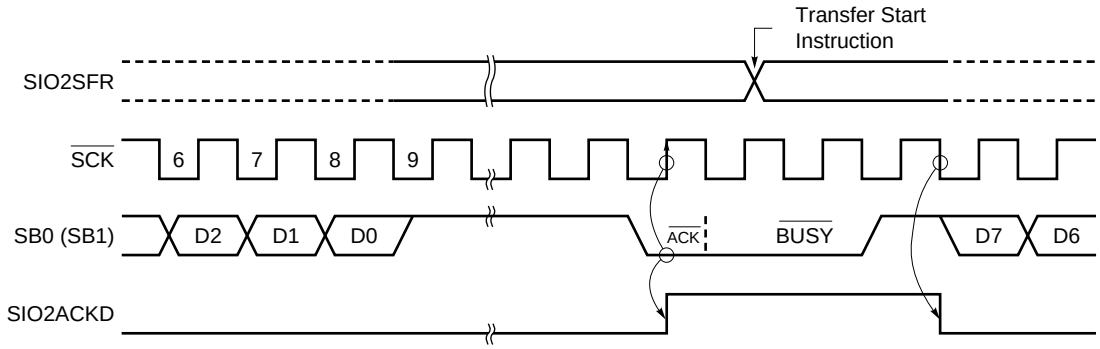


Figure 16-30. SIO2BSYE Operation

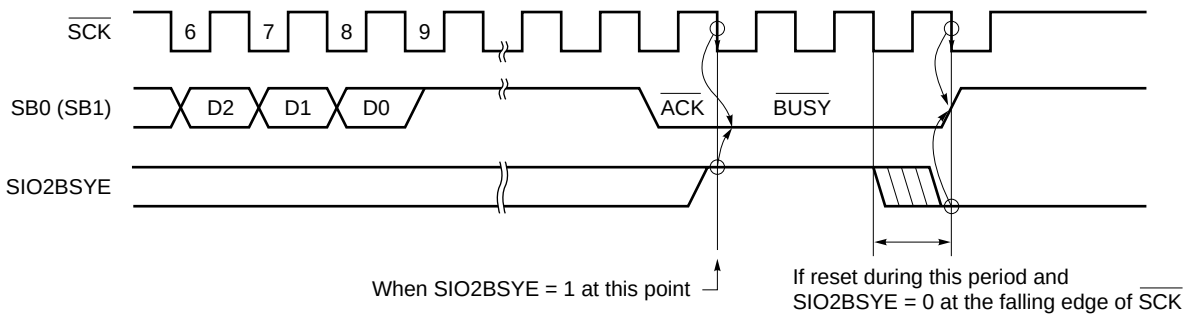


Table 16-5. Various Signals in SBI Mode (1/2)

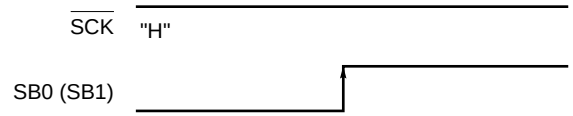
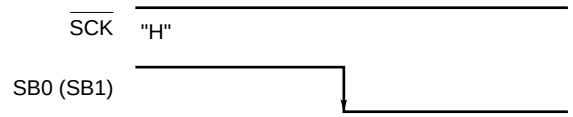
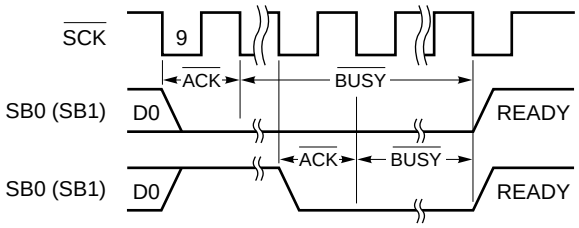
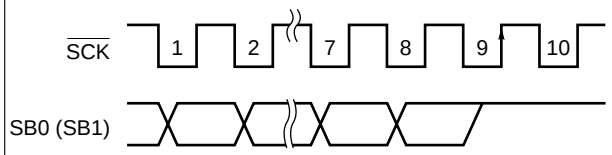
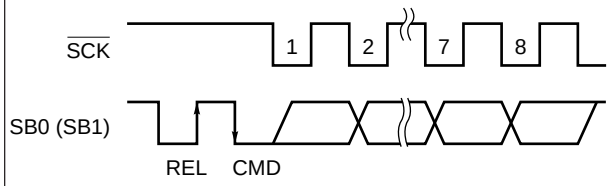
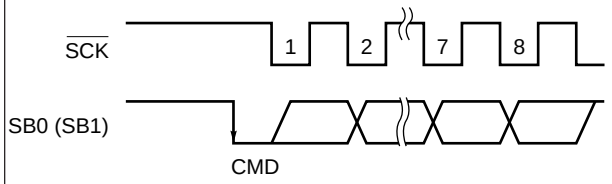
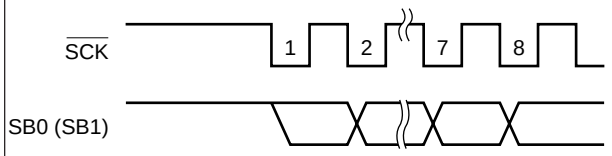
Signal Name	Output Device	Definition	Timing Chart	Output Condition	Effects on Flag	Meaning of Signal
Bus release signal (REL)	Master	SB0 (SB1) rising edge when $\overline{SCK}$ = 1		• SIO2RELT set	• SIO2RELD set • SIO2CMDD clear	CMD signal is output to indicate that transmit data is an address.
Command signal (CMD)	Master	SB0 (SB1) falling edge when $\overline{SCK}$ = 1		• SIO2CMDT set	• SIO2CMDD set	i) Transmit data is an address after REL signal output. ii) REL signal is not output and transmit data is an command.
Acknowledge signal ($\overline{ACK}$)	Master/ slave	Low-level signal to be output to SB0 (SB1) during one-clock period of $\overline{SCK}$ after completion of serial reception	[Synchronous BUSY output]	<1> SIO2ACKE = 1 <2> SIO2ACKT set	• SIO2ACKD set	Completion of reception
Busy signal ($\overline{BUSY}$)	Slave	[Synchronous BUSY signal] Low-level signal to be output to SB0 (SB1) following Acknowledge signal		• SIO2BSYE = 1	—	Serial receive disable because of processing
Ready signal (READY)	Slave	High-level signal to be output to SB0 (SB1) before serial transfer start and after completion of serial transfer		<1> SIO2BSYE = 0 <2> Execution of instruction for data write to SIO2SFR (transfer start instruction) <3> Address signal reception	—	Serial receive enable

Table 16-5. Various Signals in SBI Mode (2/2)

Signal Name	Output Device	Definition	Timing Chart	Output Condition	Effects on Flag	Meaning of Signal
Serial clock ($\overline{SCK}$)	Master	Synchronous clock to output address/command/ data, $\overline{ACK}$ signal, synchro- nous $\overline{BUSY}$ signal, etc. Address/command/data are transferred with the first eight synchronous clocks.		When SIO2CSIE = 1, execution of instruction for data write to SIO2SFR (serial transfer start instruction)Note 2	SIO2SFR set (rising edge of 9th clock of $\overline{SCK}$)Note 1	Timing of signal output to serial data bus
Address (A7 to A0)	Master	8-bit data to be transferred in synchronization with $\overline{SCK}$ after output of REL and CMD signals				Address value of slave device on the serial bus
Commands (C7 to C0)	Master	8-bit data to be transferred in synchronization with $\overline{SCK}$ after output of only CMD signal without REL signal output				Instructions and messages to the slave device
Data (D7 to D0)	Master/ slave	8-bit data to be transferred in synchronization with $\overline{SCK}$ without output of REL and CMD signals				Numeric values to be processed with slave or master device

Notes 1. When SIO2WUP = 0, CSIIF0 is set at the rising edge of the 9th clock of $\overline{SCK}$.
When SIO2WUP = 1, an address is received. Only when the address coincides the serial I/O2 slave address register (SIO2SVA) value, IRQSIO2 is set. (if the address does not coincide with the value of SIO2SVA, SIO2RELD is cleared).

2. In $\overline{BUSY}$ state, transfer starts after the READY state is set.

(4) Pin configuration

The serial clock pin $\overline{SCK}$ and serial data I/O pin SB0 (SB1) have the following configurations.

(a) $\overline{SCK}$ ..... Serial clock input/output pin

<1> Master ... CMOS and push-pull output

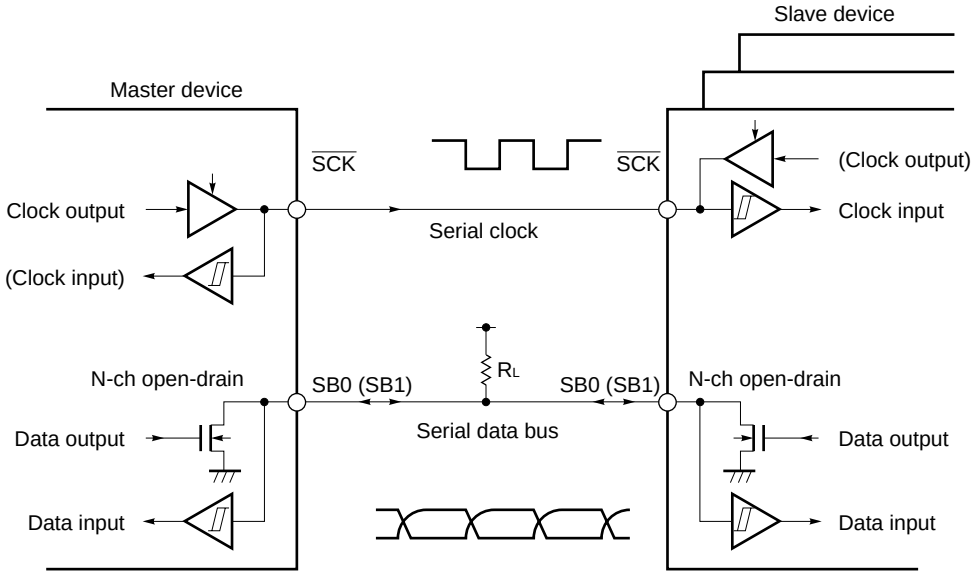
<2> Slave ..... Schmitt input

(b) SB0 (SB1) ..... Serial data input/output dual-function pin

Both master and slave devices have an N-ch open drain output and a Schmitt input.

Because the serial data bus line has an N-ch open-drain output, an external pull-up resistor is necessary.

Figure 16-31. Pin Configuration



Caution Because the N-ch open-drain must be turned off at time of data reception, write FFH to presettable shift register 2 (SIO2SFR) in advance. The N-ch open-drain can be turned off at any time of transfer. However, when the wake-up function specification bit (SIO2WUP) = 1, the N-ch transistor is always turned off. Thus, it is not necessary to write FFH to SIO2SFR.

(5) Address coincidence detection method

In the SBI mode, the master transmits a slave address to select a specific slave device.

Coincidence of the addresses can be automatically detected by hardware. IRQSIO2 is set only when the slave address transmitted by the master coincides with the address set to SIO2SVA when the wake-up function specification bit (SIO2WUP) = 1.

If the SIO2SIC of the serial I/O2 interrupt timing specification register 0 is set, the wake-up function cannot be used even if SIO2WUP is set (an interrupt request signal is generated when bus release is detected). To use the wake-up function, clear SIO2SIC to 0.

Cautions 1. Slave selection/non-selection is detected by the coincidence of the slave address received after bus release (SIO2RELD = 1).
For this coincidence detection, the coincidence detection interrupt (INTCSI0) of the address to be generated with SIO2WUP = 1 is normally used. Thus, execute selection/non-selection detection by slave address when SIO2WUP = 1.

2. When detecting selection/non-selection without the use of interrupt with SIO2WUP = 0, do so by means of transmission/reception of the command preset by program instead of using the address coincidence detection method.

(6) Error detection

In the SBI mode, the serial data bus SB0 (SB1) status being transmitted is fetched into the destination device, that is, the presettable shift register 2 (SIO2SFR). Thus, transmit errors can be detected in the following way.

(a) Method of comparing SIO2SFR data before transmission to that after transmission

In this case, if two data differ from each other, a transmit error is judged to have occurred.

(b) Method of using the serial I/O2 slave address register (SIO2SVA)

Transmit data is set to both SIO2SFR and SIO2SVA and is transmitted. After termination of transmission, SIO2COI flag (coincidence signal coming from the address comparator) of the serial I/O2 operating mode register 0 is tested. If "1", normal transmission is judged to have been carried out. If "0", a transmit error is judged to have occurred.

(7) Communication operation

In the SBI mode, the master device selects normally one slave device as communication target from among two or more devices by outputting an "address" to the serial bus.

After the communication target device has been determined, commands and data are transmitted/received and serial communication is realized between the master and slave devices.

Figures 16-32 to 16-35 show data communication timing charts.

Shift operation of the presettable shift register 2 (SIOSFR) is carried out at the falling edge of serial clock ($\overline{SCK}$). Transmit data is output with MSB set as the first bit from the SB0/P2D0 or SB1/P2D1 pin. Receive data input to the SB0 (or SB1) pin at the rising edge of $\overline{SCK}$ is latched into the SIO2SFR.

Figure 16-32. Address Transmission from Master Device to Slave Device (SIO2WUP = 1)

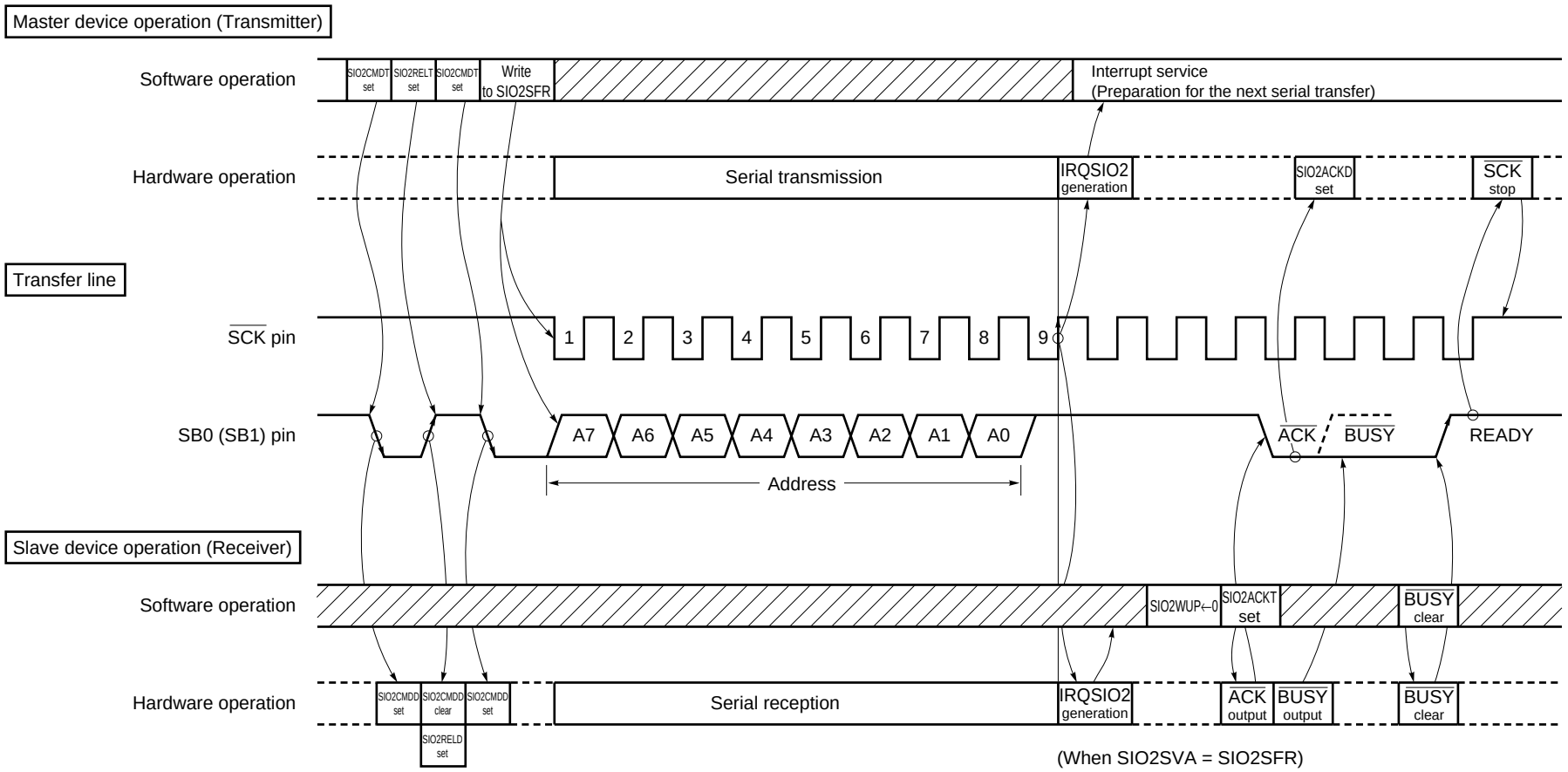


Figure 16-33. Command Transmission from Master Device to Slave Device

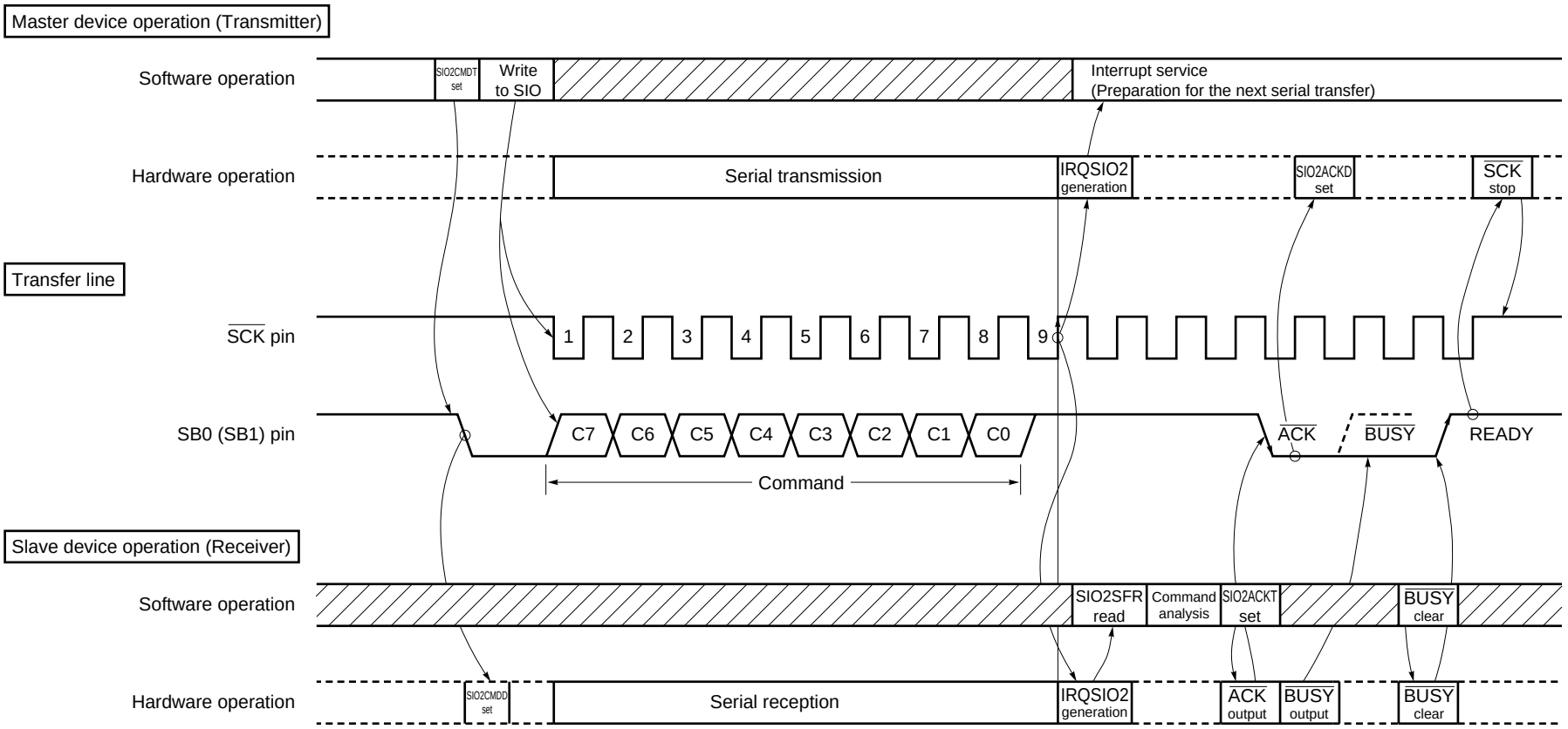


Figure 16-34. Data Transmission from Master Device to Slave Device

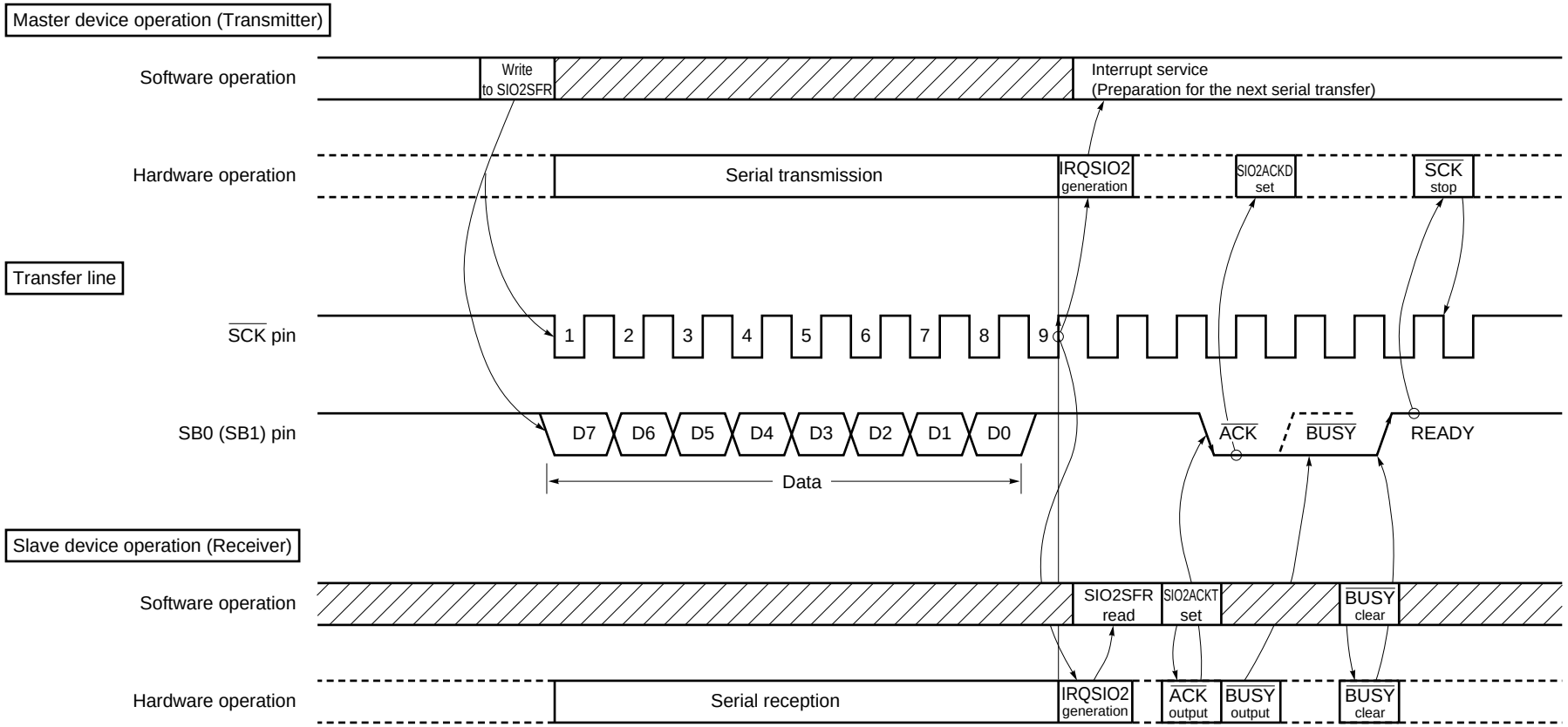
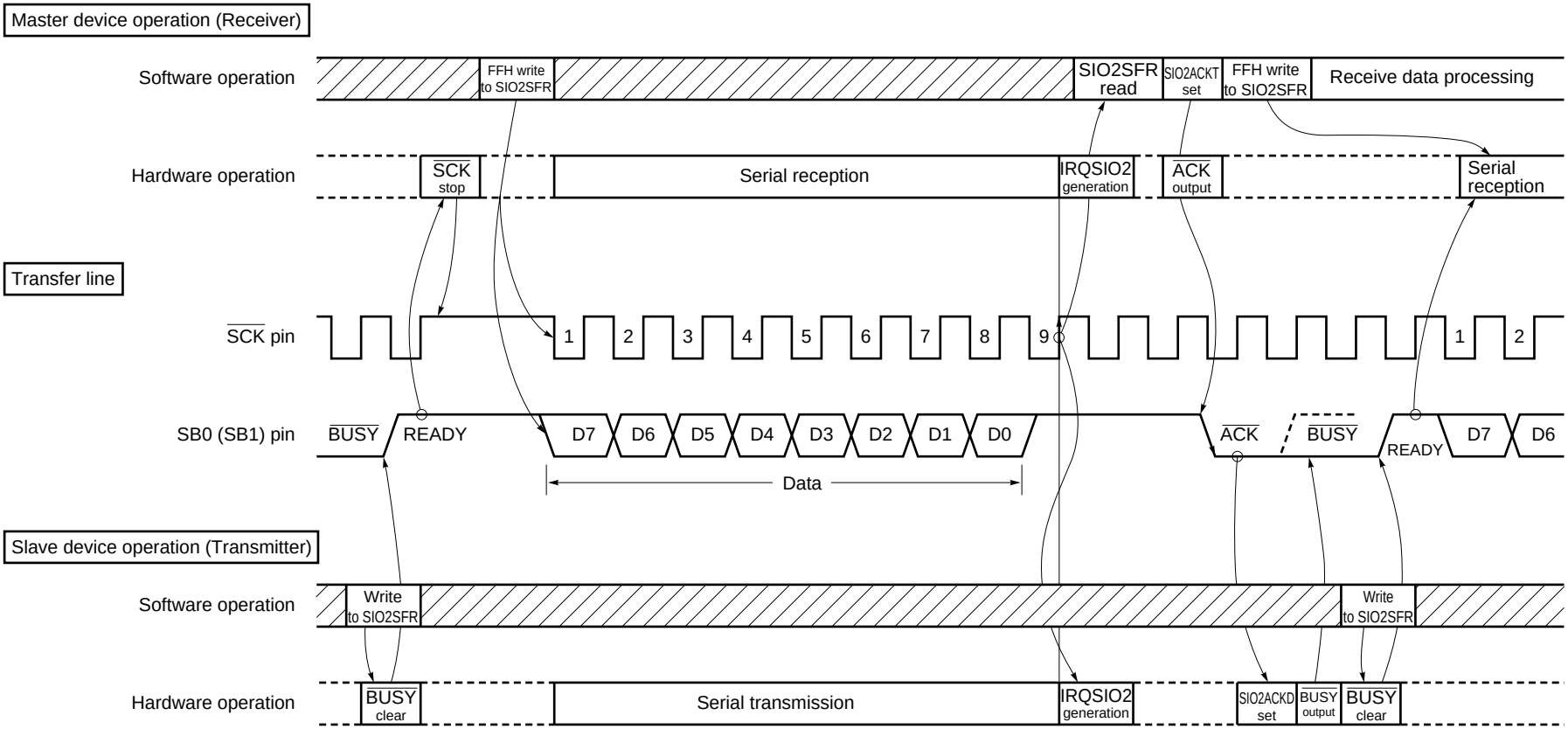


Figure 16-35. Data Transmission from Slave Device to Master Device



(8) Transfer start

Serial transfer is started by setting transfer data to the presettable shift register 2 (SIO2SFR) when the following two conditions are satisfied.

- Serial interface 2 operation control flag (SIO2CSIE) = 1
- Internal serial clock is stopped or $\overline{SCK}$ is at high level after 8-bit serial transfer.

Cautions 1. If SIO2CSIE is set to "1" after data write to SIO2SFR, transfer does not start.

2. Because the N-ch transistor must be turned off for data reception, write FFH to SIO0 in advance.
However, when the make-up function control flag (SIO2WUP) = 1, the N-ch transistor is always turned off. Thus, it is not necessary to write FFH to SIO2SFR.

3. If data is written to SIO2SFR when the slave is busy, the data is not lost.
When the busy state is cleared and SB0 (or SB1) input is set to the high level (READY) state, transfer starts.

Upon termination of 8-bit transfer, serial transfer automatically stops and the interrupt request flag (IRQSIO2) is set.

(9) SBI mode precautions

(a) Slave selection/non-selection is detected by coincidence detection of the slave address received after bus release (SIO2RELD = 1).
For this coincidence detection, match interrupt (IRQSIO2) of the address to be generated with SIO2WUP = 1 is normally used. Thus, execute selection/non-selection detection by slave address when SIO2WUP = 1.

(b) When detecting selection/non-selection without the use of interrupt with SIO2WUP = 0, do so by means of transmission/reception of the command preset by program instead of using the address coincidence detection method.

(c) If SIO2WUP is set to 1 during $\overline{\text{BUSY}}$ signal output, $\overline{\text{BUSY}}$ is not cleared. In SBI, the $\overline{\text{BUSY}}$ signal continues to be output after $\overline{\text{BUSY}}$ clear instruction generation to the falling edge of the next serial clock ($\overline{\text{SCK}}$). Before setting SIO2WUP to 1, be sure to clear $\overline{\text{BUSY}}$ and then check that the SB0 (SB1) has become high-level.

(d) For pins which are to be used for data input/output, be sure to carry out the following settings before serial transfer of the 1st byte after $\overline{\text{RESET}}$ input.

<1> Set the P2D0 and P2D1 latches to 1.
<2> Set the SIO2RELT flag of serial I/O2SBI register 1 to 1.
<3> Reset the P2D0 and P2D1 output latches from 1 to 0.

(e) When device is in the master mode, follow the procedure below to judge whether slave device is in the busy state or not.

<1> Detect acknowledge signal ($\overline{\text{ACK}}$) or interrupt request signal generation.
<2> Set the port 2D bit I/O selection register P2DBIO0 (or P2DBIO1) of the SB0/P2D0 (or SB1/P2D1) pin into the input mode.
<3> Read out the pin state (when the pin level is high, the READY state is set).

After the detection of the READY state, set the P2DBIO0 (or P2DBIO1) to 1 and return to the output mode.

16.2.9 2-wire serial I/O mode

The 2-wire serial I/O mode is validated when SIO2MD1 and 2 of the serial I/O2 operation mode register are set to 1 and SIO2WAT1 of the serial interface 2 interrupt timing specification register is cleared to 0.

(1) Outline of 2-wire serial I/O mode

In the 2-wire serial I/O mode, the SCL and SDA pins are used for communication.

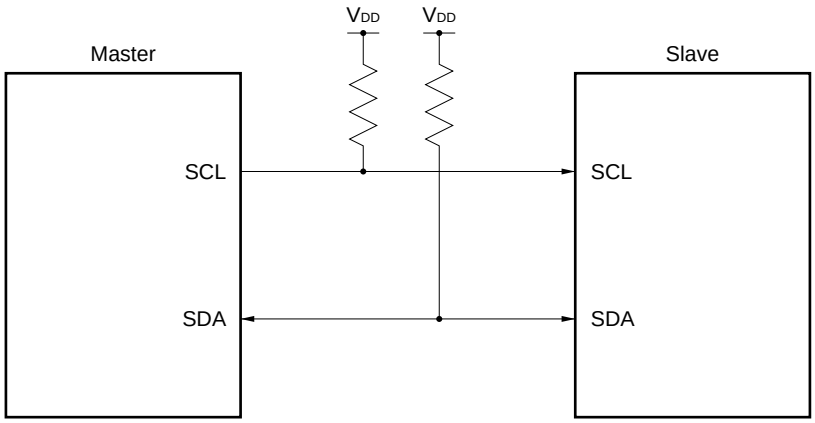
Table 16-6 shows the outline of the 2-wire serial I/O mode.

Table 16-6. Outline of 2-Wire Serial I/O Mode

Pin used for communication	• SCL pin (serial clock I/O pin) • SDA pin (serial data I/O pin)	
Transmission/reception operation	Transmit data	Sequentially output from MSB of shift register to data I/O pin in synchronization with falling of SCL pin
	Receive data	Value of data I/O pin is sequentially input from LSB of shift register in synchronization with rising of SCL pin.
Transmission/reception start	Master	Transmission/reception is started by setting transfer data to shift register after 2-wire serial I/O master mode has been set.
	Slave	Waits for clock from master with SCL pin going into high-impedance state after 2-wire serial I/O slave mode has been set.
Interrupt	Issues interrupt request IRQSIO2 at rising edge of 8th count of clock.	
Clock pin	Master	Stops output of SCL pin at rising edge of 8th count and retains high level until next transmission/reception operation is started
	Slave	Goes into high-impedance state in 2-wire serial I/O slave mode.

Caution The SIO2CMDT and SIO2RELT flags of the serial I/O2SBI register 1 are disabled from being used when the 2-wire serial I/O mode is used.

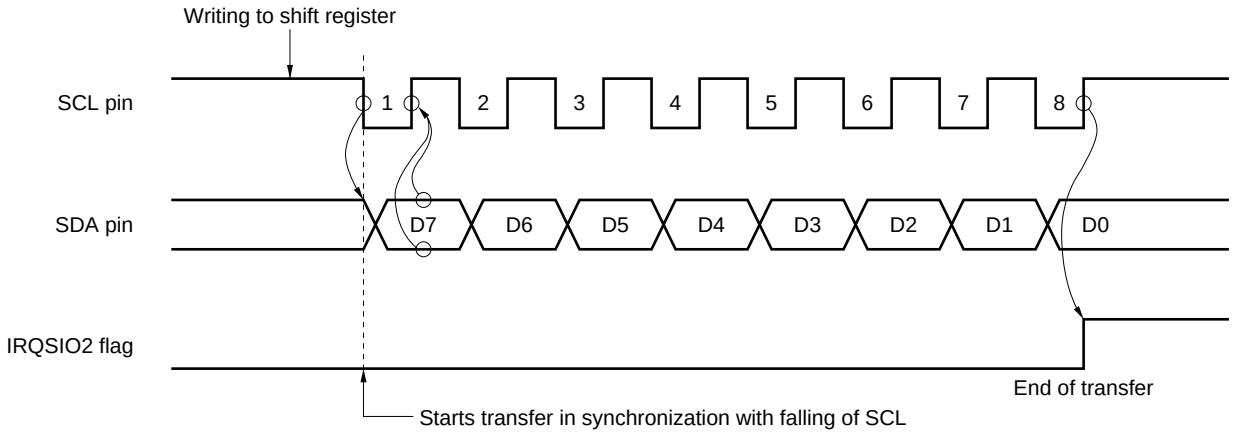
Figure 16-36. Serial Bus Configuration Example in 2-Wire Serial I/O Mode



(2) Timing chart

Figure 16-37 shows the timing chart in the 2-wire serial I/O mode.

Figure 16-37. Timing Chart in 2-Wire Serial I/O Mode



The SDA pin is an N-ch open-drain I/O pin and must be externally pulled up. Because the N-ch transistor must be turned off when data is received, write FFH to SIO2SFR in advance.

(3) Starting transfer

Serial transfer is started by setting data to the presettable shift register 2 (SIO2SFR) when the following two conditions are satisfied.

- Control flag of operation of serial interface 2 (SIO2CSIE) = 1
- When internal serial clock is stopped or SCL is low after 8-bit serial transfer

Serial transfer is automatically stopped and the interrupt request flag (IRQSIO2) is set after completion of 8-bit transfer.

(4) Detection of error

Because the status of the serial bus SDA during transmission is also input to SIO2SFR of the device that is transmitting data in the 2-wire serial I/O mode, a transmission error, if any, can be detected as follows:

(a) By comparing SIO2SFR data before and after transmission

In this case, it is assumed that a transmission error has occurred if the two data differ.

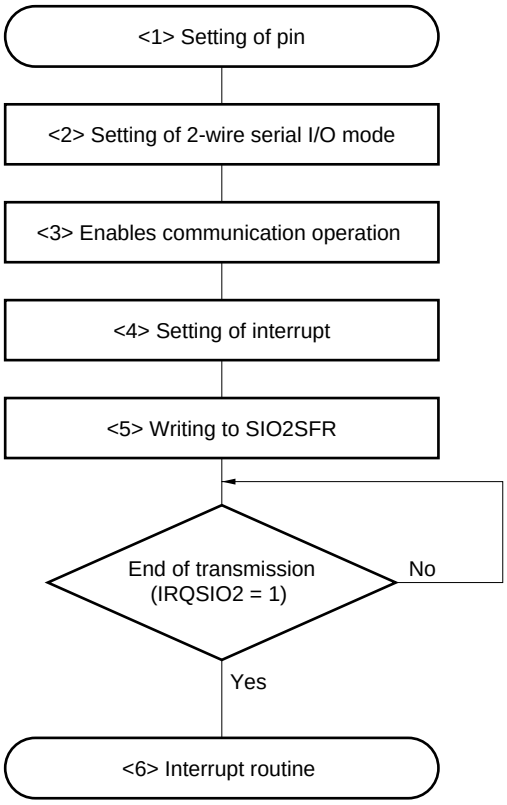
(b) By using serial interface 2 slave address register (SIO2SVA)

The transmit data is set to SIO2SFR and SIO2SVA and transmission is executed. After completion of transmission, the SIO2COI flag of the serial I/O2 operation mode register 0 (coincidence signal from the address comparator) is tested. If the flag is “1”, communication has been executed normally; if it is “0”, it is assumed that a transmission error has occurred.

(5) Program flowchart in 2-wire serial I/O mode

A program flowchart example in the 2-wire serial I/O transmission mode is shown below.

Figure 16-38. Example Flowchart in 2-Wire Serial I/O Transmission Mode



Remark To execute a 2-wire serial I/O operation with the same setting as before, start from step <5>.

<1> Setting of pin

(a) Setting of data pin in 2-wire serial I/O mode

Set the I/O control mode of the data pin to “1” (output), and the port latch of the data pin to “0”.

(b) Setting of shift clock in 2-wire serial I/O Mode

Set the I/O control mode of the shift clock to “1” (output), and the port latch of the shift clock to “1”.

<2> Setting 2-wire serial I/O transmission mode as communication mode

SIO2MD2 = 1, SIO2MD1 = 1

<3> Enabling communication operation (SIO2CSIE = “1”)

(a) To output internal clock from shift clock (SIO2MD0 = “1”)

Output the internal clock.

(b) To input external clock as shift clock (SIO2MD0 = “0”)

Input the external clock.

<4> Setting of interrupt

Execute the “EI” instruction and set the IPSIO2 flag to “1”.

<5> Setting of transmit data to SIO2SFR (PUT SIO2SFR)

The 2-wire serial I/O transmission operation is started as soon as data has been set, and the 8-bit transmit data is output from the SDA pin.

<6> Interrupt routine

When the 2-wire serial I/O transmission operation has been completed, the interrupt request flag IRQSIO2 is issued. When the interrupt is accepted, execution branches to the vector address.

Cautions 1. Transfer is not started even if SIO2CSIE is set to “1” after data has been written to SIO2SFR.

2. Write FFH to SIO2SFR in advance because the N-ch transistor must be turned off during data reception.

16.2.10 I²C bus mode

The I²C bus mode becomes valid when SIO2MD1 and 2 of the serial I/O2 operation mode register 1 are set to 1 and SIO2WAT1 of the serial I/O2 interrupt timing specification register 1 is set to 1.

- In the I²C transmission mode, clear the SIO2BSYE flag to "0".
- In the I²C reception mode, set the SIO2BSYE flag to "1".

The functions that can be used in the I2C bus mode of the uPD17717, 17718, and 17719 are listed below.

Table 16-7. Functions in I²C Bus Mode of μPD17717, 17718, and 17719

Operation Mode	Supported by serial interface 2
Multi-master	Software-supported
Single master Basic transmission/reception Acknowledge control Wait control	Hardware-supported
Slave Wait request Wake-up function	Hardware-supported

(1) Outline of I²C bus mode

In the I²C bus mode, communication is performed by using the SCL and SDA pins.

Table 16-8 shows the outline of the I²C bus mode.

Table 16-8. Outline of I²C Bus Mode

Pins used for transmission	• SCL pin (serial clock I/O pin) • SDA pin (serial data I/O pin)	
Transmission/reception operation	Transmit data	Sequentially output from MSB of shift register to data I/O pin in synchronization with falling of SCL pin.
	Receive data	Value of data I/O pin is input from LSB of shift register in synchronization with rising of SCL pin.
Transmission/reception start	Master	Transmission/reception is started by setting transfer data to shift register after I²C master mode has been set.
	Slave	Waits for clock from master with SCL pin going into high-impedance state after I²C slave mode has been set.
Interrupt	Issues interrupt request IRQSIO2 at rising of clock of 8th count.	
Clock pin	Master	9th count and those that follow are used for acknowledge.
	Slave	Goes into high-impedance state.

16.2.11 I²C bus mode operation

The I²C bus mode is provided for when communication operations are performed between a single master device and multiple slave devices. This mode configures a serial bus that includes only a single master device, and is based on the clocked serial I/O format with the addition of bus configuration functions, which allows the master device to communicate with a number of (slave) devices using only two lines: SCL and SDA. Consequently, when the user plans to configure a serial bus which includes multiple microcontrollers and peripheral devices, using this configuration results in reduction of the required number of port pins and on-board wires.

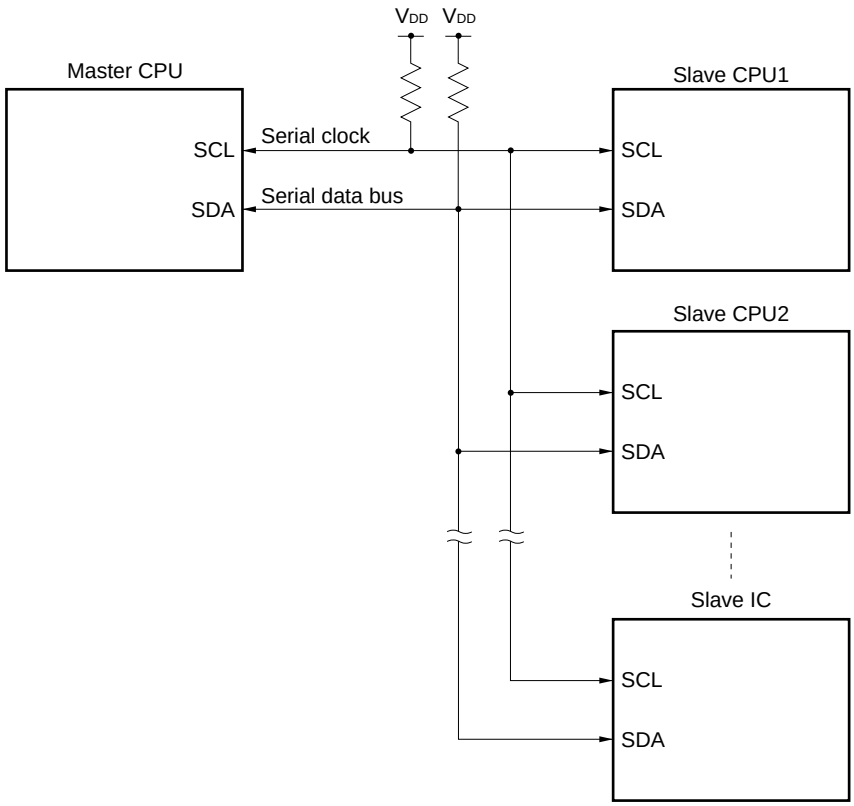
In the I²C bus specification, the master sends start condition, data, and stop condition signals to slave devices through the serial data bus, while slave devices automatically detect and distinguish the type of signals due to the signal detection function incorporated as hardware. This simplifies I²C bus control sections in the application program.

An example of a serial bus configuration is shown in Figure 16-39. This system below is composed of CPUs and peripheral ICs having serial interface hardware that complies with the I²C bus specification.

Note that pull-up resistors are required to connect to both serial clock line and serial data bus line, because open-drain buffers are used for the serial clock pin (SCL) and the serial data I/O pin (SDA) on the I²C bus.

The signals used in the I²C bus mode are described in Table 16-9.

Figure 16-39. Example of Serial Bus Configuration Using I²C Bus



(1) I^2C bus mode functions

In the I^2C bus mode, the following functions are available.

(a) Automatic identification of serial data

Slave devices automatically detect and identifies start condition, data, and stop condition signals sent in series through the serial data bus.

(b) Chip selection by specifying device addresses

The master device can select a specific slave device connected to the I^2C bus and communicate with it by sending in advance the address data corresponding to the destination device.

(c) Wake-up function

When address data is sent from the master device, slave devices compare it with the value registered in their serial I/O2 slave address registers (SIO2SVA). If the values in one of the slave devices coincide, the slave device generates an interrupt signal (the interrupt also occurs when the stop condition is detected). Therefore, CPUs other than the selected slave device on the I^2C bus can perform independent operations during the serial communication.

(d) Acknowledge signal ($\overline{\text{ACK}}$) control function

The master device and a slave device send and receive acknowledge signals to confirm that the serial communication has been executed normally.

(e) Wait signal ($\overline{\text{WAIT}}$) control function

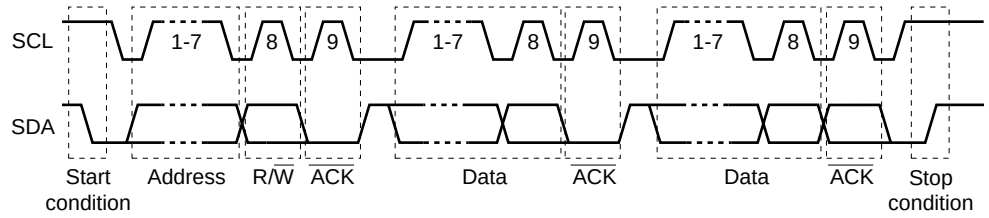
The slave device outputs a wait signal on the bus to inform the master device of the wait status.

(2) I^2C bus definition

This section describes the format of serial data communications and functions of the signals used in the I^2C bus mode.

First, the transfer timings of the start condition, data, and stop condition signals, which are output onto the signal data bus of the I^2C bus, are shown in Figure 16-40.

Figure 16-40. I^2C Bus Serial Data Transfer Timing



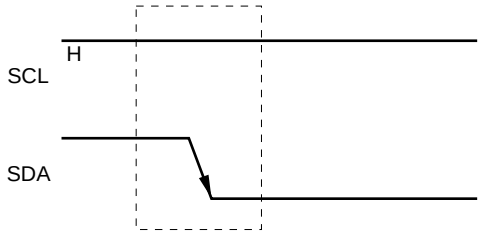
The start condition, slave address, and stop condition signals are output by the master. The acknowledge signal ($\overline{\text{ACK}}$) is output by either the master or the slave device (normally by the device which has received the 8-bit data that was sent). A serial clock (SCL) is continuously supplied from the master device.

(a) Start condition

When the SDA pin level is changed from high to low while the SCL pin is high, this transition is recognized as the start condition signal. This start condition signal, which is created using the SCL and SDA pins, is output from the master device to slave devices to initiate a serial transfer. Refer to **16.2.12 Cautions on using I²C bus mode** for details of the start condition output.

The start condition signal is detected by hardware incorporated in slave devices.

Figure 16-41. Start Condition

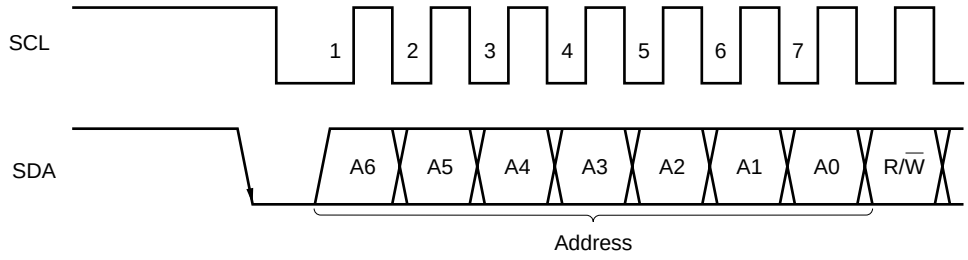


(b) Address

The 7 bits following the start condition signal are defined as an address.

The 7-bit address data is output by the master device to specify a specific slave from among those connected to the bus line. Each slave device on the bus line must therefore have a different address. Therefore, after a slave device detects the start condition, it compares the 7-bit address data received and the data of the serial I/O2 slave address register (SIO2SVA). After the comparison, only the slave device in which the data are a match becomes the communication partner, and subsequently performs communication with the master device until the master device sends a start condition or stop condition signal.

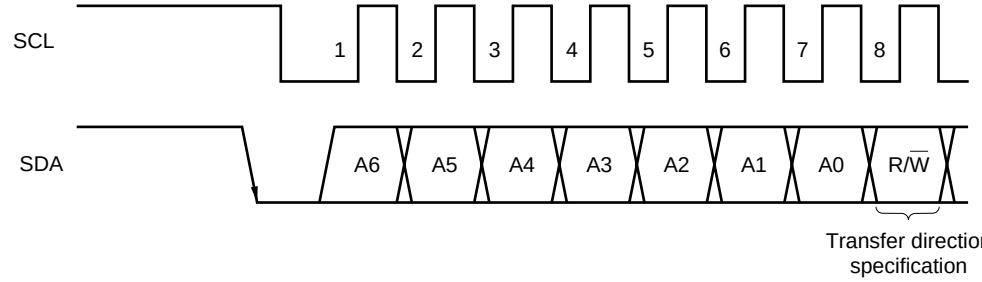
Figure 16-42. Address



(c) Transfer direction specification

The 1 bit that follows the 7-bit address data will be sent from the master device, and it is defined as the transfer direction specification bit. If this bit is 0, it is the master device which will send data to the slave. If it is 1, it is the slave device which will send data to the master.

Figure 16-43. Transfer Direction Specification

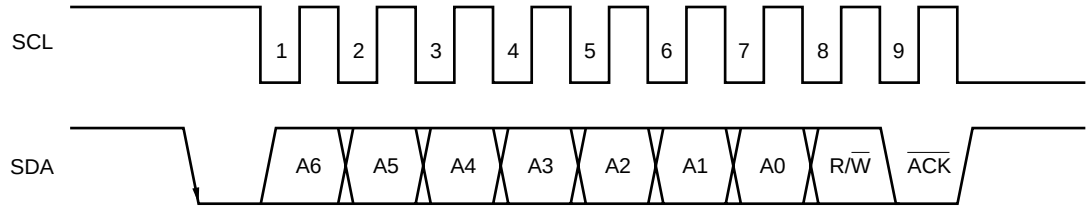


(d) Acknowledge signal ($\overline{ACK}$)

The acknowledge signal indicates that the transferred serial data has definitely been received. This signal is used between the sending side and receiving side devices for confirmation of correct data transfer. In principle, the receiving side device returns an acknowledge signal to the sending device each time it receives 8-bit data. The only exception is when the receiving side is the master device and the 8-bit data is the last transfer data; the master device outputs no acknowledge signal in this case.

The sending side that has tranferred 8-bit data waits for the acknowledge signal which will be sent from the receiving side. If the sending side device receives the acknowledge signal, which means a successful data transfer, it proceeds to the next processing. If this signal is not sent back from the slave device, this means that the data sent has not been received by the slave device, and therefore the master device outputs a stop condition signal to terminate subsequent transmissions.

Figure 16-44. Acknowledge Signal



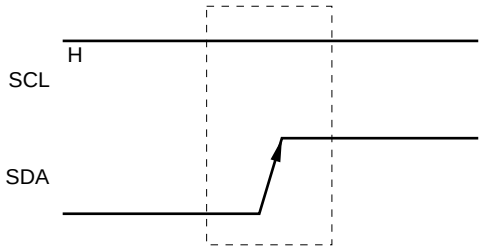
(e) Stop condition

If the SDA pin level changes from low to high while the SCL pin is high, this transition is defined as a stop condition signal.

The stop condition signal is output from the master to the slave device to terminate a serial transfer.

The stop condition signal is detected by hardware incorporated in the slave device.

Figure 16-45. Stop Condition



(f) Wait signal ($\overline{\text{WAIT}}$)

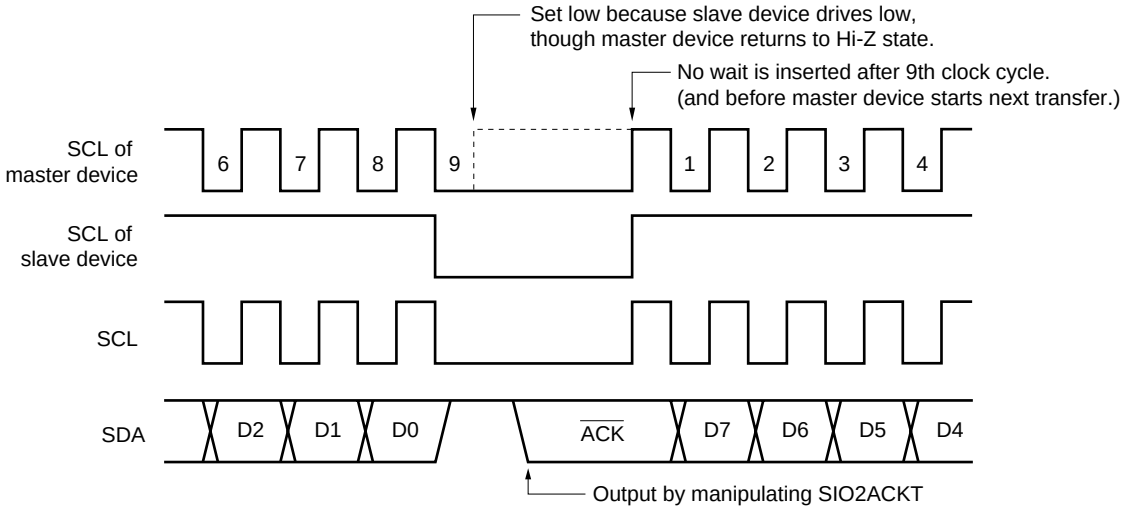
The wait signal is output by a slave device to inform the master device that the slave device is in wait state due to preparing for transmitting or receiving data.

During the wait state, the slave device continues to output the wait signal by keeping the SCL pin low to delay subsequent transfers. When the wait state is released, the master device can start the next transfer.

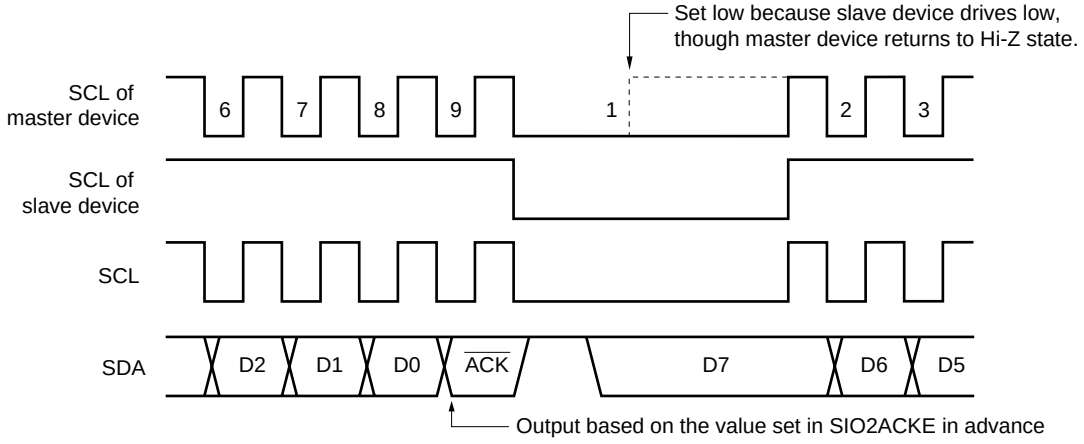
For the releasing operation of slave devices, refer to **16.2.12 Cautions on using I^2C bus mode**.

Figure 16-46. Wait Signal

(a) Wait of 8 Clock Cycles



(b) Wait of 9 Clock Cycles



(3) Various signals in I²C bus mode

A list of signals in the I²C bus mode is given in Table 16-9.

Table 16-9. Signals in I²C Bus Mode

Signal Name	Output Device	Definition	Output Condition	Affected Flag(s)	Signal Function
Start condition	Master	SDA falling edge when SCL is high[Note 1]	SIO2CMDT is set.	SIO2CMDD is set.	Indicates that sequent transmission data are address data and serial communication starts.
Stop condition	Master	SDA rising edge when SCL is high[Note 1]	SIO2RELT is set.	SIO2RELD is set. SIO2CMDD is cleared.	Indicates end of serial transmission.
Acknowledge signal ($\overline{ACK}$)	Master or slave	Low-level signal of SDA output during one SCL clock cycle after serial reception	• SIO2ACKE = 1 • SIO2ACKT is set.	SIO2ACKD is set.	Indicates completion of reception of 1 byte.
Wait ($\overline{WAIT}$)	Slave	Low-level signal output to SCL	SIO2WAT1, SIO2WAT0 = 1X.	–	Indicates state in which serial reception is not possible.
Serial clock (SCL)	Master	Synchronization clock for output of various signals	Execution of instruction for data write to SIO2SFR when SIO2CSIE = 1 (serial transfer start instruction).[Note 2]	IRQSIO2 is set.[Note 3]	Serial communication synchronization signal.
Address (A6 to A0)	Master	7-bit data output in synchronization with SCL after start condition output			Indicates address value for specification of slave on serial bus.
Transfer direction (R/$\overline{W}$)	Master	1-bit data output in synchronization with SCL after address output			Indicates whether data transmission or reception is to be performed.
Data (D7 to D0)	Master or slave	8-bit data output in synchronization with SCL, not immediately after start condition output			Indicates data actually to be sent.

Notes 1. The level of the serial clock can be controlled by SIO2CLC of serial I/O2 interrupt timing specification register 1.

2. In the wait state, the serial transfer operation will be started after the wait state is released.

3. If the 8-clock wait is selected when SIO2WUP = 0, IRQSIO2 is set at the rising edge of the 8th clock cycle of SCL. If the 9-clock wait is selected when SIO2WUP = 0, IRQSIO2 is set at the rising edge of the 9th clock cycle of SCL.

IRQSIO2 is set if an address is received and that address coincides with the value of the serial I/O2 slave address register (SIO2SVA) when SIO2WUP = 1, or if the stop condition is detected.

(4) Pin configurations

The configurations of the serial clock pin SCL and the serial data I/O pins SDA are shown below.

(a) SCL

Pin for serial clock input/output dual-function pin.

<1> Master ... N-ch open-drain output

<2> Slave ..... Schmitt input

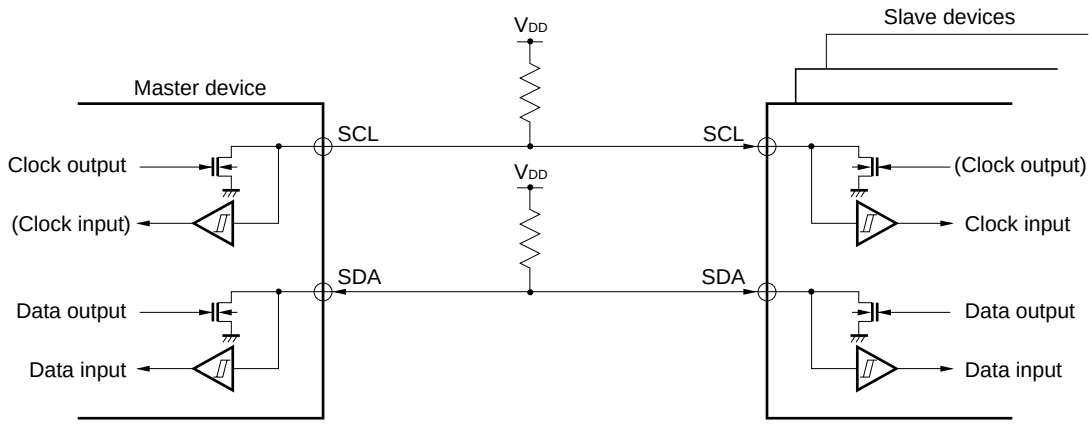
(b) SDA

Serial data input/output dual-function pin.

Uses N-ch open-drain output and Schmitt-input buffers for both master and slave devices.

Note that pull-up resistors are required to connect to both serial clock line and serial data bus line, because open-drain buffers are used for the serial clock pin (SCL) and the serial data bus pin (SDA0 or SDA1) on the I^2C bus.

Figure 16-47. Pin Configuration



Caution To receive data, the N-ch open-drain output must be in high-impedance state. Therefore, set the SIO2BSYE flag of serial I/O2SBI register 0 to 1 in advance, and write FFH to the presettable shift register 2 (SIO2SFR).

When the wake-up function is used (by setting the SIO2WUP flag of the serial I/O2 operating mode register 1, however, do not write FFH to SIO2SFR before reception. Even if FFH is not written to SIO2SFR, the N-ch open-drain output is always in high-impedance state.

(5) Address coincidence detection method

In the I^2C mode, the master can select a specific slave device by sending slave address data.

IRQSIO2 is set if the slave address transmitted by the master coincides with the value set to the serial I/O2 slave address register (SIO2SVA) when a slave device address has a serial I/O2 slave address register (SIO2SVA), and the SIO2WUP flag is 1 (IRQSIO2 is also set when the stop condition is detected).

When using the wake-up function, set SIO2SIC to 1.

Caution Slave selection/non-selection is detected by the coincidence of the data (address) received after the start condition.

For this coincidence detection, the coincidence detection interrupt (IRQSIO2) of the address to be generated with SIO2WUP = 1 is normally used. Thus, execute selection/ non-selection detection by slave address when SIO2WUP = 1.

(6) Error detection

In the I^2C bus mode, transmission error detection can be performed by the following methods because the serial data bus SDA status during transmission is also taken into the presettable shift register 2 (SIO2SFR) of the transmitting device.

(a) Comparison of SIO2SFR data before and after transmission

In this case, a transmission error is judged to have occurred if the two data values are different.

(b) Using the serial I/O2 slave address register (SIO2SVA)

Transmit data is set in SIO2SFR and SIO2SVA before transmission is performed. After transmission, the SIO2COI bit (coincidence signal from the address comparator) of serial I/O2 operation mode register 0 is tested: "1" indicates normal transmission, and "0" indicates a transmission error.

(7) Communication operation

In the I^2C bus mode, the master selects the slave device to be communicated with from among multiple devices by outputting address data onto the serial bus.

After the slave address data, the master sends the R/W bit which indicates the data transfer direction, and starts serial communication with the selected slave device.

Data communication timing charts are shown in **Figures 16-48** and **16-49**.

In the transmitting device, the presettable shift register 2 (SIO2SFR) shifts transmission data to the SO latch in synchronization with the falling edge of the serial clock (SCL), the SO0 latch outputs the data on an MSB-first basis from the SDA pin to the receiving device.

In the receiving device, the data input from the SDA pin is taken into the SIO2SFR in synchronization with the rising edge of SCL.

Figure 16-48. Data Transmission from Master to Slave (Both Master and Slave Selected 9-Clock Wait) (1 of 3)

(a) Start Condition to Address

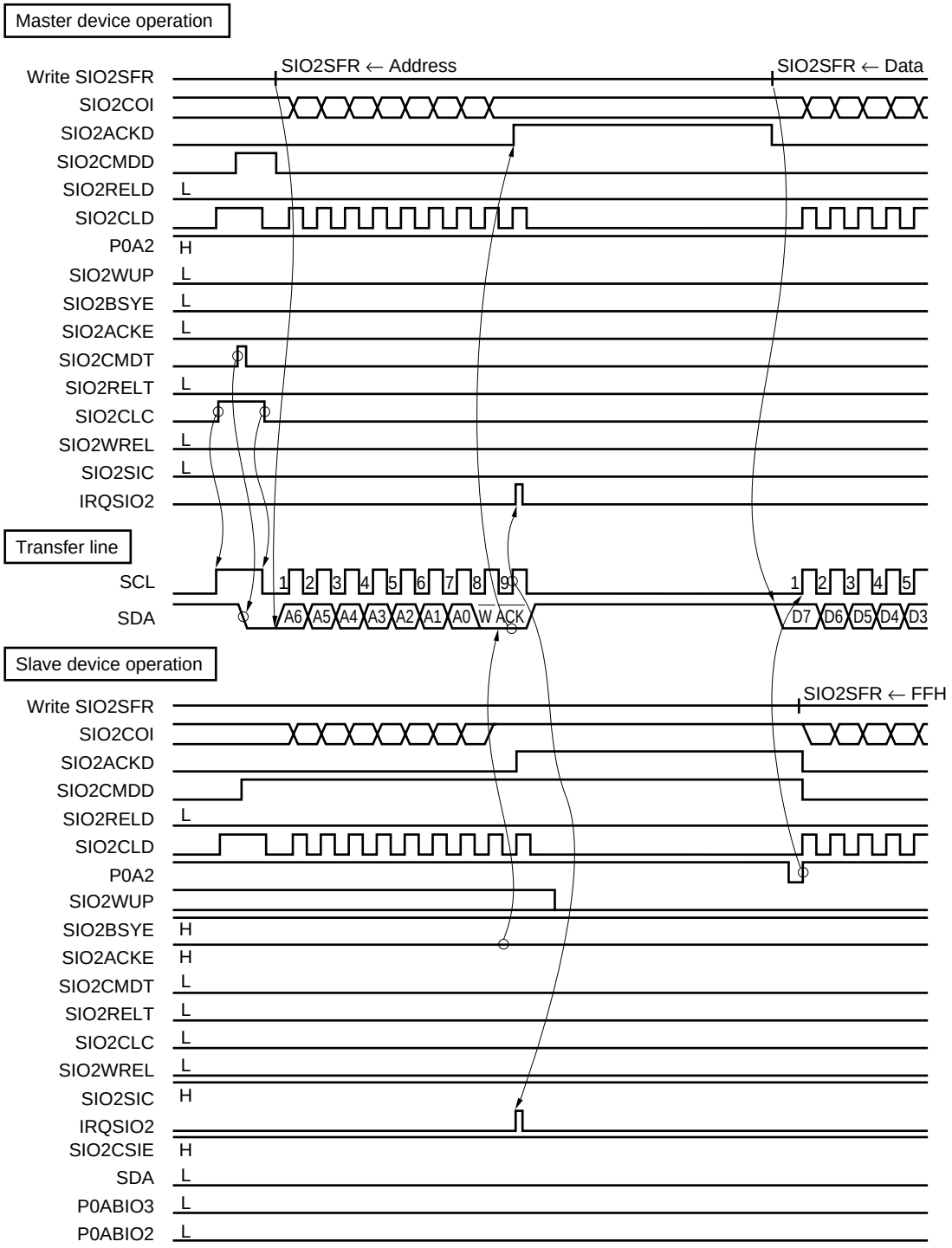


Figure 16-48. Data Transmission from Master to Slave (Both Master and Slave Selected 9-Clock Wait) (2 of 3)

(b) Data

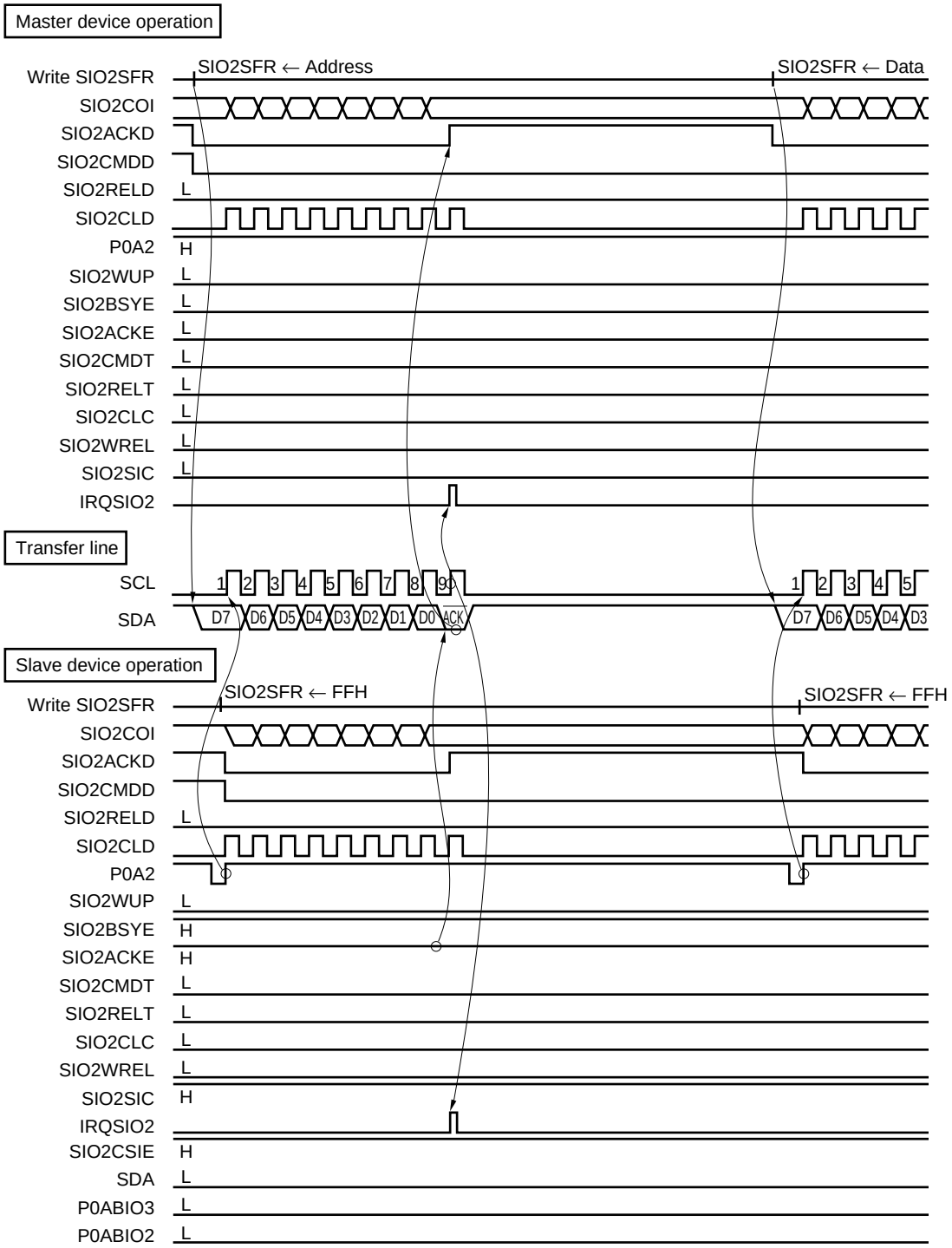


Figure 16-48. Data Transmission from Master to Slave (Both Master and Slave Selected 9-Clock Wait) (3 of 3)

(c) Stop Condition

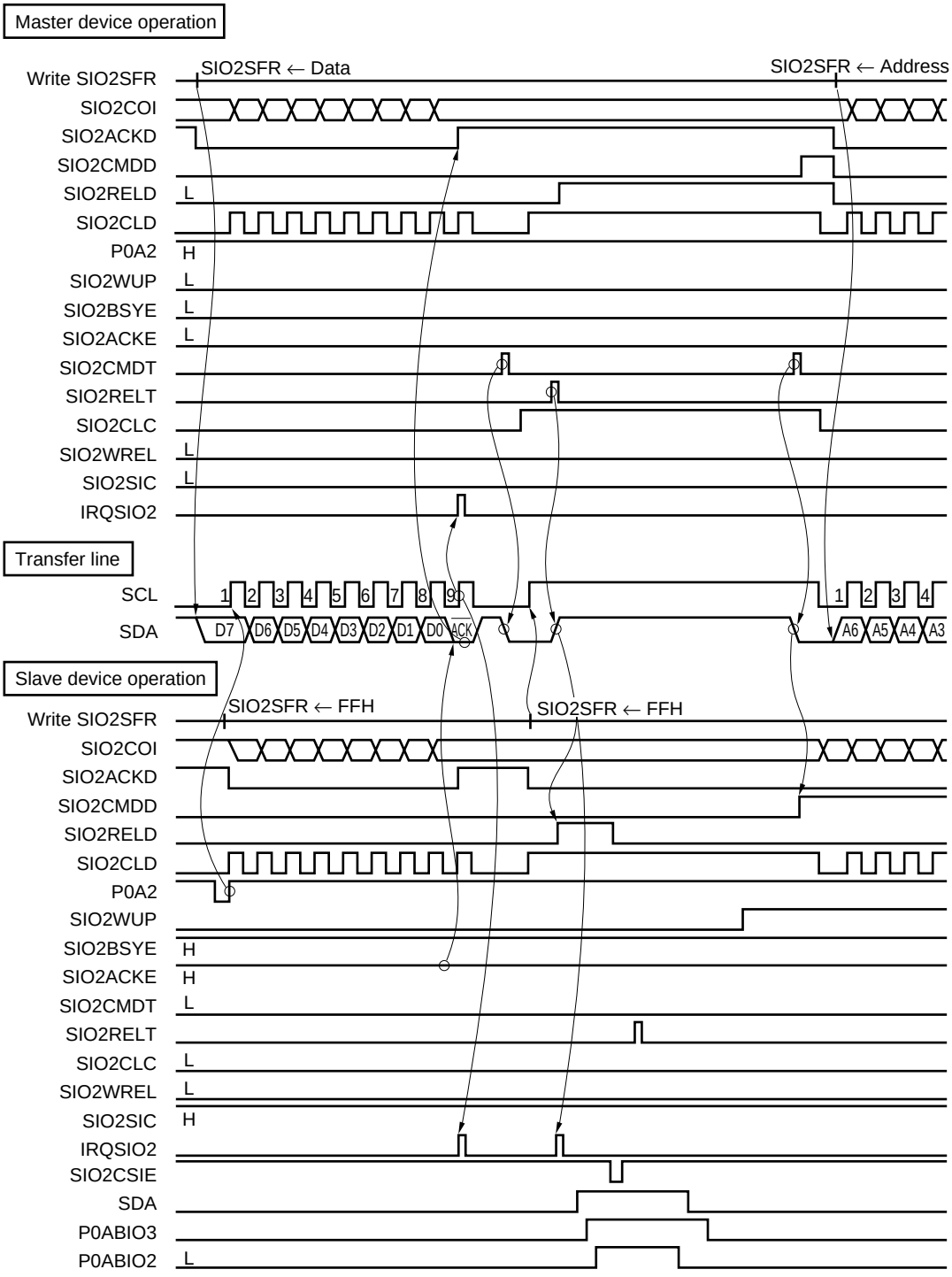


Figure 16-49. Data Transmission from Slave to Master (Both Master and Slave Selected 9-Clock Wait) (1 of 3)

(a) Start Condition to Address

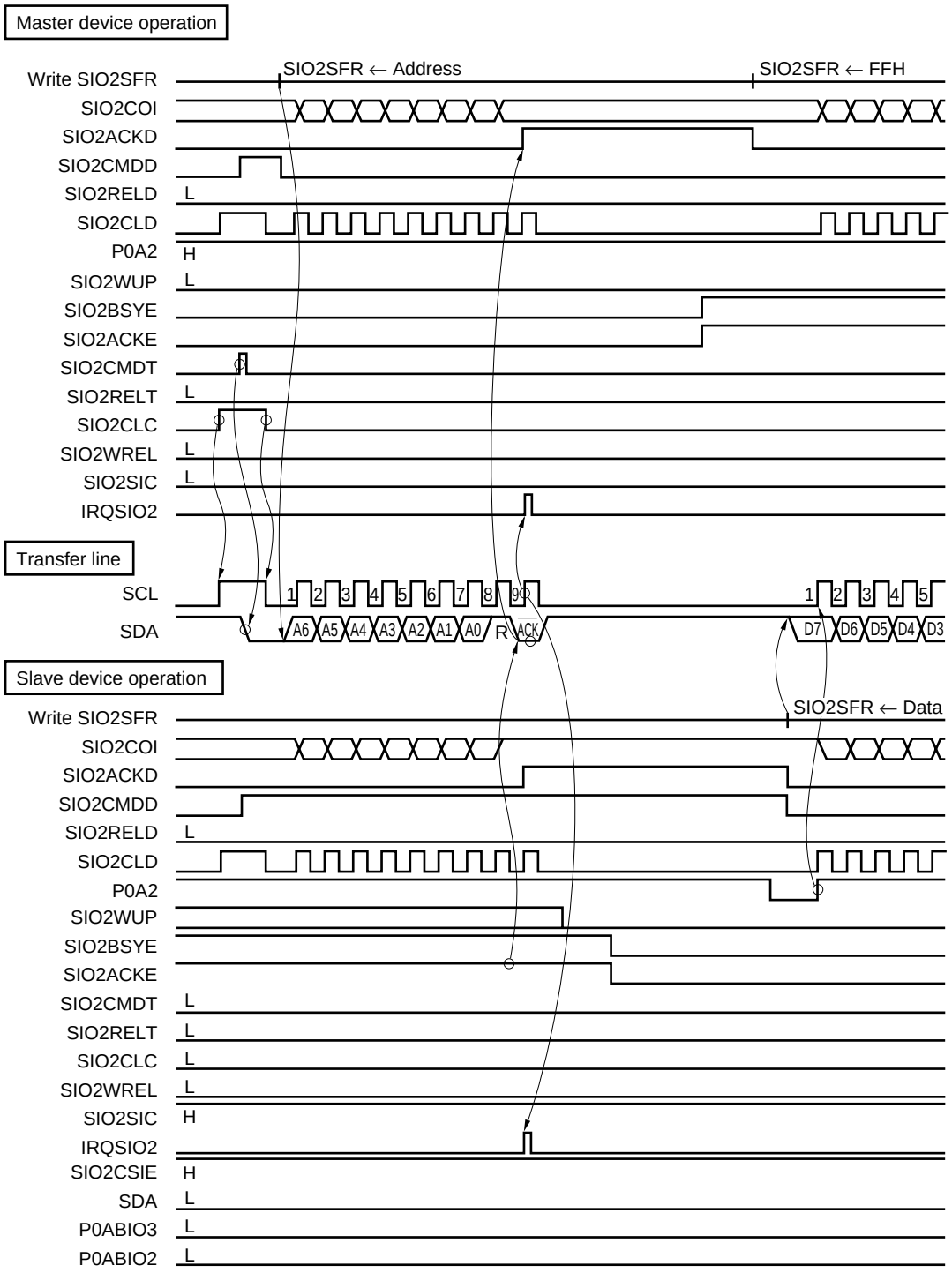


Figure 16-49. Data Transmission from Slave to Master (Both Master and Slave Selected 9-Clock Wait) (2 of 3)

(b) Data

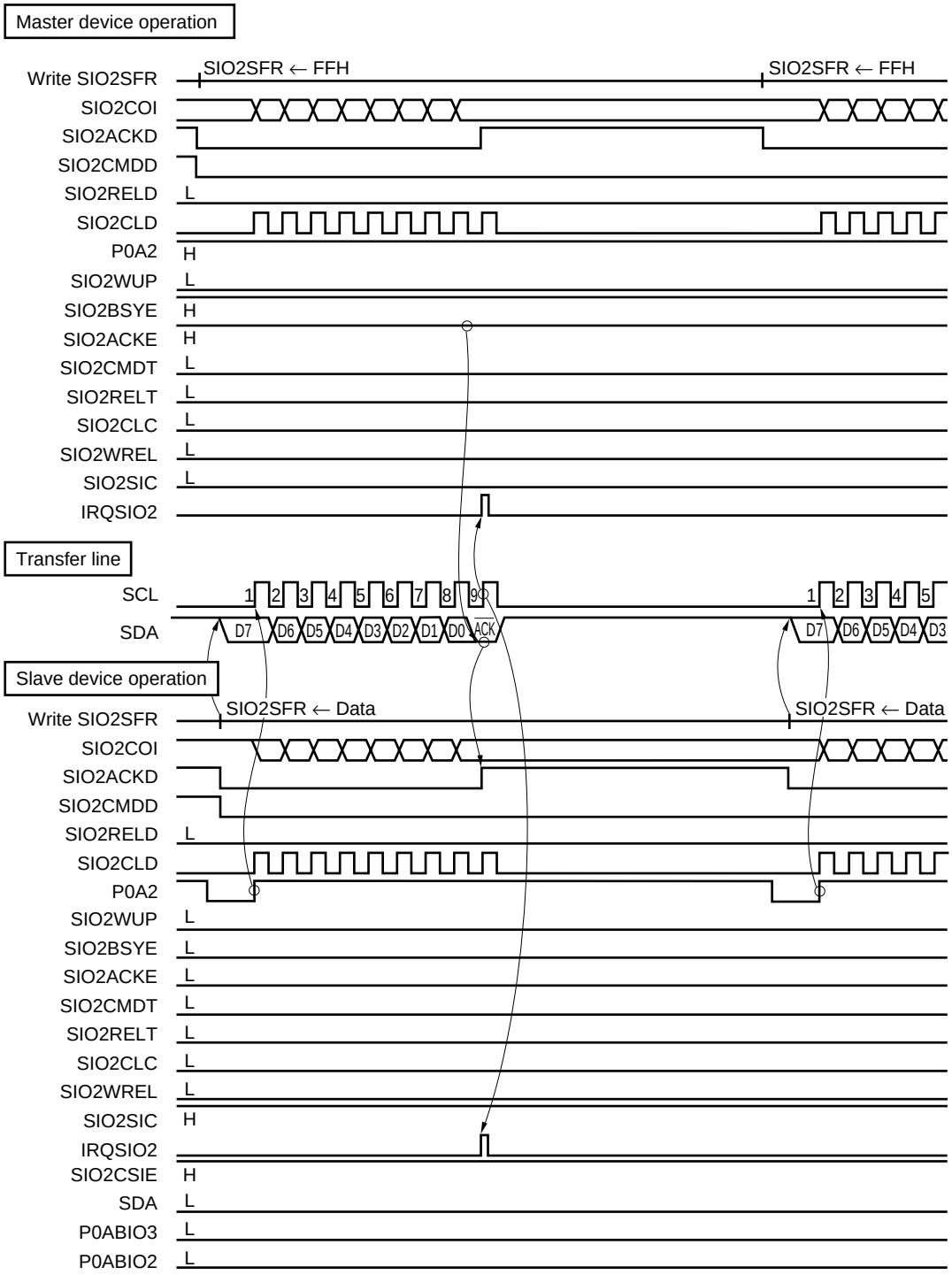
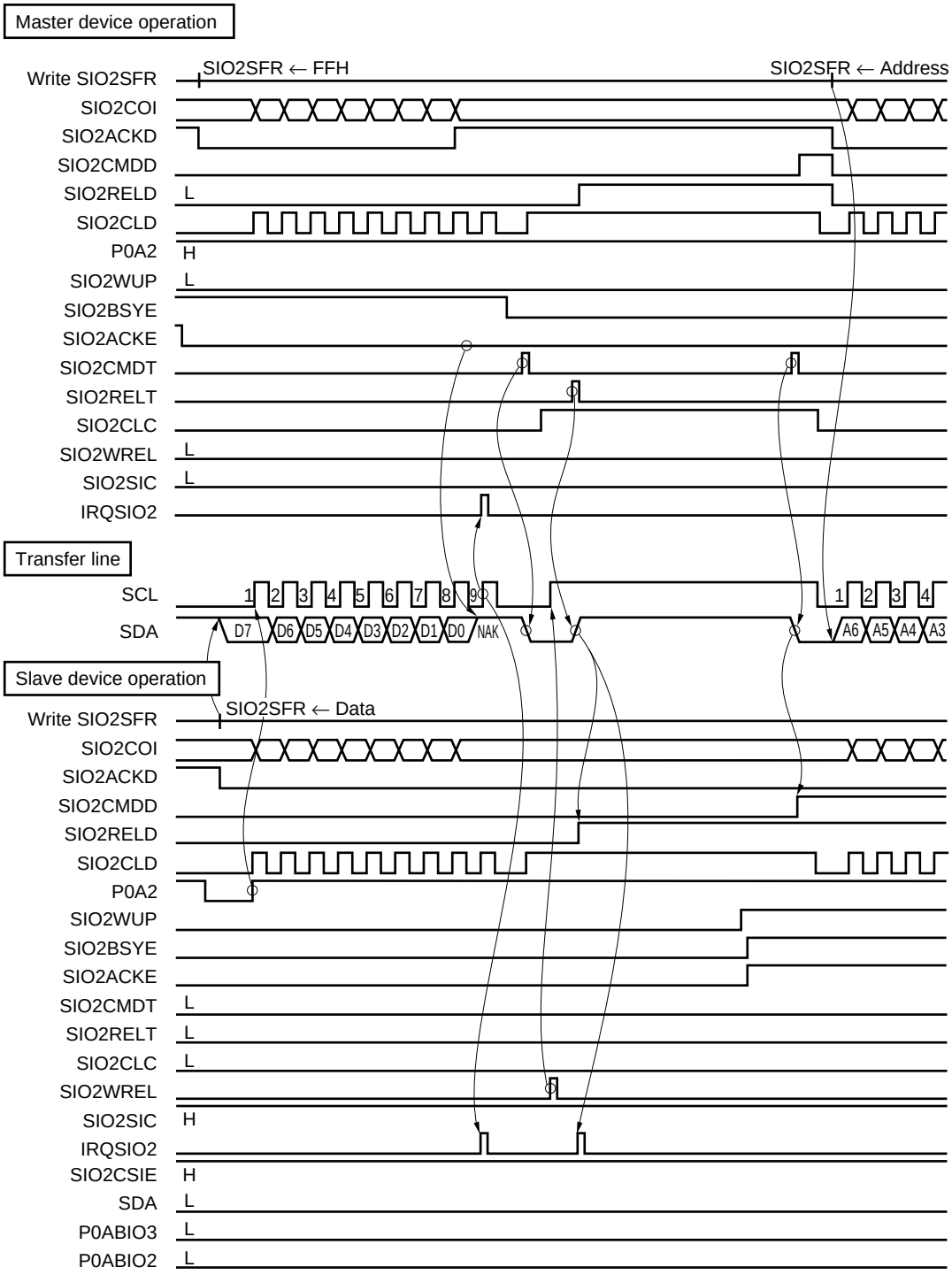


Figure 16-49. Data Transmission from Slave to Master (Both Master and Slave Selected 9-Clock Wait) (3 of 3)

(c) Stop Condition



(8) Start of transfer

A serial transfer is started by setting transfer data in the presettable shift register 2 (SIO2SFR) if the following two conditions have been satisfied:

- The serial interface 2 operation control flag (SIO2CSIE) = 1.
- After an 8-bit serial transfer, the internal serial clock is stopped or SCL is low.

Cautions 1. Setting SIO2CSIE to 1 after writing data in SIO2SFR does not initiate transfer operation.

2. Because the N-ch open-drain output must go into high-impedance during data reception, set the SIO2BSYE flag of serial I/O2SBI register 0 to 1 before writing FFH to SIO2SFR. Do not write FFH to SIO2SFR before reception when the wake-up function is used (by setting the SIO2WUP flag of serial I/O2 operation mode register 1). Even if FFH is not written to SIO2SFR, the N-ch open-drain output is always high-impedance state.

3. If data is written to SIO2SFR while the slave is in the wait state, that data is held. The transfer is started when SCL is output after the wait state is released.

When an 8-bit data transfer ends, serial transfer is stopped automatically and the interrupt request flag (IRQSIO2) is set.

16.2.12 Cautions on using I^2C bus mode

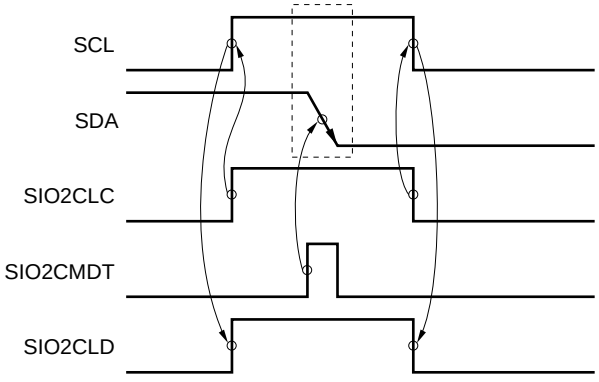
(1) Start condition output (master)

The SCL pin normally outputs a low-level signal when no serial clock is output. It is necessary to change the SCL pin to high in order to output a start condition signal. Set 1 in SIO2CLC of serial I/O2 interrupt timing specify register 1 to drive the SCL pin high.

After setting SIO2CLC, clear SIO2CLC to 0 and return the SCL pin to low. If SIO2CLC remains 1, no serial clock is output.

If it is the master device which outputs the start condition and stop condition signals, confirm that SIO2CLD is set to 1 after setting SIO2CLC to 1; a slave device may have set SCL to low (wait state).

Figure 16-50. Start Condition Output



(2) Slave wait release (slave transmission)

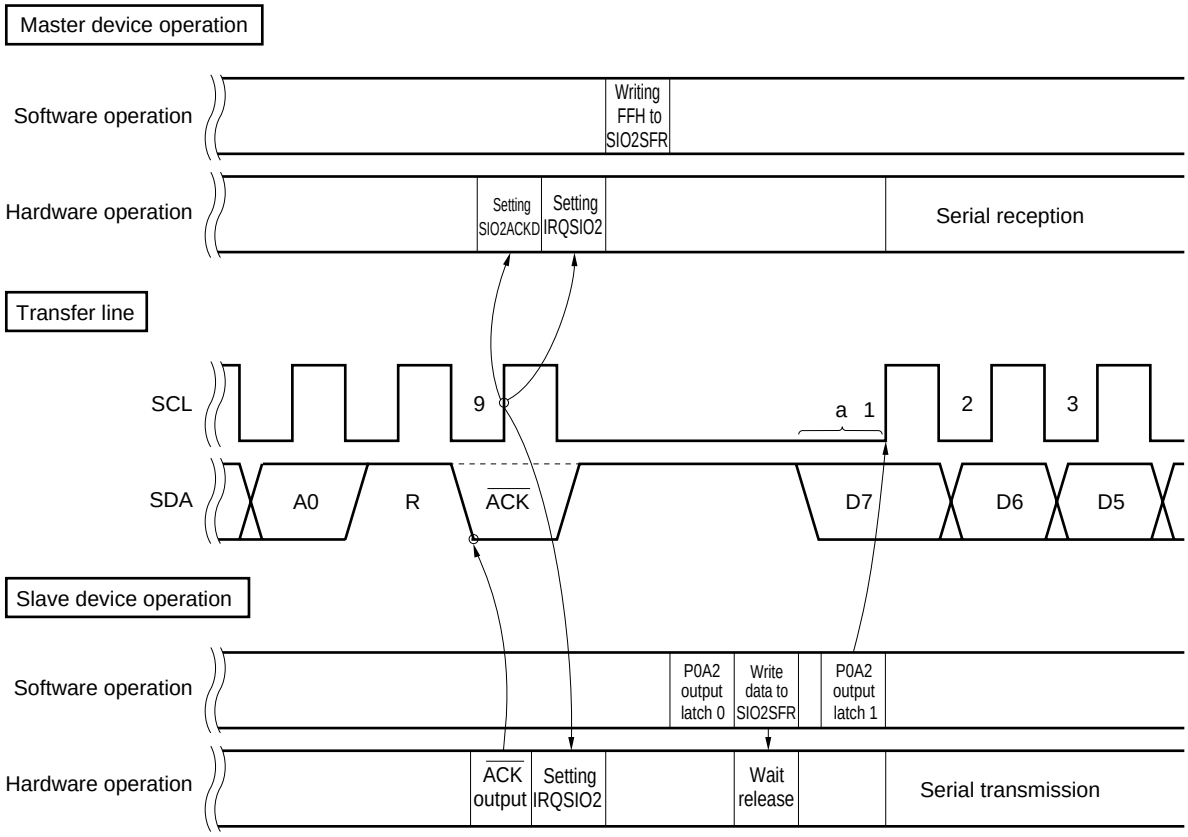
Slave wait status is released by SIO2WREL flag (bit 2 of serial I/O2 interrupt timing specification register 1) setting or execution of a presettable shift register 2 (SIO2SFR) write instruction.

If the slave sends data, the wait is immediately released by execution of an SIO2SFR write instruction and the clock rises without the start transmission bit being output in the data line. Therefore, as shown in Figure 16-51, data should be transmitted by manipulating the P0A2 output latch through the program. At this time, control the low-level width ("**a**" in **Figure 16-51**) of the first serial clock at the timing used for setting the P0A2 output latch to 1 after execution of an SIO2SFR write instruction.

In addition, if the acknowledge signal from the master is not output (if data transmission from the slave is completed), set 1 in the SIO2WREL flag and release the wait.

For these timings, refer to **Figure 16-49**.

Figure 16-51. Slave Wait Release (Transmission)



(3) Slave wait release (slave reception)

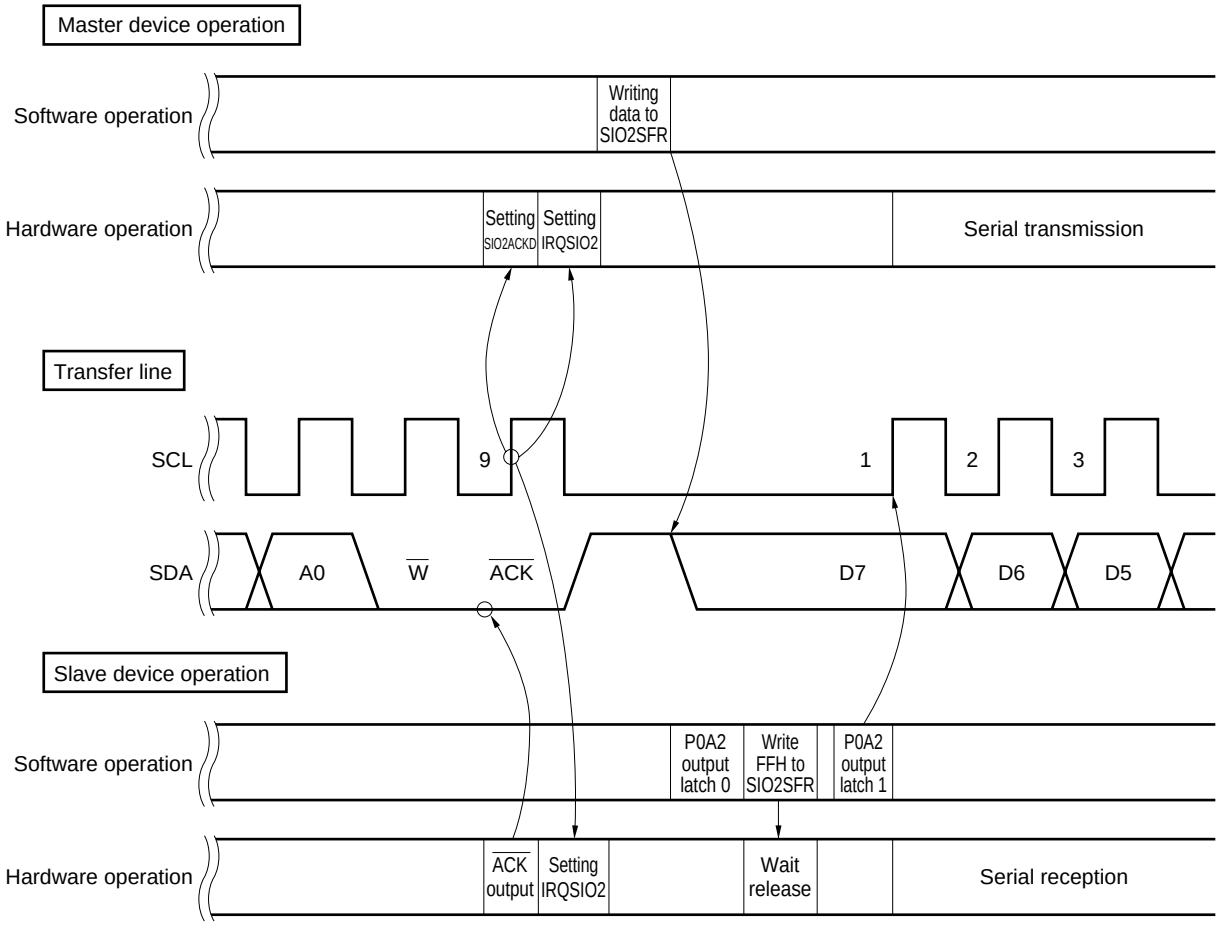
The slave is released from the wait status when the SIO2WREL flag (bit 2 of the serial I/O2 interrupt timing specification register 1) is set or when an instruction that writes data to the presettable shift register 2 (SIO2SFR) is executed.

When the slave receives data, the first bit of the data sent from the master may not be received if the SCL line immediately goes into a high-impedance state after an instruction that writes data to SIO2SFR has been executed.

This is because SIO2SFR does not start operating if the SCL line is in the high-impedance state while the instruction that writes data to SIO2SFR is executed (until the next instruction is executed).

Therefore, receive the data by manipulating the output latch of P0A2 by program, as shown in Figure 16-52. For this timing, refer to Figure 16-48.

Figure 16-52. Slave Wait Release (Reception)



(4) Reception completion of salve

In the reception completion processing of the slave, check the SIO2CMDD flag of the serial I/O2SBI register 1 and SIO2COI flag of the serial I/O2 operation mode register 0 (CSIM0) (when CMDD = 1). This is to avoid the situation where the slave cannot judge which of the start condition and data comes first and therefore, the wake-up condition cannot be used when the slave receives the undefined number of data from the master.

16.2.13 Restrictions in I^2C bus mode

The following restrictions are applied to the μPD17719.

- Restrictions when used as slave device in I^2C bus mode

Description: If the wake-up function is executed (by setting the bit 3 of the serial I/O2 operation mode register 1 to 1) in the serial transfer status**Note**, the μPD17719 checks the address of the data between the other slave and master. If that data happens to coincide with the slave address of the μPD17719, the μPD17719 takes part in communication, destroying the communication data.

Note The serial transfer status is the status since data has been written to the presettable shift register 2 (SIO2SFR) until the interrupt request flag (IRQSIO2) is set to 1 by completion of the serial transfer.

Preventive measure: The above phenomenon can be avoided by modifying the program.
Before executing the wake-up function, execute the following program that clears the serial transfer status. When executing the wake-up function, do not execute an instruction that writes data to SIO2SFR. Even if such an instruction is not executed, data can be received while the wake-up function is executed.
This program releases the serial transfer status. To release the serial transfer status, the serial interface 2 must be once disabled (by clearing the SIO2CSIE flag (bit 3 of the serial I/O2 operation mode register 0 to 0). If the serial interface 2 is disabled in the I^2C bus mode, however, the SCL pin outputs a high level, and SDA pin outputs a low level, affecting communication of the I^2C bus. Therefore, this program makes the SCL and SDA pins go into a high-impedance state to prevent the I^2C bus from being affected.
For the timing of each signal when this program is executed, refer to Figure 16-48.

- Example of program releasing serial transfer status

```
SET1   P0A3      : <1>
CLR1   P0ABIO3   : <2>
CLR1   P0ABIO2   : <3>
CLR1   SIO2CSIE  : <4>
SET1   SIO2CSIE  : <5>
SET1   SIO2RELT  : <6>
SET1   P0ABIO2   : <7>
CLR1   P0A3      : <8>
SET1   P0ABIO3   : <9>
```

<1> This instruction prevents the SDA pin from outputting a low level when the I^2C bus mode is restored by instruction <5>. The output of the SDA pin goes into a high-impedance state.

<2> This instruction sets the P0A3/SDA pin in the input mode to protect the SDA line from adverse influence when the port mode is set by instruction <4>. The P0A3/SDA pin is set in the input mode when instruction <2> is executed.

<3> This instruction sets the P0A2/SCL pin in the input mode to protect the SCL line from adverse influence when the port mode is set by instruction <4>. The P0A2/SCL pin is set in the input mode when instruction <3> is executed.

<4> This instruction changes the mode from I^2C bus mode to port mode.

<5> This instruction restores the I^2C bus mode from the port mode.

<6> This instruction prevents the SDA pin from outputting a low level when instruction <8> is executed.

<7> This instruction sets the P0A2 pin in the output mode because the P0A2 pin must be in the output mode in the I^2C bus mode.

<8> This instruction clears the output latch of the P0A3 pin to 0 because the output latch of the P0A3 pin must be set to 0 in the I^2C bus mode.

<9> This instruction sets the P0A3 pin in the output mode because the P0A3 pin must be in the output mode in the I^2C bus mode.

Remark SIO2RELT: Bit 0 of serial I/O2SBI register 1

16.2.14 SCL/P0A2 and $\overline{SCK2}$/P0A1 pins output manipulation

The SCL/P0A2 and $\overline{SCK2}$/P0A1 pins can execute static output via software, in addition to outputting the normal serial clock.

The number of serial clocks can also be arbitrarily set by software.

The SCL/P0A2 and $\overline{SCK2}$/P0A1 pins output should be manipulated as described below.

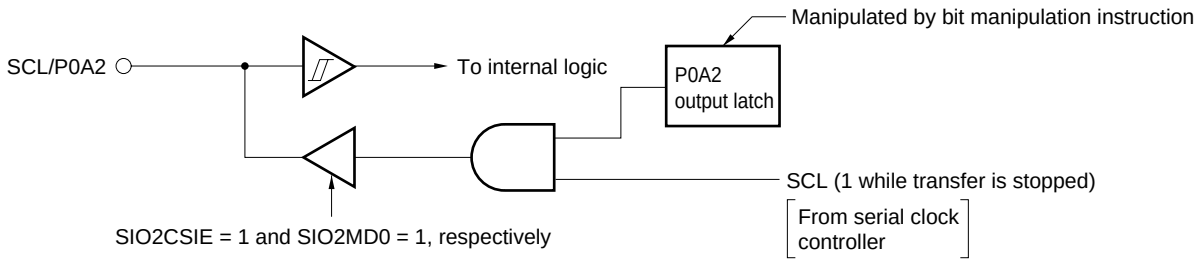
(1) In 2-wire serial I/O mode

The output level of the SCL/P0A2 pin is manipulated by the P0A2 output latch.

<1> Set the serial I/O2 operation mode register 0 and 1 (SCL pin is set in the output mode and serial operation is enabled). SCL = 1 while serial transfer is stopped.

<2> Manipulate the content of the P0A2 output latch by executing the bit manipulation instruction.

Figure 16-53. SCL/P0A2 Pin Configuration



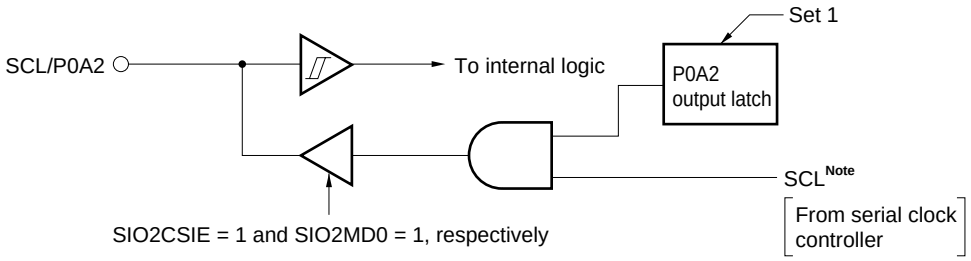
(2) In I^2C bus mode

The output level of the SCL/P0A2 pin is manipulated by the SIO2CLC flag of the serial I/O2 interrupt timing specification register 1.

<1> Set the SIO2 operation mode registers 0 and 1 (SCL pin is set in the output mode and serial operation is enabled). Set 1 to the P0A2 output latch. SCL = 0 while serial transfer is stopped.

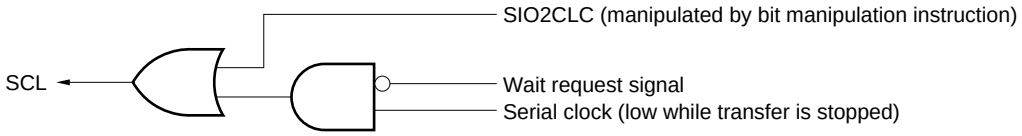
<2> Manipulate the SIO2CLC flag by executing the bit manipulation instruction.

Figure 16-54. SCL/P0A2 Pin Configuration



Note The level of the SCL signal is in accordance with the contents of the logic circuits shown in Figure 16-55.

Figure 16-55. Logic Circuit of SCL Signal



Remarks 1. This figure indicates the relation of the signals and does not indicate the internal circuit.

2. SIO2CLC: Bit 3 of serial I/O2 interrupt timing specification register 1

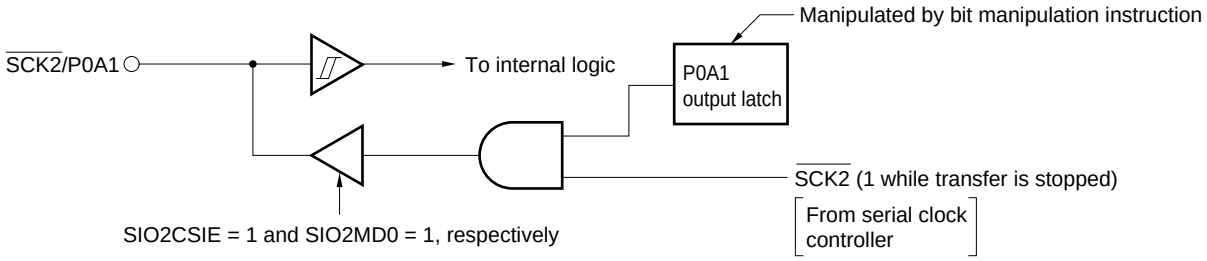
(3) In 3-wire serial I/O mode

The output level of the $\overline{SCK2}$/P0A1 pin is manipulated by the P0A1 output latch.

<1> Set the serial I/O2 operation mode registers 0 and 1 ($\overline{SCK2}$ pin is set in the output mode and serial operation is enabled). $\overline{SCK2}$ = 1 while serial transfer is stopped.

<2> Manipulate the content of the P0A1 output latch by executing the bit manipulation instruction.

Figure 16-56. $\overline{SCK2}$/P0A1 Pin Configuration



16.2.15 Status of serial interface 2 at reset

(1) At power-ON reset

Each pin is set in the general-purpose input port mode.

The contents of the presettable shift register 2 and serial I/O2 slave address register are undefined.

(2) At WDT & SP reset

Each pin is set in the general-purpose input port mode.

The contents of the presettable shift register 2 and serial I/O2 slave address register are undefined.

(3) At CE reset

Each pin retains the previous status.

The contents of the presettable shift register 2 and serial I/O2 slave address register are undefined.

(4) On execution of clock stop instruction

Each pin is set in the general-purpose input port mode.

The contents of the presettable shift register 2 and serial I/O2 slave address register are undefined.

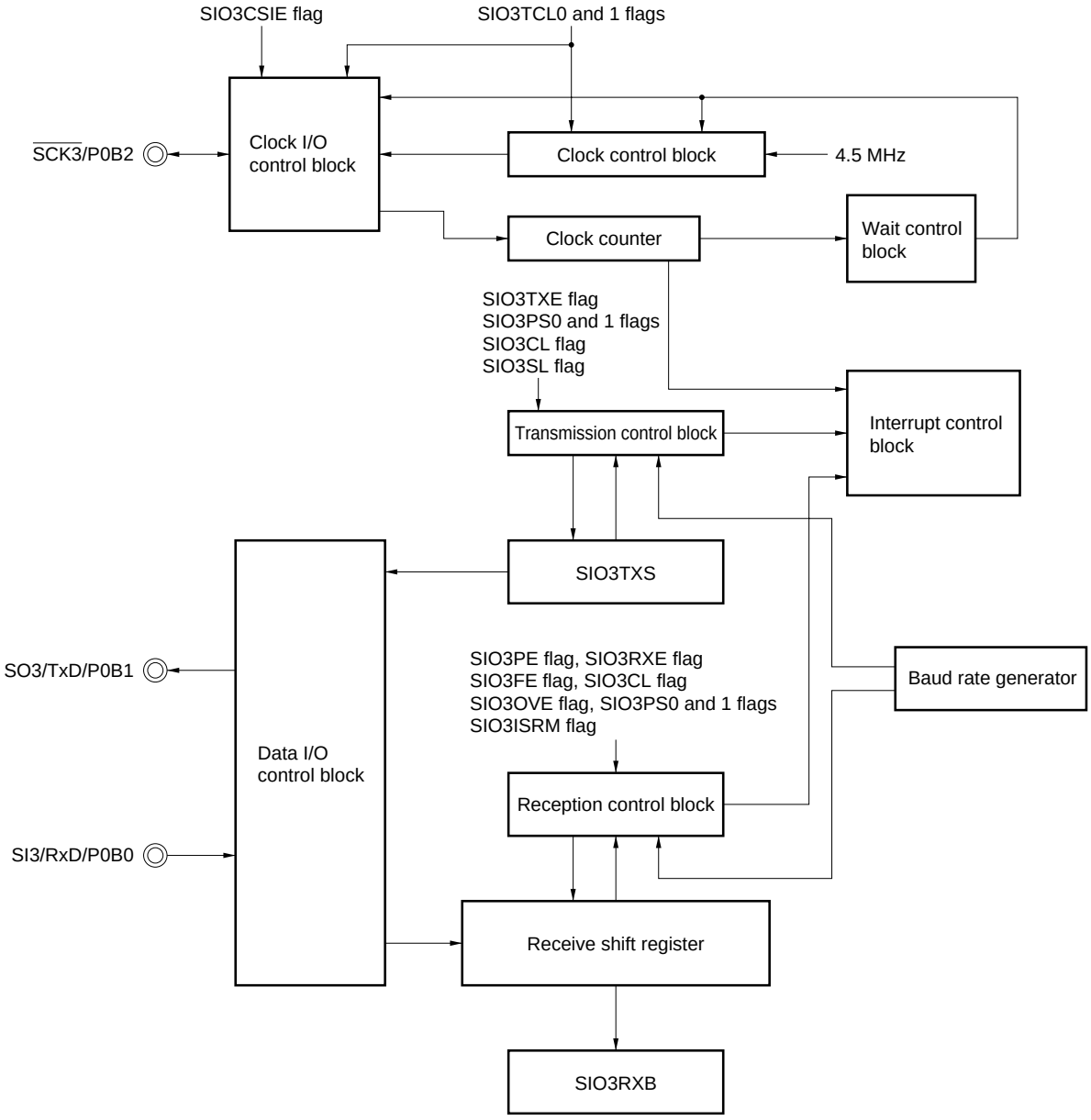
16.3 Serial Interface 3

16.3.1 Outline of serial interface 3

Figure 16-57 shows the outline of serial interface 3.

Serial interface 3 can be used in UART and 3-wire serial I/O modes.

Figure 16-57. Outline of Serial Interface 3



16.3.2 Control registers of serial interface 3

Serial interface 3 is controlled by the following four registers:

- Serial I/O3 operation mode register
- Serial I/O3 asynchronous status register
- Serial I/O3 asynchronous mode register 0
- Serial I/O3 asynchronous mode register 1

(1) Serial I/O3 operation mode register

Figure 16-58 shows the configuration of the serial I/O3 operation mode register.

This register controls the operation of 3-wire serial I/O mode, and select the clock to be used.

Figure 16-58. Configuration of Serial I/O3 Operation Mode Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Serial I/O3 operation mode register	SIO3CSIE	SIO3HIZ	SIO3TCL1	SIO3TCL0	1AH	R/W

SIO3TCL1	SIO3TCL0	Selects clock of 3-wire serial I/O
0	0	External clock
0	1	187.5 kHz
1	0	375 kHz
1	1	46.875 kHz

SIO3HIZ	Status of SO3/P0B1 pin[Note 1]
0	General-purpose I/O port
1	Serial data output[Note 2]

SIO3CSIE	Enables or stops operation of 3-wire serial I/O
0	Stops operation (wait status)
1	Enables operation of 3-wire serial I/O

		SIO3CSIE	SIO3HIZ	SIO3TCL1	SIO3TCL0
At reset	Power-ON reset	0	0	0	0
	WDT & SP reset	0	0	0	0
	CE reset	0	0	0	0
Clock stop		0	0	0	0

Notes 1. This flag is ignored in any mode other than 3-wire serial I/O mode.

2. Port 0B bit I/O select flag P0BBIO1 must be set to 1 and the port latch must be set to 1.

Caution Be sure to clear the SIO3TXE and SIO3RXE flags of the serial I/O3 asynchronous mode register 0 to "0" when using the 3-wire serial I/O mode.
When using the UART mode, be sure to clear the SIO3CSIE flag to "0".

(2) Serial I/O3 asynchronous status register

Figure 16-59 shows the configuration of the serial I/O3 asynchronous status register.

This register indicates the nature of a reception error if any when the UART mode is used.

The value of this register is cleared to “0” when data of the serial I/O3 receive buffer register (SIO3RXE) is read.

Figure 16-59. Configuration of Serial I/O3 Asynchronous Status Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Serial I/O3 asynchronous status register	0	SIO3PE	SIO3FE	SIO3OVE	1BH	R

Contents of serial I/O3 overrun error	
0	When overrun does not occur or when data is read from serial I/O3 receive buffer register
1	If data of serial I/O3 receive buffer register overlaps

Contents of serial I/O3 framing error	
0	If framing error does not occur or if data is read from serial I/O3 receive buffer register
1	If stop bit is not detected

Contents of serial I/O3 parity error	
0	If parity error does not occur, or if data is read from serial I/O3 receive buffer register
1	If parity of transmit data does not coincide

Fixed to “0”

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT & SP reset		0	0	0
	CE reset		0	0	0
Clock stop			0	0	0

(3) Serial I/O3 asynchronous mode register 0

This register sets the operation in the UART mode.

Figure 16-60 shows the configuration of the serial I/O3 asynchronous mode register 0.

Figure 16-60. Configuration of Serial I/O3 Asynchronous Mode Register 0

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Serial I/O3 asynchronous mode register 0	SIO3TXE	SIO3RXE	SIO3ISRM	0	1DH	R/W

Fixed to "0"

Enables or disables generation of reception completion interrupt on occurrence of error

b_1	
0	Enables interrupt
1	Disables interrupt

Sets operation in UART mode

b_3	b_2	
0	0	Stops operation
0	1	UART mode (reception)
1	0	UART mode (transmission)
1	1	UART mode (transmission/reception)

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT & SP reset	0	0	0	
	CE reset	0	0	0	
Clock stop		0	0	0	

Caution Be sure to clear the SIO3CSIE flag of the serial I/O3 operation mode register to "0 when using the UART mode. Clear the SIO3TXE and SIO3RXE flags to "0" when using the 3-wire serial I/O mode.

(4) Serial I/O3 asynchronous mode register 1

This register sets the parity bit, character length, and stop bit in the UART mode.

Figure 16-61 shows the configuration of the serial I/O3 asynchronous mode register 1.

Figure 16-61. Configuration of Serial I/O3 Asynchronous Mode Register 1

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Serial I/O3 asynchronous mode register 1	SIO3PS1	SIO3PS0	SIO3CL	SIO3SL	1CH	R/W

SIO3SL	Specifies number of stop bits of UART transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

SIO3CL	Specifies character length of UART
0	7 bits
1	8 bits

SIO3PS1	SIO3PS0	Specifies parity bit of UART
0	0	No parity
0	1	Transmission: Parity appended Reception: Parity error not generated
1	0	Odd parity
1	1	Even parity

		SIO3PS1	SIO3PS0	SIO3CL	SIO3SL
At reset	Power-ON reset	0	0	0	0
	WDT & SP reset	0	0	0	0
	CE reset	0	0	0	0
Clock stop		0	0	0	0

Caution Be sure to rewrite this register when the operation in the UART mode is stopped.

16.3.3 Serial I/O3 transmit register (SIO3TXS) and serial I/O3 receive buffer register (SIO3RXB)

Both the serial I/O3 transmit register (SIO3TXS) and serial I/O3 receive buffer register (SIO3RXB) are assigned to peripheral address 05H.

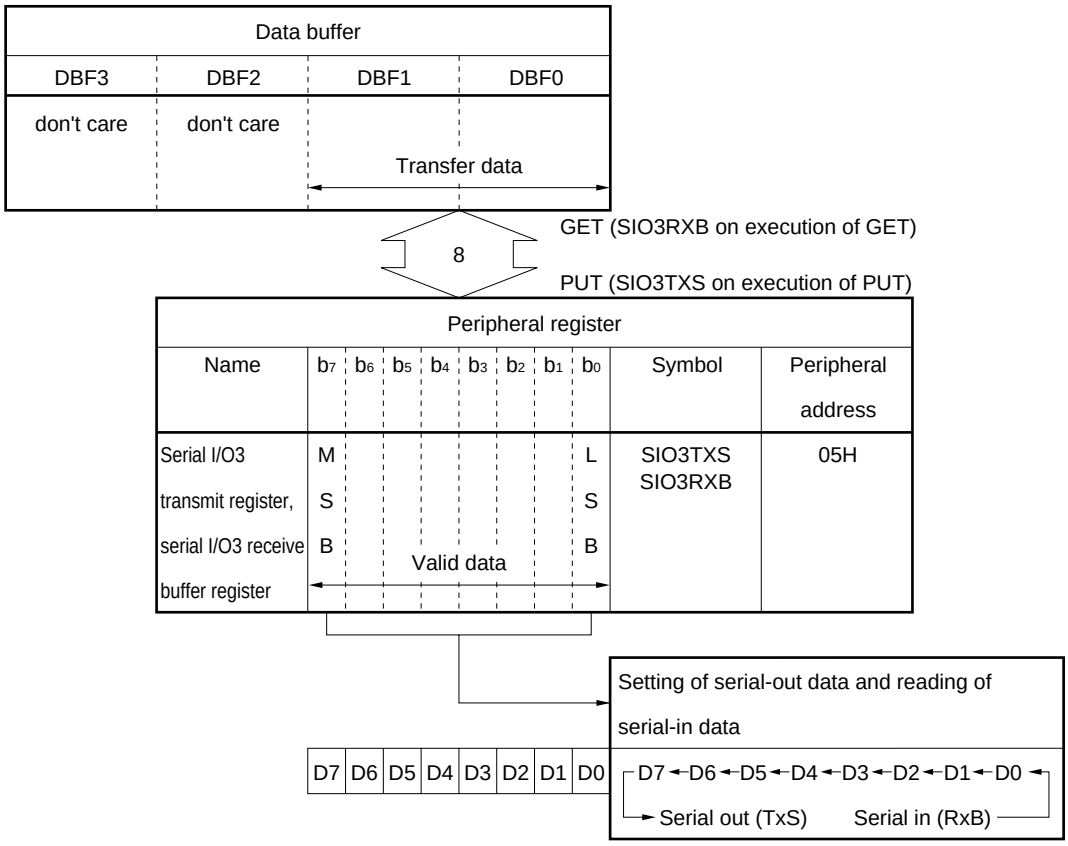
SIO3TXS is a register that sets transmit data in the 3-wire serial I/O mode and UART mode. Data b_6 through b_0 are transmitted when the character length is set to 7 bits in the UART mode.

SIO3RXB is a register that stores receive data in the 3-wire serial I/O mode and UART mode. Data b_6 through b_0 are received, and b_7 is always "0" when the character length is set to 7 bits in the UART mode.

When the PUT instruction is executed, the data of the data buffer is written to SIO3TXS. When the GET instruction is executed, the data of SIO3RXB is read to the data buffer.

Figure 16-62 shows the configuration of the serial I/O3 transmit register and serial I/O3 receive buffer register.

Figure 16-62. Configuration of Serial I/O3 Transmit Register and Serial I/O3 Receive Buffer Register



Caution Do not write data to this register during operation in the 3-wire serial I/O mode. During transmission operation in the UART mode, this register is masked and no data can be written to it.

16.3.4 Operation of serial interface 3

Serial interface 3 operates in the following two modes:

- 3-wire serial I/O mode
- UART mode

Table 16-10 shows the setting of each pin by each control flag in each operation mode.

Table 16-10. Pin Setting Status by Each Control Flag

Flag							Pin								
SIO3CSIE	SIO3TXE	SIO3RXE	Communication mode	SIO3TCL1	SIO3TCL0	Clock direction	Pin name	SIO3HIZ	P0BB2	P0BB1	P0BB0	Setting status of pin
1	×	×	3-wire serial I/O	0	0	External (slave)	$\overline{SCK3}$/P0B2		0	×				External clock input
									1	×				General-purpose output port
				0, 1 or 1, ×		Internal (master)			0	×				General-purpose input port
									1	1				Internal clock output
				×	×		SO3/TxD/P0B1	0			0	×		General-purpose input port
											1	×		General-purpose output port
								1			0	×		General-purpose input port
											1	1		Serial output
							SI3/RxD/P0B0						0	×	Serial input
													1	×	General-purpose output port
0	1	0	UART (transmission)	×	×		SO3/TxD/P0B1				0	×		General-purpose input port
											1	1		Serial output
							SI3/RxD/P0B0						0	×	General-purpose input port
													1	×	General-purpose output port
	0	1	UART (reception)	×	×		SO3/TxD/P0B1				0	×		General-purpose input port
											1	×		General-purpose output port
							SI3/RxD/P0B0						0	×	Serial input
													1	×	General-purpose output port
	1	1	UART (transmission/ reception)	×	×		SO3/TxD/P0B1				0	×		General-purpose input port
											1	1		Serial output
							SI3/RxD/P0B0						0	×	Serial input
													1	×	General-purpose output port

×: don't care

16.3.5 3-wire serial I/O mode

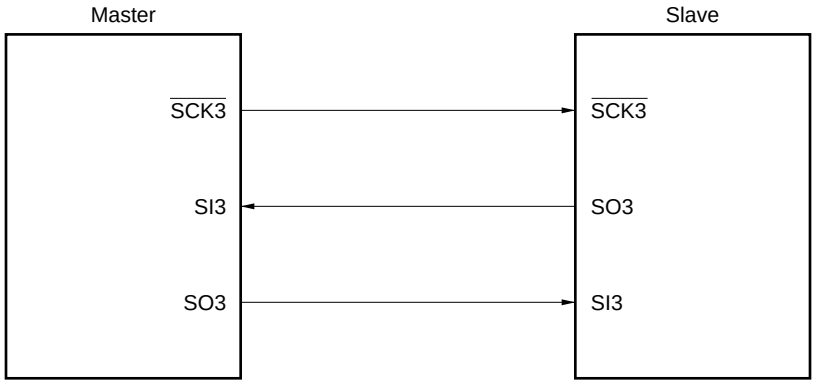
(1) Outline of 3-wire serial I/O mode

In the 3-wire serial I/O mode, communication is executed by using three pins: $\overline{\text{SCK3}}$, SI3, and SO3 pins. Table 16-11 shows the outline of the 3-wire serial I/O mode.

Table 16-11. Outline of 3-Wire Serial I/O Mode

Pin used for communication	• $\overline{\text{SCK3}}$ pin (serial clock I/O pin) • SI3 pin (serial data input pin) • SO3 pin (serial data output pin)	
Transmission/reception operation	Transmit data	Sequentially output from MSB of shift register to data output pin in synchronization with falling of $\overline{\text{SCK3}}$ pin.
	Receive data	Value of data input pin from LSB of shift register in synchronization with rising of $\overline{\text{SCK3}}$ pin.
Transmission/reception start	Master	Transmission/reception is started by setting transfer data to transmit register after 3-wire serial I/O master mode has been set.
	Slave	Waits for clock from master with $\overline{\text{SCK3}}$ going into high-impedance state after 3-wire serial I/O slave mode has been set.
Interrupt	Issues interrupt request flag IRQSIO3 at rising of clock of 8th count	
Clock pin	Master	Stops output of $\overline{\text{SCK3}}$ pin at rising of 8th count and retains high level until next transmission/reception is started
	Slave	Goes into high-impedance state

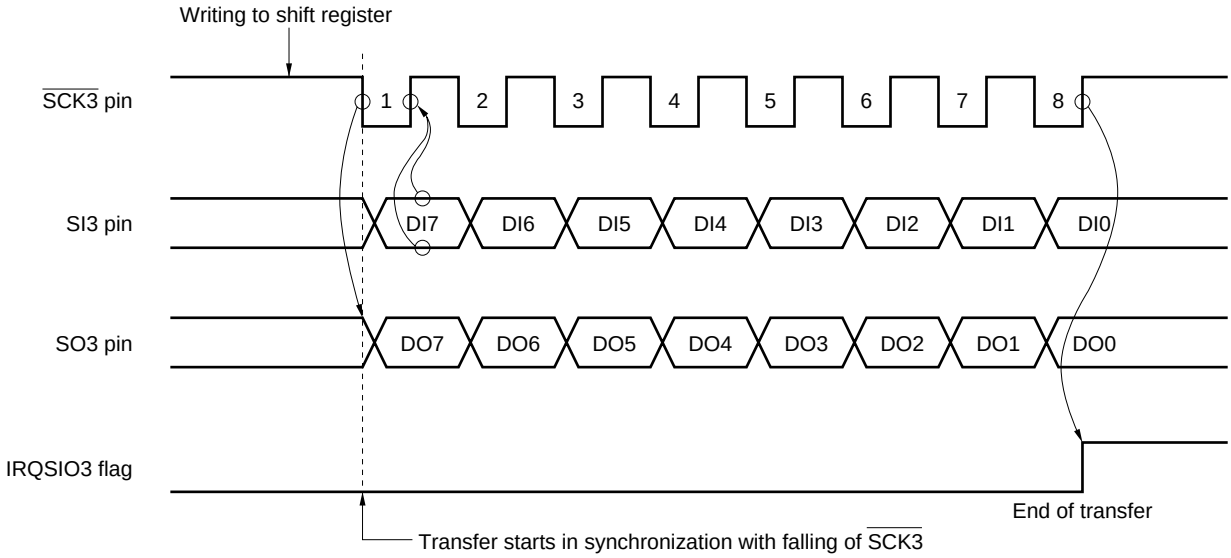
Figure 16-63. Serial Bus Configuration Example in 3-Wire Serial I/O Mode



(2) Timing chart

Figure 16-64 shows the timing chart in the 3-wire serial I/O mode.

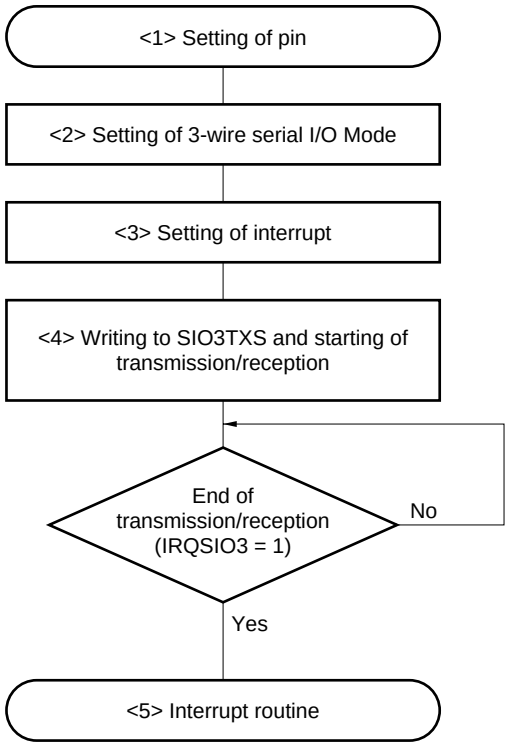
Figure 16-64. Timing Chart in 3-Wire Serial I/O Mode



(3) Program flowchart in 3-wire serial I/O mode

An example of program flow chart in the 3-wire serial I/O mode is shown below.

Figure 16-65. Example of Flow Chart in 3-Wire Serial I/O Mode



Remark To execute a 3-wire serial I/O operation with the same setting as before, start from step <4>.

<1> Setting of pin

(a) To input serial data from SI3 pin

Set the I/O control mode of the SI3 pin to “0” (input).

(b) To output serial data from SO3 pin

Set the I/O control mode of the SO3 pin to “1” (output) and the port register of the SO3 pin to “1” (output), respectively.

In addition, set the SIO3HIZ flag of the serial /O3 operation mode register to “1” (at this point, the SO3 pin outputs a high level).

(c) Setting of $\overline{\text{SCK3}}$ pin

- **To output internal clock from $\overline{\text{SCK3}}$ pin**

  Set the port register of the SO3 pin to “1”. In addition, select an internal clock by using the SIO3TCL0 and 1 flags of the serial I/O3 operation mode register in step <2>.

- **To input external clock to $\overline{\text{SCK3}}$ pin**

  Select an external clock by using the SIO3TCL0 and 1 flags of the serial I/O3 operation mode register in step <2>.

<2> Setting 3-wire serial I/O transmission mode as communication mode

Set the following three by using the serial I/O3 operation mode register.

- 3-wire serial I/O mode
- Clock
- SO3 pin

Caution Be sure to clear the SIO3TXE and SIO3RXE flags to "0".

<3> Setting of interrupt

Execute the "EI instruction" to set the IPSIO3 flag to "1".

<4> Setting of transmit data to SIO3TXS register

Start the 3-wire serial I/O transmission/reception operation as soon as the data has been set. Output 8-bit transmit data from the SO3 pin. Store the serial data input from the SI3 pin to the SIO3RXB register as 8-bit receive data.

<5> Interrupt routine

Interrupt request flag IRQSIO3 is issued when the 3-wire serial I/O transmission/reception has been completed, and if the interrupt request is accepted, execution branches to a vector address.

16.3.6 UART mode

The UART (Universal Asynchronous Receiver/Transmitter) mode is to transmit/receive 1-byte data following a start bit. In this mode, full-duplex operation can be executed.

The baud rate is fixed to 9575 bps.

(1) Outline of UART mode

Table 16-12 shows the outline of the UART mode.

Table 16-12. Outline of UART Mode

Pins used for communication	• TxD (serial data output pin. Outputs high level when transmission is not executed) • RxD (serial data input pin)	
Transfer rate	9575 bps (automatic generation)	
First bit	LSB	
Transmission/reception operation	Transmit data	Data of 7 or 8 bits is transmitted from TxD pin. Start bit, parity bit, and stop bit are automatically generated.
	Receive data	Data of 7 or 8 bits following start bit is received from RxD pin. Reception errors such as parity error, framing error, and overrun error, are detected.
Starting transmission/ reception	Master	Transmission/reception is started by setting transfer data to transmit register after UART mode has been set.
	Slave	Low level is input to RxD pin after UART mode has been set. If RxD pin remains low for about 52 μs (9575 × 2 Hz), it is recognized as start bit and reception is started.
Interrupt	Interrupt request IRQSIO3 is issued on completion of transmission, reception, or transmission/reception.	

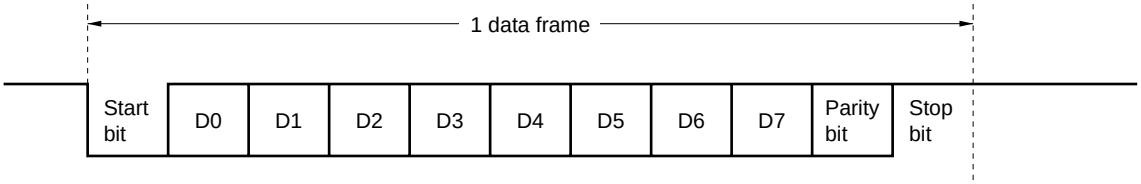
16.3.7 Data format in UART mode

(1) Data format

The format of the transmit/receive data is as shown in Figure 16-66.

One data frame consists of a start bit, character bits, a parity bit, and stop bit(s). The transfer rate is fixed to 9575 bps (automatically generated from the internal clock).

Figure 16-66. Format of Transmit/Receive Data in UART Mode



- Start bit ..... 1 bit
- Character bits .... 7 or 8 bits
- Parity bit ..... Odd parity/even parity
  - Transmission: 0 parity
  - Reception: Parity error does not occur. No parity
- Stop bit ..... 1 or 2 bits (always 1 bit for reception)

When the number of character bits is set to 7, only the low-order 7 bits (bits 0 through 6) are valid. When data is transmitted, the MSB (bit 7) of the serial I/O transmit register (SIO3TXS) is ignored. When data is received, the MSB (bit 7) of the serial I/O3 receive buffer register (SIO3RXB) is always 0.

If a reception error of serial data occurs, the nature of the error can be identified by reading the status of the serial I/O3 asynchronous status register.

(2) Type and operation of parity

The parity bit is used to detect a bit error in the communication data. Usually, the same type of parity is used at the transmission and reception sides.

Odd parity and even parity can detect an error of 1 bit (the number of 1's in data is odd). No error can be detected when 0 parity or no parity is used.

Table 16-13 shows the type and operation of parity.

Table 16-13. Type and Operation of Parity

Even parity	Transmission	• If number of 1's in transmit data is odd → Parity bit is "1". • If number of 1's in transmit data is even → Parity bit is "0". This controls number of 1's included in transmit data and parity bit to be even.
	Reception	• Counts number of 1's included in receive data and parity bit. If it is odd, parity error occurs.
Odd parity	Transmission	• If number of 1's in transmit data is odd → Parity bit is "0". • If number of 1's in transmit data is even → Parity bit is "1". This controls number of 1's included in transmit data and parity bit to be even.
	Reception	• Counts number of 1's included in receive data and parity bit. If it is even, parity error occurs.
0 parity	Transmission	Clears parity bit to "0" regardless of transmit data.
	Reception	Does not check parity bit. Therefore, parity error does not occur regardless of whether parity bit is "0" or "1".
No parity	Transmission	Parity bit is not appended.
	Reception	Reception is performed with no parity bit assumed. Because no parity bit is used, parity error does not occur.

(3) Reception error

Three types of reception errors may occur: parity error, framing error, and overrun error.

If the SIO3ISRM flag is "0" when a reception error occurs, the SIO3 interrupt request (reception completion interrupt) is issued. If the SIO3ISRM flag is "1", the SIO3 interrupt request (reception completion interrupt) is not issued.

The cause of the reception error can be detected by reading the serial I/O3 asynchronous status register after completion of the reception operation.

The serial I/O3 asynchronous status register is cleared to 0 when the serial I/O3 receive buffer register (SIO3RXB) is read.

Therefore, the serial I/O3 asynchronous status register must be read before the serial I/O3 receive buffer register is read.

Even if a reception error occurs, data is transferred to the serial I/O3 receive buffer register.

Table 16-14 describes each reception error.

Table 16-14. Reception Error

Parity error	Parity bit specified during transmission does not coincide with specified parity bit of receive data.
Framing error	Stop bit is not detected (RxD pin is low when stop bit is to be detected).
Overrun error	Reception of next data is completed before data is read from serial I/O3 receive buffer register.

(4) Detection of start bit

The reception operation is enabled when the SIO3RXE flag of the serial I/O3 asynchronous status register 0 is set to 1, and the RxD pin input is sampled.

A start bit is recognized if the RxD pin is low about 52 μs (9575 × 2 Hz) after a low level has been input to the RxD pin, and the reception operation is started.

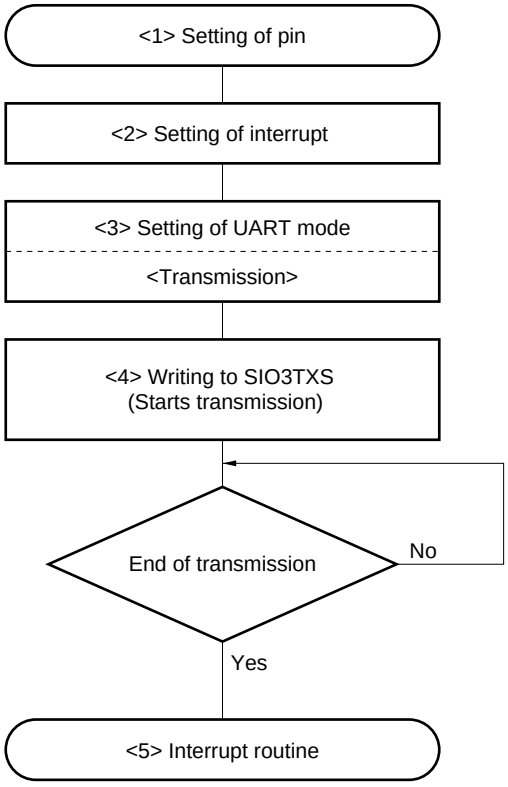
If a low level is input to the RxD pin and the RxD pin goes high after about 52 μs (9575 × 2 Hz), the start bit is not recognized, and the reception operation is not started. At this time, the reception is enabled again, and the RxD pin input is sampled.

16.3.8 Program flowchart in UART mode

(1) Flowchart in UART transmission mode

Here is an example of a program flowchart in the UART transmission mode.

Figure 16-67. Flowchart Example in UART Transmission Mode



Remark To execute transmission in the UART mode with the same setting as before, start from step <4>.

<1> Setting of pin (to output serial data from RxD pin)

1. Set the P0BBIO1 flag to “1” (output).
2. Set the port register of the TxD pin to “1” (at this point, the TxD pin outputs a high level).

<2> Setting of interrupt

Execute the “EI” instruction and set the IPSIO3 flag to “1”.

<3> Setting of UART

1. Set the following in the serial I/O3 asynchronous mode register 1.
   - Parity bit
   - Character length
   - Stop bit
2. Set the UART mode (transmission) by using the serial I/O3 asynchronous mode register 0.

Caution Be sure to clear the SIO3CSIE flag to “0”.

<4> Set transmit data to the SIO3TXS register (start transmission)

UART transmission is started as soon as data has been set. The TxD pin outputs the start bit, transmit data (7 or 8 bits), parity bit, and stop bit (1 or 2 bits), and the transmission is completed.

If the character length is 7 bits, however, the bit 7 (MSB) of the SIO3TXS register is ignored.

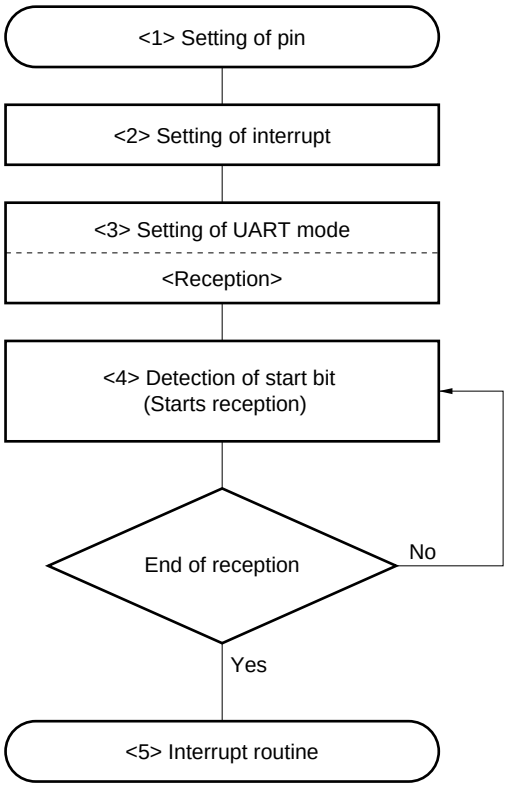
<5> Interrupt routine

When the UART transmission operation is completed, the interrupt request flag IRQSIO3 is issued. When this interrupt is accepted, execution branches to the vector address.

(2) Flowchart in UART reception mode

Here is an example of a program flowchart in the UART reception mode.

Figure 16-68. Flowchart Example in UART Reception Mode



Remark To execute reception in the UART mode with the same setting as before, start from step <4>.

<1> Setting of pin (to input serial data from RxD pin)

Set the P0BBIO1 flag to “0” (input).

<2> Setting of interrupt

Execute the “EI” instruction and set the IPSIO3 flag to “1”.

<3> Setting of UART

1. Set the following in the serial I/O3 asynchronous mode register 1.
   - Parity bit
   - Character length

   The number of stop bits is 1 during reception, regardless of the setting.
2. Set the following two to the serial I/O3 asynchronous mode register 0.
   - UART mode (reception)
   - Reception completion interrupt in case of reception error

Caution Be sure to clear the SIO3CSIE flag to “0”.

<4> Detection of start bit

UART reception is started as soon as the start bit has been detected from the RxD pin. The RxD pin inputs the start bit, transmit data (7 or 8 bits), parity bit, and stop bit (1 bit) in that order. The received data is stored to the SIO3RXB register and the reception is completed.

<5> Interrupt routine

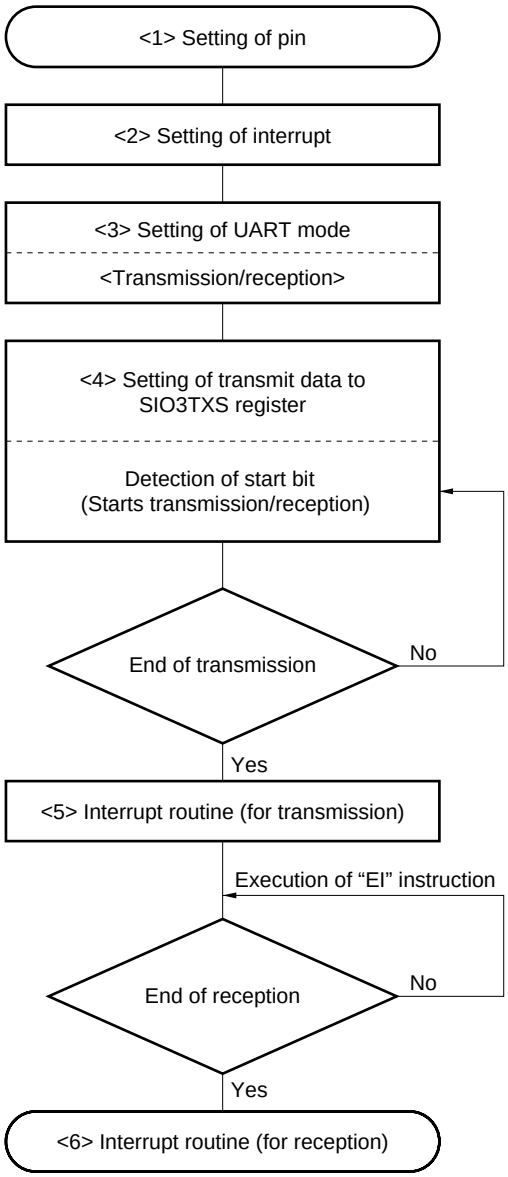
When the UART transmission operation is completed, the interrupt request flag IRQSIO3 is issued. When this interrupt is accepted, execution branches to the vector address.

Caution Because the serial I/O3 asynchronous status register is cleared to "0" when the SIO3RXB register has been read, read the serial I/O3 asynchronous status register and then the SIO3RXB register.

(3) Flowchart in UART transmission/reception mode

Here is an example of a program flowchart in the UART transmission/reception mode.

Figure 16-69. Flowchart Example in UART Transmission/Reception Mode



Remark To execute reception in the UART mode with the same setting as before, start from step <4>.

Caution This program flowchart shows an example where transmission and then reception have been completed in that order after transmission and reception have been started.
In the following cases, the interrupt request IRQSIO3 flag of serial I/O3 may not be detected two times (completion of transmission/reception), unlike in the above flowchart:

- **If transmission is completed before the IRQSIO3 flag is cleared to 0 after completion of transmission .**
- **If transmission is completed before the IRQSIO3 flag is cleared to 0 after completion of reception.**

<1> Setting of pin (to output serial data from TxD pin and input serial data from RxD pin)

1. Set the P0BBIO1 flag to “1” (output).
2. Set the port register of the TxD pin to “1” (at this point, the TxD pin outputs a high level).
3. Set the P0BBIO0 flag to “0” (input).

<2> Setting of interrupt

Execute the “EI” instruction and set the IPSIO3 flag to “1”.

<3> Setting of UART

1. Set the following in the serial I/O3 asynchronous mode register 1.
   - Parity bit
   - Character length
   - Stop bit

   The number of stop bits is 1 during reception, regardless of the setting.
2. Set the following in the serial I/O3 asynchronous mode register 0.
   - UART mode (transmittion/reception)
   - Reception completion interrupt in case of reception error

Caution Be sure to clear the SIO3CSIE flag to “0”.

<4> Set transmit data to the SIO3TXS register (start transmission)

UART transmission is started as soon as data has been set. The TxD pin outputs the start bit, transmit data (7 or 8 bits), parity bit, and stop bit (1 or 2 bits) in that order, and the transmission is completed.

If the character length is 7 bits, however, the bit 7 (MSB) of the SIO3TXS register is ignored.

- Detection of start bit

  UART reception is started as soon as the start bit has been detected from the RxD pin. The RxD pin inputs the start bit, transmit data (7 or 8 bits), parity bit, and stop bit (1 bit) in that order. The received data is stored to the SIO3RXB register and the reception is completed.

<5> Interrupt routine (for transmission)

When the UART transmission operation is completed, the interrupt request flag IRQSIO3 is issued. When this interrupt is accepted, execution branches to the vector address.

<6> Interrupt routine (for reception)

When the UART transmission operation is completed, the interrupt request flag IRQSIO3 is issued, and data is set to the serial I/O3 asynchronous status register (however, only if a reception error occurs). When this interrupt is accepted, execution branches to the vector address.

Caution Because the serial I/O3 asynchronous status register is cleared to “0” when the SIO3RXB register has been read, read the serial I/O3 asynchronous status register and then the SIO3RXB register.

16.3.9 Cautions on UART mode

The data of SIO3TXS is other than “FFH” after the following UART operation. To execute UART transmission after that, be sure to set “FFH” to SIO3TXS and then set the SIO3TXE flag to 1.

This is because the UART transmit shift clock operates and the data of SIO3TXS is output from the TxD pin if the SIO3TXE flag is set to 1.

- If SIO3TXE is cleared to 0 during UART transmission.
- If SIO3RXE is cleared to 0 during UART reception.

After completion of transmission in the UART mode, and after completion of the operation in the 3-wire serial I/O mode, the data of SIO3TXS is “FFH”.

16.3.10 Status of serial interface 3 at reset

(1) At power-ON reset

Each pin is set in the general-purpose input port mode.

The contents of the serial I/O3 transmit register (SIO3TXS) and serial I/O3 receive buffer register (SIO3RXB) are FFH.

(2) At WDT & SP reset

Each pin is set in the general-purpose input port mode.

The contents of the serial I/O3 transmit register and serial I/O3 receive buffer register are FFH.

(3) At CE reset

Each pin is set in the general-purpose input port mode.

The contents of the serial I/O3 transmit register and serial I/O3 receive buffer register are FFH.

(4) On execution of clock stop instruction

Each pin is set in the general-purpose input port mode.

The contents of the serial I/O3 transmit register and serial I/O3 receive buffer register are FFH.

17. PLL FREQUENCY SYNTHESIZER

The PLL (Phase Locked Loop) frequency synthesizer is used to lock a frequency in the MF (Medium Frequency), HF (High Frequency), and VHF (Very High Frequency) to a constant frequency by means of phase difference comparison.

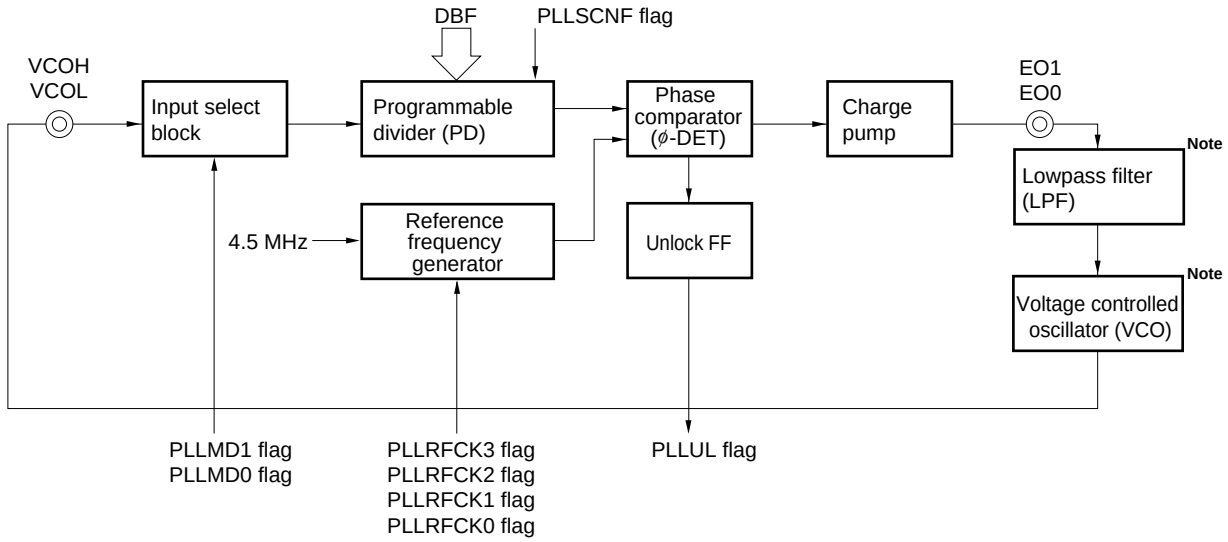
17.1 Outline of PLL Frequency Synthesizer

Figure 17-1 outlines the PLL frequency synthesizer. A PLL frequency synthesizer can be configured by connecting an external lowpass filter (LPF) and voltage controlled oscillator (VCO).

The PLL frequency synthesizer divides a signal input from the VCOH or VCOL pin by using a programmable divider and outputs a phase difference between this signal and a reference frequency from the EO0 and EO1 pins.

The PLL frequency synthesizer operates only while the CE pin is high. It is disabled when the CE pin is low. For the details of the disabled status of the PLL frequency synthesizer, refer to **17.5 PLL Disabled Status**.

Figure 17-1. Outline of PLL Frequency Synthesizer



Note External circuit

Remarks 1. PLLMD1 and PLLMD0 (bits 1 and 0 of PLL mode selection register: refer to **Figure 17-3**) selects a division mode of the PLL frequency synthesizer.

2. PLLSCNF (bit 3 of PLL mode selection register: refer to **Figure 17-3**) selects the least significant bit of the swallow counter.

3. PLLRFCK3 through PLLRFCK0 (bits 3 through 0 of PLL reference frequency selection register: refer to **Figure 17-6**) selects a reference frequency fr of the PLL frequency synthesizer.

4. PLLUL (bit 0 of PLL unlock FF register: refer to **Figure 17-9**) detects the PLL unlock FF status.

17.2 Input Selection Block and Programmable Divider

17.2.1 Configuration and function of input selection block and programmable divider

Figure 17-2 shows the configuration of the input selection block and programmable divider.

The input selection block selects an input pin and division mode of the PLL frequency synthesizer.

The VCOH or VCOL pin can be selected as the input pin.

The voltage on the selected pin is at the intermediate level (approx. 1/2 V_{DD}). The pin not selected is internally pulled down.

Because these pins are connected to an internal AC amplifier, cut the DC component of the input signal by connecting a capacitor in series to the pin.

Direct division mode and pulse swallow mode can be selected as division modes.

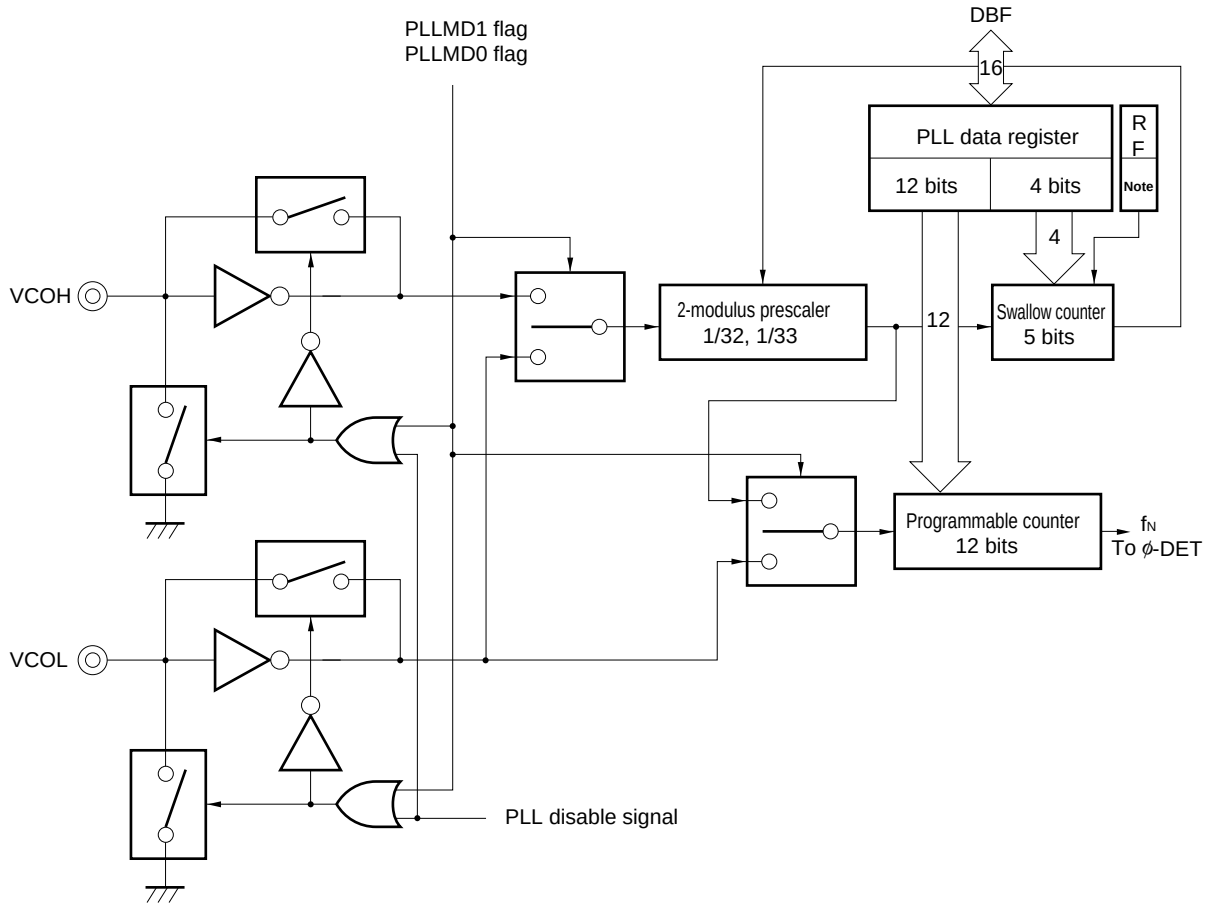
The programmable divider divides the frequency of the input signal according to the value set to the swallow counter and programmable counter.

The pin and division mode to be used are selected by the PLL mode selection register.

Figure 17-3 shows the configuration of the PLL mode selection register.

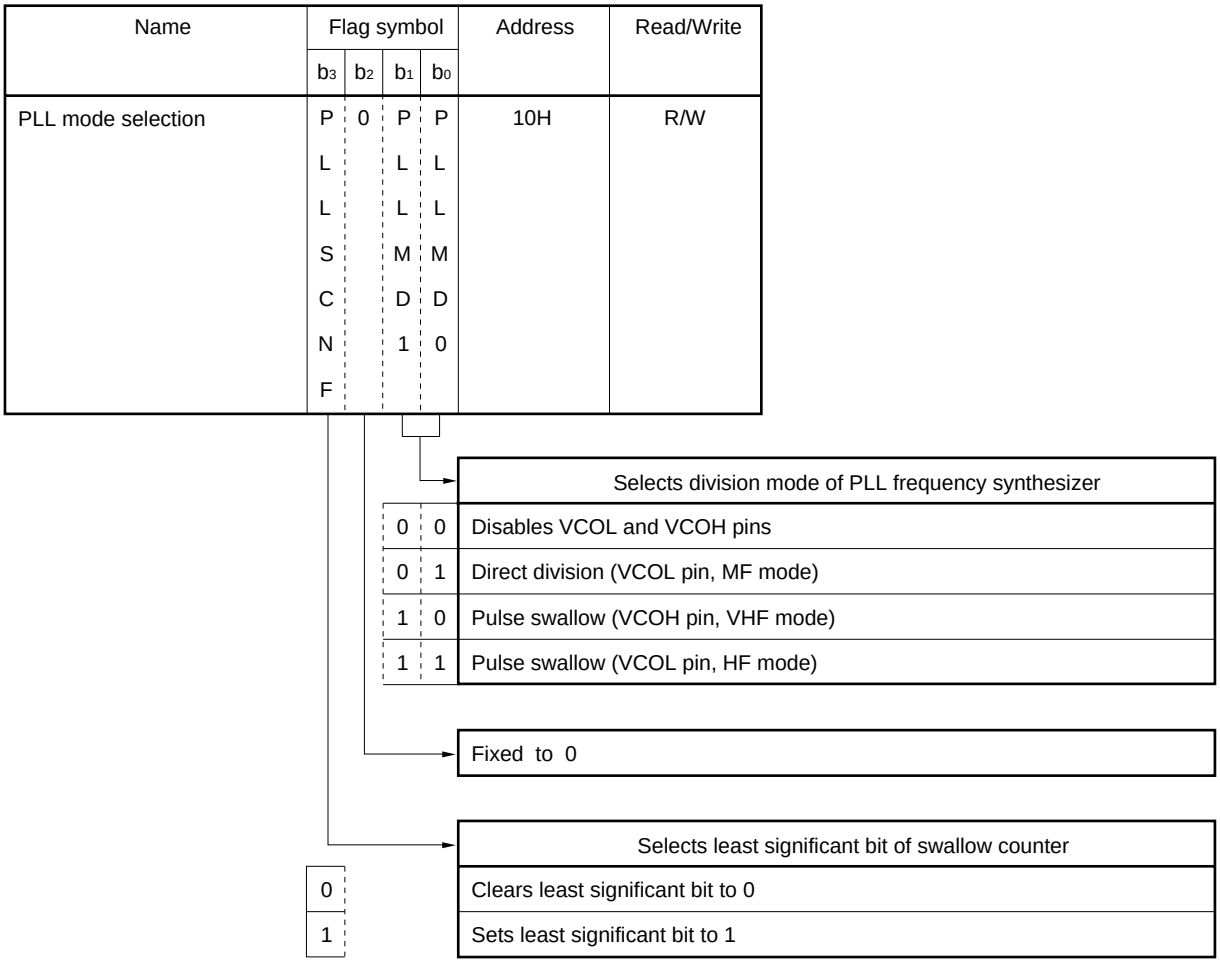
The value of the programmable divider is set by using the PLL data register via data buffer.

Figure 17-2. Configuration of Input Selection Block and Programmable Divider



Note PLLSCNF flag

Figure 17-3. Configuration of PLL Mode Selection Register



At reset	Power-ON reset	U	0	0	0
	WDT&SP reset	U		0	0
	CE reset 1	R		0	0
Clock stop		R		0	0

U: Undefined R: Retained

17.2.2 Outline of each division mode

(1) Direct division mode (MF)

In this mode, the VCOL pin is used.

The VCOH pin is pulled down.

In this mode, only the programmable counter is used for frequency division.

(2) Pulse swallow mode (HF)

In this mode, the VCOL pin is used.

The VCOH pin is pulled down.

In this mode, the swallow counter and programmable counter are used for frequency division.

(3) Pulse swallow mode (VHF)

In this mode, the VCOH pin is used.

The VCOL pin is pulled down.

In this mode, the swallow counter and programmable counter are used for frequency division.

(4) VCOL and VCOH pin disabled

In this mode, only the VCOL and VCOH pins are internally pulled down, but the other blocks operate.

17.2.3 Programmable divider and PLL data register

The programmable divider consists of a 5-bit swallow counter and a 12-bit programmable counter. Each counter is a 17-bit binary down counter.

The programmable counter is allocated to the high-order 12 bits of the PLL data register, and the swallow counter is allocated to the low-order 4 bits. Data are set to these counters via data buffer.

The least significant bit of the swallow counter sets data to the PLLSCNF flag of the control register.

The value by which the input signal frequency is to be divided is called “N value”.

For how to set a division value (N value) in each division mode, refer to **17.6 Using PLL Frequency Synthesizer**.

(1) PLL data register and data buffer

Figure 17-4 shows the relationships between the PLL data register and data buffer.

In the direct division mode, the high-order 12 bits of the PLL data register are valid, and all 17 bits of the register are valid in the pulse swallow mode.

In the direct division mode, all 12 bits are used as a programmable counter.

In the pulse swallow mode, the high-order 12 bits are used as a programmable counter, and the low-order 5 bits are used as a swallow counter.

(2) Relationship between division value N of programmable divider and divided output frequency

The relationship between the value “N” set to the PLL data register and the signal frequency “f_N” divided and output by the programmable divider is as shown below.

For details, refer to **17.6 Using PLL Frequency Synthesizer**.

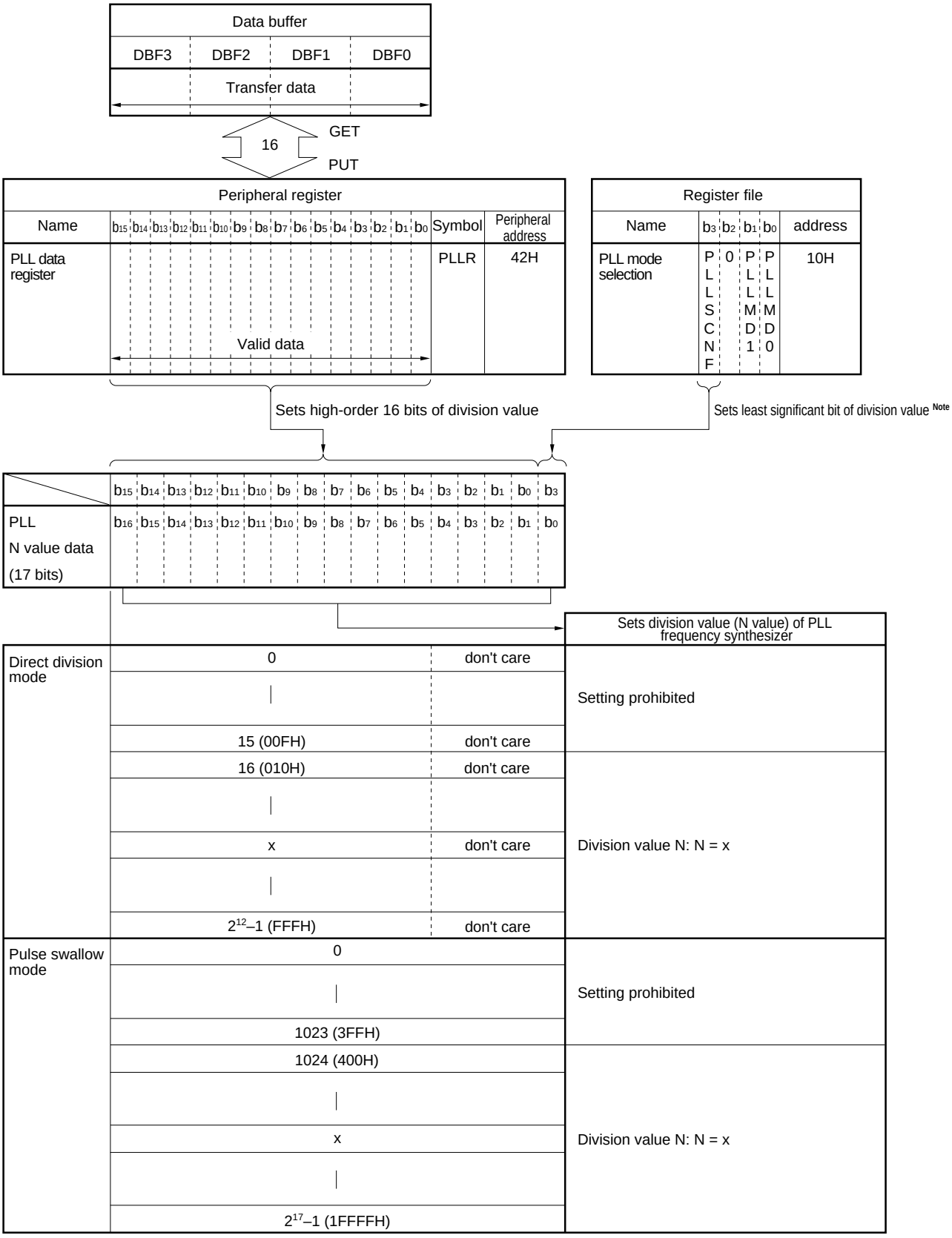
(a) Direct division mode (MF)

$$f_{IN} = \frac{f_{IN}}{N} \qquad \text{N: 12 bits}$$

(b) Pulse swallow mode (HF, VHF)

$$f_{IN} = \frac{f_{IN}}{N} \qquad \text{N: 17 bits}$$

Figure 17-4. Setting Division Value (N Value) of PLL Frequency Synthesizer



Note The value of PLLSCNF flag is transferred when a write (PUT) instruction is executed to the PLL data register (PLLR). Therefore, data must be set to the PLLSCNF flag before executing the write instruction to the PLL data register.

17.3 Reference Frequency Generator

Figure 17-5 shows the configuration of the reference frequency generator.

The reference frequency generator generates the reference frequency “fr” of the PLL frequency synthesizer by dividing the 4.5 MHz output of a crystal oscillator.

Thirteen frequencies can be selected as reference frequency fr: 1, 1.25, 2.5, 3, 5, 6.25, 9, 10, 12.5, 18, 20, 25, and 50 kHz.

The reference frequency fr is selected by the PLL reference frequency selection register.

Figure 17-6 shows the configuration and function of the PLL reference frequency selection registerion.

Figure 17-5. Configuration of Reference Frequency Generator

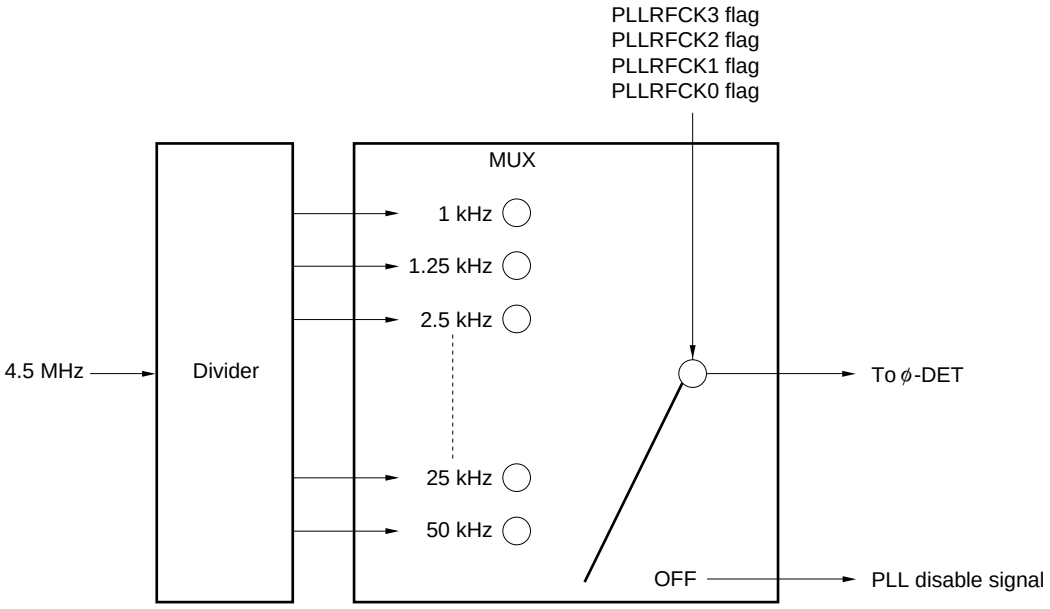


Figure 17-6. Configuration of PLL Reference Frequency Selection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
PLL reference frequency selection	PLLRFCK3	PLLRFCK2	PLLRFCK1	PLLRFCK0	11H	R/W

PLLRFCK3	PLLRFCK2	PLLRFCK1	PLLRFCK0	Sets reference frequency f_r of PLL frequency synthesizer
0	0	0	0	1.25 kHz
0	0	0	1	2.5 kHz
0	0	1	0	5 kHz
0	0	1	1	10 kHz
0	1	0	0	6.25 kHz
0	1	0	1	12.5 kHz
0	1	1	0	25 kHz
0	1	1	1	50 kHz
1	0	0	0	3 kHz
1	0	0	1	9 kHz
1	0	1	0	18 kHz
1	0	1	1	Setting prohibited
1	1	0	0	1 kHz
1	1	0	1	20 kHz
1	1	1	0	Setting prohibited
1	1	1	1	PLL disable

		PLLRFCK3	PLLRFCK2	PLLRFCK1	PLLRFCK0
At reset	Power-ON reset	1	1	1	1
	WDT&SP reset	1	1	1	1
	CE reset	1	1	1	1
Clock stop		1	1	1	1

Remark When the PLL frequency synthesizer is disabled by the PLL reference frequency selection register, the VCOH and VCOL pins are internally pulled down. The EO1 and EO0 pins are floated.

17.4 Phase Comparator (φ-DET), Charge Pump, and Unlock FF

17.4.1 Configuration of phase comparator, charge pump, and unlock FF

Figure 17-7 shows the configuration of the phase comparator, charge pump, and unlock FF.

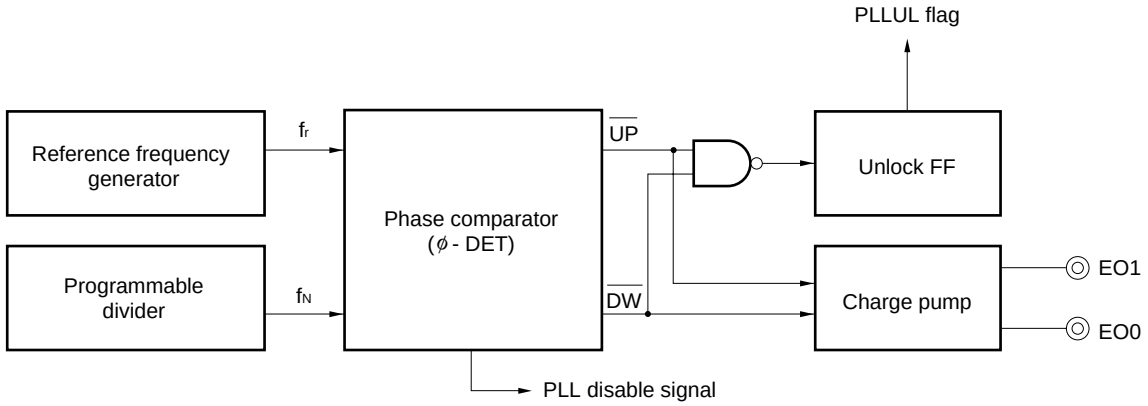
The phase comparator compares the phase of the divided frequency "f_N" output by the programmable divider with the phase of the reference frequency "fr" output by the reference frequency generator, and outputs an up ($\overline{UP}$) or down ($\overline{DW}$) request signal.

The charge pump outputs the output of the phase comparator from an error out pin (EO1 and EO0 pins).

The unlock FF detects the unlock status of the PLL frequency synthesizer.

17.4.2 through 17.4.4 describe the operations of the phase comparator, charge pump, and unlock FF.

Figure 17-7. Configuration of Phase Comparator, Charge Pump, and Unlock FF



17.4.2 Function of phase comparator

As shown in Figure 17-7, the phase comparator compares the phases of the divided frequency "f_N" output by the programmable divider and the reference frequency "fr", and outputs an up or down request signal.

If the divided frequency f_N is lower than reference frequency fr, the up request signal is output. If f_N is higher than fr, the down request signal is output.

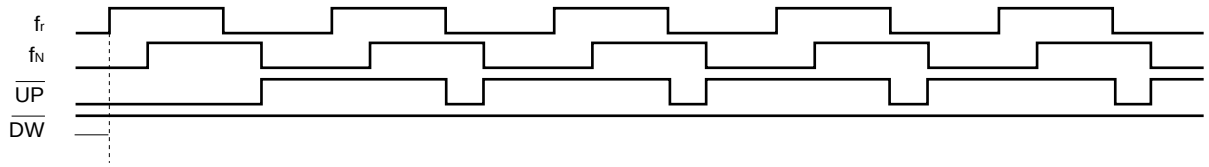
Figure 17-8 shows the relationship between reference frequency fr, divided frequency f_N, up request signal, and down request signal.

When the PLL frequency synthesizer is disabled, neither the up request nor the down request signal is output.

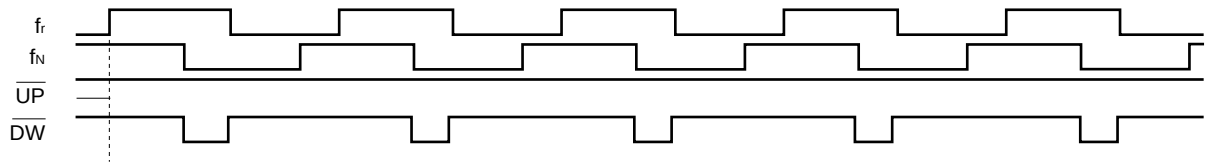
The up and down request signals are input to the charge pump and unlock FF, respectively.

Figure 17-8. Relationship between fr, f_N, $\overline{UP}$, and $\overline{DW}$

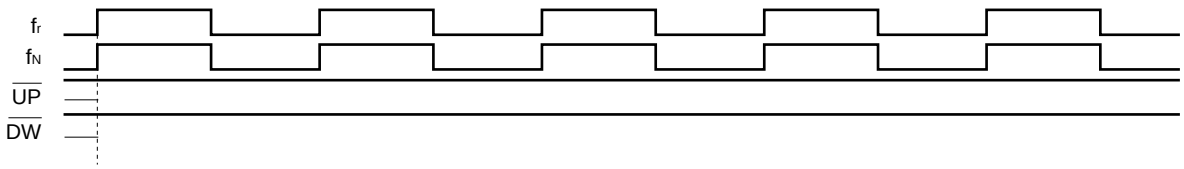
(a) If f_N lags behind fr



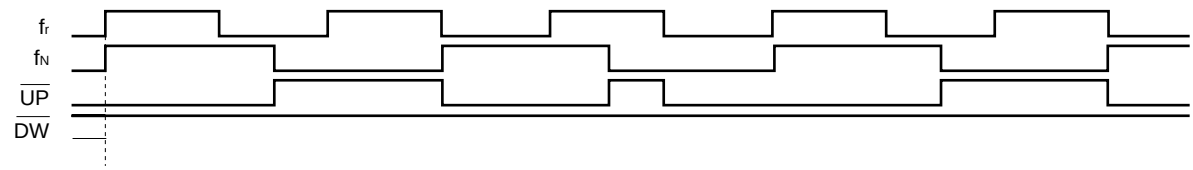
(b) If f_N leads fr



(c) If f_N and fr are in phase



(d) If f_N is lower than fr



17.4.3 Charge pump

As shown in Figure 17-7, the charge pump outputs the up request and down request signals output by the phase comparator, from the error out pins (EO1 and EO0 pins).

Therefore, the relationship between the output of the error out pins, divided frequency f_N and reference frequency fr is as follows:

Where reference frequency fr > divided frequency f_N: Low-level output
Where reference frequency fr < divided frequency f_N: High-level output
Where reference frequency fr = divided frequency f_N: Floating

17.4.4 Unlock FF

As shown in Figure 17-7, the unlock FF detects the unlock status of the PLL frequency synthesizer from the up request and down request signals of the phase comparator.

Because either the up request or down request signal is low in the unlock status, the unlock status is detected by this low-level signal.

In the unlock status, the unlock FF is set to 1.

The unlock FF is set in the cycle of the reference frequency fr selected at that time. When the contents of the PLL unlock FF register are read (by the PEEK instruction), the unlock FF is reset (Read & Reset).

Therefore, the unlock FF must be detected in a cycle longer than cycle 1/fr of the reference frequency fr.

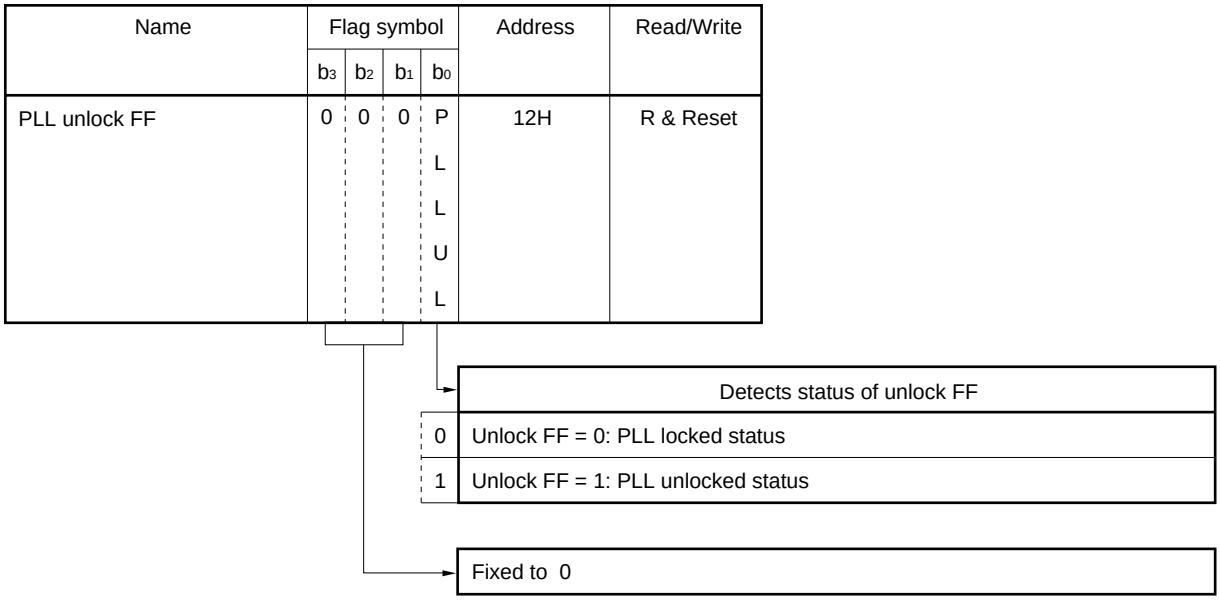
The status of the unlock FF is detected by the PLL unlock FF register. Figure 17-9 shows the configuration of the PLL unlock FF register.

Because this register is a read-only register, its contents can be read to the window register by the “PEEK” instruction.

Because the unlock FF is set in a cycle of the reference frequency fr, the contents of the PLL unlock FF register are read to the window register in a cycle longer than cycle 1/fr of the reference frequency.

The delay time of the up and down request signals of the phase comparator are fixed to 0.8 to 1.0 μs.

Figure 17-9. Configuration of PLL Unlock FF Register



At reset	Power-ON reset	0	0	0	U
	WDT&SP reset				U
	CE reset				R
Clock stop					R

U: Undefined R: Retained

17.5 PLL Disabled Status

The PLL frequency synthesizer stops (is disabled) while the CE pin is low.

Likewise, it also stops when PLL disabled status is selected by the PLL reference frequency register (RF address 11H).

Table 17-1 shows the operation of each block in the PLL disabled status.

When the VCOL and VCOH pins are disabled by the PLL mode selection register, only the VCOL and VCOH pins are internally pulled down, and the other blocks operate.

Because the PLL frequency selection register and PLL mode selection register are not initialized at CE reset (hold the previous status), these registers return to the previous status when the CE pin has gone low, the PLL frequency synthesizer has been disabled, and then CE pin has gone high.

To disable the PLL frequency synthesizer at CE reset, initialize these registers in software.

At power-ON reset, the PLL frequency synthesizer is disabled.

Table 17-1. Operation of Each Block under Each PLL Disable Condition

Condition / Each Block	CE Pin = Low Level (PLL disabled)	CE Pin = High Level	
		PLL reference frequency selection register = 1111B (PLL disabled)	PLL mode selection register = 0000B (VCOH and VCOL disabled)
VCOL, VCOH pins	Internally pulled down	Internally pulled down	Internally pulled down
Programmable divider	Division stopped	Division stopped	Operates
Reference frequency generator	Output stopped	Output stopped	Operates
Phase comparator	Output stopped	Output stopped	Operates
Charge pump	Error out pins are floated	Error out pins are floated	Operates. However, usually outputs low level because no signal is input

17.6 Using PLL Frequency Synthesizer

To control the PLL frequency synthesizer, the following data is necessary.

(1) Division mode : Direct division (MF), pulse swallow (HF, VHF)
(2) Pins used : VCOL and VCOH pins
(3) Reference frequency : fr
(4) Division value : N

17.6.1 through 17.6.3 below describe how to set PLL data in each division mode (MF, HF, and VHF).

17.6.1 Direct division mode (MF)

(1) Selecting division mode

Select the direct division mode by using the PLL mode selection register.

(2) Pins used

The VCOL pin is enabled to operate when the direct division mode is selected.

(3) Selecting reference frequency fr

Select the reference frequency by using the PLL reference frequency selection register.

(4) Calculation of division value N

Calculate N as follows:

$$N = \frac{f_{VCOL}}{fr}$$

f_{VCOL} : Input frequency of VCOL pin
fr : Reference frequency

(5) Example of setting PLL data

How to set data to receive broadcasting in the following MW band is described below.

Reception frequency : 1422 kHz (MW band)
Reference frequency : 9 kHz
Intermediate frequency : +450 kHz

Division value N is calculated as follows:

$$N = \frac{f_{VCOL}}{fr} = \frac{1422 + 450}{9} = 208 \text{ (decimal)} = 0D0H \text{ (hexadecimal)}$$

Set data to the PLL data register, PLL mode selection register, and PLL reference frequency selection register as follows:

PLL data register (PLLR)												
0	0	0	0	1	1	0	1	0	0	0	0	don't care
0				D				0				

Note 1	PLL mode selection register			PLL reference frequency selection register			
Note 2	0	0	1	1	1	0	1
	MF			9 kHz			

Notes 1. PLLSCNF flag

2. don’t care

17.6.2 Pulse swallow mode (HF)

(1) Selecting division mode

Select the pulse swallow mode by using the PLL mode selection register.

(2) Pins used

The VCOL pin is enabled to operate when the pulse swallow mode is selected.

(3) Selecting reference frequency fr

Select the reference frequency by using the PLL reference frequency selection register.

(4) Calculation of division value N

Calculate N as follows:

$$N = \frac{f_{VCOL}}{fr}$$

f_{VCOL} : Input frequency of VCOL pin
fr : Reference frequency

(5) Example of setting PLL data

How to set data to receive broadcasting in the following SW band is described below.

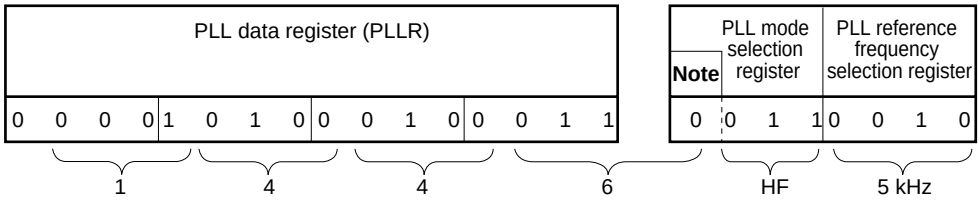
Reception frequency : 25.50 MHz (SW band)
Reference frequency : 5 kHz
Intermediate frequency: +450 kHz

Division value N is calculated as follows:

$$N = \frac{f_{VCOL}}{fr} = \frac{25500 + 450}{5} = 5190 \text{ (decimal)} = 1446H \text{ (hexadecimal)}$$

Set data to the PLL data register, PLL mode selection register, and PLL reference frequency selection register as follows:

Caution The division value N is 17 bits long when the pulse swallow mode is selected, and the least significant bit of the swallow counter is the bit 3 of the PLL mode selection register (PLLSCNF). To set "1446H" as the division value N, the value to be actually set to the PLL data register is "0A23H".



Note PLLSCNF flag

17.6.3 Pulse swallow mode (VHF)

(1) Selecting division mode

Select the pulse swallow mode by using the PLL mode selection register.

(2) Pins used

The VCOH pin is enabled to operate when the pulse swallow mode is selected.

(3) Selecting reference frequency fr

Select the reference frequency by using the PLL reference frequency selection register.

(4) Calculation of division value N

Calculate N as follows:

$$N = \frac{f_{VCOH}}{fr}$$

f_{VCOH} : Input frequency of VCOH pin
fr : Reference frequency

(5) Example of setting PLL data

How to set data to receive broadcasting in the following FM band is described below.

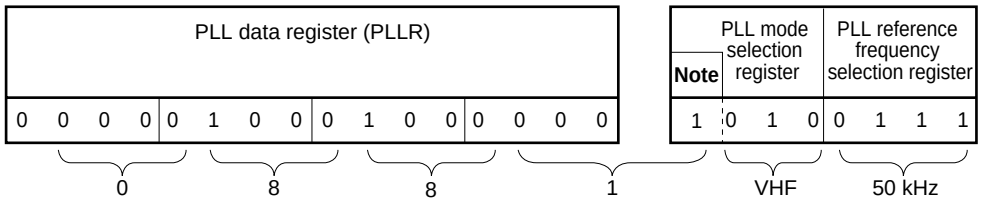
Reception frequency : 98.15 MHz (FM band)
Reference frequency : 50 kHz
Intermediate frequency : +10.7 MHz

Division value N is calculated as follows:

$$N = \frac{f_{VCOH}}{fr} = \frac{98.15 + 10.7}{0.050} = 2177 \text{ (decimal)} = 0881\text{H (hexadecimal)}$$

Set data to the PLL data register, PLL mode selection register, and PLL reference frequency selection register as follows:

Caution The division value N is 17 bits long when the pulse swallow mode is selected, and the least significant bit of the swallow counter is the bit 3 of the PLL mode selection register (PLLSCNF). To set "0881H" as the division value N, the value to be actually set to the PLL data register is "0440H".



Note PLLSCNF flag

Note that data must be set to the PLLSCNF flag before a write (PUT) instruction is executed to the PLL data register (PLLR).

Example

```
SET1    PLLSCNF
MOV     DBF0, #0
MOV     DBF1, #4
MOV     DBF2, #4
PUT     PLLR, DBF
```

17.7 Status at Reset

17.7.1 At power-ON reset

The PLL frequency synthesizer is disabled because the PLL reference frequency selection register is initialized to 1111B.

17.7.2 At WDT&SP reset

The PLL frequency synthesizer is disabled because the PLL reference frequency selection register is initialized to 1111B.

17.7.3 On execution of clock stop instruction

The PLL frequency synthesizer is disabled because the PLL reference frequency selection register is initialized to 1111B.

17.7.4 At CE reset

The PLL frequency synthesizer is disabled because the PLL reference frequency selection register is initialized to 1111B.

17.7.5 In halt status

The set status is retained if the CE pin is high.

18. FREQUENCY COUNTER

18.1 Outline of Frequency Counter

Figure 18-1 outlines the frequency counter.

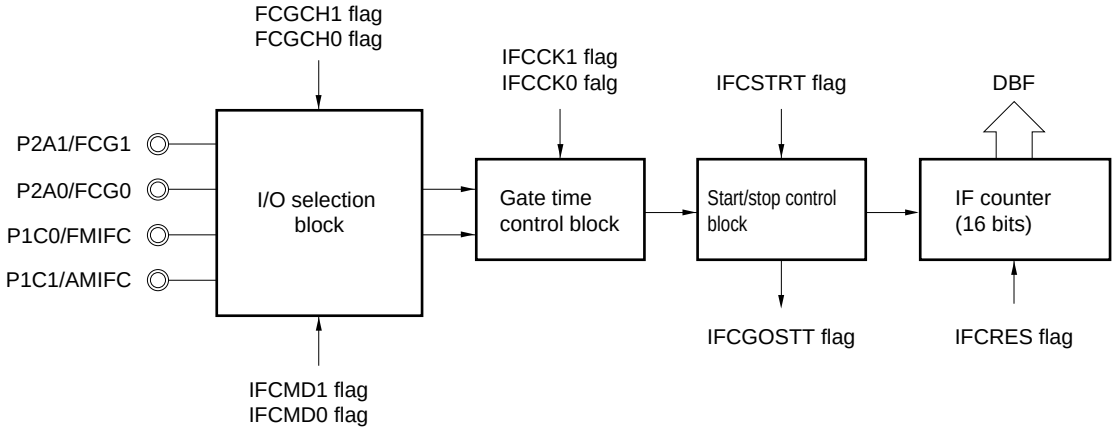
The frequency counter has an IF counter function to count the intermediate frequency (IF) of an external input signal and an external gate counter (FCG: Frequency Counter for external Gate signal) to detect the pulse width of an external input signal.

The IF counter function counts the frequency input to the P1C0/FMIFC or P1C1/AMIFC pin at fixed intervals (1 ms, 4 ms, 8 ms, or open) by using a 16-bit counter.

The external gate counter function counts the frequency of the internal clock (1 kHz, 100 kHz, 900 kHz) from the rising to the falling of the signal input to the P2A1/FCG1 or P2A0/FCG0 pin.

The IF counter and external gate counter functions cannot be used at the same time.

Figure 18-1. Outline of Frequency Counter



Remarks 1. FCGCH1 and FCGCH0 (bits 1 and 0 of FCG channel selection register: refer to **Figure 18-4**) select the pin used for the external gate counter function.

2. IFCMD1 and IFCMD0 (bits 3 and 2 of IF counter mode selection register: refer to **Figure 18-3**) select the IF counter or external gate counter function.

3. IFCCK1 and IFCCK0 (bits 1 and 0 of IF counter mode selection register: refer to **Figure 18-3**) select the gate time of the IF counter function and the reference frequency of the external gate counter function.

4. IFCSTRT (bit 1 of IF counter control register: refer to **Figure 18-6**) control starting of the IF counter and external gate counter functions.

5. IFCGOSTT (bit 0 of IF counter gate status detection register: refer to **Figure 18-7**) detects opening/closing the gate of the IF counter function.

6. IFCRES (bit 0 of IF counter control register: refer to **Figure 18-6**) reset the count value of the IF counter.

18.2 Input/Output Selection Block and Gate Time Control Block

Figure 18-2 shows the configuration of the input/output selection block and gate time control block.

The input/output selection block consists of an IF counter input selection block and FCG I/O selection block.

The IF counter input selection block selects whether the frequency counter is used as an IF counter or an external gate counter, by using the IF counter mode register. When the frequency counter is used as the IF counter, either P1C0/FMIFC or P1C1/AMIFC pin and a count mode are selected. The pin not used for the IF counter is used as a general-purpose input port pin.

The FCG I/O selection block selects the P2A1/FCG1 or P2A0/FCG0 pin by using the FCG channel selection register, when the frequency counter is used as the external gate counter. The pin not used is used as a general-purpose I/O port pin.

When using the frequency counter as the external gate counter, the pin to be used must be set in the input mode by using the port 2A bit I/O selection register. This is because the pin is set in the general-purpose output port mode if it is set in the output mode even if the external gate counter function is selected by the IF counter mode selection register and FCG channel selection register.

The gate time control block selects gate time by using the IF counter mode selection register when the frequency counter is used as the IF counter, or a count frequency when the frequency counter is used as the external gate counter.

Figure 18-3 shows the configuration of the IF counter mode selection register.

Figure 18-4 shows the configuration of the FCG channel selection register.

Figure 18-2. Configuration of I/O Selection Block and Gate Time Control Block

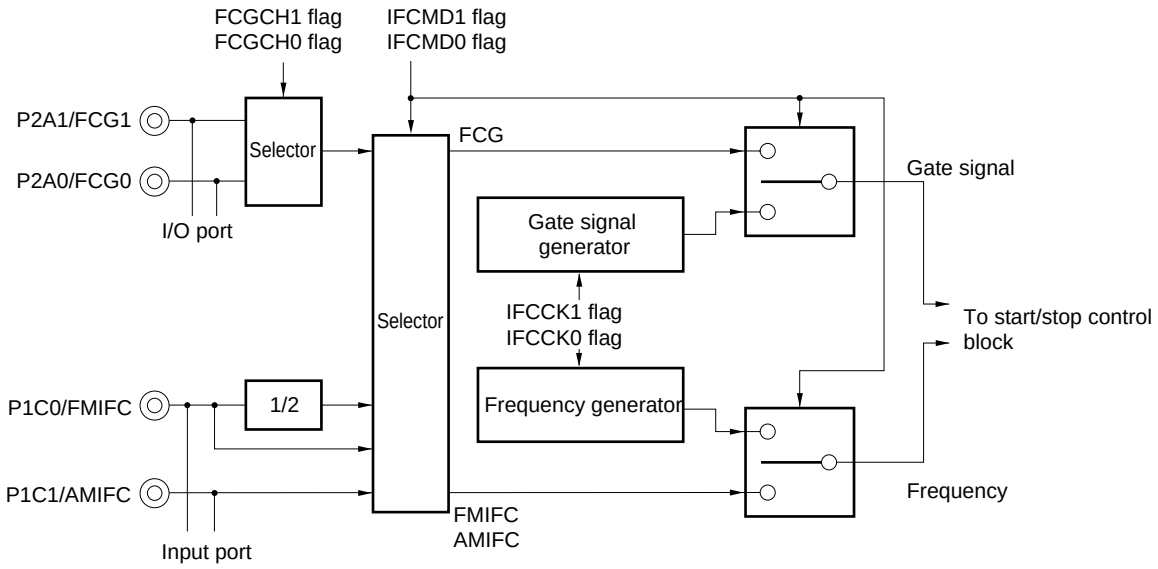


Figure 18-3. Configuration of IF Counter Mode Selection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
IF counter mode selection	IFCMD1	IFCMD0	IFCCK1	IFCCK0	22H	R/W

Selects gate time of IF counter and reference frequency of external gate counter

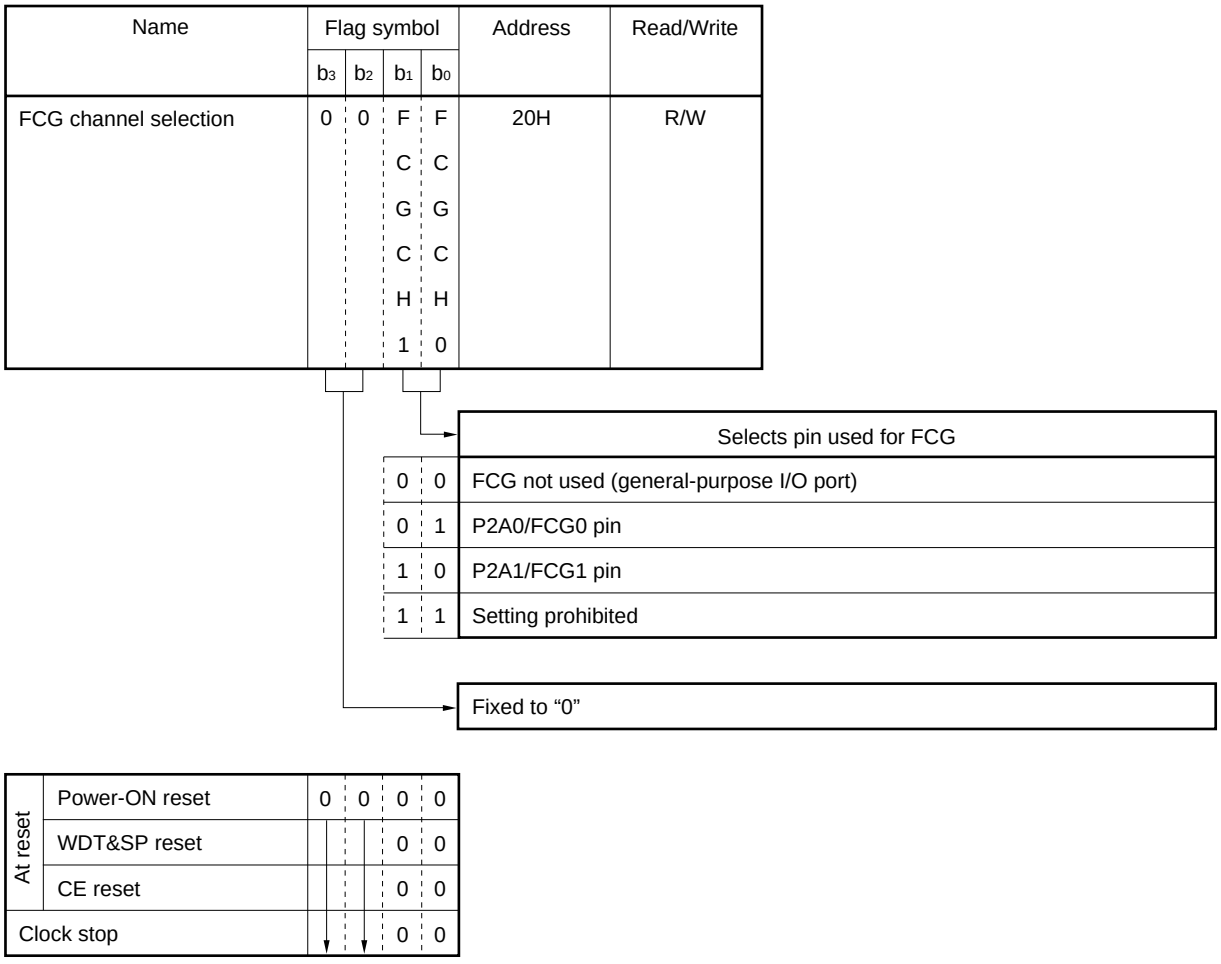
IFCCK1	IFCCK0	Gate time of IF counter	Reference frequency of external gate counter
0	0	1 ms	1 kHz
0	1	4 ms	100 kHz
1	0	8 ms	900 kHz
1	1	Open	Setting prohibited

IFCMD1	IFCMD0	Selects function of IF counter or external gate counter
0	0	External gate counter (FCG)
0	1	IF counter (AMIFC pin, AMIF count mode)
1	0	IF counter (FMIFC pin, FMIF count mode, 1/2 division)
1	1	IF counter (FMIFC pin, AMIF count mode)

		IFCMD1	IFCMD0	IFCCK1	IFCCK0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset	0	0	0	0
	CE reset	0	0	0	0
Clock stop		0	0	0	0

Caution The IF counter and external gate counter functions cannot be used at the same time.

Figure 18-4. Configuration of FCG Channel Selection Register



18.3 Start/Stop Control Block and IF Counter

18.3.1 Configuration of start/stop control block and IF counter

Figure 18-5 shows the configuration of the start/stop control block and IF counter.

The start/stop control block starts the frequency counter or detects the end of counting.

The counter is started by the IF counter control register.

The end of counting is detected by the IF counter gate status detection register. When the external gate counter function is used, however, the end of counting cannot be detected by the IF counter gate status detection register.

Figure 18-6 shows the configuration of the IF counter control register.

Figure 18-7 shows the configuration of the IF counter gate status detection register.

18.3.2 and 18.3.3 describe the gate operation when the IF counter function is selected and that when the external gate counter function is selected.

The IF counter is a 16-bit binary counter that counts up the input frequency when the IF counter function or external gate counter function is selected.

When the IF counter function is selected, the frequency input to a selected pin is counted while the gate is opened by an internal gate signal. The frequency count is counted without alteration in the AMIF count mode. In the FMIF counter mode, however, the frequency input to the pin is halved and counted.

When the external gate counter function is selected, the internal frequency is counted while the gate is opened by the signal input to the pin.

When the IF counter counts up to FFFFH, it remains at FFFFH until reset.

The count value is read by the IF counter data register (IFC) via data buffer.

The count value is reset by the IF counter control register.

Figure 18-8 shows the configuration of the IF counter data register.

Figure 18-5. Configuration of Start/Stop Control Block and IF Counter

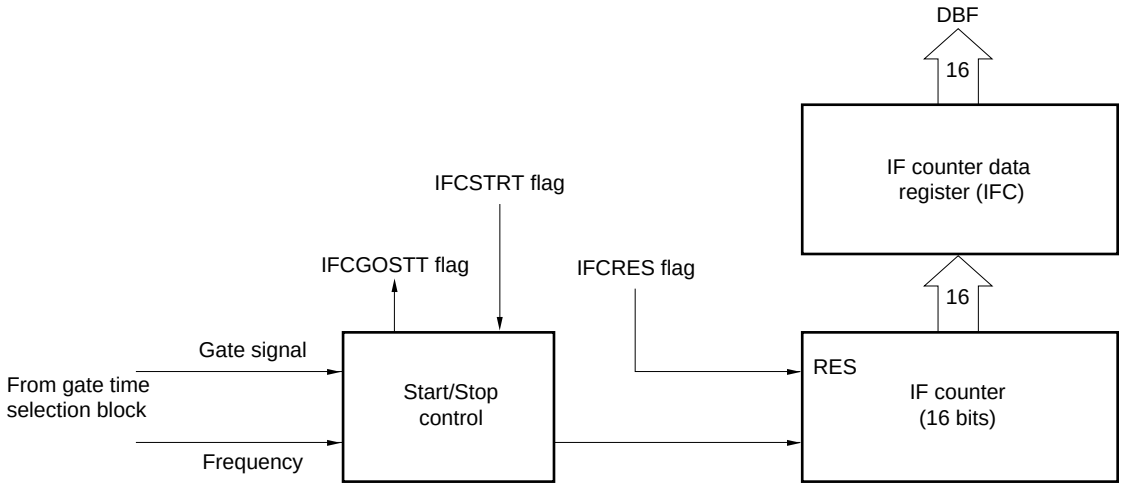


Figure 18-6. Configuration of IF Counter Control Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
IF counter control	0	0	IFCSTRT	IFCRES	23H	W

IFCRES — Resets data of IF counter and external gate counter

0	Nothing is affected
1	Resets counter

IFCSTRT — Start IF counter and external gate counter

0	Nothing is affected
1	Resets counter

Fixed to “0”

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset			0	0
	CE reset			0	0
Clock stop				0	0

Figure 18-7. Configuration of IF Counter Gate Status Detection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
IF counter gate status detection	0	0	0	IFCGOSTT	21H	R

IFCGOSTT	When IF counter function is selected	When external gate counter function is selected
0	Sets IFCSTRT flag to “1” and is set to 1 until gate is closed	Sets IFCSTRT flag to “1” and is set to 1 while gate is open, regardless of input of P2A0/FCG0 and P2A1/FCG1 pins
1		

Fixed to “0”

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset				0
	CE reset				0
Clock stop					0

Cautions 1. Do not read the contents of the IF counter data register (IFC) to the data buffer while the IFCGOSTT flag is set to 1.

2. The gate of the external gate counter cannot be opened or closed by the IFCGOSTT flag. Use the IFCSTRT flag to open or close the gate.

18.3.2 Operation of gate when IF counter function is selected

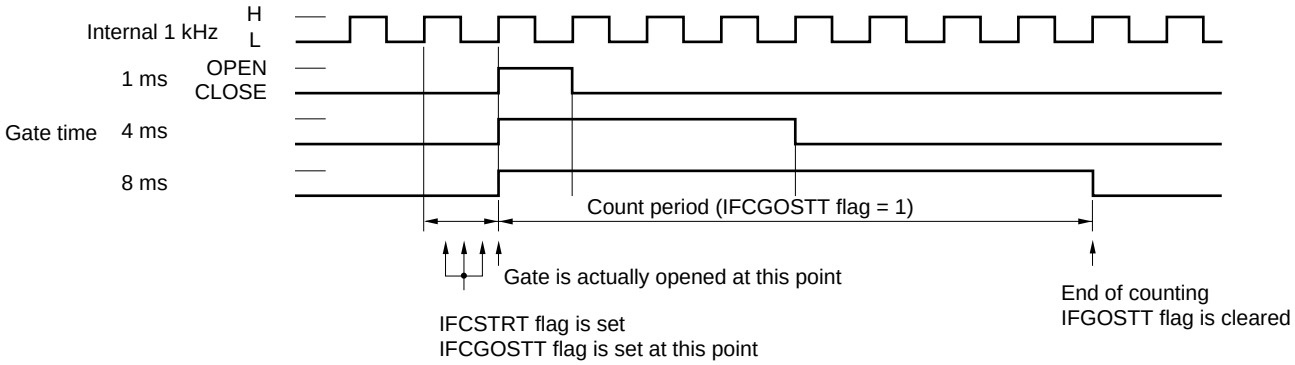
(1) When gate time of 1, 4, or 8 ms is selected

The gate is opened for 1, 4, or 8 ms from the rising of the internal 1-kHz signal after the IFCSTRT flag has been set to 1, as illustrated below.

While this gate is open, the frequency input from a selected pin is counted by a 16-bit counter.

When the gate is closed, the IFCG flag is cleared to 0.

The IFCGOSTT flag is automatically set to 1 when the IFCSTRT flag is set.



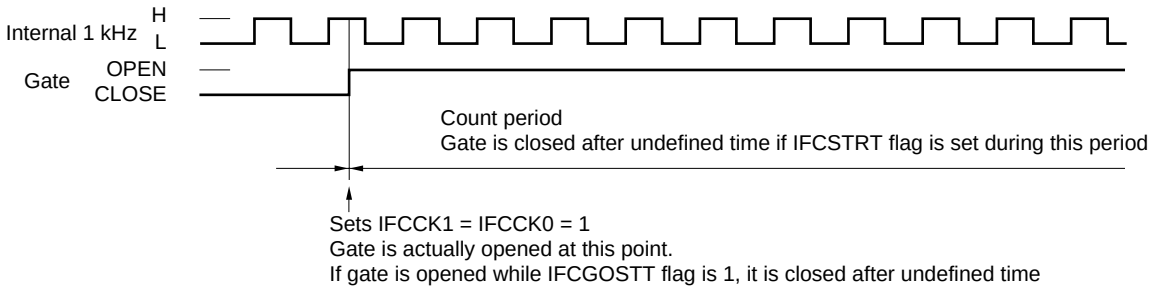
(2) When gate is open

If opening of the gate is selected by the IFCCK1 and IFCCK0 flags, the gate is opened as soon as its opening has been selected, as illustrated below.

If the counter is started by using the IFCSTRT flag while the gate is open, the gate is closed after undefined time.

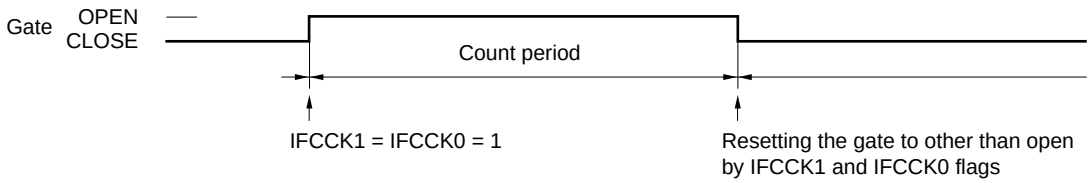
To open the gate, therefore, do not set the IFCSTRT flag to 1.

However, the counter can be reset by the IFCRES flag.



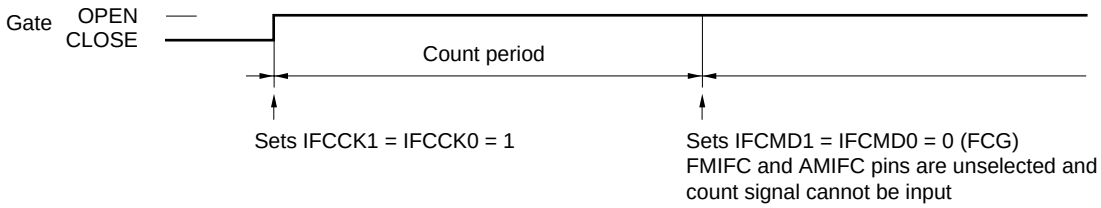
The gate is opened or closed in the following two ways when opening the gate is selected as the gate time.

(a) Resetting the gate to other than open by using IFCCK1 and IFCCK0 flags



(b) Unselect pin used by using IFCMD1 and IFCMD0 flags

In this way, the gate remains open, and counting is stopped by disabling input from the pin.



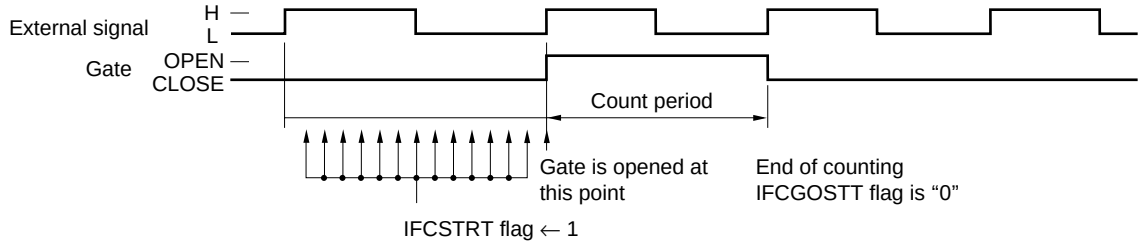
18.3.3 Gate operation when external gate counter function is selected

The gate is opened from the rising to the next rising of the signal input to a selected pin after the IFCSTRT flag has been set to 1, as illustrated below.

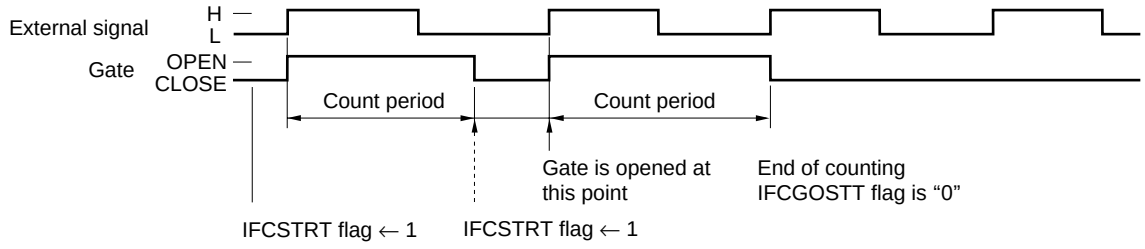
While the gate is open, the internal frequency (1 kHz, 100 kHz, 900 kHz) is counted by a 16-bit counter.

The IFCGOSTT flag is set to 1 from the rising to the next rising of the external signal after the IFCSTRT flag has been set.

In other words, the opening or closing of the gate cannot be detected by the IFCG flag when the external gate counter function is selected.



If reset and started while gate is open



18.3.4 Function and operation of 16-bit counter

The 16-bit counter counts up the frequency input within selected gate time.

The 16-bit counter can be reset by writing "1" to the IFCRES flag of the IF counter control register.

Once the 16-bit counter has counted up to FFFFH, it remains at FFFFH until it is reset.

The following paragraphs (1) and (2) describe the operations when the IF counter function is selected and when the external gate counter function is selected.

The value of the IF counter data register is read via data buffer.

Figure 18-8 shows the configuration and function of the IF counter data register.

(1) When IF counter is selected

The frequency input to the P1C0/FMIFC or P1C1/AMIFC pin is counted while the gate is open. Note, however, that the frequency input to the P1C0/FMIFC is divided by two and counted.

The relationship between count value "x (decimal)" and input frequencies (f_{FMIFC} and f_{AMIFC}) is shown below.

- FMIFC

$$f_{FMIFC} = \frac{x}{t_{GATE}} \times 2 \text{ (kHz)}$$ t_{GATE}: gate time (1 ms, 4 ms, 8 ms)

- AMIFC

$$f_{AMIFC} = \frac{x}{t_{GATE}} \text{ (kHz)}$$ t_{GATE}: gate time (1 ms, 4 ms, 8 ms)

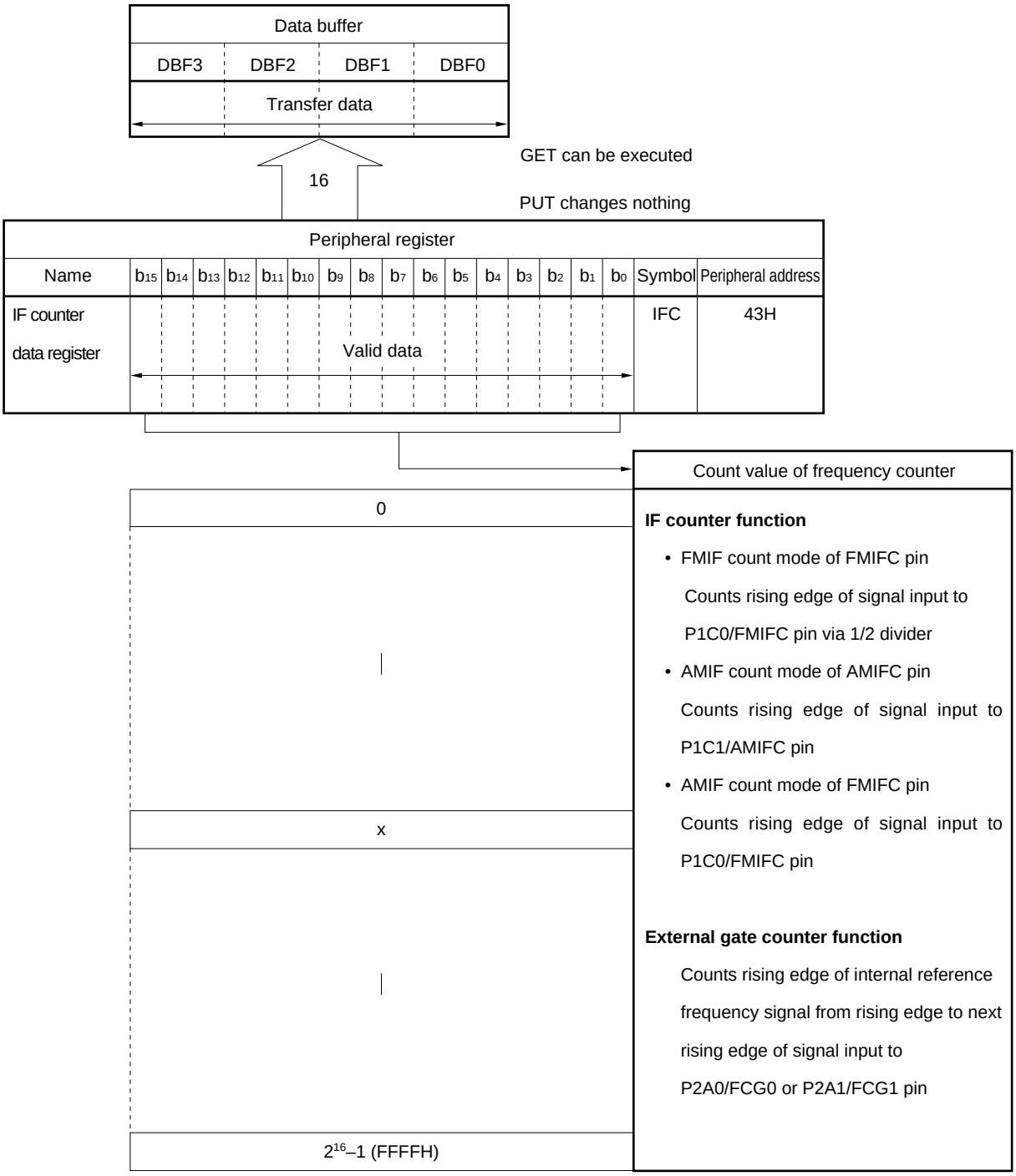
(2) When external gate counter (FCG) is selected

The internal frequency is counted while the gate is opened by the signal input to the P2A1/FCG1 or P2A0/FCG0 pin.

The relationship between the count value "x (decimal)" and the gate width t_{GATE} of the input signal is shown below.

$$t_{GATE} = \frac{x}{fr} \text{ (ms)}$$ fr: internal frequency (1, 100, 900 kHz)

Figure 18-8. Configuration of IF Counter Data Register



Once the IF counter data register has counted up to FFFFH, it remains at FFFFH until the counter is reset.

18.4 Using IF Counter

The following sections 18.4.1 through 18.4.3 describe how to use the hardware of the IF counter, a program example, and count error.

18.4.1 Using hardware of IF counter

Figure 18-9 shows the block diagram when the P1C0/FMIFC and P1C1/AMIFC pins.

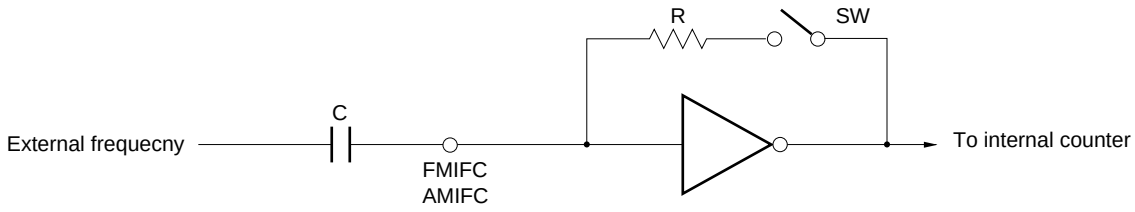
As shown in the figure, the IF counter uses an input pin with an AC amplifier, the DC component of the input signal must be cut with a capacitor.

When the P1C0/FMIFC or P1C1/AMIFC pin is selected for the IF counter function, switch SW turns ON, and the voltage level on each pin reaches about $1/2V_{DD}$.

If the voltage has not risen to a sufficient intermediate level at this time, the IF counter does not operate normally because the AC amplifier is not in the normal operating range.

Therefore, make sure that a sufficient wait time elapses after each pin has been specified to be used for the IF counter until counting is started.

Figure 18-9. IF Count Function Block Diagram of Each Pin



18.4.2 Program example of IF counter

A program example of the IF counter is shown below.

As shown in this example, make sure that a wait time elapses after an instruction that selects the P1C0/FMIFC or P1C1/AMIFC pin for the IF counter function has been executed until counting is started.

This is because, as described in 18.4.1, the internal AC amplifier does not operate normally immediately after a pin has been selected for the IF counter.

Example To count the frequency input to the P1C0/FMIFC pin (FMIF count mode) (gate time: 8 ms)

```
        INITFLG IFCMD1, NOT IFCMD0, IFCCK1, NOT IFCCK0
                                ; Selects FMIFC pin (FMIF count mode), and sets gate time to 8 ms

        [ Wait ]                ; Internal AC amplifier stabilization time

        SET1    IFCRES          ; Resets counter
        SET1    IFCSTRT         ; Starts counting
LOOP:
        SKT1    IFCG0STT        ; Detects opening or closing of gate
        BR      READ            ; Branches to READ: if gate is closed

        [ Processing A ]

        BR      LOOP            ; Do not read data of IF counter with this processing A
READ:
        GET     DBF, IFC        ; Reads value of IF counter data register to data buffer
```

18.4.3 Error of IF counter

The errors of the IF counter include a gate time error and a count error. The following paragraphs (1) and (2) describe each of these errors.

(1) Gate time error

The gate time of the IF counter is created by dividing the 4.5-MHz clock. Therefore, if the system clock is shifted from 4.5 MHz by “+x” ppm, the gate time is shifted by “–x” ppm.

(2) Count error

The IF counter counts frequency by the rising edge of the input signal.

If a high level is input to the pin when the gate is open, therefore, one excess pulse is counted.

If the gate is closed, however, a count error due to the status of the pin does not occur.

Therefore, the count error is “+1, –0”.

18.5 Using External Gate Counter

18.5.1 Program example of external gate counter

A program example of the external gate counter is shown below.

Example To use the P2A0/FCG0 pin as external gate input pin

```
        INITFLG NOT IFCMD1, NOT IFCMD0, IFCCK1, NOT IFCCK0
                                        ; Selects external gate counter function and sets gate
                                          time to 8 ms
        INITFLG  NOT FCGCH1, FCGCH0     ; Selects FCG0 pin as external gate input pin
        SET1     IFCRES                 ; Resets counter
        SET1     IFCSTRT                ; Starts counting
LOOP:
        SKF1     IFCGOSTT               ; Detects opening or closing of gate
        BR       READ                   ; Branches to READ: if gate is closed

        Processing A                    ; Do not read data of IF counter with this processing A

        BR       LOOP
READ:
        GET      DBF, IFC               ; Reads value of IF counter data register to data buffer
```

18.5.2 Error of external gate counter

The errors of the external gate counter include an internal frequency error and a count error. The following paragraphs (1) and (2) describe each of these errors.

(1) Internal frequency error

The internal frequency of the external gate counter is created by dividing the 4.5-MHz clock. Therefore, if the system clock is shifted from 4.5 MHz by "+x" ppm, the gate time is shifted by "–x" ppm.

(2) Count error

The external gate counter counts the frequency by the rising edge of the internal frequency.

If the internal frequency is low when the gate is opened (when the signal input to the pin rises), one excess pulse is counted.

If the gate is closed (when the signal rises next time), the excess pulse is not counted due to the count level of the internal frequency.

Therefore, the count error is "+1, –0".

18.6 Status at Reset

18.6.1 At power-ON reset

The P1C0/FMIFC, P1C1/AMIFC, P2A0/FCG0, and P2A1/FCG1 pins are set in the general-purpose input port mode.

18.6.2 At WDT&SP reset

The P1C0/FMIFC, P1C1/AMIFC, P2A0/FCG0, and P2A1/FCG1 pins are set in the general-purpose input port mode.

18.6.3 On execution of clock stop instruction

The P1C0/FMIFC and P1C1/AMIFC pins are set in the general-purpose input port mode.

The P2A0/FCG0 and P2A1/FCG1 pins are set in the general-purpose I/O port mode, and retain the previous input or output status.

18.6.4 At CE reset

The P1C0/FMIFC and P1C1/AMIFC pins are set in the general-purpose input port mode.

The P2A0/FCG0 and P2A1/FCG1 pins are set in the general-purpose I/O port mode, and retain the previous input or output status.

18.6.5 In halt status

The P1C0/FMIFC, P1C1/AMIFC, P2A0/FCG0, and P2A1/FCG1 pins retain the status immediately before the halt mode is set.

19. BEEP

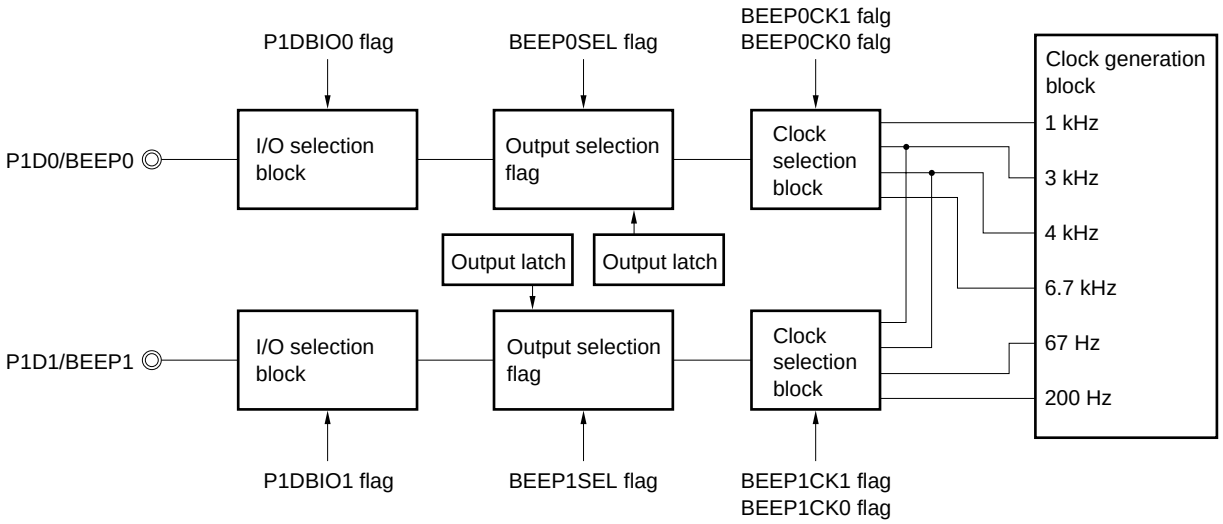
19.1 Outline of BEEP

Figure 19-1 outlines BEEP.

BEEP outputs a clock of 1, 3, 4, or 6.7 kHz from the P1D0/BEEP0 pin, and a clock of 4 kHz, 3 kHz, 200 Hz, or 67 Hz from the P1D1/BEEP1 pin.

The duty factor of the BEEP output is 50%.

Figure 19-1. Outline of BEEP



Remarks 1. BEEP0CK1 and BEEP0CK0 (bits 1 and 0 of BEEP clock selection register: refer to **Figure 19-4**) select the output frequency of BEEP0.

2. BEEP1CK1 and BEEP1CK0 (bits 3 and 2 of BEEP clock selection register: refer to **Figure 19-4**) select the output frequency of BEEP1.

3. BEEP1SEL and BEEP0SEL (bits 1 and 0 of BEEP/general-purpose port pin function selection register: refer to **Figure 19-3**) select general-purpose I/O port and BEEP.

4. P1DBIO1 and P1DBIO0 (bits 1 and 0 of port 1D bit I/O selection register: refer to **Figure 19-2**) select the input or output mode of the port.

19.2 I/O Selection Block and Output Selection Block

The I/O selection block selects the input or output mode of the P1D0/BEEP0 and P1D1/BEEP1 pins by using the port 1D bit I/O selection register. Set the pin to be used as a BEEP pin in the output mode.

The output selection block sets the P1D0/BEEP0 and P1D1/BEEP1 pins in the general-purpose output port mode or BEEP output mode by using the BEEP/general-purpose port pin function selection register.

Figure 19-2 shows the configuration of the port 1D bit I/O selection register.

Figure 19-3 shows the configuration of the BEEP/general-purpose port pin function selection registerion.

Figure 19-2. Configuration of Port 1D Bit I/O Selection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Port 1D bit I/O selection	P1DBIO3	P1DBIO2	P1DBIO1	P1DBIO0	(BANK15) 6CH	R/W

At reset		b_3	b_2	b_1	b_0
	Power-ON reset	0	0	0	0
	WDT&SP reset	0	0	0	0
	CE reset	Retained			
Clock stop		Retained			

P1D0/BEEP0	Selects input or output port mode
0	Sets P1D0/BEEP0 pin in input mode
1	Sets P1D0/BEEP0 pin in output mode.

P1D1/BEEP1	Selects input or output port mode
0	Sets P1D1/BEEP1 pin in input mode
1	Sets P1D1/BEEP1 pin in output mode

P1D2	Selects input or output port mode
0	Sets P1D2 pin in input mode
1	Sets P1D2 pin in output mode

P1D3	Selects input or output port mode
0	Sets P1D3 pin in input mode
1	Sets P1D3 pin in output mode

Figure 19-3. Configuration of BEEP/General-Purpose Port Pin Function Selection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
BEEP/general-purpose port pin function selection	0	0	BEEP1SEL	BEEP0SEL	13H	R/W

BEEP0SEL	Selects general-purpose I/O port or BEEP
0	Uses P1D0/BEEP0 pin as general-purpose I/O port
1	Uses P1D0/BEEP0 pin for BEEP

BEEP1SEL	Selects general-purpose I/O port or BEEP
0	Uses P1D1/BEEP1 pin as general-purpose I/O port
1	Uses P1D1/BEEP1 pin for BEEP

Fixed to 0

At reset		b_3	b_2	b_1	b_0
	Power-ON reset	0	0	0	0
	WDT&SP reset			0	0
	CE reset			0	0
Clock stop				0	0

19.3 Clock Selection Block and Clock Generation Block

The clock selection block selects the output frequency of BEEP1 and BEEP0 by using the BEEP clock selection register.

The clock generation block generates the clock to be output to BEEP0 and BEEP1.

The clock frequency generated is 1 kHz, 3 kHz, 4 kHz, 6.7 kHz, 67 Hz, or 200 Hz.

Figure 19-4. Configuration of BEEP Clock Selection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
BEEP clock selection	BEEP1CK1	BEEP1CK0	BEEP0CK1	BEEP0CK0	14H	R/W

BEEP0CK1	BEEP0CK0	Sets output frequency of BEEP0
0	0	1 kHz
0	1	3 kHz
1	0	4 kHz
1	1	6.7 kHz

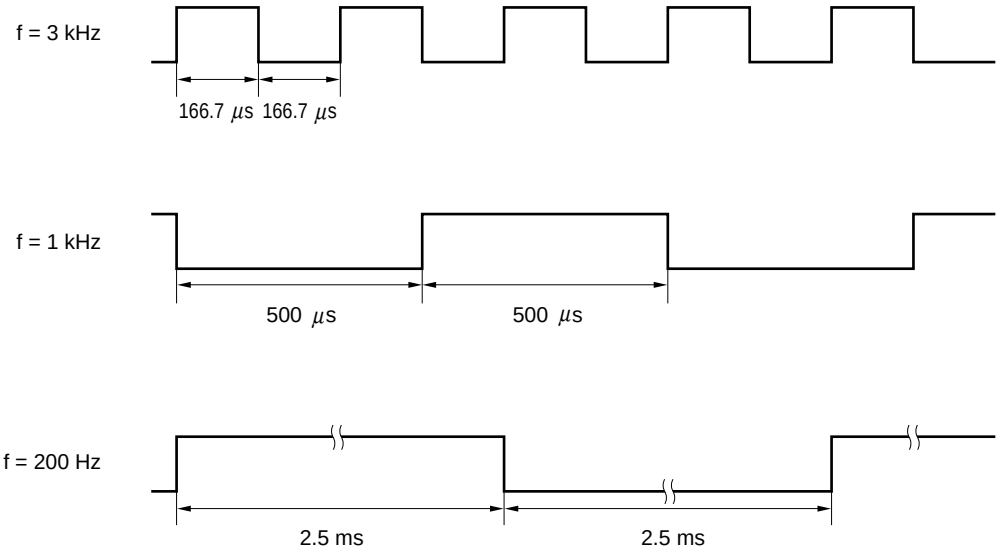
BEEP1CK1	BEEP1CK0	Sets output frequency of BEEP1
0	0	4 kHz
0	1	3 kHz
1	0	200 Hz
1	1	67 Hz

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset	0	0	0	0
	CE reset	0	0	0	0
Clock stop		0	0	0	0

19.4 Output Waveform of BEEP

The duty factor of the BEEP output waveform is 50%.

Example



f: output frequency of BEEP

19.5 Status at Reset

19.5.1 At power-ON reset

The P1D0/BEEP0 and P1D1/BEEP1 pins are set in the general-purpose input port mode.

19.5.2 At WDT&SP reset

The P1D0/BEEP0 and P1D1/BEEP1 pins are set in the general-purpose input port mode.

19.5.3 On execution of clock stop instruction

The P1D0/BEEP0 and P1D1/BEEP1 pins are set in the general-purpose I/O port mode, and retain the previous input or output status.

19.5.4 At CE reset

The P1D0/BEEP0 and P1D1/BEEP1 pins are set in the general-purpose I/O port mode, and retain the previous input or output status.

19.5.5 In halt status

The previous status is retained.

20. STANDBY

The standby function is used to reduce the current consumption of the device while the device is backed up.

20.1 Outline of Standby Function

Figure 20-1 outlines the standby block.

The standby function reduces the current consumption of the device by partly or totally stopping the device operation.

The following three types of standby functions are available for selection as the application requires.

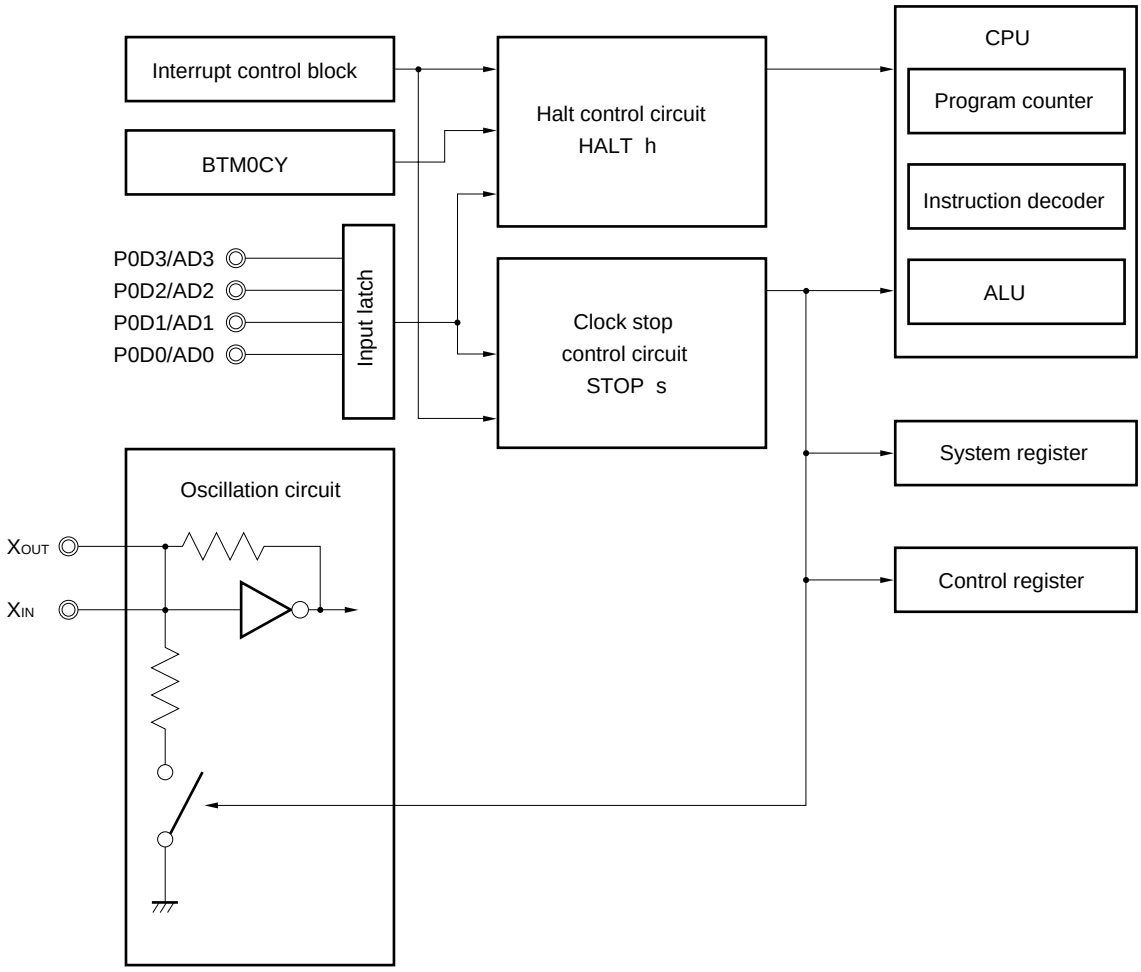
- Halt function
- Clock stop function
- Device operation control function by CE pin

The halt function reduces the current consumption of the device by stopping the CPU operation by using a dedicated instruction “HALT h”.

The clock stop function reduces the current consumption of the device by stopping the oscillation of the oscillation circuit by using a dedicated instruction “STOP s”.

The CE pin can be said to be one of the standby functions because it can be used to control the operation of the PLL frequency synthesizer and to reset the device.

Figure 20-1. Outline of Standby Block



20.2 Halt Function

20.2.1 Outline of halt function

The halt function stops the operating clock of the CPU by executing the “HALT h” instruction.

When this instruction is executed, the program is stopped until the halt status is later released. Therefore, the current consumption of the device in the halt status is reduced by the operating current of the CPU.

The halt status is released by using basic timer 0 carry FF, interrupt, or port input (P0D).

The release condition is specified by operand “h” of the “HALT h” instruction.

20.2.2 Halt status

In the halt status, all the operations of the CPU are stopped. In other words, execution of the program is stopped at the “HALT h” instruction. However, the peripheral hardware units continue the operation specified before execution of the “HALT h” instruction.

For the operation of each peripheral hardware unit, refer to **20.4 Device Operation in Halt and Clock Stop Status**.

20.2.3 Halt release condition

Figure 20-2 shows the halt release condition.

The halt release condition is specified by 4-bit data specified by operand “h” of the “HALT h” instruction.

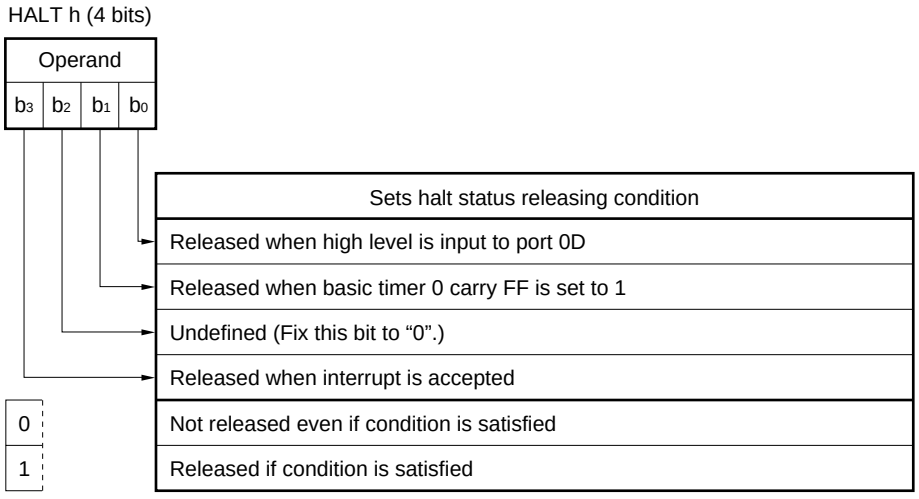
The halt status is released when the condition specified by “1” in operand “h”.

When the halt status is released, program execution is started from the instruction after the “HALT h” instruction. If the halt status is released by an interrupt, the operation to be performed after the halt status has been released differs depending on whether the interrupts are enabled (EI status) or disabled (DI status) when an interrupt source (IRQxxx = 1) is issued with the interrupt (IPxxx = 1) enabled.

If two or more releasing conditions are specified, the halt status is released when one of the specified condition is satisfied.

If 0000B is set as halt release condition “h”, no releasing condition is set. If the device is reset (by means of power-ON reset, WDT&SP reset, or CE reset) at this time, the halt status is released.

Figure 20-2. Halt Release Condition



20.2.4 Releasing halt by input port (P0D)

The halt releasing condition using an input port is specified by the “HALT 0001B” instruction.

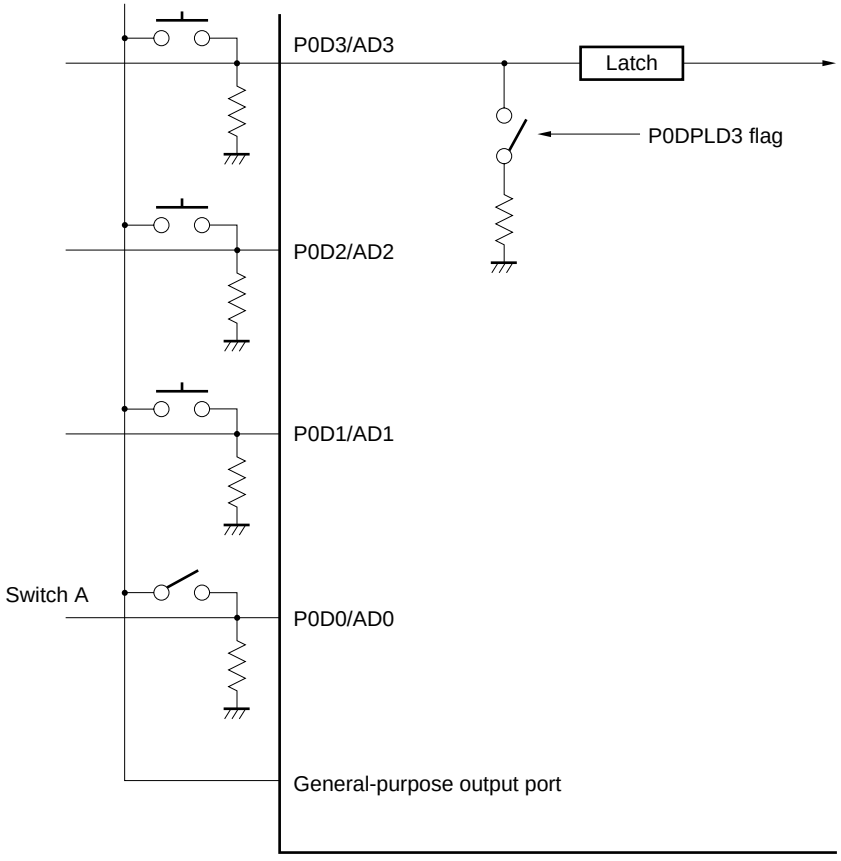
When the halt releasing condition using an input port is specified, the halt status is released if a high level is input to one of the P0D0 through P0D3 pins.

The P0D0 through P0D3 pins are multiplexed with the A/D converter input pins AD0 through AD3, and the halt status is not released when these pins are used as A/D converter input pins.

An example is given below.

- **To use as key matrix**

  The P0D0 through P0D3 pins are general-purpose input port pins which can be set in the input or output mode in 1-bit units and can be connected to an internal pull-down resistor. If connection of the internal pull-down resistor is specified by software, an external resistor can be eliminated as shown in this example (the internal-pull down resistor is connected at power-ON reset).



The “HALT 0001B” instruction is executed after the general-purpose output ports for key source signal are made high. Note that if an alternate switch is used as shown by switch A in the above figure, the halt status is released immediately because a high level is input to the P0D0/AD0 pin while switch A is closed.

20.2.5 Releasing halt status by basic timer 0 carry FF

Releasing the halt status by using the basic timer 0 carry FF is specified by the “HALT 0010B” instruction.

When releasing the halt status by the basic timer 0 carry FF is specified, the halt status is released as soon as the basic timer 0 carry FF has been set to 1.

The basic timer 0 carry FF corresponds to the BTM0CY flag on a one-to-one basis and is set at fixed time intervals (100, 50, 20, or 10 ms). Therefore, the halt status can be released at fixed time intervals.

Example To release halt status every 100 ms to execute processing A

```
HLTTMR  DAT       0010B                        ; Symbol definition
        INITFLG   NOT BTM0CK1, NOT BTM0CK0     ; Sets time interval of basic timer 0 to 100 ms
LOOP:
        HALT      HLTTMR                       ; Specifies setting of basic timer 0 carry FF as halt releasing condition
        SKT1      BTMOCY                       ; Embedded macro
        BR        LOOP                         ; Branches to LOOP if BTM0CY flag is not set

        Processing A                           ; Executes processing A if carry occurs

        BR        LOOP
```

20.2.6 Releasing halt status by interrupt

Releasing the halt status by an interrupt is specified by the “HALT 1000B” instruction.

When releasing the halt status by an interrupt is specified, the halt status is released as soon as the interrupt has been accepted.

Many interrupt sources are available as described in 12. INTERRUPTS. Which interrupt source is used to release the halt status must be specified in advance in software.

To accept an interrupt, each interrupt request must be issued from each interrupt source and each interrupt must be enabled (by setting the corresponding interrupt enable flag).

Therefore, the interrupt is not accepted even if the interrupt request is issued, and the halt status is not released.

When the halt status is released by accepting an interrupt, the program flow branches to the vector address of the interrupt.

When the RETI instruction is executed after interrupt servicing, the program flow is restored to the instruction after the HALT instruction.

If all the interrupts are disabled (DI status), the halt status is released by enabling an interrupt (IPxxx = 1) and issuing an interrupt source (IRQxxx = 1), and the flow of the program goes to the instruction after the HALT instruction.

Example Releasing halt status by timer 0 and INT0 pin interrupts

In this example, the halt status is released and processing B is executed when timer 0 interrupt is accepted. And processing A is executed when INT0 pin interrupt is accepted.
Each time the halt status has been released, processing C is executed.

```
HLTINT    DAT     1000B                ; Symbol definition
START:                                 ; Address 0000H
          BR      MAIN
;*** Interrupt vector address ***
          NOP                          ; SI03
          NOP                          ; SI02
          NOP                          ; TIMER3
          NOP                          ; TIMER2
          NOP                          ; TIMER1
          BR      INTTM0               ; Branches to timer 0 interrupt processing
          NOP                          ; INT4
          NOP                          ; INT3
          NOP                          ; INT2
          NOP                          ; INT1
          BR      INT0                 ; Branches to INT0 interrupt processing
          NOP                          ; CE DOWN EDGE
INT0:                                  ; INT0 pin interrupt vector address (000BH)

          [ Processing A ]             ; INT0 pin interrupt processing

          EI
          RETI
INTMM0:

          [ Processing B ]             ; Timer 0 interrupt processing

          EI
          RETI
MAIN:
          INITFLG   NOT TMOCK1, TM0CK0 ; Sets timer 0 count clock to 100 μs

          MOV       DBF1, #0
          MOV       DBF0, #0AH
          PUT       TM0M,DBF           ; Sets time interval of timer 0 interrupt to 1 ms
          SET2      TM0RES, TM0EN      ; Resets and starts timer 0
          SET2      IPTM0, IP0         ; Enables INT0 and timer 0 interrupts
LOOP:

          [ Processing C ]             ; Main routine processing

          EI                           ; Enables all interrupts
          HALT      HLTINT             ; Specifies releasing halt status by interrupt
    ;<1>
          BR        LOOP
```

If the INT0 pin interrupt request and timer 0 interrupt request are issued simultaneously in the halt status, processing A for the INT0 pin, which has the higher hardware priority, is executed.
After execution of processing A and when "RETI" is executed, the program branches to the "BR LOOP" instruction of <1>. However, the "BR LOOP" instruction is not executed, and timer 0 interrupt is immediately accepted.
When the "RETI" instruction is executed after processing B of timer 0 interrupt has been executed, the "BR LOOP" instruction is executed.

Caution To reset the interrupt request flag (IRQxxx) once before the halt instruction is executed, insert a NOP instruction (or one or more other instructions) between the HALT instruction and the instruction that resets the interrupt request flag (IRQxxx) as shown below. If a NOP instruction (or one or more other instructions) is not inserted, the interrupt request flag is not reset, and therefore, the halt status is released immediately.

Example

```
:
:                       ; IRQxxx is set at certain timing
:
CLR1    IRQxxx          ; Resets IRQxxx flag once
NOP                     ; Resets IRQxxx flag at this timing
                        ; Unless this period is missing, the IRQxxx flag is not reset,
                        ; and the next HALT instruction is immediately released
HALT    1000B           ;
```

20.2.7 If two or more releasing conditions are specified at same time

If two or more halt releasing conditions are specified at same time, the halt status is released when one of the conditions is satisfied.

The following program example shows how the releasing conditions are identified if two or more conditions are satisfied at the same time.

Example

```
            HLTINT    DAT     1000B
            HLTBTM    DAT     0010B
            HLTP0D    DAT     0001B
            P0D       MEM     0.73H
START:
            BR        MAIN
;*** Interrupt vector address ***
            NOP                       ; SI03
            NOP                       ; SI02
            NOP                       ; TIMER3
            NOP                       ; TIMER2
            NOP                       ; TIMER1
            NOP                       ; TIMER0
            NOP                       ; INT4
            NOP                       ; INT3
            NOP                       ; INT2
            NOP                       ; INT1
            BR        INT0            ; Branches to INT0 interrupt processing
            NOP                       ; CE DOWN EDGE

INT0:                                 ; INT0 pin interrupt vector address (000BH)

            Processing A              ; INT0 pin interrupt processing

            EI
            RETI
BTMOUP:                               ; Timer carry FF processing

            Processing B

            RET
P0DP:                                 ; P0D input processing

            Processing C

            RET
MAIN:
            INITFLG NOT BTM0CK1, NOT BTM0CK0
                                      ; Selects 100 ms as clock of basic timer 0
            SET1      IP0             ; Enables INT0 pin interrupt
            EI
LOOP:
            HALT HLTINT OR HLTBTM OR HLTP0C
                                      ; Selects interrupt, timer carry FF, and P0D input as halt releasing conditions
            SKF1      BTM0CY          ; Detects BTM0CY flag
            CALL      BTM0UP          ; Timer carry FF processing if flag is set to 1
            SKF       P0D, 1111B      ; Detects P0D input
            CALL      P0DP            ; Port input processing if P0D is high
            BR        LOOP
```

In the above example, three halt status releasing conditions, INT0 pin interrupt, 100-ms basic timer 0 carry FF, and port 0D input, are specified.

To identify which condition has released the halt status, a vector address (interrupt), BTM0CY flag (timer carry FF), and port register (port input) are detected.

To use two or more releasing conditions, the following two points must be noted.

- When the halt status is released, all the specified releasing conditions must be detected.
- The releasing condition with the higher priority must be detected first.

20.3 Clock Stop Function

20.3.1 Outline of clock stop function

The clock stop function stops the oscillation circuit of a 4.5-MHz crystal resonator by executing the "STOP s" instruction (clock stop status).

Therefore, the current consumption of the device is reduced to 30 μA MAX.

20.3.2 Clock stop status

In the clock stop status, all the device operations of the CPU and peripheral hardware units are stopped because the generation circuit of the crystal resonator is stopped.

For the operations of the CPU and peripheral hardware units, refer to **20.4 Device Operation in Halt and Clock Stop Status**.

In the clock stop status, the power failure detection circuit does not operate even if the supply voltage V_{DD} of the device is raised to 2.2 V. Therefore, the data memory can be backed up at a low voltage. For the power failure detection circuit, refer to **21. RESET**.

20.3.3 Releasing clock stop status

Figure 20-3 shows the stop status releasing conditions.

The stop status releasing condition is specified by 4-bit data specified by operand "s" of the "STOP s" instruction.

The stop status is released when the condition specified by "1" in operand "s" is satisfied.

When the stop status has been released, a halt period which is half the time (t_{SET}/2) specified by the basic timer 0 clock selection register as oscillation circuit stabilization wait time has elapsed, and the program execution is started from the instruction next to the "STOP s" instruction. If releasing the stop status by an interrupt is specified, however, the program operation after the stop status has been released differs depending on whether the interrupt is enabled (EI status) or disabled (DI status) when an interrupt source is issued (IRQxxx = 1) with the interrupt enabled (IPxxx = 1).

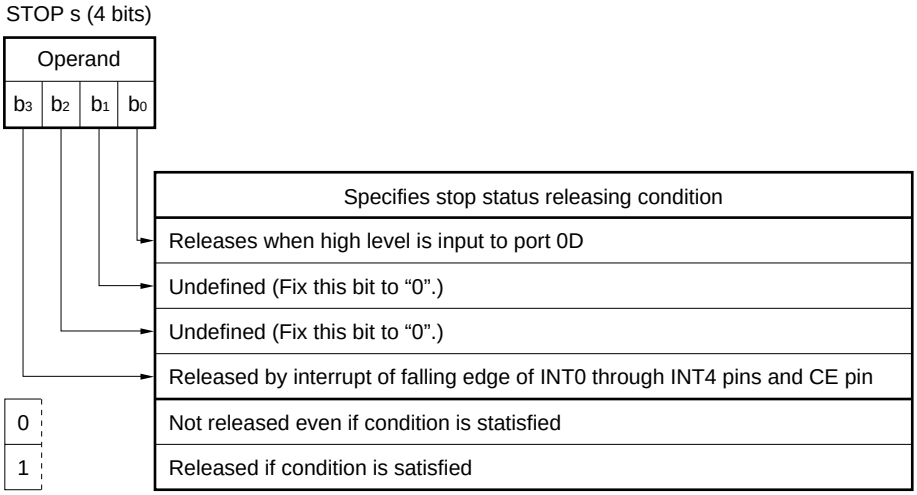
If all the interrupts are enabled (EI status), the stop status is released when the interrupt is enabled (IPxxx = 1) and the interrupt source is issued (IRQxxx = 1), and the program flow returns to the instruction next to the STOP instruction.

If all the interrupts are disabled (DI status), the stop status is released when the interrupt is enabled (IPxxx = 1) and the interrupt resource is issued (IRQxxx = 1), and the program flow returns to the instruction next to the STOP instruction.

If two or more releasing conditions are specified at one time, and if one of the conditions is satisfied, the stop status is released.

If 0000B is specified as stop releasing condition "s", no releasing condition is satisfied. If the device is reset at this time (by means of power-ON reset, or CE reset), the stop status is released.

Figure 20-3. Stop Releasing Conditions



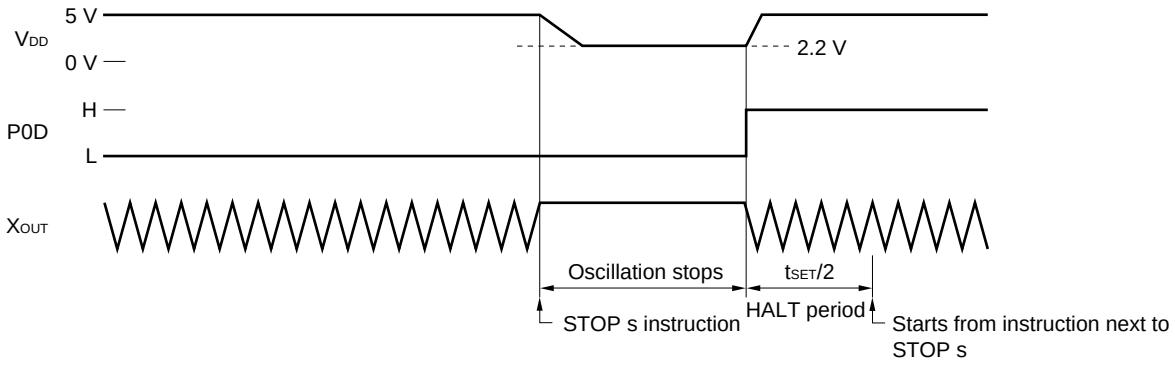
The “STOP s” instruction is executed as a “NOP” instruction when the CE pin rises and when the CE reset counter operates.

The operating status of the CE reset counter can be detected by the CECNTSTT flag (for the CE reset counter, refer to **21. RESET**).

20.3.4 Releasing clock stop status by high level input of port 0D

Figure 20-4 illustrates how the clock stop status is released by the high level input to port 0D.

Figure 20-4. Releasing Clock Stop Status By High Level Input of Port 0D



t_{SET}: basic timer 0 setting time

20.3.5 Cautions on releasing clock stop status

For the cautions on releasing the clock stop status, refer to **(2) Releasing from clock stop status in 21.4.4 Cautions on raising supply voltage V_{DD}**.

20.4 Device Operation in Halt and Clock Stop Status

Table 20-1 shows the operations of the CPU and peripheral hardware units in the halt and clock stop status.

In the halt status, all the peripheral hardware units continue the normal operation until instruction execution is stopped.

In the clock stop status, all the peripheral hardware units stop operation.

The control registers that control the operations of the peripheral hardware units operate normally (not initialized) in the halt status, but are initialized to specified values when the clock stop instruction is executed.

In other words, all peripheral hardware continues the operation specified by the control register in the halt status, and the operation is determined by the initialized value of the control register in the clock stop status.

For the values of the control registers in the clock stop status, refer to **8. REGISTER FILE (RF)**.

Table 20-1. Device Operation in Halt and Clock Stop Status

Peripheral Hardware	Status	
	Halt	Clock stop
Program counter	Stops at address of HALT instruction	Stops at address of STOP instruction
System register	Retained	Retained
Peripheral register	Retained	Partly initialized[Note 1]
Control register	Retained	Partly initialized[Note 1]
Timer	Normal operation	Operation stops
PLL frequency synthesizer	Normal operation[Note 2]	Operation stops
A/D converter	Normal operation	Operation stops
D/A converter	Normal operation	Stops operation and used as general-purpose output port
Serial interface	Stops operation when internal clock (master) is selected and continues operation when external clock (slave) is selected	Stops operation and used as general-purpose I/O port
Frequency counter	Normal operation	Stops operation and used as general-purpose input port
BEEP output	Normal operation	Stops operation and used as general-purpose I/O port
General-purpose I/O port	Normal operation	Retained
General-purpose input port	Normal operation	Input port
General-purpose output port	Normal operation	Retains output latch

Notes 1. For the value to which these registers are initialized, refer to **5. SYSTEM REGISTER (SYSREG)** and **8. REGISTER FILE (RF)**.

2. The PLL frequency synthesizer is automatically disabled by the low level input to the CE pin.

20.5 Cautions on Processing of Each Pin in Halt and Clock Stop Status

The halt status is used to reduce the current consumption when, say, only the watch is used.

The clock stop function is used to reduce the current consumption of the device to only use the data memory.

Therefore, the current consumption must be reduced as much as possible in the halt status or clock stop status.

At this time, the current consumption significantly varies depending on the status of each pin, and the points shown in Table 20-2 must be noted.

Table 20-2. Status of Each Pin in Halt and Clock Stop Status and Cautions (1/2)

Pin Function		Pin Symbol	Status of Each Pin and Cautions on Processing	
			Halt status	Clock stop status
General-purpose I/O port	Port 0A	P0A3/SDA P0A2/SCL P0A1/$\overline{SCK2}$ P0A0/SO2	Retains status before halt **(1) When specified as output pin** Current consumption increases if pin is externally pulled down while it outputs high level, or externally pulled up while it outputs low level. Exercise care in using N-ch open-drain output (P0A3, P0A2, P1B3 through P1B0, P2D1, P2D0) **(2) When specified as input pin** Current consumption increases due to noise if pin is floated **(3) Port 0D (P0D3/AD3 through P0D0/AD0)** Current consumption increases if pin is externally pulled up because it is provided with pull-down resistor selectable by software **(4) Port 1C (P1C3/AD5, P1C2/AD4, P1C1/AMIFC, P1C0/FMIFC)** When P1C1/AMIFC or P1C0/FMIFC pin is used for IF counter, current consumption increases because internal amplifier operates	All port pins are set in general-purpose port mode (except P0D3/AD3 through P0D0/AD0, P1A3/INT4, P1A2/INT3, P1C3/AD5, and P1C2/AD4) Input or output mode of general-purpose I/O port set before clock stop status is retained. **(1) When specified as general-purpose output port** Current consumption increases due to noise if pin is floated **(2) When specified as general-purpose input port** Current consumption does not increase due to noise even if pin is floated **(3) P1A3/INT4, P1A2/INT3** Set as interrupt pin and current consumption increases due to external noise if pin is floated **(4) P0D3/AD3 through P0D0/AD0, P1C3/AD5, P1C2/AD4** Pin used for A/D converter is retained as is. Pull-down resistor of P0D3 through P0D0 pin retains previous status
	Port 0B	P0B3/SI2 P0B2/$\overline{SCK3}$ P0B1/SO3/TxD P0B0/SI3/RxD		
	Port 0C	P0C3-P0C0		
	Port 1D	P1D3 P1D2 P1D1/BEEP1 P1D0/BEEP0		
	Port 2A	P2A2 P2A1/FCG1 P2A0/FCG0		
	Port 2B	P2B3-P2B0		
	Port 2C	P2C3-P2C0		
	Port 2D	P2D2/$\overline{SCK}$ P2D1/SB1 P2D0/SB0		
	Port 3A	P3A3-P3A0		
	Port 3B	P3B3-P3B0		
	Port 3C	P3C30P3C0		
	Port 3D	P3D3-P3D0		
General-purpose input port	Port 0D	P0D3/AD3 | P0D0/AD0		
	Port 1A	P1A3/INT4 P1A2/INT3 P1A1 P1A0/TM0G		
	Port 1C	P1C3/AD5 P1C2/AD4 P1C1/AMIFC P1C0/FMIFC		
General-purpose output port	Port 1B	P1B3 P1B2/PWM2 | P1B0/PWM0		Specified as general-purpose output port. Output contents are retained as is. If pin is externally pulled down while it outputs high level or externally pulled up while it outputs low level, current consumption increases

Table 20-2. Status of Each Pin in Halt and Clock Stop Status and Cautions (2/2)

Pin Function	Pin Symbol	Status of Each Pin and Cautions on Processing	
		Halt status	Clock stop status
External interrupt	INT4-INT0	Current consumption increases due to noise if pin is floated	
PLL frequency synthesizer	VCOL VCOH EO0 EO1	Current consumption increases during PLL operation. When PLL is disabled, pin is in following status: VCOH, VCOL : internally pulled down EO1, EO0 : floated PLL is automatically disabled if CE pin goes low	PLL is disabled VCOH, VCOL : internally pulled down EO1, EO0 : floated
Crystal oscillation circuit	X_{IN} X_{OUT}	Current consumption changes due to oscillation waveform of crystal oscillation circuit. The higher oscillation amplitude, the lower current consumption. Oscillation amplitude must be evaluated because it is influenced by crystal resonator or load capacitor used	X_{IN} pin is internally pulled down, and X_{OUT} pin outputs high level

20.6 Device Operation Control Function of CE Pin

The CE pin controls the following functions by the input level and rising edge of the signal input from an external source.

- PLL frequency synthesizer
- Interrupt by falling edge of CE pin
- Resetting of device

20.6.1 Controlling operation of PLL frequency synthesizer

The PLL frequency synthesizer can operate only when the CE pin is high.

It is automatically disabled when the CE pin is low.

When the synthesizer is disabled, the VCOH and VCOL pins are internally pulled down, and the EO0 and EO1 pins are floated. For details, refer to **17.5 PLL Disabled Status**.

The PLL frequency synthesizer can be disabled in software even when the CE pin is high.

20.6.2 Controlling interrupt by falling edge input of CE pin

An interrupt can be generated by the falling edge of the CE pin. For details, refer to **12. INTERRUPTS**.

20.6.3 Resetting device

The device can be reset (CE reset) by raising the CE pin.

The device can also be reset as follows:

- Power-ON reset on application of supply voltage V_{DD}
- Watchdog timer reset for software hang-up detection and stack overflow/underflow reset
- Reset by $\overline{\text{RESET}}$ pin

For details, refer to **21. RESET**.

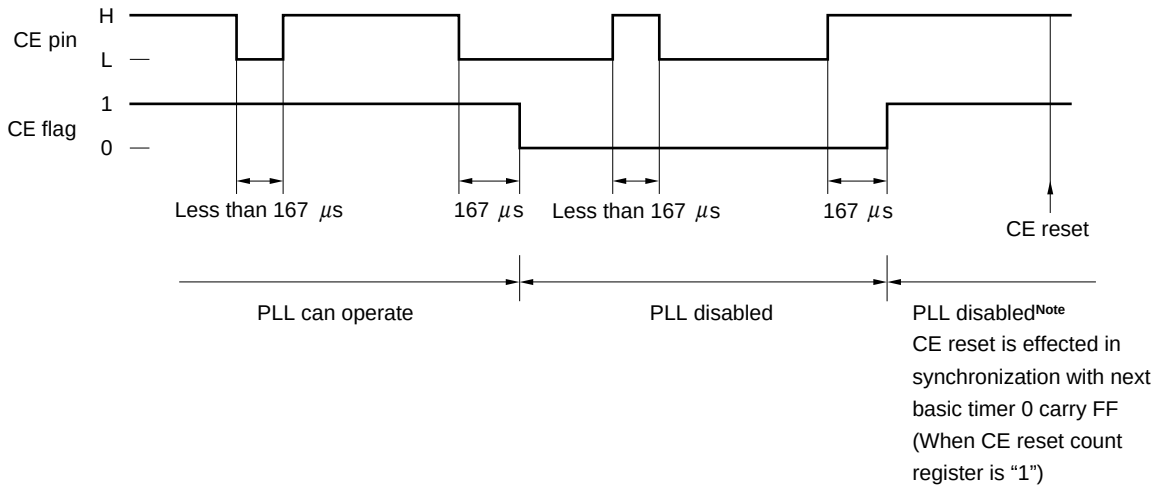
20.6.4 Signal input to CE pin

The CE pin does not accept a low level or high level of less than 167 μs to prevent malfunctioning due to noise.

The level of the signal input to the CE pin can be detected by the CE pin status detection flag of the CE pin interrupt request register (RF address 3FH).

Figure 20-5 shows the relationship between the input signal and CE flag.

Figure 20-5. Relationship between Input Signal of CE Pin and CE Flag



Note Unless the PLL mode selection register and PLL reference frequency selection register are rewritten by software, the PLL disabled status is retained.

20.6.5 Configuration and function of CE pin interrupt request register

The CE pin interrupt request register detects the input signal level of the CE pin.

Figure 20-6 shows the configuration of the CE pin interrupt request register.

Figure 20-6. Configuration of CE Pin Interrupt Request Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
CE pin interrupt request	CE	0	CECNTSTT	IRQCE	3FH	R[Note]

IRQCE — Sets interrupt request issuance status of CE pin

0	No interrupt request
1	Interrupt request

CECNTSTT — Detects status of CE reset counter

0	Stops
1	Operates

b_2 — Fixed to “0”

CE — Detects status of CE pin

0	Low level is input
1	High level is input

		CE	b_2	CECNTSTT	IRQCE
At Reset	Power-ON reset	U	0	0	0
	WDT&SP reset	U		0	0
	CE reset	U		0	R
Clock stop		U		0	R

U: Undefined R: Retained

Note IRQCE is a R/W flag.

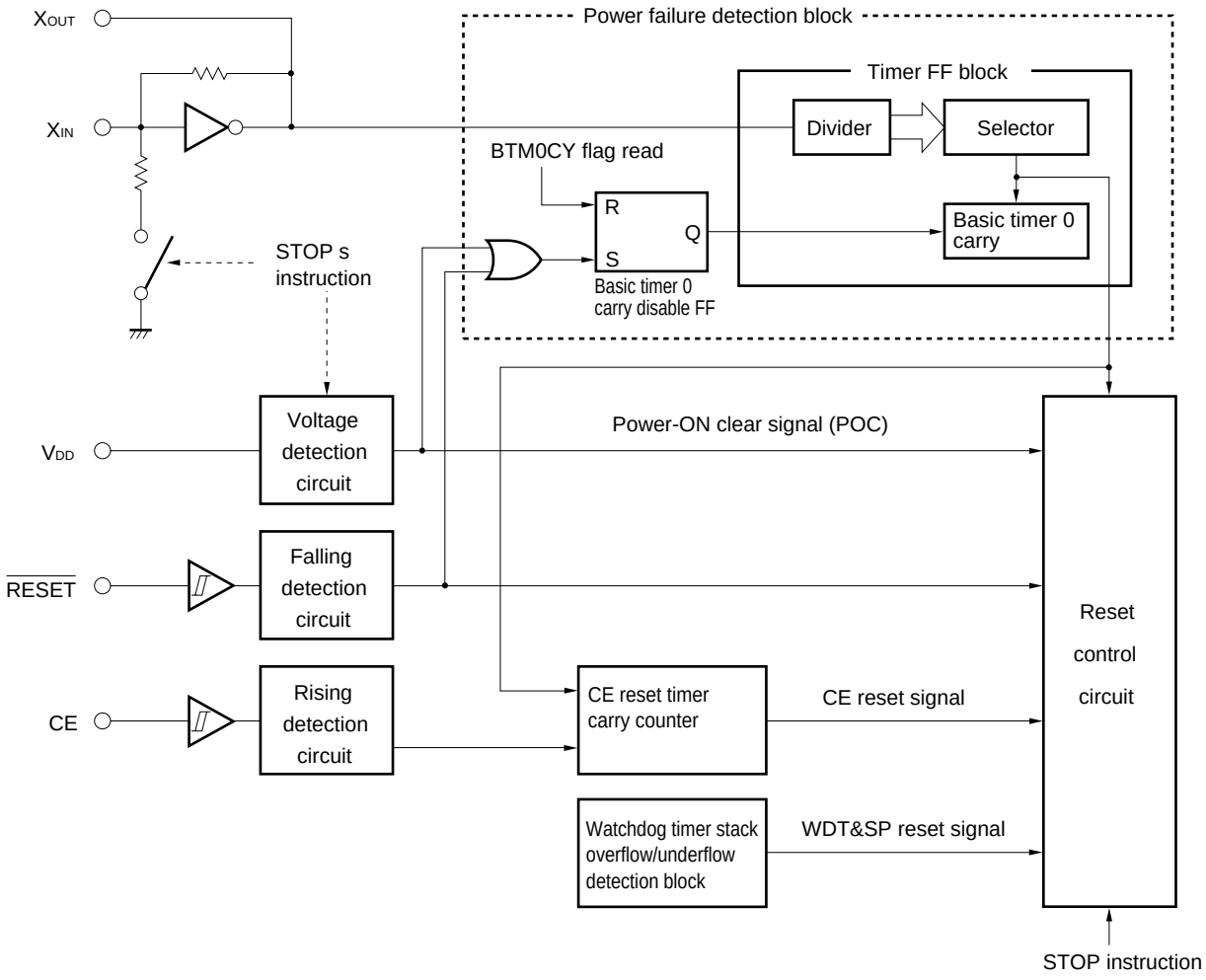
21. RESET

21.1 Outline of Reset

The reset function is used to initialize the device.
The μPD17719 can be reset in the following ways:

- CE reset
- Power-ON reset
- Reset by $\overline{\text{RESET}}$ pin
- WDT&SP reset

Figure 21-1. Configuration of Reset Block



21.2 CE Reset

CE reset is effected by raising the CE pin.

When the CE pin goes high, the next rising edge of the basic timer 0 carry FF setting pulse is counted. When the count value coincides with the value set to the CE reset timer carry counter register (1 to 15 counts), the reset signal is generated.

When CE reset is effected, the program counter, stack, system registers, and some of the control registers are initialized to the initial values, and program execution is started from address 0000H. For the initial value of each register, refer to the **description** of **each register**.

Figure 21-2. Configuration of CE Reset Timer Carry Counter Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
CE reset timer carry counter	CECNT3	CECNT2	CECNT1	CECNT0	06H	R/W

				Sets number of counts of timer carry counter for CE reset
0	0	0	0	Setting prohibited
0	0	0	1	1 count
0	0	1	0	2 counts
0	0	1	1	3 counts
0	1	0	0	4 counts
0	1	0	1	5 counts
0	1	1	0	6 counts
0	1	1	1	7 counts
1	0	0	0	8 counts
1	0	0	1	9 counts
1	0	1	0	10 counts
1	0	1	1	11 counts
1	1	0	0	12 counts
1	1	0	1	13 counts
1	1	1	0	14 counts
1	1	1	1	15 counts

At reset	Power-ON reset	0	0	0	1
	WDT&SP reset	Retained			
	CE reset	Retained			
Clock stop		0	0	0	1

The operation of CE reset differs depending on whether the clock stop instruction is used or not.

This difference is described in 21.2.1 and 21.2.2 below.

21.2.3 describes the points to be noted when CE reset is effected.

21.2.1 CE reset without clock stop (STOP s) instruction

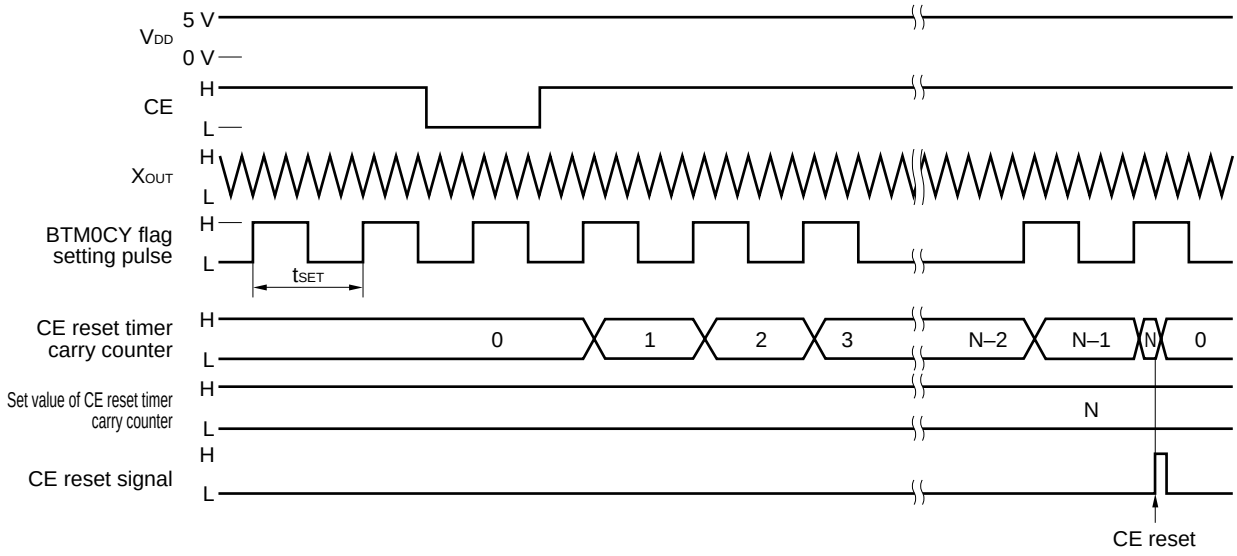
Figure 21-2 shows the operation.

When the CE pin has gone high, the CE reset timer carry counter starts counting at the rising edge of the basic timer 0 carry FF setting pulse.

Figure 21-3. CE Reset Operation without Clock Stop Instruction (1/2)

(a) Normal operation

- **When "N" is set to CE reset timer carry counter**



- **When "1" is set to CE reset timer carry counter**

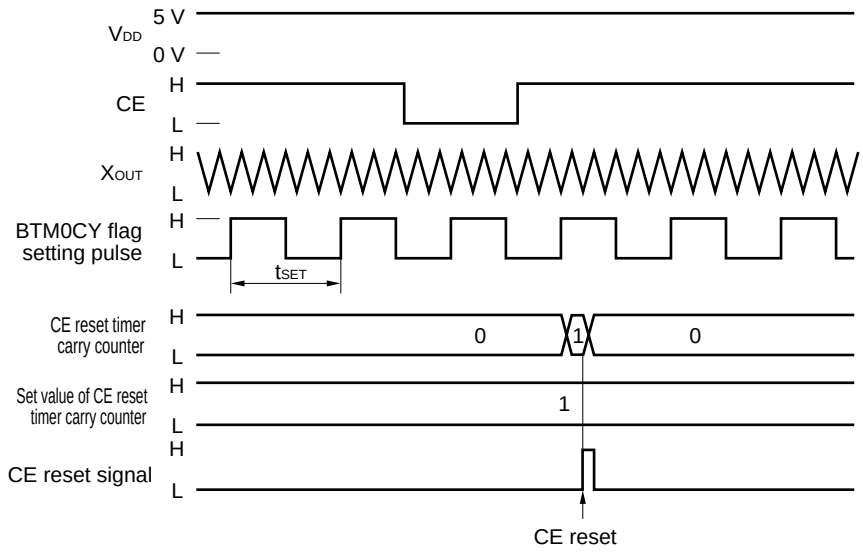
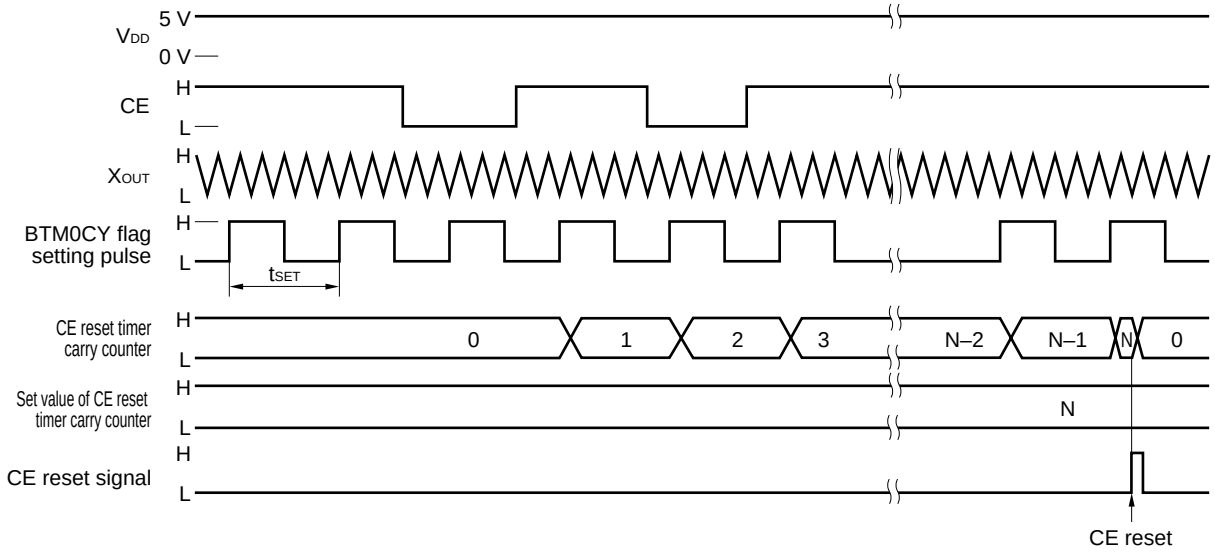


Figure 21-3. CE Reset Operation without Clock Stop Instruction (2/2)

(b) If status of CE pin changes while CE counter operates

At this time, the CE reset timer carry counter status is not affected.



21.2.2 CE reset with clock stop (STOP s) instruction used

Figure 21-4 shows the operation.

When the clock stop instruction is used, the clock stop signal is output when the “STOP s” instruction is executed, and oscillation is stopped and the device operation is stopped.

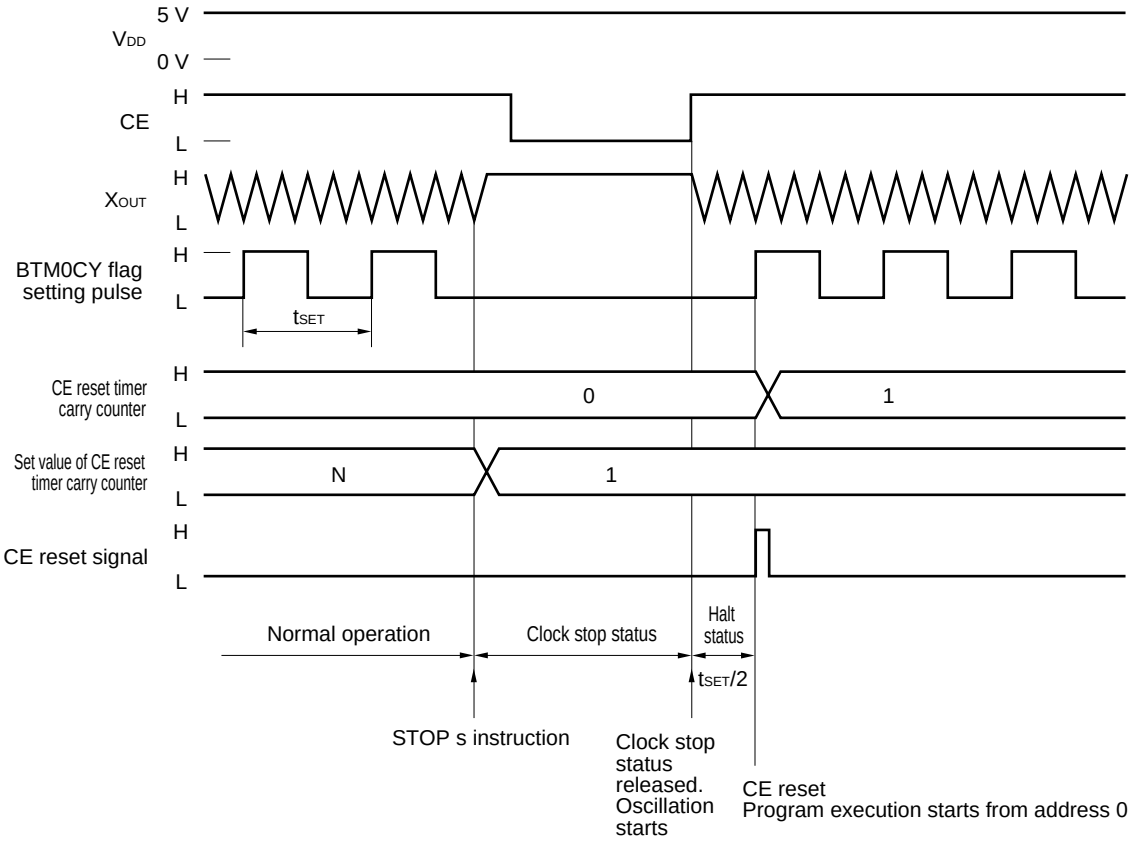
When the CE pin goes high, the clock stop status is released, and oscillation is started (high level input of P0D or INT pin interrupt can also be used as the clock stop status releasing conditions. For details, refer to **20. STANDBY**).

If the basic timer 0 carry FF setting pulse goes high after the CE pin has gone high, the halt status is released, and program execution is started from address 0 (CE reset).

As the set time (t_{SET}) of the basic timer 0 carry FF setting pulse, the value immediately before the clock stop instruction is executed is retained.

Because the set value of the CE reset timer carry counter is initialized to 1, CE reset is effected $t_{SET}/2$ after the CE pin has gone high.

Figure 21-4. CE Reset Operation with Clock Stop Instruction



21.2.3 Cautions on CE reset

Because CE reset is effected regardless of the instruction under execution, the following points (1) and (2) must be noted.

(1) Time to execute timer processing such as watch

When creating a watch program by using the basic timer 0 carry, the processing time of the program must be kept to within a specific time.

For details, refer to **13.2.6 Cautions on using basic timer 0**.

(2) Processing of data and flags used in program

Exercise care in rewriting the data and flags whose contents must not be changed even when CE reset is effected, such as security code.

An example is shown below.

Example 1.

```
R1      MEM     0.01H           ; 1st digit of key input data of security code
R2      MEM     0.02H           ; 2nd digit of key input data of security code
R3      MEM     0.03H           ; 1st digit data when security code is changed
R4      MEM     0.04H           ; 2nd digit data when security code is changed
M1      MEM     0.11H           ; 1st digit of current security code
M2      MEM     0.12H           ; 2nd digit of current security code

START:
        Key input processing
          R1 ← contents of key A    ; Security code input wait mode
          R2 ← contents of key B    ; Substitutes contents of pressed key to R1 and R2

        SET2    CMP, Z          ; <1> ; Compares security code and input data
        SUB     R1, M1
        SUB     R2, M2
        SKT1    Z
        BR      ERROR           ; Input data differs from security code
MAIN:
        Key input processing
          R3 ← contents of key C    ; Security code rewriting mode
          R4 ← contents of key D    ; Substitutes contents of pressed key to R3 and R4

        ST      M1, R3          ; <2> ; Rewrites security code
        ST      M2, R4          ; <3>
        BR      MAIN
ERROR:
        Must not operate
```

Suppose the security code is “12H” in the program in Example 1. The contents of data memory addresses M1 and M2 are “1H” and “2H”, respectively.

If CE reset is effected, the contents of key input and security code “12H” are compared in <1>. If the two are the same, the normal processing is performed.

If the security code is changed in the main processing, the new code is written to M1 and M2 in <2> and <3>.

Suppose the security code is changed to “34H”. Then “3H” and “4H” are written to M1 and M2 in <2> and <3>.

If CE reset is effected as soon as <2> has been executed, program execution is started from address 0000H, without <3> being executed.

Consequently, the security code is set to “32H”, making it impossible to clear the security system.

In this case, create the program shown in Example 2.

Example 2.

```
R1          MEM     0.01H           ; 1st digit of key input data of security code
R2          MEM     0.02H           ; 2nd digit of key input data of security code
R3          MEM     0.03H           ; 1st digit data when security code is changed
R4          MEM     0.04H           ; 2nd digit data when security code is changed
M1          MEM     0.11H           ; 1st digit of current security code
M2          MEM     0.12H           ; 2nd digit of current security code
CHANGE      FLG     0.13H.0         ; "1" while security code is changed

START:
            Key input processing
              R1 ← contents of key A  ; Security code input wait mode
              R2 ← contents of key B  ; Substitutes contents of pressed key to R1 and R2

            SKT1    CHANGE      ; <4> ; If CHANGE flag is "1"
            BR      SECURITY_CHK
            ST      M1, R3          ; Rewrites M1 and M2
            ST      M2, R4
            CLR1    CHANGE
SECURITY_CHK:
            SET2    CMP, Z      ; <1> ; Compares security code and input data
            SUB     R1, M1
            SUB     R2, M2
            SKT1    Z
            BR      ERROR           ; Input data differs from security code
MAIN:
            Key input processing
              R3 ← contents of key C  ; Security code rewriting mode
              R4 ← contents of key D  ; Substitutes contents of pressed key to R3 and R4

            SET1    CHANGE      ; <5> ; Until security code is changed,
                                      ; Sets CHANGE flag to "1"
            ST      M1, R3      ; <2> ; Rewrites security code
            ST      M2, R4      ; <3>
            CLR1    CHANGE          ; If security code has been changed,
                                    ; Sets CHANGE flag to "0"
            BR      MAIN
ERROR:
            Must not operate
```

The program in Example 2 sets the CHANGE flag to "1" in <5> before the security code is rewritten in <2> and <3>.

Therefore, even if CE reset is effected before <3> is executed, the security code is rewritten in <4>.

21.3 Power-ON Reset

Power-ON reset is effected by raising the supply voltage V_{DD} of the device from a specific level (called a power-ON clear voltage).

If supply voltage V_{DD} is lower than the power-ON clear voltage, a power-ON clear signal (POC) is output from the voltage detection circuit shown in Figure 21-1.

When the power-ON clear signal is input to the reset control circuit, the crystal oscillation circuit is stopped and consequently, the device operation is stopped.

At this time, the program counter, stack, system registers, and control registers are initialized (for the initial value, refer to the **description** of **each register**).

If supply voltage V_{DD} exceeds the power-ON clear voltage, the power-ON clear signal is deasserted, crystal oscillation is started, and the device waits for release of the halt status by the basic timer 0 carry which has been initialized to 100 ms. Program execution is started from address 0 at the rising edge of the basic timer 0 carry FF setting signal 50 ms after the supply voltage has exceeded the power-ON clear voltage.

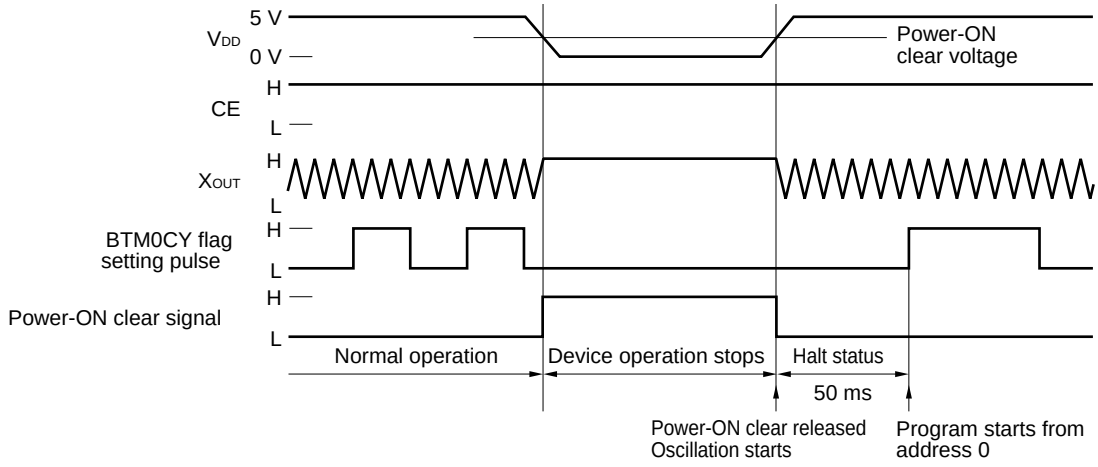
Normally, the power-ON clear voltage is 3.5 V, but it is 2.2 V in the clock stop status.

The operations of power-ON reset are described in 21.3.1 and 21.3.2.

The operation when supply voltage V_{DD} is raised from 0 V is described in 21.3.3.

Caution Although it is stated that the normal power-ON clear voltage is 3.5 V (MAX.) and that in the clock stop status is 2.2 V (MAX.), the actual power-ON clear voltage does not exceed these maximum values.

Figure 21-5. Operation of Power-ON Reset



21.3.1 Power-ON reset during normal operation

Figure 21-6 (a) shows the operation.

As shown, the power-ON clear signal is output and the device operation is stopped if the supply voltage V_{DD} drops below 3.5 V, regardless of the input level of the CE pin.

If V_{DD} rises beyond 3.5 V again, program execution starts from address 0000H after a halt of 50 ms.

Normal operation means operation without the clock stop instruction, and includes the halt status set by the halt instruction.

21.3.2 Power-ON reset in clock stop status

Figure 21-6 (b) shows the operation.

As shown, the power-ON clear signal is output and the device operation is stopped when supply voltage V_{DD} drops below 2.2 V.

However, it does not appear that device operation has changed because the device is in the clock stop status.

If V_{DD} rises beyond 3.5 V, program execution starts from address 0000H after a halt of 50 ms.

21.3.3 Power-ON reset when supply voltage V_{DD} rises from 0 V

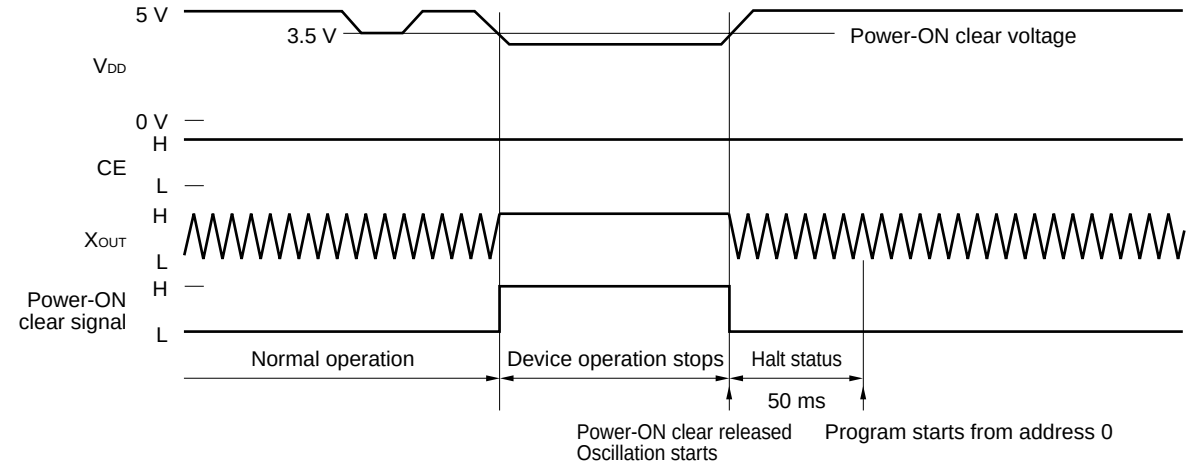
Figure 21-6 (c) shows the operation.

As shown, the power-ON clear signal is output until supply voltage V_{DD} rises from 0 V to 3.5 V.

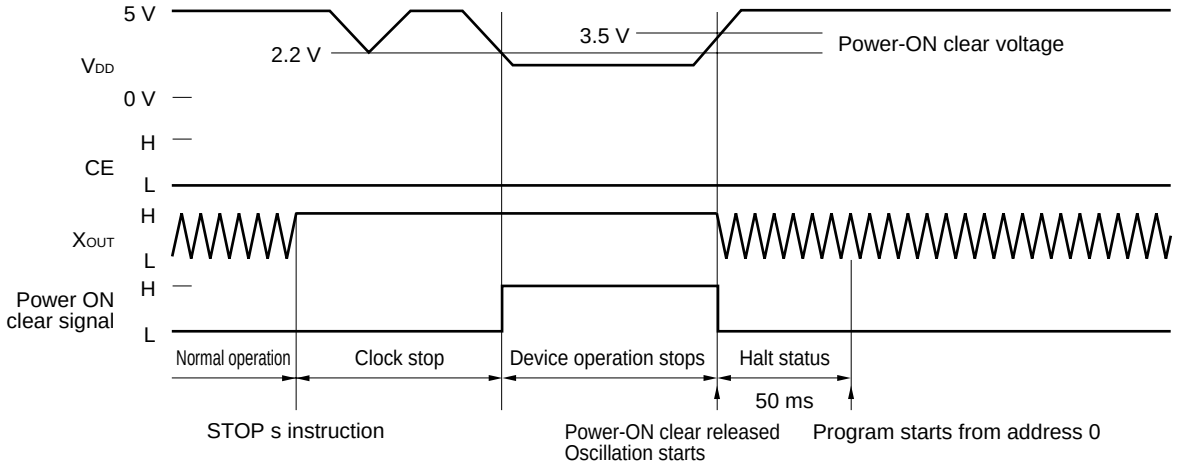
When V_{DD} exceeds the power-ON clear voltage, the crystal oscillation circuit starts operating, and program execution starts from address 0000H after a half of 50 ms.

Figure 21-6. Power-ON Reset and Supply Voltage V_{DD}

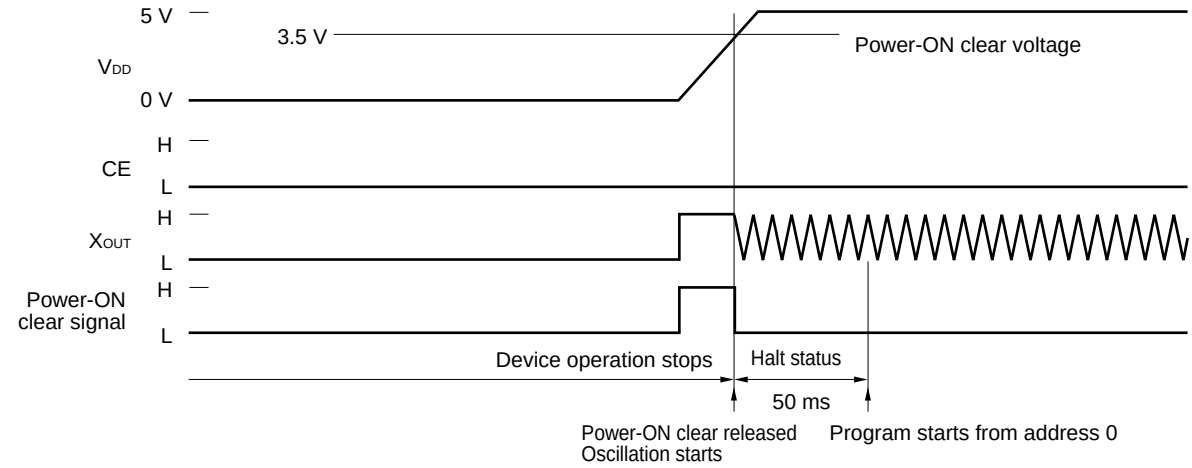
(a) Normal operation (including halt status)



(b) In clock stop status



(c) If supply voltage V_{DD} rises from 0 V



21.4 Relationship between CE Reset and Power-ON Reset

On the first application of supply voltage V_{DD}, power-ON reset and CE reset are performed at the same time.
The reset operations at this time are described in 21.4.1 through 21.4.3.
21.4.4 describes the points to be noted when raising supply voltage V_{DD}.

21.4.1 If V_{DD} pin and CE pin go high at the same time

Figure 21-7 (a) shows the operation.
At this time, the program starts from address 0000H because of power-ON reset.

21.4.2 If CE pin rises in forced halt status set by power-ON reset

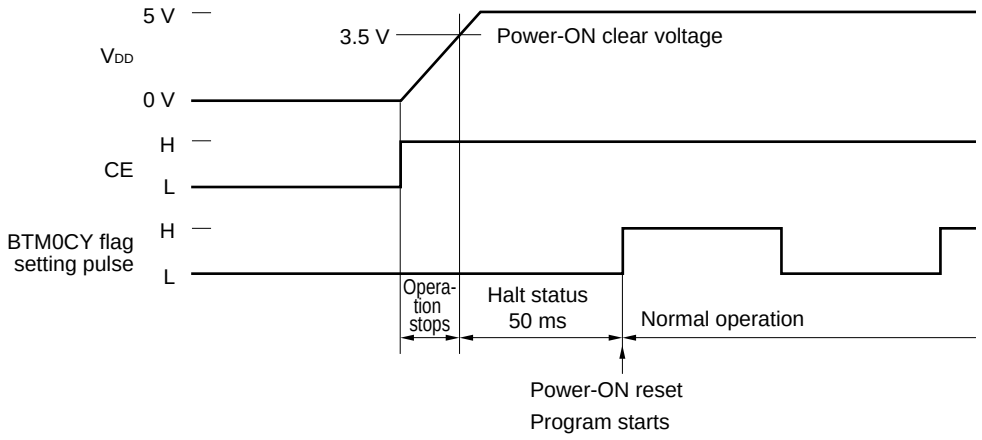
Figure 21-7 (b) shows the operation.
At this time, the program starts from address 0000H because of power-ON reset, in the same manner as 21.4.1.

21.4.3 If CE pin rises after power-ON reset

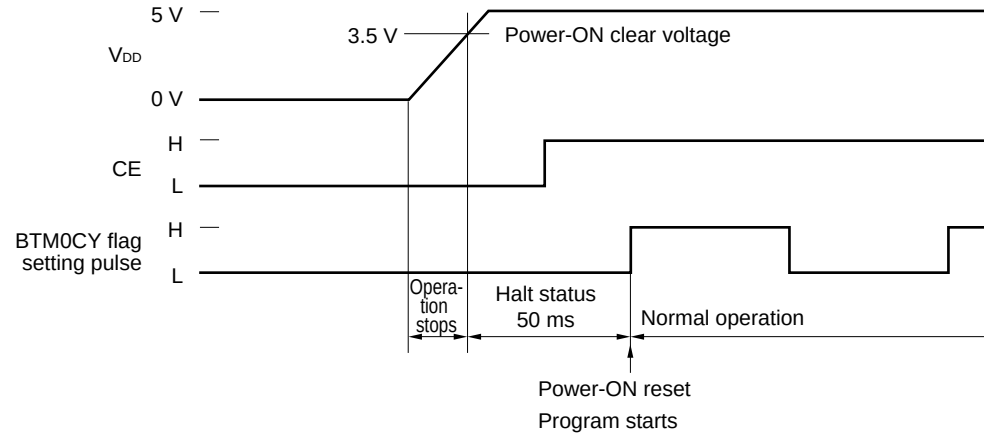
Figure 21-7 (c) shows the operation.
At this time, the program starts from address 0000H because of power-ON reset, and the program starts from address 0000H again at the rising edge of the next basic timer 0 carry FF setting signal because of CE reset.

Figure 21-7. Relationship between Power-ON Reset and CE Reset

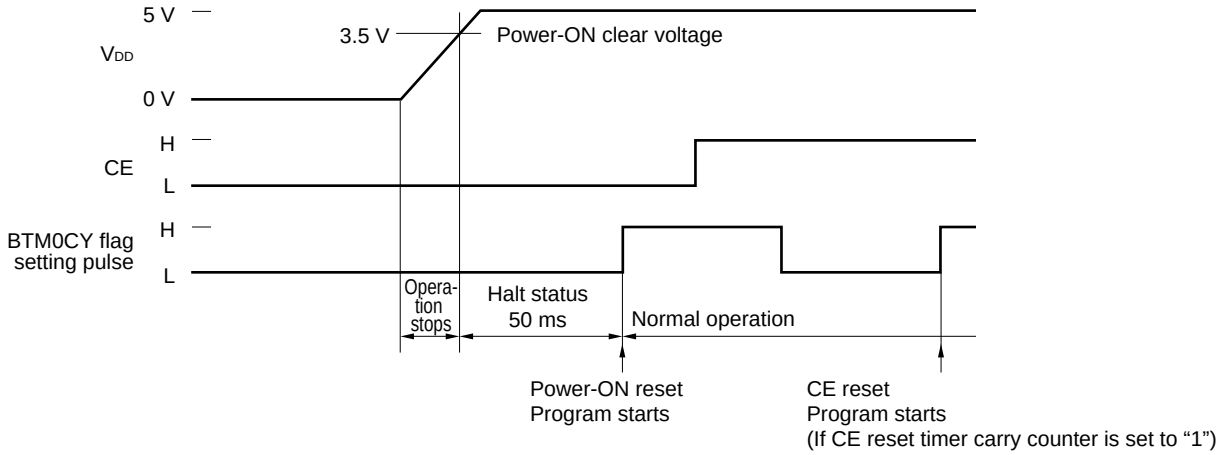
(a) When V_{DD} and CE pin rise at the same time



(b) If CE pin rises in halt status



(c) If CE pin rises after power-ON reset



21.4.4 Cautions on raising supply voltage V_{DD}

The following points (1) and (2) must be noted when raising supply voltage V_{DD}.

(1) To raise supply voltage V_{DD} from level lower than power-ON clear voltage

Supply voltage V_{DD} must be raised once to a level higher than 3.5 V.

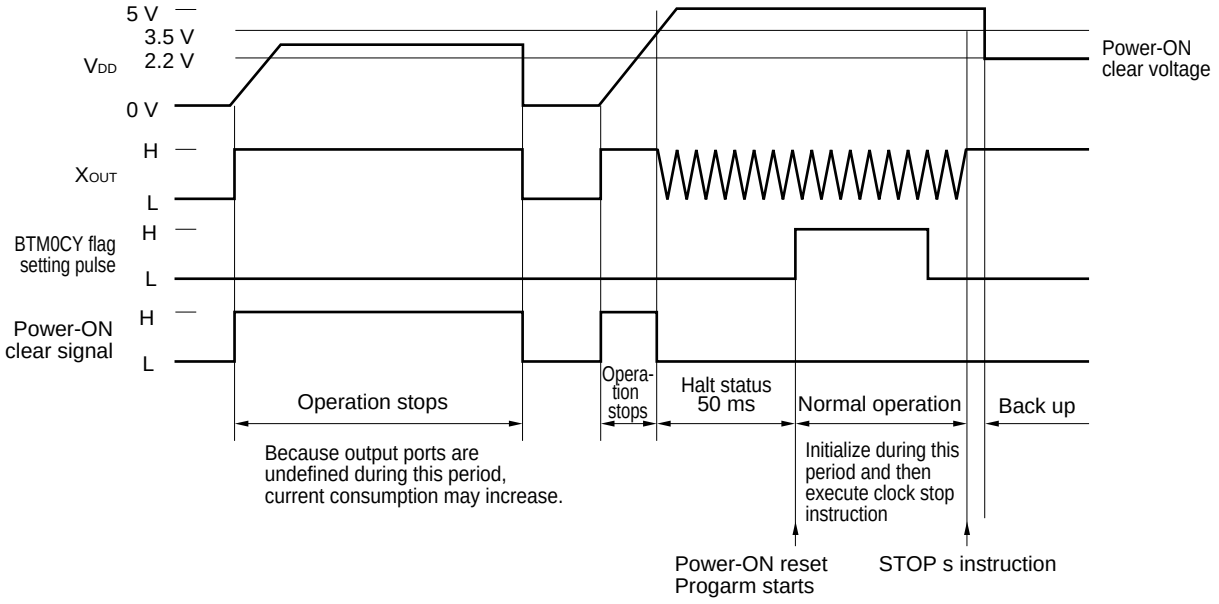
Figure 21-8 illustrates this.

As shown in the figure, if a voltage less than 3.5 V is applied on application of V_{DD} in a program that backs up V_{DD} at 2.2 V by using the clock stop instruction, the power-ON clear signal remains output, and the program is not executed.

At this time, the output ports of the device output undefined values, increasing the current consumption in some cases.

Consequently, the backup time when the device is backed up by batteries is substantially shortened.

Figure 21-8. Cautions on Raising V_{DD}



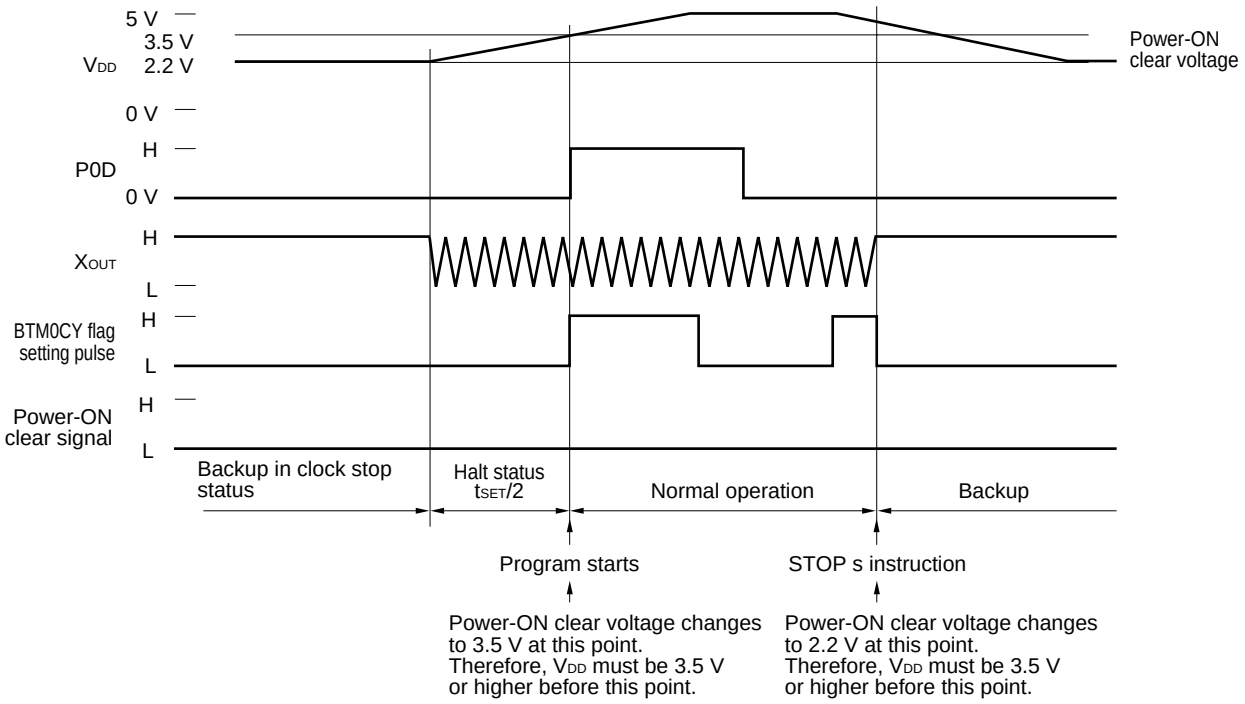
(2) Releasing from clock stop status

If the device is released from the backup status when supply voltage V_{DD} is backed up at 2.2 V by using the clock stop status, V_{DD} must be raised to 3.5 V or more within t_{SET}/2 after the clock stop status has been released by INT pin interrupt or high level input to port 0D.

As shown in Figure 21-9, the device is released from the clock stop status by means of CE reset. However, because the power-ON clear voltage is changed to 3.5 V t_{SET}/2 after the clock stop status has been released, power-ON reset is effected unless V_{DD} is 3.5 V or higher.

The same applies when V_{DD} is raised.

Figure 21-9. Releasing from Clock Stop Status



t_{SET}: basic timer 0 setting time

21.5 Reset by $\overline{\text{RESET}}$ Pin

The device is reset by the $\overline{\text{RESET}}$ pin in the following cases:

- To reset the device at voltage higher than power-ON clear voltage
- External reset input in case of software hang-up

Caution If reset is executed by using the $\overline{\text{RESET}}$ pin during program execution, the data of the data memory may be destroyed. Therefore, exercise care when executing reset by using the $\overline{\text{RESET}}$ pin.

The reset operation is the same as that performed at power-ON reset.

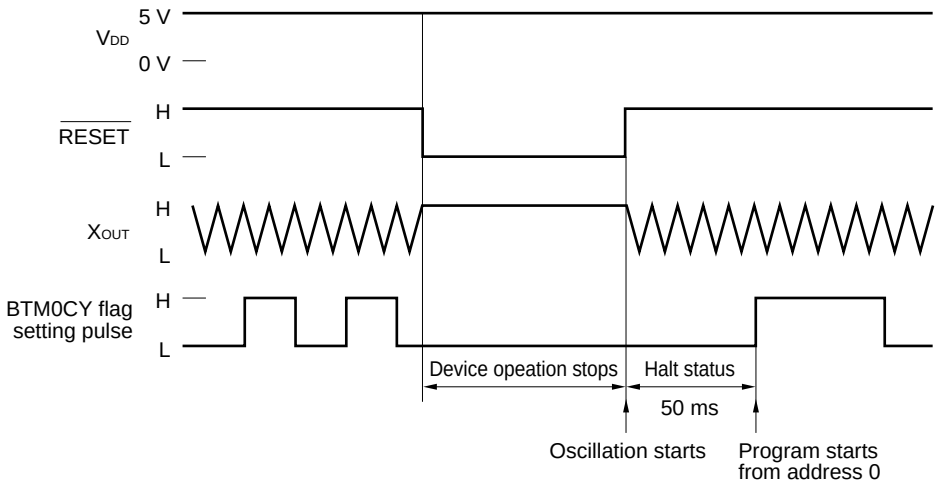
When a low level is input to the $\overline{\text{RESET}}$ pin, an internal reset signal is generated, the crystal oscillation circuit is stopped, and the device stops operation.

At this point, the program counter, stack, system registers, and control registers are initialized (for the initial value, refer to the **description** of **each register**).

When the $\overline{\text{RESET}}$ pin is raised next time, the crystal oscillation is started, and the device waits to be released from the halt wait status by the basic timer 0 carry which has been initialized to a 100-ms cycle. The program starts from address 0 at the rising edge of the basic timer 0 carry FF setting signal 50 ms after a high level has been input to the $\overline{\text{RESET}}$ pin.

Because the μPD17719 has a power-ON reset function, connect the $\overline{\text{RESET}}$ pin to V_{DD} via resistor if the $\overline{\text{RESET}}$ pin is not used for the above application.

Figure 21-10. Reset Operation by $\overline{\text{RESET}}$ Pin

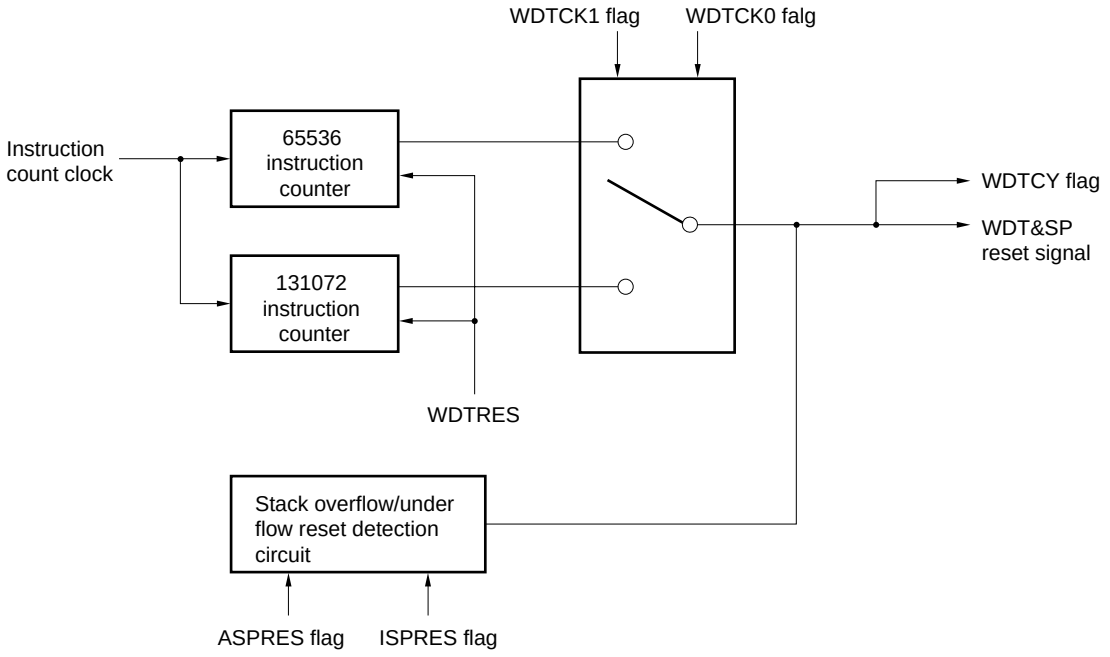


21.6 WDT&SP Reset

WDT&SP reset includes the following:

- Watchdog timer reset
- Stack pointer overflow/underflow reset

Figure 21-11. Outline of WDT&SP Reset



21.6.1 Watchdog timer reset

The watchdog timer is a circuit that generates a reset signal when the execution sequence of the program is abnormal (hung-up).

Hanging-up means that the program jumps to an unexpected routine due to external noise, entering a specific infinite loop and causing the system to be deadlocked. By using the watchdog timer, the program can be restored from this hang-up status because a reset signal is generated from the watchdog timer at fixed time intervals and program execution is started from address 0.

The watchdog timer does not function in the clock stop mode and halt mode.

Resetting by the watchdog timer initializes all the registers except the stack overflow selection register, watchdog timer counter reset register, basic timer 0 carry register, and CE reset timer carry counter.

The watchdog timer reset is detected by the WDTCY flag (R&Reset).

21.6.2 Watchdog timer setting flags

These flags can be set only once after power-ON reset on power application or reset by the $\overline{\text{RESET}}$ pin.

The WDTCK0 and WDTCK1 flags select an interval at which the reset signal is output.

The reference time can be selected to the following three conditions:

- 655356 instructions
- 131072 instructions
- Watchdog timer not set

On power application, 131072 instructions are selected.

If the reset signal generation interval is specified to be 131072 instructions, the watchdog timer FF must be reset at intervals not exceeding 131072 instructions. The valid reset period is from 1 to 131071 instructions.

If the reset signal generation interval is 65536 instructions, the watchdog timer FF must be reset at intervals not exceeding 65536 instrutions. The valid reset period is from 1 to 65535 instructions.

Figure 21-12. Configuration of Watchdog Timer Clock Selection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Watchdog timer clock selection	0	0	WDTCK1	WDTCK0	02H	R/W[Note]

Selects clock of watchdog timer	
0 0	Does not set watchdog timer
0 1	65536 instructions
1 0	Setting prohibited
1 1	131072 instructions

Fixed to “0”

At reset	Power-ON reset	0	0	1	1
	WDT&SP reset				Retained
	CE reset				Retained
Clock stop					Retained

Note Can be written only once.

The WDTRES flag is used to reset the watchdog timer counter.

When this flag is set to 1, the watchdog timer counter is automatically reset.

If the WDTRES flag is set to 1 once within a reference time in which the WDTCK0 and WDTCK1 flags are set, the reset signal is not output by the watchdog timer.

Figure 21-13. Configuration of Watchdog Timer Counter Reset Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Watchdog timer counter reset	WDTRES	0	0	0	03H	W&Reset

Fixed to "0"

	Resets watchdog timer counter
0	Invalid
1	Resets watchdog timer counter

At reset	Power-ON reset	U	0	0	0
	WDT&SP reset	U			
	CE reset	U			
Clock stop		U			

U: Undefined

21.6.3 Stack pointer overflow/underflow reset

A reset signal is generated if the address or interrupt stack overflows or underflows.

Stack pointer overflow/underflow reset can be used to detect a program hang-up in the same manner as watchdog timer reset.

The reset signal is generated under the following conditions:

- Interrupt due to overflow or underflow of interrupt stack (4 levels)
- Interrupt due to overflow or underflow of address stack (15 levels)

Reset by stack pointer overflow or underflow initializes all the registers, except the stack overflow selection register, watchdog timer counter reset register, basic timer 0 carry register, and CE reset timer carry counter.

Generation of stack pointer overflow or underflow reset is detected by the WDTCY flag (R&Reset).

21.6.4 Stack pointer setting flag

The stack overflow/underflow reset selection register can be set only once after power-ON reset on power application or reset by the $\overline{\text{RESET}}$ pin. This register specifies whether reset by address stack overflow or underflow and reset by interrupt stack overflow or underflow are enabled or disabled.

Figure 21-14. Configuration of Stack Overflow/Underflow Reset Selection Register

Name	Flag symbol				Address	Read/Write
	b_3	b_2	b_1	b_0		
Stack overflow/underflow reset selection	0	0	ISPRES	ASPRES	05H	R/W[Note]

ASPRES	Selects address stack overflow/underflow reset
0	Disables reset
1	Enables reset

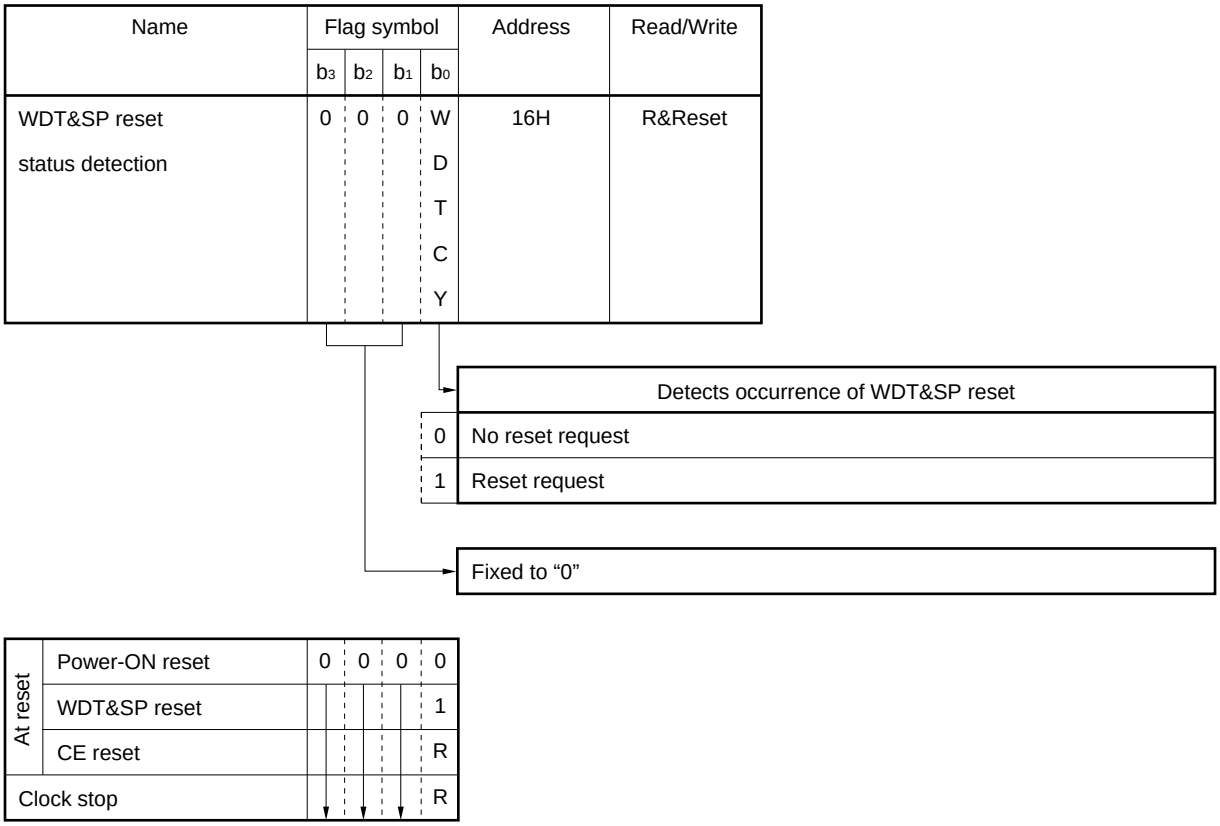
ISPRES	Selects interrupt stack overflow/underflow reset
0	Disables reset
1	Enables reset

Fixed to “0”

		b_3	b_2	b_1	b_0
At reset	Power-ON reset	0	0	1	1
	WDT&SP reset				Retained
	CE reset				Retained
Clock stop					Retained

Note Can be written only once.

Figure 21-15. Configuration of WDT&SP Reset Selection Register



		b3	b2	b1	b0
At reset	Power-ON reset	0	0	0	0
	WDT&SP reset				1
	CE reset				R
Clock stop					R

R: Retained

21.7 Power Failure Detection

Power failure detection is used to identify whether the device has been reset by application of supply voltage V_{DD}, $\overline{\text{RESET}}$ pin, or CE pin.

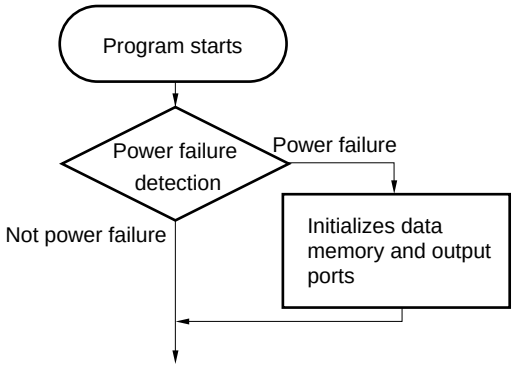
Because the contents of the data memory and output ports are “undefined” on power application, these contents are initialized by using power failure detection.

Power failure detection can be performed in two ways: by detecting the BTM0CY flag and the contents of the data memory (RAM judgment).

21.7.1 and 21.7.2 describe the power failure detection circuit and power failure detection by using the BTM0CY flag.

21.7.3 and 21.7.4 describe power failure detection by RAM judgment method.

Figure 21-16. Power Failure Detection Flowchart



21.7.1 Power failure detection circuit

The power failure detection circuit consists of a voltage detection circuit, and basic timer 0 carry disable flip-flop that is set by the output (power-ON clear signal) of the voltage detection circuit, and timer carry, as shown in Figure 21-1.

The basic timer 0 carry disable FF is set to 1 by the power-ON clear signal, and is reset to 0 when an instruction that reads the BTM0CY flag is executed.

When the basic timer 0 carry disable FF is set to 1, the BTM0CY flag is not set to 1.

If the power-ON clear signal is output (at power-ON reset), the program starts with the BTM0CY flag reset. After that, the BTM0CY flag is disabled from being set until an instruction that reads the flag is executed.

Once the instruction that reads this flag has been executed, the BTM0CY flag is set each time the basic timer 0 carry FF setting pulse rises. Therefore, by detecting the content of the BTM0CY flag when the device is reset, whether the device has been reset by power-ON reset (power failure) or CE reset (not power failure) can be identified. That is, the device has been reset by power-ON reset if the BTM0CY flag has been reset to 0. It has been reset by CE reset if the flag has been set to 1.

Because the voltage at which a power failure can be detected is the same as that at which power-ON reset is executed, V_{DD} = 3.5 V during crystal oscillation and V_{DD} = 2.2 V in the clock stop status.

The operation of the BTM0CY flag is the same regardless of whether the device has been reset by the $\overline{\text{RESET}}$ pin or by power-ON reset.

21.7.2 Cautions on detecting power failure by BTM0CY flag

The following points must be noted when counting the watch timer by using the BTM0CY flag.

(1) Updating watch

When creating a watch program using the timer carry, the watch must be updated after a power failure has been detected.

This is because the BTM0CY flag is reset to 0 because it is read after a power failure has been detected. As a result, counting of the watch is overlooked once.

(2) Watch updating processing time

Updating the watch must be completed before the next basic timer 0 carry FF setting pulse rises.

This is because CE reset is executed before the watch updating processing has been completed if the CE pin goes high during watch updating processing.

For the details of (1) and (2), refer to **(3) Compensating basic timer 0 carry at CE reset** in **13.2.6**.

The following points must be noted when performing processing in case of a power failure.

(3) Timing to detect power failure

When counting the watch by using the BTM0CY flag, the BTM0CY flag must be read to detect a power failure before the next basic timer 0 carry FF setting pulse rises after the program has been started from address 0000H.

This is because, if the basic timer 0 carry FF setting time is set to, say, 10 ms, and if the power failure is detected 11 ms after the program has been started, the BTM0CY flag is overlooked once.

For further information, refer to **(3) Compensating basic timer 0 carry at CE reset** in **13.2.6**.

Power failure detection and initial processing must be performed within the time in which the basic timer 0 carry FF is set, as shown in the example below.

This is because, if the CE pin rises and CE reset is executed during power failure processing or initial processing, the processing is stopped in midway, causing a problem.

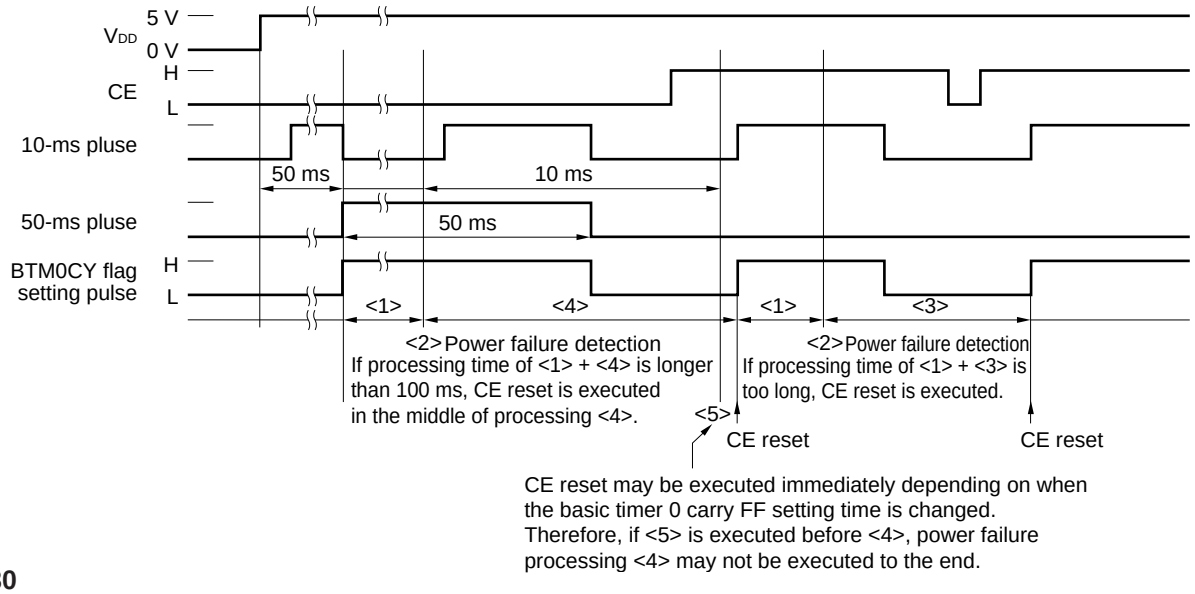
To update the basic timer 0 carry FF setting time in the initial processing, the instruction that changes the setting time must be executed at the end of the initial processing.

This is because, if the basic timer 0 carry FF setting time is changed before the initial processing, the initial processing may not be executed to the end because CE reset may be executed.

Example

```
START:                              ; Program address 0000H
; <1>
    Processing at reset
; <2>
    SKT1    BTM0CY                  ; Power failure detection
    BR      INITIAL
BACKUP:
; <3>
    Watch updating
    BR      MAIN
INITIAL:
; <4>
    Initial processing
; <5>
    INITFLG BTM0CK1, BTM0CK0        ; Embedded macro
                                    ; Sets basic timer 0 carry FF
                                    ; Sets time to 10 ms
MAIN:
    Main processing
    SKT1    BTM0CY
    BR      MAIN
    Watch updating
    BR      MAIN
```

Operation example (if CE reset timer counter is set to “1”)



21.7.3 Power failure detection by RAM judgment method

By the RAM judgment method, a power failure is detected by judging whether the contents of the data memory at a specific address are a specific value when the device has been reset.

An example of a program that detects a power failure by RAM judgment method is shown below.

By the RAM judgment method, a power failure is detected by comparing an "undefined" value and a "specific" value because the contents of the data memory are "undefined" on application of supply voltage V_{DD}.

Therefore, a power failure may be judged by mistake by this method as described in **21.7.4 Cautions on power failure detection by RAM judgment method**.

Example Program example of power failure detection by RAM judgment method

```
M012    MEM     0.12H
M034    MEM     0.34H
M056    MEM     0.56H
M107    MEM     1.07H
M128    MEM     1.28H
M16F    MEM     1.6FH
DATA0   DAT     1010B
DATA1   DAT     0101B
DATA2   DAT     0110B
DATA3   DAT     1001B
DATA4   DAT     1100B
DATA5   DAT     0011B

START:
        SET2    CMP, Z
        SUB     M012, #DATA0        ; If M012 = DATA0, and
        SUB     M034, #DATA1        ; M034 = DATA1, and
        SUB     M056, #DATA2        ; M056 = DATA2, and
        BANK1
        SUB     M107, #DATA3        ; M107 = DATA3, and
        SUB     M128, #DATA4        ; M128 = DATA4, and
        SUB     M16F, #DATA5        ; M16F = DATA5,
        BANK0
        SKF1    Z
        BR      BACKUP              ; branches to BACKUP
; INITIAL:
        Initial processing

        MOV     M012, #DATA0
        MOV     M034, #DATA1
        MOV     M056, #DATA2
        BANK1
        MOV     M107, #DATA3
        MOV     M128, #DATA4
        MOV     M16F, #DATA5
        BR      MAIN
BACKUP:
        Backup processing

MAIN:
        Main processing
```

21.7.4 Cautions on power failure detection by RAM judgment method

Because the values of the data memory on application of supply voltage V_{DD} are basically "undefined", the following points (1), (2), and (3) must be noted.

(1) Data to be compared

Where the number of bits of the data memory to be compared by the RAM judgment method is "n bits", the probability that the value of the data memory happens to coincide the value to be compared on application of V_{DD} is $(1/2)^n$.

In other words, a power failure detected by the RAM judgment method may be judged as backup at a probability of $(1/2)^n$.

To minimize this probability, compare as many bits as possible.

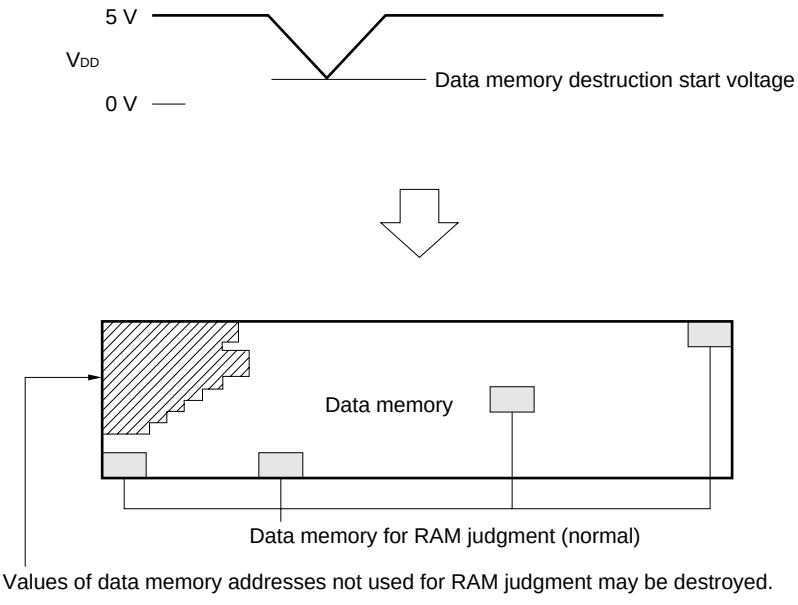
Because the contents of the data memory on application of V_{DD} are likely to be the same value such as "0000B" and "1111B", it is recommended that the data to be compared consist of a combination of "0"s and "1"s, such as "1010B" and "0110B".

(2) Cautions on program

If V_{DD} rises from a level at which the contents of the data memory are destroyed as shown in Figure 21-17, even if the value of the data memory to be compared is normal, the other parts of the data memory may be destroyed.

If a power failure detection is performed by the RAM judgment method at this time, it is judged to be a backup. Therefore, the program must be designed so that a hang-up does not occur even if the contents of the data memory are destroyed.

Figure 21-17. V_{DD} and Destruction of Data Memory Contents



(3) Cautions on $\overline{\text{RESET}}$ pin

If reset is executed by using the $\overline{\text{RESET}}$ pin during program execution, the data of the data memory may be destroyed. Therefore, exercise care when executing reset by using the $\overline{\text{RESET}}$ pin.

22. INSTRUCTION SET

22.1 Outline of Instruction Set

b_{14}-b_{11} / b_{15}		0		1	
BIN	HEX				
0000	0	ADD	r,m	ADD	m,#n4
0001	1	SUB	r,m	SUB	m, #n4
0010	2	ADDC	r,m	ADDC	m,#n4
0011	3	SUBC	r,m	SUBC	m,#n4
0100	4	AND	r,m	AND	m,#n4
0101	5	XOR	r,m	XOR	m,#n4
0110	6	OR	r,m	OR	m,#n4
0111	7	INC INC RORC MOVT PUSH POP GET PUT PEEK POKE BR CALL SYSCAL RET RETSK RETI EI DI STOP HALT NOP	AR IX r DBF,@AR AR AR DBF,p p,DBF WR,rf rf,WR @AR @AR entry s h 		
1000	8	LD	r,m	ST	m,r
1001	9	SKE	m,#n4	SKGE	m,#n4
1010	A	MOV	@r,m	MOV	m,@r
1011	B	SKNE	m,#n4	SKLT	m,#n4
1100	C	BR	addr (page 0)	CALL	addr (page 0)
1101	D	BR	addr (page 1)	MOV	m,#n4
1110	E	BR	addr (page 2)	SKT	m,#n4
1111	F	BR	addr (page 3)	SKF	m,#n

22.2 Legend

AR	:	Address register
ASR	:	Address stack register indicated by stack pointer
addr	:	Program memory address (low-order 11 bits)
BANK	:	Bank register
CMP	:	Compare flag
CY	:	Carry flag
DBF	:	Data buffer
entry	:	Program memory address (bits 10 through 8, bits 3 through 0)
$entry_H$	:	Program memory address (bits 10 through 8)
$entry_L$	:	Program memory address (bits 3 through 0)
h	:	Halt release condition
INTEF	:	Interrupt enable flag
INTR	:	Register automatically saved to stack when interrupt occurs
INTSK	:	Interrupt stack register
IX	:	Index register
MP	:	Data memory row address pointer
MPE	:	Memory pointer enable flag
m	:	Data memory address indicated by m_R, m_C
m_R	:	Data memory row address (high-order)
m_C	:	Data memory column address (low-order)
n	:	Bit position (4 bits)
n4	:	Immediate data (4 bits)
PAGE	:	Page (bits 12 and 11 of program counter)
PC	:	Program counter
P	:	Peripheral address
p_H	:	Peripheral address (high-order 3 bits)
p_L	:	Peripheral address (low-order 4 bits)
r	:	General register column address
rf	:	Register file address
rf_R	:	Register file row address (high-order 3 bits)
rf_C	:	Register file column address (low-order 4 bits)
SGR	:	Segment register (bit 13 of program counter)
SP	:	Stack pointer
s	:	Stop release condition
WR	:	Window register
(x)	:	Contents addressed by x

22.3 Instruction List

Instructions	Mnemonic	Operand	Operation	Instruction Code			
				Op code	Operand		
Add	ADD	r,m	(r) ← (r) + (m)	00000	m_R	m_C	r
		m,#n4	(m) ← (m) + n4	10000	m_R	m_C	n4
	ADDC	r,m	(r) ← (r) + (m) + CY	00010	m_R	m_C	r
		m,#n4	(m) ← (m) + n4 + CY	10010	m_R	m_C	n4
	INC	AR	AR ← AR + 1	00111	000	1001	0000
		IX	IX ← IX + 1	00111	000	1000	0000
Subtract	SUB	r,m	(r) ← (r) – (m)	00001	m_R	m_C	r
		m,#n4	(m) ← (m) – n4	10001	m_R	m_C	n4
	SUBC	r,m	(r) ← (r) – (m) – CY	00011	m_R	m_C	r
		m,#n4	(m) ← (m) – n4 – CY	10011	m_R	m_C	n4
Logical operation	OR	r,m	(r) ← (r) ∨ (m)	00110	m_R	m_C	r
		m,#n4	(m) ← (m) ∨ n4	10110	m_R	m_C	n4
	AND	r,m	(r) ← (r) ∧ (m)	00100	m_R	m_C	r
		m,#n4	(m) ← (m) ∧ n4	10100	m_R	m_C	n4
	XOR	r,m	(r) ← (r) ∀ (m)	00101	m_R	m_C	r
		m,#n4	(m) ← (m) ∀ n4	10101	m_R	m_C	n4
Judge	SKT	m,#n	CMP ← 0, if (m) ∧ n = n, then skip	11110	m_R	m_C	n
	SKF	m,#n	CMP ← 0, if (m) ∧ n = 0, then skip	11111	m_R	m_C	n
Compare	SKE	m,#n4	(m) – n4, skip if zero	01001	m_R	m_C	n4
	SKNE	m,#n4	(m) – n4, skip if not zero	01011	m_R	m_C	n4
	SKGE	m,#n4	(m) – n4, skip if not borrow	11001	m_R	m_C	n4
	SKLT	m,#n4	(m) – n4, skip if borrow	11011	m_R	m_C	n4
Rotate	RORC	r	CY ← (r) b_3 ← (r) b_2 ← (r) b_1 ← (r) b_0	00111	000	0111	r
Transfer	LD	r,m	(r) ← (m)	01000	m_R	m_C	r
	ST	m,r	(m) ← (r)	11000	m_R	m_C	r
	MOV	@r,m	if MPE = 1 : (MP, (r)) ← (m) if MPE = 0 : (BANK, m_R, (r)) ← (m)	01010	m_R	m_C	r
		m, @r	if MPE = 1 : (m) ← (MP, (r)) if MPE = 0 : (m) ← (BANK, m_R, (r))	11010	m_R	m_C	r
		m,#n4	(m) ← n4	11101	m_R	m_C	n4
	MOVT	DBF,@AR	SP ← SP – 1, ASR ← PC, PC ← AR, DBF ← (PC), PC ← ASR, SP ← SP + 1	00111	000	0001	0000
	PUSH	AR	SP ← SP – 1, ASR ← AR	00111	000	1101	0000
	POP	AR	AR ← ASR, SP ← SP + 1	00111	000	1100	0000
	GET	DBF,p	DBF ← (p)	00111	p_H	1011	p_L
	PUT	p,DBF	(p) ← DBF	00111	p_H	1010	p_L
	PEEK	WR,rf	WR ← (rf)	00111	rf_R	0011	rf_C
	POKE	rf,WR	(rf) ← WR	00111	rf_R	0010	rf_C

Instructions	Mnemonic	Operand	Operation	Instruction Code			
				Op code	Operand		
Branch	BR	addr	PC_{10-0} ← addr, PAGE ← 0	01100	addr		
			PC_{10-0} ← addr, PAGE ← 1	01101			
			PC_{10-0} ← addr, PAGE ← 2	01110			
			PC_{10-0} ← addr, PAGE ← 3	01111			
		@AR	PC ← AR	00111	000	0100	0000
Subroutine	CALL	addr	SP ← SP – 1, ASR ← PC PC_{11} ← 0, PC_{10-0} ← addr	11100	addr		
		@AR	SP ← SP – 1, ASR ← PC PC ← AR	00111	000	0101	0000
	SYSCAL	entry	SP ← SP – 1, ASR ← PC, SGR ← 1 $PC_{12, 11}$ ← 0, PC_{10-8} ← $entry_H$, PC_{7-4} ← 0, PC_{3-0} ← $entry_L$	00111	$entry_H$	0010	$entry_L$
	RET		PC ← ASR, SP ← SP + 1	00111	000	1110	0000
	RETSK		PC ← ASR, SP ← SP + 1 and skip	00111	001	1110	0000
	RETI		PC ← ASR, INTR ← INTSK, SP ← SP + 1	00111	010	1110	0000
Interrupt	EI		INTEF ← 1	00111	000	1111	0000
	DI		INTEF ← 0	00111	001	1111	0000
Others	STOP	s	STOP	00111	010	1111	s
	HALT	h	HALT	00111	011	1111	h
	NOP		No operation	00111	100	1111	0000

22.4 Assembler (RA17K) Embedded Macro Instruction

Legend

flag n : FLG symbol
n : Bit number
< > : Can be omitted

	Mnemonic	Operand	Operation	n
Embedded macro	SKTn	flag 1, ··· flag n	if (flag1) ~ (flag n) = all “1”, then skip	1 ≤ n ≤ 4
	SKFn	flag 1, ··· flag n	if (flag 1) ~ (flag n) = all “0”, then skip	1 ≤ n ≤ 4
	SETn	flag 1, ··· flag n	(flag 1) ~ (flag n) ← 1	1 ≤ n ≤ 4
	CLRn	flag 1, ··· flag n	(flag 1) ~ (flag n) ← 0	1 ≤ n ≤ 4
	NOTn	flag 1, ··· flag n	if (flag n) = “0”, then (flag n) ← 1 if (flag n) = “1”, then (flag n) ← 0	1 ≤ n ≤ 4
	INITFLG	<NOT> flag 1, ··· <<NOT> flag n>	if description = NOT flag n, then (flag n) ← 0 if description = flag n, then (flag n) ← 1	1 ≤ n ≤ 4
	BANKn		(BANK) ← n	0 ≤ n ≤ 15
Expanded instruction	BRX	Label	Jump Label	—
	CALLX	function-name	CALL sub-routine	—
	SYSCALX	function-name or expression	CALL system sub-routine	—
	INITFLGX	<NOT/INV> flag 1, ··· <NOT/INV> flag n	if description = NOT (or INV) flag, (flag) ← 0 if description = flag, (flag) ← 1	n ≤ 4

23. RESERVED SYMBOLS

23.1 Data Buffer (DBF)

Symbol Name	Attribute	Value	R/W	Description
DBF3	MEM	0.0CH	R/W	Bits 15 through 12 of data buffer
DBF2	MEM	0.0DH	R/W	Bits 11 through 8 of data buffer
DBF1	MEM	0.0EH	R/W	Bits 7 through 4 of data buffer
DBF0	MEM	0.0FH	R/W	Bits 3 through 0 of data buffer

23.2 System Registers (SYSREG)

Symbol Name	Attribute	Value	R/W	Description
AR3	MEM	0.74H	R/W	Bits 15 through 12 of address register
AR2	MEM	0.75H	R/W	Bits 11 through 8 of address register
AR1	MEM	0.76H	R/W	Bits 7 through 4 of address register
AR0	MEM	0.77H	R/W	Bits 3 through 0 of address register
WR	MEM	0.78H	R/W	Window register
BANK	MEM	0.79H	R/W	Bank register
IXH	MEM	0.7AH	R/W	Bits 10 through 8 of index register
MPH	MEM	0.7AH	R/W	Bits 6 through 4 of memory pointer
MPE	FLG	0.7AH.3	R/W	Memory pointer enable flag
IXM	MEM	0.7BH	R/W	Bits 7 through 4 of index register
MPL	MEM	0.7BH	R/W	Bits 3 through 0 of memory pointer
IXL	MEM	0.7CH	R/W	Bits 3 through 0 of index register
RPH	MEM	0.7DH	R/W	Bits 6 through 3 of general register pointer
RPL	MEM	0.7EH	R/W	Bits 2 through 0 of general register pointer
BCD	FLG	0.7EH.0	R/W	BCD operation flag
PSW	MEM	0.7FH	R/W	Program status word
CMP	FLG	0.7FH.3	R/W	Compare flag
CY	FLG	0.7FH.2	R/W	Carry flag
Z	FLG	0.7FH.1	R/W	Zero flag
IXE	FLG	0.7FH.0	R/W	Index enable flag

23.3 Port Registers

Symbol Name	Attribute	Value	R/W	Description
P0A3	FLG	0.70H.3	R/W	Bit 3 of port 0A
P0A2	FLG	0.70H.2	R/W	Bit 2 of port 0A
P0A1	FLG	0.70H.1	R/W	Bit 1 of port 0A
P0A0	FLG	0.70H.0	R/W	Bit 0 of port 0A
P0B3	FLG	0.71H.3	R/W	Bit 3 of port 0B
P0B2	FLG	0.71H.2	R/W	Bit 2 of port 0B
P0B1	FLG	0.71H.1	R/W	Bit 1 of port 0B
P0B0	FLG	0.71H.0	R/W	Bit 0 of port 0B
P0C3	FLG	0.72H.3	R/W	Bit 3 of port 0C
P0C2	FLG	0.72H.2	R/W	Bit 2 of port 0C
P0C1	FLG	0.72H.1	R/W	Bit 1 of port 0C
P0C0	FLG	0.72H.0	R/W	Bit 0 of port 0C
P0D3	FLG	0.73H.3	R[Note]	Bit 3 of port 0D
P0D2	FLG	0.73H.2	R[Note]	Bit 2 of port 0D
P0D1	FLG	0.73H.1	R[Note]	Bit 1 of port 0D
P0D0	FLG	0.73H.0	R[Note]	Bit 0 of port 0D
P1A3	FLG	1.70H.3	R[Note]	Bit 3 of port 1A
P1A2	FLG	1.70H.2	R[Note]	Bit 2 of port 1A
P1A1	FLG	1.70H.1	R[Note]	Bit 1 of port 1A
P1A0	FLG	1.70H.0	R[Note]	Bit 0 of port 1A
P1B3	FLG	1.71H.3	R/W	Bit 3 of port 1B
P1B2	FLG	1.71H.2	R/W	Bit 2 of port 1B
P1B1	FLG	1.71H.1	R/W	Bit 1 of port 1B
P1B0	FLG	1.71H.0	R/W	Bit 0 of port 1B
P1C3	FLG	1.72H.3	R[Note]	Bit 3 of port 1C
P1C2	FLG	1.72H.2	R[Note]	Bit 2 of port 1C
P1C1	FLG	1.72H.1	R[Note]	Bit 1 of port 1C
P1C0	FLG	1.72H.0	R[Note]	Bit 0 of port 1C

Note These are input ports. However, even if an instruction that outputs data to these ports is described, the assembler and in-circuit emulator do not output an error message. Moreover, nothing is affected in terms of operation even if such an instruction is actually executed on the device.

Symbol Name	Attribute	Value	R/W	Description
P1D3	FLG	1.73H.3	R/W	Bit 3 of port 1D
P1D2	FLG	1.73H.2	R/W	Bit 2 of port 1D
P1D1	FLG	1.73H.1	R/W	Bit 1 of port 1D
P1D0	FLG	1.73H.0	R/W	Bit 0 of port 1D
P2A2	FLG	2.70H.2	R/W	Bit 2 of port 2A
P2A1	FLG	2.70H.1	R/W	Bit 1 of port 2A
P2A0	FLG	2.70H.0	R/W	Bit 0 of port 2A
P2B3	FLG	2.71H.3	R/W	Bit 3 of port 2B
P2B2	FLG	2.71H.2	R/W	Bit 2 of port 2B
P2B1	FLG	2.71H.1	R/W	Bit 1 of port 2B
P2B0	FLG	2.71H.0	R/W	Bit 0 of port 2B
P2C3	FLG	2.72H.3	R/W	Bit 3 of port 2C
P2C2	FLG	2.72H.2	R/W	Bit 2 of port 2C
P2C1	FLG	2.72H.1	R/W	Bit 1 of port 2C
P2C0	FLG	2.72H.0	R/W	Bit 0 of port 2C
P2D2	FLG	2.73H.2	R/W	Bit 2 of port 2D
P2D1	FLG	2.73H.1	R/W	Bit 1 of port 2D
P2D0	FLG	2.73H.0	R/W	Bit 0 of port 2D
P3A3	FLG	3.70H.3	R/W	Bit 3 of port 3A
P3A2	FLG	3.70H.2	R/W	Bit 2 of port 3A
P3A1	FLG	3.70H.1	R/W	Bit 1 of port 3A
P3A0	FLG	3.70H.0	R/W	Bit 0 of port 3A
P3B3	FLG	3.71H.3	R/W	Bit 3 of port 3B
P3B2	FLG	3.71H.2	R/W	Bit 2 of port 3B
P3B1	FLG	3.71H.1	R/W	Bit 1 of port 3B
P3B0	FLG	3.71H.0	R/W	Bit 0 of port 3B
P3C3	FLG	3.72H.3	R/W	Bit 3 of port 3C
P3C2	FLG	3.72H.2	R/W	Bit 2 of port 3C
P3C1	FLG	3.72H.1	R/W	Bit 1 of port 3C
P3C0	FLG	3.72H.0	R/W	Bit 0 of port 3C
P3D3	FLG	3.73H.3	R/W	Bit 3 of port 3D
P3D2	FLG	3.72H.2	R/W	Bit 2 of port 3D
P3D1	FLG	3.73H.1	R/W	Bit 1 of port 3D
P3D0	FLG	3.73H.0	R/W	Bit 0 of port 3D

23.4 Register File (Control Registers)

Symbol Name	Attribute	Value	R/W	Description
SP	MEM	0.81H	R/W	Stack pointer
WDTCK	MEM	0.82H	R/W	Watchdog timer clock selection flag (can be set only once after power application)
WDTCK1	FLG	0.82H.1	R/W	Watchdog timer clock selection flag (can be set only once after power application)
WDTCK0	FLG	0.82H.0	R/W	Watchdog timer clock selection flag (can be set only once after power application)
WDTRES	FLG	0.83H.3	R/W	Watchdog timer counter reset (when read: 0)
DBFSP	MEM	0.84H	R	DBF stack pointer
SPRSEL	MEM	0.85H	R/W	Stack overflow/underflow reset selection flag (can be set only once after power application)
ISPRES	FLG	0.85H.1	R/W	Stack overflow/underflow reset selection flag (can be set only once after power application)
ASPRES	FLG	0.85H.0	R/W	Stack overflow/underflow reset selection flag (can be set only once after power application)
CECNT3	FLG	0.86H.3	R/W	CE reset timer carry counter
CECNT2	FLG	0.86H.2	R/W	CE reset timer carry counter
CECNT1	FLG	0.86H.1	R/W	CE reset timer carry counter
CECNT0	FLG	0.86H.0	R/W	CE reset timer carry counter
MOVTSEL1	FLG	0.87H.1	R/W	MOVT bit selection flag
MOVTSEL0	FLG	0.87H.0	R/W	MOVT bit selection flag
SYSRSP	MEM	0.88H	R	System register stack pointer
SIO2CLC	FLG	0.8AH.3	R/W	Serial interface 2 clock level control flag
SIO2WREL	FLG	0.8AH.2	R/W	Serial interface 2 wait release control flag
SIO2WAT1	FLG	0.8AH.1	R/W	Serial interface 2 interrupt generation timing/wait control flag
SIO2WAT0	FLG	0.8AH.0	R/W	Serial interface 2 interrupt generation timing/wait control flag
SIO2CLD	FLG	0.8BH.2	R	Serial interface 2 clock pin level detection flag
SIO2SIC	FLG	0.8BH.1	R/W	Serial interface 2 interrupt source selection flag
SIO2SVAM	FLG	0.8BH.0	R/W	Serial interface 2 address mask function specification flag
SIO2CMDD	FLG	0.8CH.3	R	Serial interface 2 command signal detection flag
SIO2RELD	FLG	0.8CH.2	R	Serial interface 2 bus release signal detection flag
SIO2CMDT	FLG	0.8CH.1	R/W	Serial interface 2 command signal trigger output control flag
SIO2RELT	FLG	0.8CH.0	R/W	Serial interface 2 bus release signal trigger output control flag
SIO2BSYE	FLG	0.8DH.3	R/W	Serial interface 2 synchronization busy signal enable flag
SIO2ACKD	FLG	0.8DH.2	R	Serial interface 2 acknowledge detection flag
SIO2ACKE	FLG	0.8DH.1	R/W	Serial interface 2 acknowledge enable flag
SIO2ACKT	FLG	0.8DH.0	R/W	Serial interface 2 acknowledge signal trigger output control flag
SIO2WUP	FLG	0.8EH.3	R/W	Serial interface 2 wake-up function specification flag
SIO2MD2	FLG	0.8EH.2	R/W	Serial interface 2 operation mode selection flag
SIO2MD1	FLG	0.8EH.1	R/W	Serial interface 2 operation mode selection flag
SIO2MD0	FLG	0.8EH.0	R/W	Serial interface 2 clock direction selection flag

Symbol Name	Attribute	Value	R/W	Description
SIO2CSIE	FLG	0.8FH.3	R/W	Serial interface 2 operation enable/disable flag
SIO2COI	FLG	0.8FH.2	R	Coincidence signal detection flag from serial interface 2 address comparator
SIO2TCL1	FLG	0.8FH.1	R/W	Serial interface 2 clock selection flag
SIO2TCL0	FLG	0.8FH.0	R/W	Serial interface 2 clock selection flag
PLLSCNF	FLG	0.90H.3	R/W	Swallow counter least significant bit setting flag
PLLMD1	FLG	0.90H.1	R/W	PLL mode selection flag
PLLMD0	FLG	0.90H.0	R/W	PLL mode selection flag
PLLRFCK3	FLG	0.91H.3	R/W	PLL reference frequency selection flag
PLLRFCK2	FLG	0.91H.2	R/W	PLL reference frequency selection flag
PLLRFCK1	FLG	0.91H.1	R/W	PLL reference frequency selection flag
PLLRFCK0	FLG	0.91H.0	R/W	PLL reference frequency selection flag
PLLUL	FLG	0.92H.0	R&Reset	PLL unlock FF flag
BEEP1SEL	FLG	0.93H.1	R/W	BEEP1/general-purpose port pin function selection flag
BEEP0SEL	FLG	0.93H.0	R/W	BEEP0/general-purpose port pin function selection flag
BEEP1CK1	FLG	0.94H.3	R/W	BEEP1 clock selection flag
BEEP1CK0	FLG	0.94H.2	R/W	BEEP1 clock selection flag
BEEP0CK1	FLG	0.94H.1	R/W	BEEP0 clock selection flag
BEEP0CK0	FLG	0.94H.0	R/W	BEEP0 clock selection flag
WDTCY	FLG	0.96H.0	R	Watchdog timer/stack pointer reset status detection flag
BTM0CY	FLG	0.97H.0	R	Basic timer 0 carry flag
BTM0CK1	FLG	0.98H.1	R/W	Basic timer 0 clock selection flag
BTM0CK0	FLG	0.98H.0	R/W	Basic timer 0 clock selection flag
SIO3CSIE	FLG	0.9AH.3	R/W	Serial interface 3 operation enable/disable flag
SIO3HIZ	FLG	0.9AH.2	R/W	Serial interface 3 SO3 pin status setting flag
SIO3TCL1	FLG	0.9AH.1	R/W	Serial interface 3 clock selection flag
SIO3TCL0	FLG	0.9AH.0	R/W	Serial interface 3 clock selection flag
SIO3PE	FLG	0.9BH.2	R	Serial interface 3 parity error flag
SIO3FE	FLG	0.9BH.1	R	Serial interface 3 framing error flag
SIO3OVE	FLG	0.9BH.0	R	Serial interface 3 overrun error flag
SIO3PS1	FLG	0.9CH.3	R/W	Parity bit specification flag of UART
SIO3PS0	FLG	0.9CH.2	R/W	Parity bit specification flag of UART
SIO3CL	FLG	0.9CH.1	R/W	Character length specification flag of UART
SIO3SL	FLG	0.9CH.0	R/W	Number of stop bits specification flag of UART transmission data
SIO3TXE	FLG	0.9DH.3	R/W	UART transmission mode enable flag
SIO3RXE	FLG	0.9DH.2	R/W	UART reception mode enable flag
SIO3ISRM	FLG	0.9DH.1	R/W	Reception completion interrupt enable flag in the case of error
IEG4	FLG	0.9EH.3	R/W	Edge direction selection flag for INT4 pin interrupt request detection
INT4SEL	FLG	0.9EH.2	R/W	INT4 pin interrupt request flag setting disable
IEG3	FLG	0.9EH.1	R/W	Edge direction selection flag for INT3 pin interrupt request detection
INT3SEL	FLG	0.9EH.0	R/W	INT3 pin interrupt request flag setting disable

Symbol Name	Attribute	Value	R/W	Description
IEG2	FLG	0.9FH.2	R/W	Edge direction selection flag for INT2 pin interrupt request detection
IEG1	FLG	0.9FH.1	R/W	Edge direction selection flag for INT1 pin interrupt request detection
IEG0	FLG	0.9FH.0	R/W	Edge direction selection flag for INT0 pin interrupt request detection
FCGCH1	FLG	0.0A0H.1	R/W	FGC channel selection flag
FCGCH0	FLG	0.0A0H.0	R/W	FGC channel selection flag
IFCGOSTT	FLG	0.0A1H.0	R	IF counter gate status detection flag (1: Open, 0: Closed)
IFCMD1	FLG	0.0A2H.3	R/W	IF counter mode selection flag (10: AMIF, 11: FCG)
IFCMD0	FLG	0.0A2H.2	R/W	IF counter mode selection flag (00: CGP, 11: FMIF)
IFCCK1	FLG	0.0A2H.1	R/W	IF counter clock selection flag
IFCCK0	FLG	0.0A2H.0	R/W	IF counter clock selection flag
IFCSTRT	FLG	0.0A3H.1	W	IF counter count start flag
IFCRES	FLG	0.0A3H.0	W	IF counter reset flag
ADCCH3	FLG	0.0A4H.3	R/W	A/D converter channel selection flag (dummy)
ADCCH2	FLG	0.0A4H.2	R/W	A/D converter channel selection flag
ADCCH1	FLG	0.0A4H.1	R/W	A/D converter channel selection flag
ADCCH0	FLG	0.0A4H.0	R/W	A/D converter channel selection flag
ADCMD	FLG	0.0A5H.2	R/W	A/D converter compare mode selection flag
ADCSTT	FLG	0.0A5H.1	R	A/D converter operation status detection flag (0: End of conversion, 1: Conversion in progress)
ADCCMP	FLG	0.0A5H.0	R	A/D converter compare result detection flag
PWMBIT	FLG	0.0A6H.2	R/W	PWM counter bit selection flag (0: 8 bits, 1: 9 bits)
PWMCK	FLG	0.0A6H.0	R/W	PWM timer output clock selection flag
PWM2SEL	FLG	0.0A7H.2	R/W	PWM2/general-purpose port pin function selection flag
PWM1SEL	FLG	0.0A7H.1	R/W	PWM1/general-purpose port pin function selection flag
PWM0SEL	FLG	0.0A7H.0	R/W	PWM0/general-purpose port pin function selection flag
TM3SEL	FLG	0.0A8H.3	R/W	PWM/modulo timer 3 selection flag
TM3EN	FLG	0.0A8H.1	R/W	Modulo timer 3 count start flag
TM3RES	FLG	0.0A8H.0	R/W	Modulo timer 3 reset flag (when read: 0)
TM2EN	FLG	0.0A9H.3	R/W	Modulo timer 2 count start flag
TM2RES	FLG	0.0A9H.2	R/W	Modulo timer 2 reset flag (when read: 0)
TM2CK1	FLG	0.0A9H.1	R/W	Modulo timer 2 clock selection flag
TM2CK0	FLG	0.0A9H.0	R/W	Modulo timer 2 clock selection flag
TM1EN	FLG	0.0AAH.3	R/W	Modulo timer 1 count start flag
TM1RES	FLG	0.0AAH.2	R/W	Modulo timer 1 reset flag (when read: 0)
TM1CK1	FLG	0.0AAH.1	R/W	Modulo timer 1 clock selection flag
TM1CK0	FLG	0.0AAH.0	R/W	Modulo timer 1 clock selection flag
TM0EN	FLG	0.0ABH.3	R/W	Modulo timer 0 count start flag
TM0RES	FLG	0.0ABH.2	R/W	Modulo timer 0 reset flag (when read: 0)
TM0CK1	FLG	0.0ABH.1	R/W	Modulo timer 0 clock selection flag
TM0CK0	FLG	0.0ABH.0	R/W	Modulo timer 0 clock selection flag

Symbol Name	Attribute	Value	R/W	Description
TM0OVF	FLG	0.0ACH.3	R	Modulo timer 0 overflow detection flag
TM0GCEG	FLG	0.0ACH.2	R/W	Modulo timer 0 gate close input signal edge selection flag
TM0GOEG	FLG	0.0ACH.1	R/W	Modulo timer 0 gate open input signal edge selection flag
TM0MD	FLG	0.0ACH.0	R/W	Modulo timer 0 modulo counter/gate counter selection flag
IPSIO3	FLG	0.0ADH.3	R/W	Serial interface 3 interrupt enable flag
IPSIO2	FLG	0.0ADH.2	R/W	Serial interface 2 interrupt enable flag
IPTM3	FLG	0.0ADH.1	R/W	PWM timer interrupt enable flag
IPTM2	FLG	0.0ADH.0	R/W	Modulo timer 2 interrupt enable flag
IPTM1	FLG	0.0AEH.3	R/W	Modulo timer 1 interrupt enable flag
IPTM0	FLG	0.0AEH.2	R/W	Modulo timer 0 interrupt enable flag
IP4	FLG	0.0AEH.1	R/W	INT4 pin interrupt enable flag
IP3	FLG	0.0AEH.0	R/W	INT3 pin interrupt enable flag
IP2	FLG	0.0AFH.3	R/W	INT2 pin interrupt enable flag
IP1	FLG	0.0AFH.2	R/W	INT1 pin interrupt enable flag
IP0	FLG	0.0AFH.1	R/W	INT0 pin interrupt enable flag
IPCE	FLG	0.0AFH.0	R/W	CE pin interrupt enable flag
IRQSIO3	FLG	0.0B4H.0	R/W	Serial interface 3 interrupt request detection flag
IRQSIO2	FLG	0.0B5H.0	R/W	Serial interface 2 interrupt request detection flag
IRQTM3	FLG	0.0B6H.0	R/W	PWM timer interrupt request detection flag
IRQTM2	FLG	0.0B7H.0	R/W	Modulo timer 2 interrupt request detection flag
IRQTM1	FLG	0.0B8H.0	R/W	Modulo timer 1 interrupt request detection flag
IRQTM0	FLG	0.0B9H.0	R/W	Modulo timer 0 interrupt request detection flag
INT4	FLG	0.0BAH.3	R	INT4 pin status detection flag
IRQ4	FLG	0.0BAH.0	R/W	INT4 pin interrupt request detection flag
INT3	FLG	0.0BBH.3	R	INT3 pin status detection flag
IRQ3	FLG	0.0BBH.0	R/W	INT3 pin interrupt request detection flag
INT2	FLG	0.0BCH.3	R	INT2 pin status detection flag
IRQ2	FLG	0.0BCH.0	R/W	INT2 pin interrupt request detection flag
INT1	FLG	0.0BDH.3	R	INT1 pin status detection flag
IRQ1	FLG	0.0BDH.0	R/W	INT1 pin interrupt request detection flag
INT0	FLG	0.0BEH.3	R	INT0 pin status detection flag
IRQ0	FLG	0.0BEH.0	R/W	INT0 pin interrupt request detection flag
CE	FLG	0.0BFH.3	R	CE pin status detection flag
CECNTSTT	FLG	0.0BFH.1	R	CE reset counter status detection flag
IRQCE	FLG	0.0BFH.0	R/W	CE pin interrupt request detection flag
P0DPLD3	FLG	15.66H.3	R/W	P0D3 pin pull-down resistor selection flag
P0DPLD2	FLG	15.66H.2	R/W	P0D2 pin pull-down resistor selection flag
P0DPLD1	FLG	15.66H.1	R/W	P0D1 pin pull-down resistor selection flag
P0DPLD0	FLG	15.66H.0	R/W	P0D0 pin pull-down resistor selection flag

Symbol Name	Attribute	Value	R/W	Description
P3DGIO	FLG	15.67H.3	R/W	P3D input/output selection flag
P3CGIO	FLG	15.67H.2	R/W	P3C input/output selection flag
P3BGIO	FLG	15.67H.1	R/W	P3B input/output selection flag
P3AGIO	FLG	15.67H.0	R/W	P3A input/output selection flag
P2DBIO3	FLG	15.68H.3	R/W	P2D3 input/output selection flag (dummy)
P2DBIO2	FLG	15.68H.2	R/W	P2D2 input/output selection flag
P2DBIO1	FLG	15.68H.1	R/W	P2D1 input/output selection flag
P2DBIO0	FLG	15.68H.0	R/W	P2D0 input/output selection flag
P2CBIO3	FLG	15.69H.3	R/W	P2C3 input/output selection flag
P2CBIO2	FLG	15.69H.2	R/W	P2C2 input/output selection flag
P2CBIO1	FLG	15.69H.1	R/W	P2C1 input/output selection flag
P2CBIO0	FLG	15.69H.0	R/W	P2C0 input/output selection flag
P2BBIO3	FLG	15.6AH.3	R/W	P2B3 input/output selection flag
P2BBIO2	FLG	15.6AH.2	R/W	P2B2 input/output selection flag
P2BBIO1	FLG	15.6AH.1	R/W	P2B1 input/output selection flag
P2BBIO0	FLG	15.6AH.0	R/W	P2B0 input/output selection flag
P2ABIO3	FLG	15.6BH.3	R/W	P2A3 input/output selection flag (dummy)
P2ABIO2	FLG	15.6BH.2	R/W	P2A2 input/output selection flag
P2ABIO1	FLG	15.6BH.1	R/W	P2A1 input/output selection flag
P2ABIO0	FLG	15.6BH.0	R/W	P2A0 input/output selection flag
P1DBIO3	FLG	15.6CH.3	R/W	P1D3 input/output selection flag
P1DBIO2	FLG	15.6CH.2	R/W	P1D2 input/output selection flag
P1DBIO1	FLG	15.6CH.1	R/W	P1D1 input/output selection flag
P1DBIO0	FLG	15.6CH.0	R/W	P1D0 input/output selection flag
P0CBIO3	FLG	15.6DH.3	R/W	P0C3 input/output selection flag
P0CBIO2	FLG	15.6DH.2	R/W	P0C2 input/output selection flag
P0CBIO1	FLG	15.6DH.1	R/W	P0C1 input/output selection flag
P0CBIO0	FLG	15.6DH.0	R/W	P0C0 input/output selection flag
P0BBIO3	FLG	15.6EH.3	R/W	P0B3 input/output selection flag
P0BBIO2	FLG	15.6EH.2	R/W	P0B2 input/output selection flag
P0BBIO1	FLG	15.6EH.1	R/W	P0B1 input/output selection flag
P0BBIO0	FLG	15.6EH.0	R/W	P0B0 input/output selection flag
P0ABIO3	FLG	15.6FH.3	R/W	P0A3 input/output selection flag
P0ABIO2	FLG	15.6FH.2	R/W	P0A2 input/output selection flag
P0ABIO1	FLG	15.6FH.1	R/W	P0A1 input/output selection flag
P0ABIO0	FLG	15.6FH.0	R/W	P0A0 input/output selection flag

23.5 Peripheral Hardware Registers

Symbol Name	Attribute	Value	R/W	Description
ADCR	DAT	02H	R/W	A/D converter reference voltage setting register
SIO2SFR	DAT	03H	R/W	Presettable shift register 2
SIO2SVA	DAT	04H	R/W	Serial interface 2 slave address register
SIO3TXS	DAT	05H	W	Serial interface 3 transmission register
SIO3RXB	DAT	05H	R	Serial interface 3 receive buffer register
TM0M	DAT	1AH	R/W	Timer modulo 0 register
TM0C	DAT	1BH	R	Timer modulo 0 counter
TM1M	DAT	1CH	R/W	Timer modulo 1 register
TM1C	DAT	1DH	R	Timer modulo 1 counter
TM2M	DAT	1EH	R/W	Timer modulo 2 register
TM2C	DAT	1FH	R	Timer modulo 2 counter
AR	DAT	40H	R/W	Address register
DBFSTK	DAT	41H	R/W	DBF stack register
PLLR	DAT	42H	R/W	PLL data register
IFC	DAT	43H	R	IF counter data register
PWMR0	DAT	44H	R/W	PWM0 data register
PWMR1	DAT	45H	R/W	PWM1 data register
PWMR2	DAT	46H	R/W	PWM2 data register
TM3M	DAT	46H	R/W	Timer modulo 3 register

23.6 Others

Symbol Name	Attribute	Value	Description
DBF	DAT	0FH	Operand of GET/PUT/MOVT/MOVTH/MOVL instruction (DBF)
IX	DAT	01H	Operand of INC instruction (IX)
AR_EPA1	DAT	8040H	Operand of CALL/BR/MOVT/MOVTH/MOVTL instruction (EPA bit on)
AR_EPA0	DAT	4040H	Operand of CALL/BR/MOVT/MOVTH/MOVTL instruction (EPA bit off)

24. ELECTRICAL CHARACTERISTICS (PRELIMINARY)

Absolute Maximum Ratings (T_A = 25 °C)

Parameter	Symbol	Condition	Rating	Unit
Supply voltage	V_{DD}		–0.3 to +6.0	V
Input voltage	V_I	Other than CE, INT0 through INT4, and $\overline{\text{RESET}}$ pins	–0.3 to V_{DD}+0.3	V
		CE, INT0 through INT4, and $\overline{\text{RESET}}$ pins	–0.3 to V_{DD}+0.6	V
Output voltage	V_O	Except P1B0 through P1B3	–0.3 to V_{DD}+0.3	V
High-level output current	I_{OH}	1 pin	–8.0	mA
		Total of P2A0 through P2A2, P3A0 through P3A3, and P3B0 through P3B3	–15.0	mA
		Total of P0A0, P0A1, P0B0 through P0B3, P0C0 through P0C3, P1D0 through P1D3, P2B0 through P2B3, P2C0 through P2C3, P2D2, P3C0 through P3C3, and P3D0 through P3D3	–25.0	mA
Low-level output current	I_{OL}	1 pin of P1B0 through P1B3	12.0	mA
		1 pin of P1B0 through P1B3	8.0	mA
		Total of P2A0 through P2A2, P3A0 through P3A3, and P3B0 through P3B3	15.0	mA
		Total of P0A0 through P0A3, P0B0 through P0B3, P0C0 through P0C3, P1D0 through P1D3, P2B0 through P2B3, P2C0 through P2C3, P2D0 through P2D2, P3C0 through P3C3, and P3D0 through P3D3	25.0	mA
		Total of P1B0 through P1B3 pins	25.0	mA
Output voltage	V_{BDS}	P1B0-P1B3	14.0	V
Total power dissipation	P_t		200	mW
Operating ambient temperature	T_A		–40 to +85	°C
Storage temperature	T_{stg}		–55 to +125	°C

Caution If the rated value of even one of the above parameters is exceeded even momentarily, the quality of the product may be degraded. The absolute maximum ratings define the rated values exceeding which the product may be physically damaged. Never exceed these ratings.

Recommended Operating Range (T_A = –40 to +85 °C)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Supply voltage	V_{DD1}	When CPU and PLL are operating	4.5	5.0	5.5	V
	V_{DD2}	When CPU and PLL are stopped	3.5	5.0	5.5	V

Recommended Output Voltage (T_A = –40 to +85 °C)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Output voltage	V_{BDS}	P1B0-P1B3			12	V

DC Characteristics (T_A = –40 to +85 °C, V_{DD} = 3.5 to 5.5 V)

Parameter	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Supply current	I_{DD1}	When CPU is operating and PLL is stopped with sine wave input to X_{IN} pin. (f_{IN} = 4.5 MHz±1%, V_{IN} = V_{DD})			1.5	3.0	mA
	I_{DD2}	When CPU and PLL are stopped with sine wave input to X_{IN} pin. (f_{IN} = 4.5 MHz±1%, V_{IN} = V_{DD}) With HALT instruction			0.7	1.5	mA
Data retention voltage	V_{DDR1}	Crystal oscillation		3.5		5.5	V
	V_{DDR2}	Crystal oscillation stops	Power failure detection by timer FF	2.2		5.5	V
	V_{DDR3}		Data memory retained	2.0		5.5	V
Data retention current	I_{DDR1}	Crystal oscillation stops	V_{DD} = 5 V, T_A = 25 °C		2.0	4.0	μA
	I_{DDR2}				2.0	30.0	μA
High-level input voltage	V_{IH1}	P0A0, P0B1, P0C0-P0C3, P1A0, P1A1, P1C0-P1C3, P1D0-P1D3, P2A2, P2B0-P2B3, P2C0-P2C3, P3A0-P3A3, P3B0-P3B3, P3C0-P3C3, P3D0-P3D3		0.7V_{DD}		V_{DD}	V
	V_{IH2}	P0A1-P0A3, P0B0, P0B2, P0B3, P2A0, P2A1, P2D0-P2D2, CE, INT0-INT4, $\overline{\text{RESET}}$		0.8V_{DD}		V_{DD}	V
	V_{IH3}	P0D0-P0D3		0.55V_{DD}		V_{DD}	V
Low-level input voltage	V_{IL1}	P0A0, P0B1, P0C0-P0C3, P1A0, P1A1, P1C0-P1C3, P1D0-P1D3, P2A2, P2B0-P2B3, P2C0-P2C3, P3A0-P3A3, P3B0-P3B3, P3C0-P3C3, P3D0-P3D3		0		0.3V_{DD}	V
	V_{IL2}	P0A1-P0A3, P0B0, P0B2, P0B3, P2A0, P2A1, P2D0-P2D2, CE, INT0-INT4, $\overline{\text{RESET}}$		0		0.2V_{DD}	V
	V_{IL3}	P0D0-P0D3		0		0.15V_{DD}	V
High-level output current	I_{OH1}	P0A0, P0A1, P0B0-P0B3, P0C0-P0C3, P1D0-P1D3, P2A0-P2A2, P2B0-P2B3, P2C0-P2C3, P2D2, P3A0-P3A3, P3B0-P3B3, P3C0-P3C3, P3D0-P3D3	V_{OH} = V_{DD}–1 V	–1.0			mA
	I_{OH2}	EO0, EO1	V_{DD} = 4.5 to 5.5 V, V_{OH} = V_{DD}–1 V	–3.0			mA
Low-level output current	I_{OL1}	P0A0-P0A3, P0B0-P0B3, P0C0-P0C3, P1D0-P1D3, P2A0-P2A2, P2B0-P2B3, P2C0-P2C3, P2D0-P2D2, P3A0-PA3A, P3B0-P3B3, P3C0-P3C3, P3D0-P3D3	V_{OL} = 1 V	1.0			mA
	I_{OL2}	EO0, EO1	V_{DD} = 4.5 to 5.5 V, V_{OL} = 1 V	3.0			mA
	I_{OL3}	P1B0-P1B3	V_{OL} = 1 V	7.0			mA
High-level input current	I_{IH}	P0D0 through P0D3 pulled down	V_{IN} = V_{DD}	5.0		150	μA
Output off leakage current	I_{LO1}	P1B0-P1B3	V_{IN} = 12 V			1.0	μA
	I_{LO2}	EO0, EO1	V_{IN} = V_{DD}, V_{IN} = 0 V			±1.0	μA
High-level input leakage current	I_{LIH}	Input pin	V_{IN} = V_{DD}			1.0	μA
Low-level input leakage current	I_{LIL}	Input pin	V_{IN} = 0 V			–1.0	μA

AC Characteristics (T_A = –40 to +85 °C, V_{DD} = 5 V±10%)

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Operating frequency	f_{IN1}	VCOL pin, MF mode, sine wave input V_{IN} = 0.1 V_{p-p} Note	0.5		3	MHz
	f_{IN2}	VCOL pin, HF mode, sine wave input V_{IN} = 0.1 V_{p-p} Note	10		40	MHz
	f_{IN3}	VCOH pin, VHF mode, sine wave input V_{IN} = 0.1 V_{p-p} Note	60		130	MHz
	f_{IN4}	AMIFC pin, sine wave input V_{IN} = 0.15 V_{p-p} Note	0.4		0.5	MHz
	f_{IN5}	FMIFC pin, FMIF count mode, sine wave input V_{IN} = 0.20 V_{p-p}	10		11	MHz
	f_{IN6}	FMIFC pin, AMIF count mode, sine wave input V_{IN} = 0.15 V_{p-p}	0.4		0.5	MHz
SIO2 input frequency	f_{IN7}	External clock			1	MHz
SIO3 input frequency	f_{IN8}	External clock			0.7	MHz

Note The condition of sine wave input V_{IN} = 0.1 V_{p-p} is the rated value when the μPD17717, 17718, or 17719 alone is operating. Where influence of noise must be taken into consideration, operation under input amplitude condition of V_{IN} = 0.15 V_{p-p} is recommended.

A/D Converter Characteristics (T_A = –40 to +85 °C, V_{DD} = 5 V±10%)

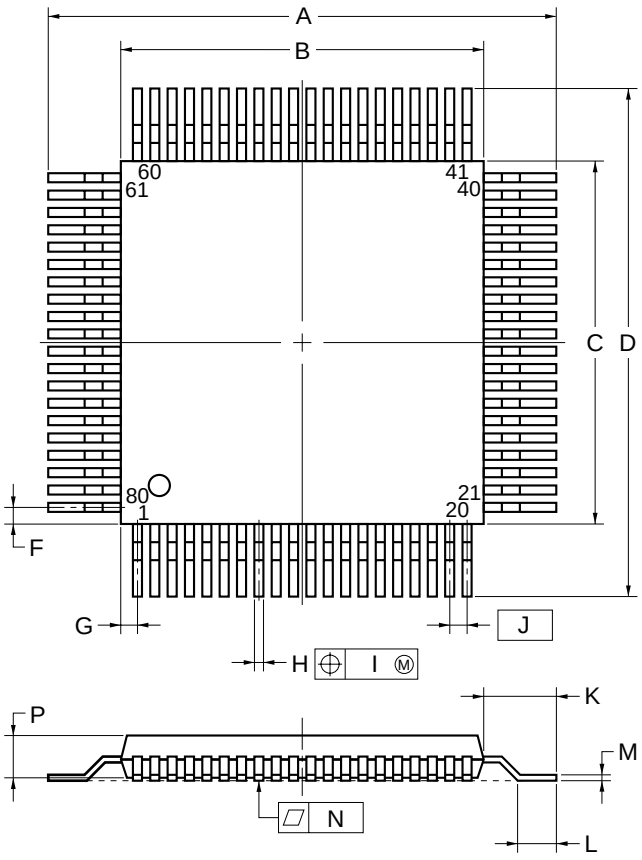
Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
A/D conversion total error		8 BIT			±3.0	LSB
A/D conversion total error		8 BIT T_A = 0 to 85 °C			±2.5	LSB

Reference Characteristics (T_A = +25 °C, V_{DD} = 5.0 V)

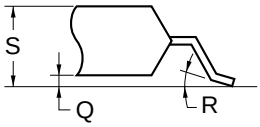
Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Supply current	I_{DD3}	When CPU and PLL are operating with sine wave input to VCOH pin (f_{IN} = 130 MHz, V_{IN} = 0.3 V_{p-p})		6.0	12.0	mA

25. PACKAGE DRAWING

80 PIN PLASTIC QFP (14×14)



detail of lead end



NOTE

Each lead centerline is located within 0.13 mm (0.005 inch) of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS	INCHES
A	17.2±0.4	0.677±0.016
B	14.0±0.2	$0.551^{+0.009}_{-0.008}$
C	14.0±0.2	$0.551^{+0.009}_{-0.008}$
D	17.2±0.4	0.677±0.016
F	0.825	0.032
G	0.825	0.032
H	0.30±0.10	$0.012^{+0.004}_{-0.005}$
I	0.13	0.005
J	0.65 (T.P.)	0.026 (T.P.)
K	1.6±0.2	0.063±0.008
L	0.8±0.2	$0.031^{+0.009}_{-0.008}$
M	$0.15^{+0.10}_{-0.05}$	$0.006^{+0.004}_{-0.003}$
N	0.10	0.004
P	2.7	0.106
Q	0.1±0.1	0.004±0.004
R	5°±5°	5°±5°
S	3.0 MAX.	0.119 MAX.

S80GC-65-3B9-4

Remark The dimensions and materials of the ES model are the same as those of the mass-produced model.

26. RECOMMENDED SOLDERING CONDITIONS

Solder the μPD17719 under the following recommended conditions.

For the details of the recommended soldering conditions, refer to **"Semiconductor Device Mounting Technology Manual" (C10535E)**.

For the soldering method and conditions other than those recommended, consult NEC.

Table 26-1. Soldering Conditions of Surface Mount Type

μPD17717GC-xxx-3B9: 80-pin plastic QFP (14 × 14 mm, 0.65 mm pitch)
μPD17718GC-xxx-3B9: 80-pin plastic QFP (14 × 14 mm, 0.65 mm pitch)
μPD17719GC-xxx-3B9: 80-pin plastic QFP (14 × 14 mm, 0.65 mm pitch)

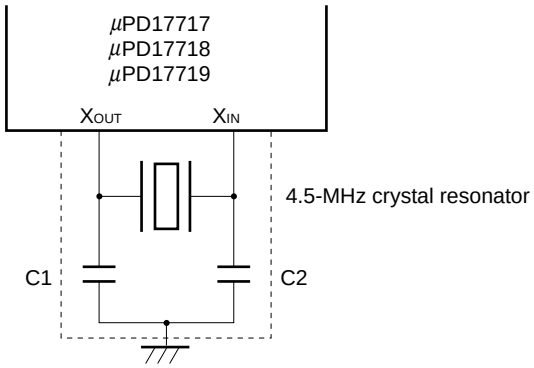
Soldering Method	Soldering Condition	Symbol of Recommended Condition
Infrared reflow	Package peak temperature: 235 °C, Time: 30 seconds MAX. (210 °C MIN.) Number of times: 3 MAX.	IR35-00-3
VPS	Package peak temperature: 215 °C, Time: 40 seconds MAX. (200 °C MIN.) Number of times: 3 MAX.	VP15-00-3
Wave soldering	Soldering bath temperature: 260 °C MAX., Time: 10 seconds MAX., Number of times: 1, Preheating temperature: 120 °C MAX. (package surface temperature)	WS60-00-1
Partial heating	Pin temperature: 300 °C MAX., Time: 3 seconds MAX. (per side of device)	—

Caution Do not use two or more soldering methods in combination (except partial heating method).

APPENDIX A. CAUTIONS ON CONNECTING CRYSTAL RESONATOR

When using the system clock oscillation circuit, wire the portion enclosed by the dotted line in the figure below as follows to prevent adverse influence from wiring capacity.

- Keep the wiring length as short as possible.
- If capacitances C1 and C2 are too high, the oscillation start characteristics may be degraded or current consumption may increase.
- Generally, connect a trimmer capacitor for adjusting the oscillation frequency to the X_{IN} pin. Depending on the crystal resonator to be used, however, the oscillation stability differs. Therefore, evaluate the crystal resonator actually used.
- The crystal oscillation frequency cannot be accurately adjusted when an emulation probe is connected to the X_{OUT} and X_{IN} pin, because of the capacitance of the probe. Adjust the frequency while measuring the VCO oscillation frequency.



APPENDIX B. DEVELOPMENT TOOLS

The following development tools are available for development of programs for the μPD17719.

Hardware

Name	Outline
In-circuit emulator (IE-17K IE-17K-ET[Note 1] EMU-17K[Note 2])	IE-17K, IE-17K-ET, and EMU-17K are in-circuit emulators that can be used with any model in the 17K series. IE-17K and IE-17K-ET are connected to a host machine, which is PC-9800 series or IBM PC/AT™, with RS-232C. EMU-17K is mounted to the expansion slot of a host machine, PC-9800 series. By using these in-circuit emulators with a system evaluation board (SE board) corresponding to each model, these emulators operate dedicated to the model. When man-machine interface software *SIMPLEHOST*™ is used, a more sophisticated debugging environment can be created. EMU-17K also has a function to allow you to check the contents of the data memory real-time.
SE board (SE-17709)	SE-17709 is an SE board for the μPD17719 subseries. This board can be used alone to evaluate a system, or in combination with an in-circuit emulator for debugging.
Emulation probe (EP-17K80GC)	EP-17K80GC is an emulation probe for the μPD17719 subseries. By using this probe with EV-9200GC-80[Note 3], the SE board and target system are connected.
Conversion socket (EV-9200GC-80[Note 3])	EV-9200GC-80 is a conversion socket for 80-pin plastic QFP (14 × 14 mm). It is used to connect EP-17K80GC and target system.
PROM programmer (PG-1500)	PG-1500 is a PROM programmer supporting μPD17P719. It can program μPD17P719 when connected with PG-1500 adapter PA-17KDZ and programmer adapter PA-17P709GC.
Programmer adapter (PA-17P709GC)	PA-17P709GC is an adapter to program μPD17P719. It is used with PG-1500.

Notes 1. Low-price model: external power supply type

2. This is a product of Naito Densei Machida Mfg. Co., Ltd. For details, consult Naito Densei Machida Mfg. Co., Ltd. ((044) 822-3813).

3. One EV-9200GC-80 is supplied with the EP-17K80GC. Five EV-9200GC-80 are also available as a set.

Remark Third party PROM programmers AF-9703, AF-9704, AF-9705, and AF-9706 are available from Ando Electric Co., Ltd. Use these programmers with programmer adapter PA-17P709GC. For details, consult Ando Electric Co., Ltd. ((03) 3733-1163).

Software

Name	Outline	Host Machine	OS		Media	Parts Number
17K series assembler (AS17K)	AS17K is an assembler that can be commonly used with 17K series. To develop programs for the μPD17719, this AS17K and a device file (AS17707) are used in combination.	PC-9800 series	MS-DOS™		5” 2HD	μS5A10AS17K
					3.5” 2HD	μS5A13AS17K
		IBM PC/AT	PC DOS™		5” 2HC	μS7B10AS17K
					3.5” 2HC	μS7B13AS17K
Device file (AS17707)	AS17707 is a device file for the μPD17719 subseries. It is used with the assembler common to the 17K series (AS17K).	PC-9800 series	MS-DOS		5” 2HD	μS5A10AS17707
					3.5” 2HD	μS5A13AS17707
		IBM PC/AT	PC DOS		5” 2HC	μS7B10AS17707
					3.5” 2HC	μS7B13AS17707
Support software (*SIMPLEHOST*)	*SIMPLEHOST* is man-machine interface software that runs on Windows™ when a program is developed by using an in-circuit emulator and personal computer.	PC-9800 series	MS-DOS	Windows	5” 2HD	μS5A10IE17K
					3.5” 2HD	μS5A13IE17K
		IBM PC/AT	PC DOS		5” 2HC	μS7B10IE17K
					3.5” 2HC	μS7B13IE17K

Remark The version of the supported OS is as follows:

OS	Version
MS-DOS	Ver.3.30 to Ver.5.00A[Note]
PC DOS	Ver.3.1 to Ver.5.0[Note]
Windows	Ver.3.0 to Ver.3.1

Note MS-DOS Ver. 5.00/5.00A and PC DOS Ver. 5.0 have a task swap function, but this function cannot be used with this software.

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note: Strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note: No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS device behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using a pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND with a resistor, if it is considered to have a possibility of being an output pin. All handling related to the unused pins must be judged device by device and related specifications governing the devices.

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note: Power-on does not necessarily define initial status of MOS device. Production process of MOS does not define the initial operation status of the device. Immediately after the power source is turned ON, the devices with reset function have not yet been initialized. Hence, power-on does not guarantee out-pin levels, I/O settings or contents of registers. Device is not initialized until the reset signal is received. Reset operation must be executed immediately after power-on for devices having reset function.

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 800-366-9782
Fax: 800-729-9288

NEC Electronics (Germany) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 02
Fax: 0211-65 03 490

NEC Electronics (UK) Ltd.
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Italiana s.r.1.
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

NEC Electronics (Germany) GmbH
Benelux Office
Eindhoven, The Netherlands
Tel: 040-2445845
Fax: 040-2444580

NEC Electronics (France) S.A.
Velizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

NEC Electronics (France) S.A.
Spain Office
Madrid, Spain
Tel: 01-504-2787
Fax: 01-504-2860

NEC Electronics (Germany) GmbH
Scandinavia Office
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Singapore Pte. Ltd.
United Square, Singapore 1130
Tel: 253-8311
Fax: 250-3583

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-719-2377
Fax: 02-719-5951

NEC do Brasil S.A.
Sao Paulo-SP, Brasil
Tel: 011-889-1680
Fax: 011-889-1689

J96. 8

Purchase of NEC I^2C components conveys a license under the Philips I^2C Patent Rights to use these components in an I^2C system, provided that the system conforms to the I^2C Standard Specification as defined by Philips.

***SIMPLEHOST* is a trademark of NEC Corporation.**
MS-DOS and Windows are trademarks of Microsoft Corporation.
PC/AT and PC DOS are trademarks of IBM Corporation.

The export of this product from Japan is regulated by the Japanese government. To export this product may be prohibited without governmental license, the need for which must be judged by the customer. The export or re-export of this product from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

M4 96.5