Renesas Electronics America Inc - UPD78F0442GB-GAH-AX Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	78K/0
Core Size	8-Bit
Speed	10MHz
Connectivity	3-Wire SIO, LINbus, UART/USART
Peripherals	LCD, LVD, POR, PWM, WDT
Number of I/O	46
Program Memory Size	24KB (24K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd78f0442gb-gah-ax

Correspondence between the address values and block numbers in the flash memory are shown below.

Table 3-2. Correspondence Between Address Values and Block Numbers in Flash Memory

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
0000H to 03FFH	00H	4000H to 43FFH	10H	8000H to 83FFH	20H	C000H to C3FFH	30H
0400H to 07FFH	01H	4400H to 47FFH	11H	8400H to 87FFH	21H	C400H to C7FFH	31H
0800H to 0BFFH	02H	4800H to 4BFFH	12H	8800H to 8BFFH	22H	C800H to CBFFH	32H
0C00H to 0FFFH	03H	4C00H to 4FFFH	13H	8C00H to 8FFFH	23H	CC00H to CFFFH	33H
1000H to 13FFH	04H	5000H to 53FFH	14H	9000H to 93FFH	24H	D000H to D3FFH	34H
1400H to 17FFH	05H	5400H to 57FFH	15H	9400H to 97FFH	25H	D400H to D7FFH	35H
1800H to 1BFFH	06H	5800H to 5BFFH	16H	9800H to 9BFFH	26H	D800H to DBFFH	36H
1C00H to 1FFFH	07H	5C00H to 5FFFH	17H	9C00H to 9FFFH	27H	DC00H to DFFFH	37H
2000H to 23FFH	08H	6000H to 63FFH	18H	A000H to A3FFH	28H	E000H to E3FFH	38H
2400H to 27FFH	09H	6400H to 67FFH	19H	A400H to A7FFH	29H	E400H to E7FFH	39H
2800H to 2BFFH	0AH	6800H to 6BFFH	1AH	A800H to ABFFH	2AH	E800H to EBFFH	3AH
2C00H to 2FFFH	0BH	6C00H to 6FFFH	1BH	AC00H to AFFFH	2BH	EC00H to EFFFH	3BH
3000H to 33FFH	0CH	7000H to 73FFH	1CH	B000H to B3FFH	2CH		
3400H to 37FFH	0DH	7400H to 77FFH	1DH	B400H to B7FFH	2DH		
3800H to 3BFFH	0EH	7800H to 7BFFH	1EH	B800H to BBFFH	2EH		
3C00H to 3FFFH	0FH	7C00H to 7FFFH	1FH	BC00H to BFFFH	2FH		

Remark μPD78F0441, 78F0451, 78F0461: Block numbers 00H to 0FH
μPD78F0442, 78F0452, 78F0462: Block numbers 00H to 17H
μPD78F0443, 78F0453, 78F0463: Block numbers 00H to 1FH
μPD78F0444, 78F0454, 78F0464: Block numbers 00H to 2FH
μPD78F0445, 78F0455, 78F0465: Block numbers 00H to 3BH

3.1.1 Internal program memory space

The internal program memory space stores the program and table data. Normally, it is addressed with the program counter (PC).

78K0/LE3 products incorporate internal ROM (flash memory), as shown below.

Table 3-3. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD78F0441, 78F0451, 78F0461	Flash memory	16384 × 8 bits (0000H to 3FFFH)
μPD78F0442, 78F0452, 78F0462		24576 × 8 bits (0000H to 5FFFH)
μPD78F0443, 78F0453, 78F0463		32768 × 8 bits (0000H to 7FFFH)
μPD78F0444, 78F0454, 78F0464		49152 × 8 bits (0000H to BFFFH)
μPD78F0445, 78F0455, 78F0465		61440 × 8 bits (0000H to EFFFH)

The internal program memory space is divided into the following areas.

(1) Vector table area

The 64-byte area 0000H to 003FH is reserved as a vector table area. The program start addresses for branch upon reset or generation of each interrupt request are stored in the vector table area.

Of the 16-bit address, the lower 8 bits are stored at even addresses and the higher 8 bits are stored at odd addresses.

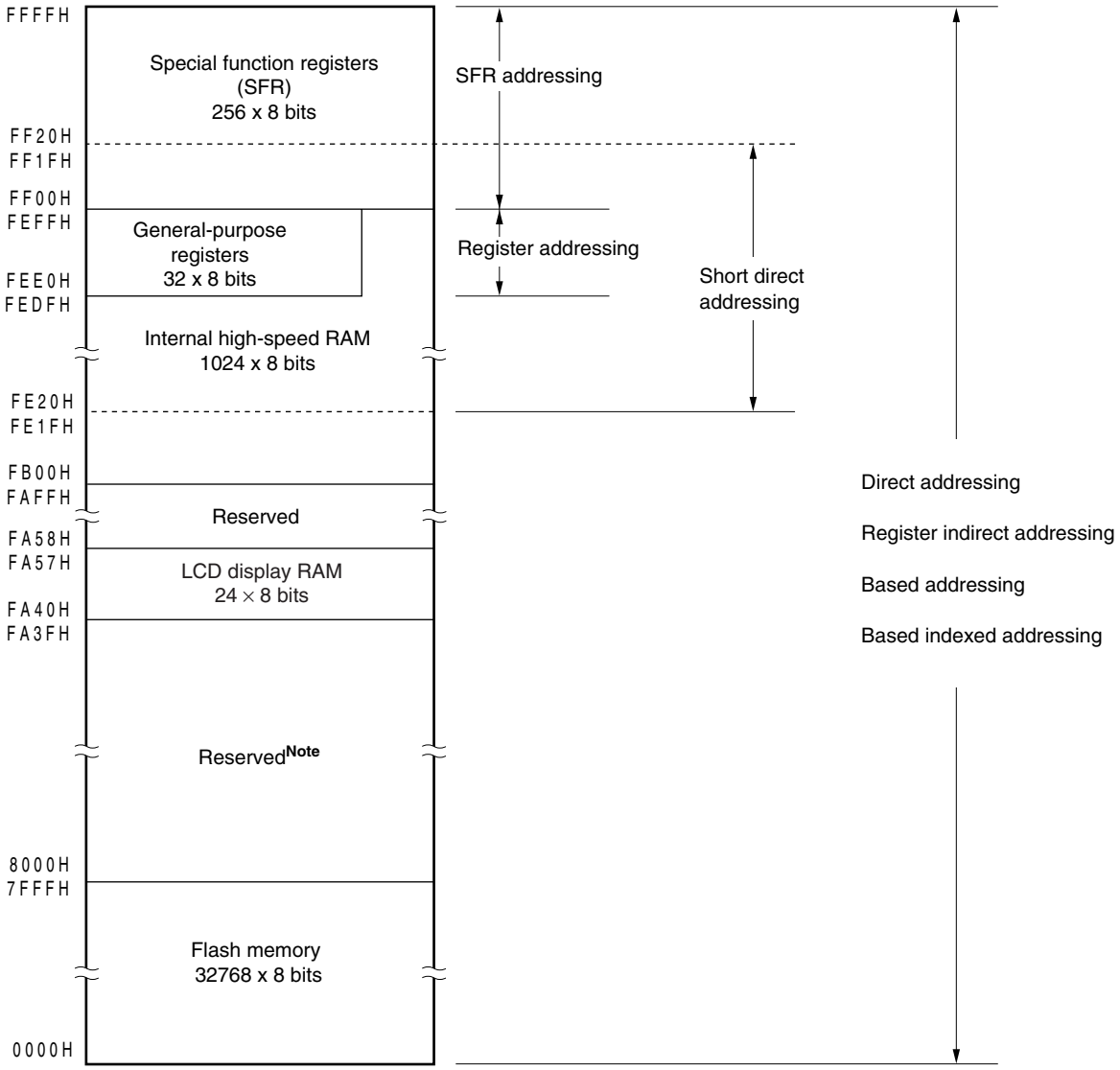
Table 3-4. Vector Table

Vector Table Address	Interrupt Source	Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input, POC, LVI, WDT	0022H	INTTM010
0004H	INTLVI	0024H[Note 1]	INTAD[Note 1]
0006H	INTP0	0026H	INTSR0
0008H	INTP1	0028H	INTRTC
000AH	INTP2	002AH	INTTM51
000CH	INTP3	002CH	INTKR
000EH	INTP4	002EH	INTRTCI
0012H	INTSRE6	0030H[Note 2]	INTDSAD[Note 2]
0014H	INTSR6	0032H	INTTM52
0016H	INTST6	0034H	INTTMH2
0018H	INTCSI10/INTST0	0036H	INTMCG
001AH	INTTMH1	0038H	INTRIN
001CH	INTTMH0	003AH	INTRERR/INTGP/INTREND /INTDFULL
001EH	INTTM50	003EH	BRK
0020H	INTTM000		

Notes 1. μPD78F045x and 78F046x only.

2. μPD78F046x only.

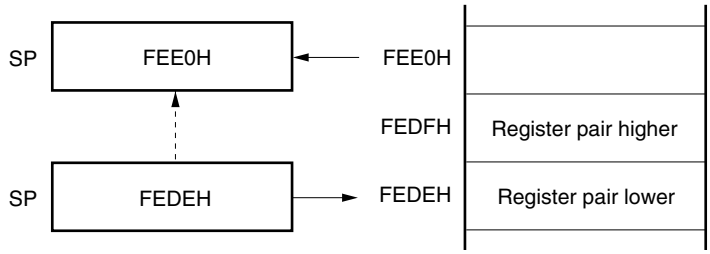
Figure 3-16. Correspondence Between Data Memory and Addressing (μPD78F0463)



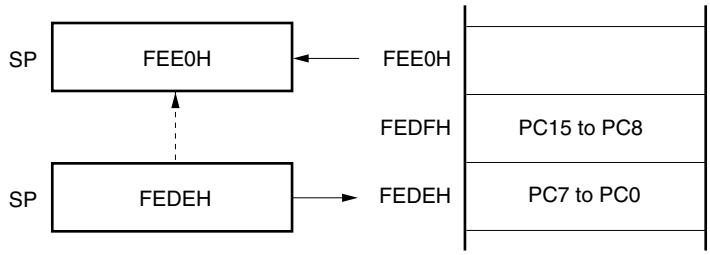
<R> **Note** However, FA26H and FA27H can be used (See **13.3 Registers Used in 16-Bit ΔΣ Type A/D Converter**).

Figure 3-25. Data to Be Restored from Stack Memory

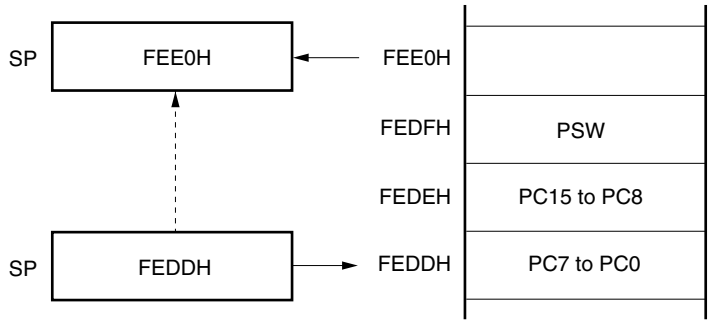
(a) POP rp instruction (when SP = FEDEH)



(b) RET instruction (when SP = FEDEH)



(c) RETI, RETB instructions (when SP = FEDDH)



4.2.5 Port 8

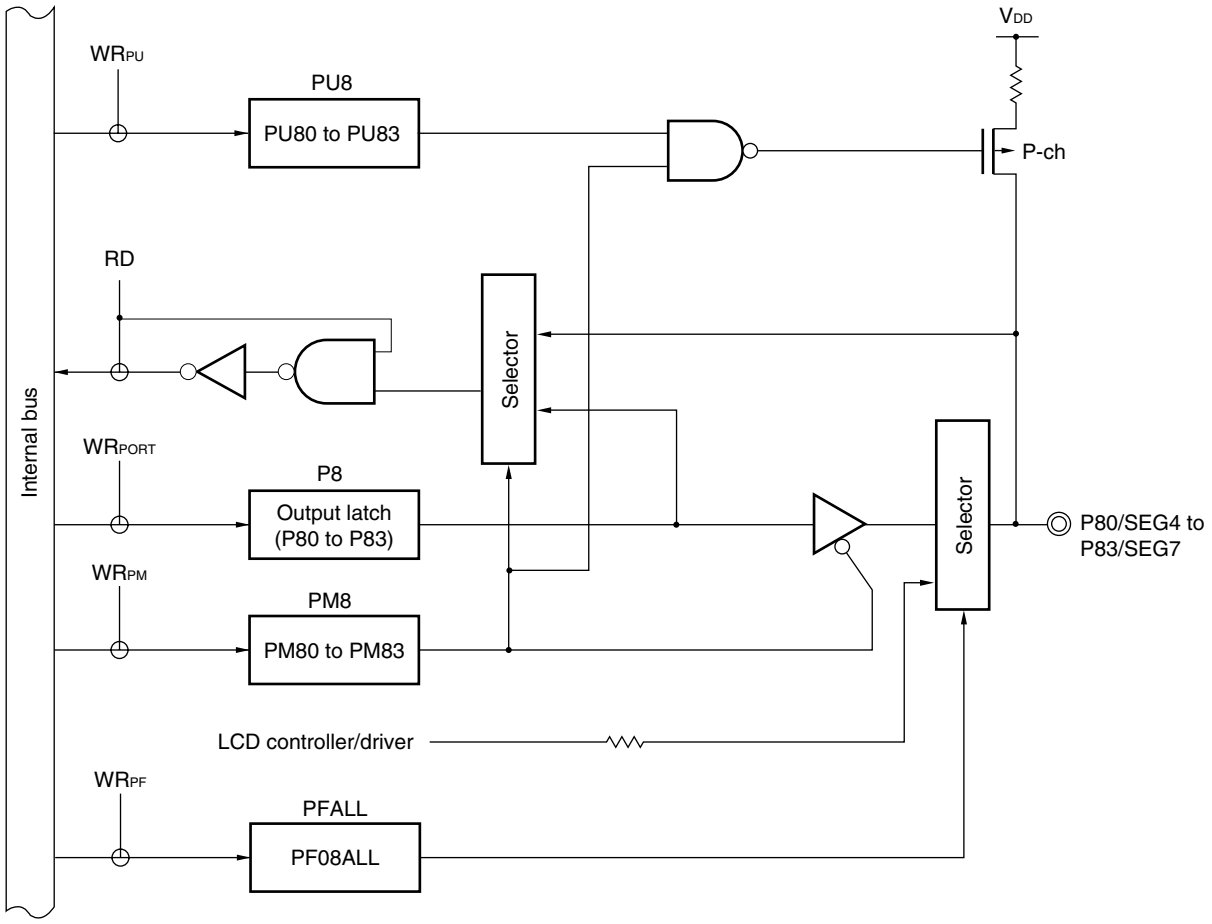
Port 8 is a 4-bit I/O port with an output latch. Port 8 can be set to the input mode or output mode in 1-bit units using port mode register 8 (PM8). When the P80 to P83 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 8 (PU8).

This port can also be used for segment output.

Reset signal generation sets port 8 to input mode.

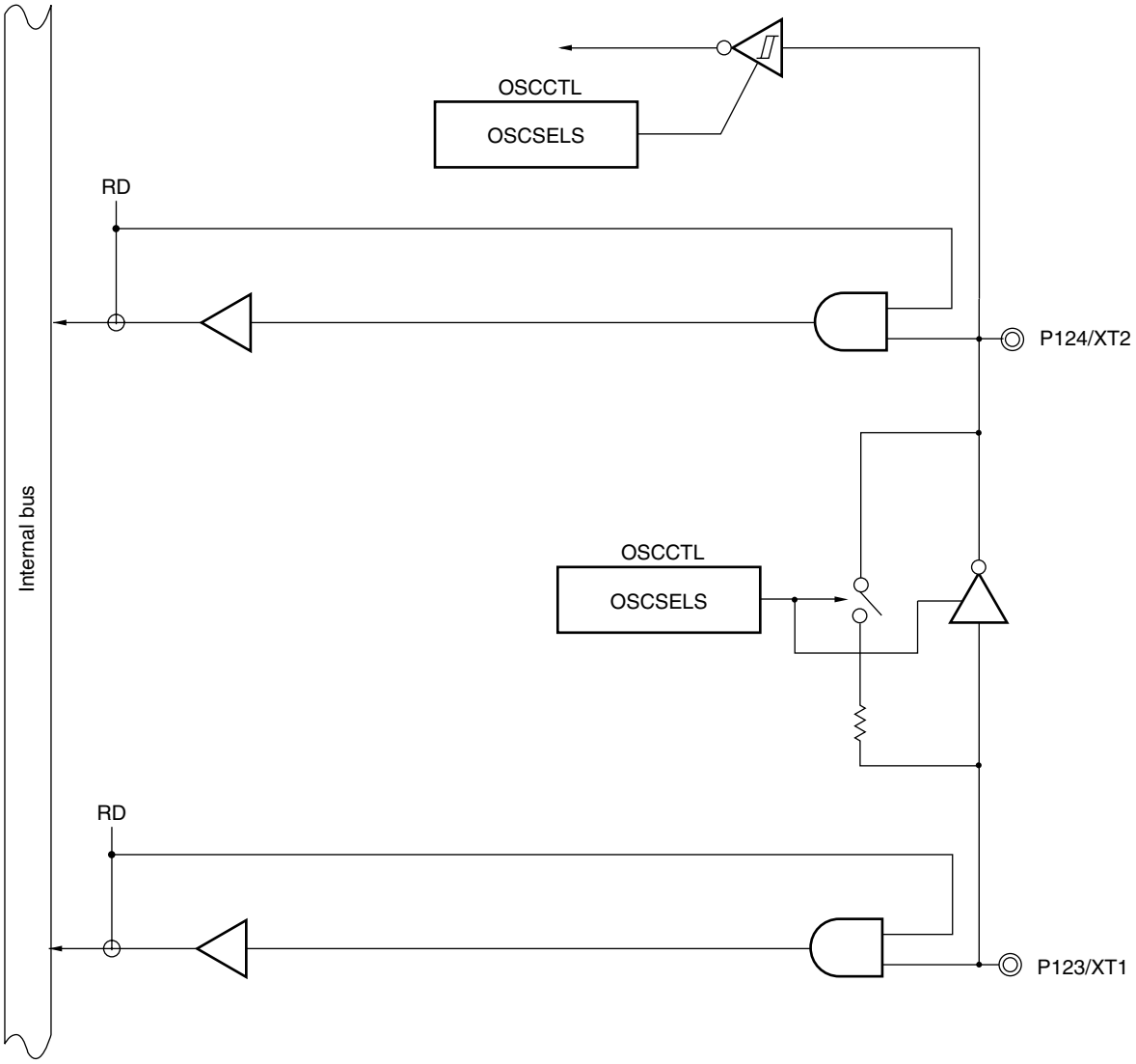
Figure 4-11 shows a block diagram of port 8.

Figure 4-11. Block Diagram of P80 to P83



P8: Port register 8
PU8: Pull-up resistor option register 8
PM8: Port mode register 8
PFALL: Port function register ALL
RD: Read signal
WR××: Write signal

Figure 4-18. Block Diagram of P123 and P124



OSCCTL: Clock operation mode select register
RD: Read signal

(7) A/D port configuration register 0 (ADPC0) (μPD78F045x and 78F046x only)

This register switches the P20/ANI0 to P27/ANI7 pins to analog input of A/D converter or digital I/O of port.

ADPC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 08H.

Figure 4-27. Format of A/D Port Configuration Register 0 (ADPC0)

Address: FF8FH After reset: 08H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC0	0	0	0	0	ADPC03	ADPC02	ADPC01	ADPC00

ADPC03	ADPC02	ADPC01	ADPC00	Digital I/O (D)/analog input (A: successive approximation type, Δ: ΔΣ type) switching							
				P27/ ANI7/ REF+	P26/ ANI6/ REF-	P25/ ANI5/ DS2+	P24/ ANI4/ DS2-	P23/ ANI3/ DS1+	P22/ ANI2/ DS1-	P21/ ANI1/ DS0+	P20/ ANI0/ DS0-
0	0	0	0	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ
0	0	0	1	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A	D
0	0	1	0	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	A/Δ	D	D
0	0	1	1	A/Δ	A/Δ	A/Δ	A/Δ	A	D	D	D
0	1	0	0	A/Δ	A/Δ	A/Δ	A/Δ	D	D	D	D
0	1	0	1	A	A	A	D	D	D	D	D
0	1	1	0	A	A	D	D	D	D	D	D
0	1	1	1	A	D	D	D	D	D	D	D
1	0	0	0	D	D	D	D	D	D	D	D
Other than above				Setting prohibited							

Cautions 1. Set the channel used for A/D conversion to the input mode by using port mode register 2 (PM2).

2. The pin to be set as a digital I/O via ADPC, must not be set via ADS, ADDS1 or ADDS0.

3. If data is written to ADPC0, a wait cycle is generated. Do not write data to ADPC0 when the CPU is operating on the subsystem clock and the peripheral hardware clock is stopped. For details, see CHAPTER 33 CAUTIONS FOR WAIT.

4. If pins ANI0/P20/SEG31 to ANI7/P27/SEG24 are set to segment output via the PF2 register, output is set to segment output, regardless of the ADPC0 setting (for μPD78F045x only).

(1) Example of setting procedure when restarting oscillation of the internal high-speed oscillation clock[Note 1]

<1> Setting restart of oscillation of the internal high-speed oscillation clock (RCM register)
When RSTOP is cleared to 0, the internal high-speed oscillation clock starts operating.

<2> Waiting for the oscillation accuracy stabilization time of internal high-speed oscillation clock (RCM register)
Wait until RSTS is set to 1[Note 2].

Notes 1. After a reset release, the internal high-speed oscillator automatically starts oscillating and the internal high-speed oscillation clock is selected as the CPU clock.

2. This wait time is not necessary if high accuracy is not necessary for the CPU clock and peripheral hardware clock.

(2) Example of setting procedure when using internal high-speed oscillation clock as CPU clock, and internal high-speed oscillation clock or high-speed system clock as peripheral hardware clock

<1> • Restarting oscillation of the internal high-speed oscillation clock[Note]
(See **5.6.2 (1) Example of setting procedure when restarting internal high-speed oscillation clock**).

• Oscillating the high-speed system clock[Note]
(This setting is required when using the high-speed system clock as the peripheral hardware clock. See **5.6.1 (1) Example of setting procedure when oscillating the X1 clock** and **(2) Example of setting procedure when using the external main system clock.**)

Note The setting of <1> is not necessary when the internal high-speed oscillation clock or high-speed system clock is already operating.

<2> Selecting the clock supplied as the main system clock and peripheral hardware clock (MCM register)
Set the main system clock and peripheral hardware clock using XSEL and MCM0.

XSEL	MCM0	Selection of Main System Clock and Clock Supplied to Peripheral Hardware	
		Main System Clock (f_{XP})	Peripheral Hardware Clock (f_{PRS})
0	0	Internal high-speed oscillation clock (f_{RH})	Internal high-speed oscillation clock (f_{RH})
0	1		
1	0		High-speed system clock (f_{XH})

<3> Selecting the CPU clock division ratio (PCC register)
When CSS is cleared to 0, the main system clock is supplied to the CPU. To select the CPU clock division ratio, use PCC0, PCC1, and PCC2.

CSS	PCC2	PCC1	PCC0	CPU Clock (f_{CPU}) Selection
0	0	0	0	f_{XP}
	0	0	1	$f_{XP}/2$ (default)
	0	1	0	$f_{XP}/2^2$
	0	1	1	$f_{XP}/2^3$
	1	0	0	$f_{XP}/2^4$
	Other than above			Setting prohibited

(2) Operation in clear & start mode entered by TI000 pin valid edge input (CR000: compare register, CR010: capture register)

Figure 6-25. Block Diagram of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Capture Register)

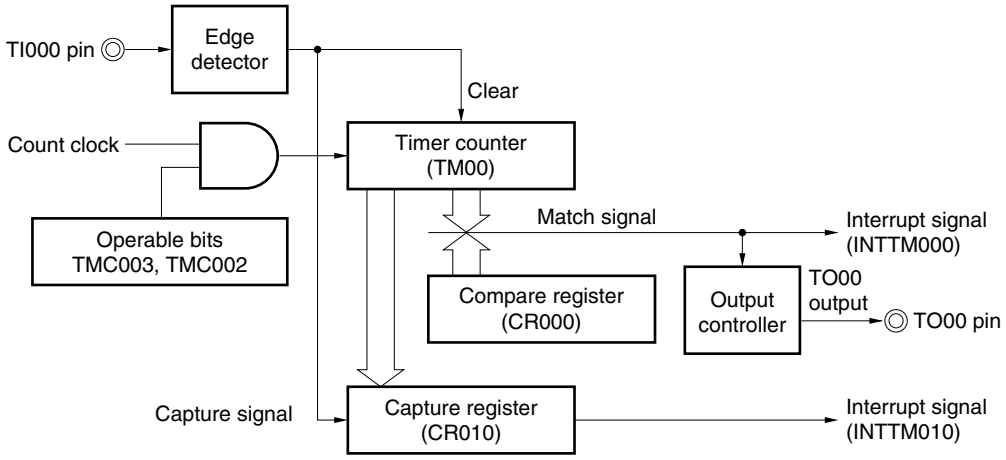
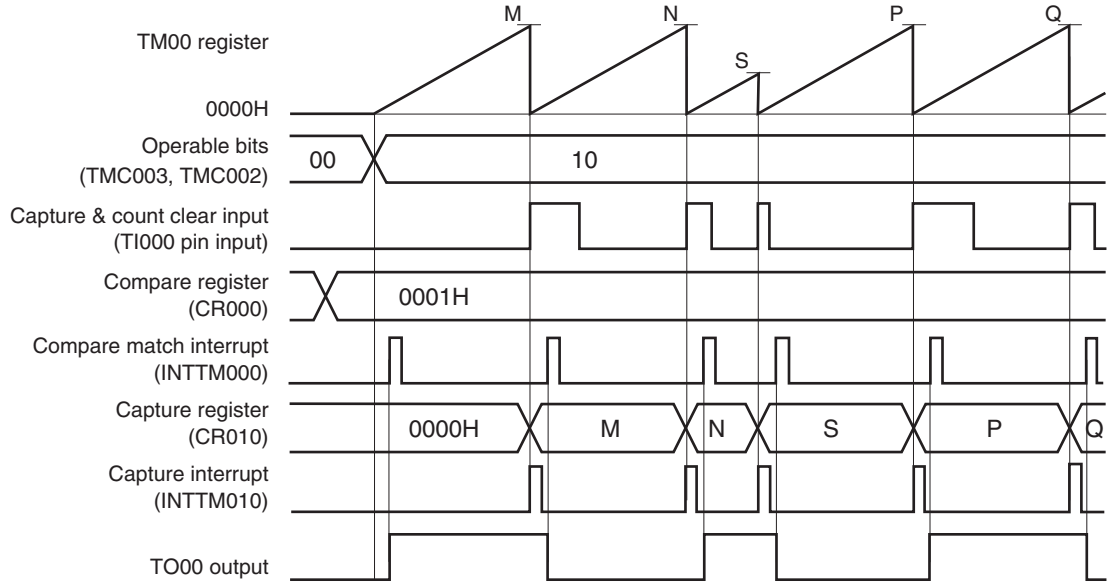


Figure 6-26. Timing Example of Clear & Start Mode Entered by TI000 Pin Valid Edge Input (CR000: Compare Register, CR010: Capture Register) (1/2)

(a) TOC00 = 13H, PRM00 = 10H, CRC00, = 04H, TMC00 = 08H, CR000 = 0001H



This is an application example where the TO00 output level is inverted when the count value has been captured & cleared.

The count value is captured to CR010 and TM00 is cleared (to 0000H) when the valid edge of the TI000 pin is detected. When the count value of TM00 is 0001H, a compare match interrupt signal (INTTM000) is generated, and the TO00 output level is inverted.

8.4.3 Carrier generator operation (8-bit timer H1 only)

In the carrier generator mode, the 8-bit timer H1 is used to generate the carrier signal of an infrared remote controller, and the 8-bit timer/event counter 51 is used to generate an infrared remote control signal (time count).

The carrier clock generated by the 8-bit timer H1 is output in the cycle set by the 8-bit timer/event counter 51.

In carrier generator mode, the output of the 8-bit timer H1 carrier pulse is controlled by the 8-bit timer/event counter 51, and the carrier pulse is output from the TOH1 output.

(1) Carrier generation

In carrier generator mode, the 8-bit timer H compare register 01 (CMP01) generates a low-level width carrier pulse waveform and the 8-bit timer H compare register 11 (CMP11) generates a high-level width carrier pulse waveform.

Rewriting the CMP11 register during the 8-bit timer H1 operation is possible but rewriting the CMP01 register is prohibited.

(2) Carrier output control

Carrier output is controlled by the interrupt request signal (INTTM51) of the 8-bit timer/event counter 51 and the NRZB1 and RMC1 bits of the 8-bit timer H carrier control register (TMCYC1). The relationship between the outputs is shown below.

RMC1 Bit	NRZB1 Bit	Output
0	0	Low-level output
0	1	High-level output at rising edge of INTTM51 signal input
1	0	Low-level output
1	1	Carrier pulse output at rising edge of INTTM51 signal input

Figure 9-3. Format of Real-Time Counter Control Register 0 (RTCC0)

Address: FF89H After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	2	1	0
RTCC0	RTCE	0	RCLOE1	RCLOE0	AMPM	CT2	CT1	CT0

RTCE	Real-time counter operation control
0	Stops counter operation.
1	Starts counter operation.

RCLOE1	RTC1HZ pin output control
0	Disables output of RTC1HZ pin (1 Hz).
1	Enables output of RTC1HZ pin (1 Hz).

RCLOE0[Note]	RTCCL pin output control
0	Disables output of RTCCL pin (32.768 kHz).
1	Enables output of RTCCL pin (32.768 kHz).

AMPM	Selection of 12-/24-hour system
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system
• To change the value of AMPM, set RWAIT (bit 0 of RTCC1) to 1, and re-set the hour count register (HOUR). • Table 9-2 shows the displayed time digits.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use constant-period interrupt function.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)
After changing the values of CT2 to CT0, clear the interrupt request flag.			

Note RCLOE0 and RCLOE2 must not be enabled at the same time.

Caution If RCLOE0 and RCLOE1 are changed when RTCE = 1, a pulse with a narrow width may be generated on the 32.768 kHz and 1 Hz output signals.

Remark ×: don't care

10.4 Operation of Watchdog Timer

10.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (0080H).

   - Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (0080H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 26**).

WDTON	Operation Control of Watchdog Timer Counter/Illegal Access Detection
0	Counter operation disabled (counting stopped after reset), illegal access detection operation disabled
1	Counter operation enabled (counting started after reset), illegal access detection operation enabled

   - Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (0080H) (for details, see **10.4.2** and **CHAPTER 26**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (0080H) (for details, see **10.4.3** and **CHAPTER 26**).

2. After a reset release, the watchdog timer starts counting.
3. By writing “ACH” to WDTE after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.
4. After that, write WDTE the second time or later after a reset release during the window open period. If WDTE is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without “ACH” written to WDTE, an internal reset signal is generated.
   A internal reset signal is generated in the following cases.

   - If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
   - If data other than “ACH” is written to WDTE
   - If the instruction is fetched from an area not set by the IMS and IXS registers (detection of an invalid check during a CPU program loop)
   - If the CPU accesses an area not set by the IMS and IXS registers (excluding FB00H to FFFFH) by executing a read/write instruction (detection of an abnormal access during a CPU program loop)

Cautions 1. The first writing to WDTE after a reset release clears the watchdog timer, if it is made before the overflow time regardless of the timing of the writing, and the watchdog timer starts counting again.

2. If the watchdog timer is cleared by writing “ACH” to WDTE, the actual overflow time may be different from the overflow time set by the option byte by up to 2/f_{RL} seconds.

3. The watchdog timer can be cleared immediately before the count value overflows (FFFFH).

12.3 Registers Used in 10-Bit Successive Approximation Type A/D Converter

The A/D converter uses the following seven registers.

- A/D converter mode register (ADM)
- A/D port configuration register 0 (ADPC0)
- Analog input channel specification register (ADS)
- Port function register 2 (PF2) (μPD78F045x only)
- Port mode register 2 (PM2)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)

(1) A/D converter mode register (ADM)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.

ADM can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12-3. Format of A/D Converter Mode Register (ADM)

Address: FF8DH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	FR3 Note 1	FR2 Note 1	FR1 Note 1	FR0 Note 1	LV1 Note 1	LV0 Note 1	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation
1	Enables conversion operation

ADCE	Comparator operation control Note 2
0	Stops comparator operation
1	Enables comparator operation

Notes 1. For details of FR3 to FR0, LV1, LV0, and A/D conversion, see **Table 12-2 A/D Conversion Time Selection**.

2. The operation of the comparator is controlled by ADCS and ADCE, and it takes 1 μs from operation start to operation stabilization. Therefore, when ADCS is set to 1 after 1 μs or more has elapsed from the time ADCE is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

Table 12-1. Settings of ADCS and ADCE

ADCS	ADCE	A/D Conversion Operation
0	0	Stop status (DC power consumption path does not exist)
0	1	Conversion waiting mode (comparator operation, only comparator consumes power)
1	0	Conversion mode (comparator operation stopped Note)
1	1	Conversion mode (comparator operation)

Note Ignore data of the first conversion.

13.6 How to Read ΔΣ Type A/D Converter Characteristics Table

Here, special terms unique to the ΔΣ type A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 16 bits.

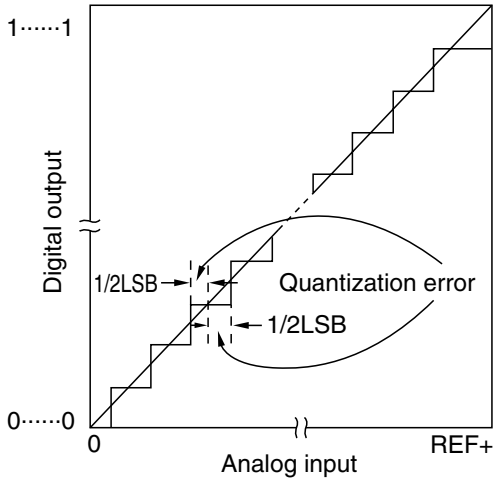
$1LSB = 1/2^{16} = 1/65536$
$\cong 0.0015\%FSR$

(2) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the offset, gain error, integral linearity error, and differential linearity error in the characteristics table.

Figure 13-15. Quantization Error



15.4.4 Calculation of baud rate

(1) Baud rate calculation expression

The baud rate can be calculated by the following expression.

- Baud rate = $\frac{f_{XCLK6}}{2 \times k}$ [bps]

f_{XCLK6}: Frequency of base clock selected by TPS63 to TPS60 bits of CKSR6 register

k: Value set by MDL67 to MDL60 bits of BRGC6 register (k = 4, 5, 6, ..., 255)

Table 15-4. Set Value of TPS63 to TPS60

TPS63	TPS62	TPS61	TPS60	Base Clock (f_{XCLK6}) Selection[Note 1]				
					f_{PRS} = 2 MHz	f_{PRS} = 5 MHz	f_{PRS} = 8 MHz	f_{PRS} = 10 MHz
0	0	0	0	f_{PRS}[Note 2]	2 MHz	5 MHz	8 MHz	10 MHz
0	0	0	1	$f_{PRS}/2$	1 MHz	2.5 MHz	4 MHz	5 MHz
0	0	1	0	$f_{PRS}/2^2$	500 kHz	1.25 MHz	2 MHz	2.5 MHz
0	0	1	1	$f_{PRS}/2^3$	250 kHz	625 kHz	1 MHz	1.25 MHz
0	1	0	0	$f_{PRS}/2^4$	125 kHz	312.5 kHz	500 kHz	625 kHz
0	1	0	1	$f_{PRS}/2^5$	62.5 kHz	156.25 kHz	250 kHz	312.5 kHz
0	1	1	0	$f_{PRS}/2^6$	31.25 kHz	78.13 kHz	125 kHz	156.25 kHz
0	1	1	1	$f_{PRS}/2^7$	15.625 kHz	39.06 kHz	62.5 kHz	78.13 kHz
1	0	0	0	$f_{PRS}/2^8$	7.813 kHz	19.53 kHz	31.25 kHz	39.06 kHz
1	0	0	1	$f_{PRS}/2^9$	3.906 kHz	9.77 kHz	15.625 kHz	19.53 kHz
1	0	1	0	$f_{PRS}/2^{10}$	1.953 kHz	4.88 kHz	7.813 kHz	9.77 kHz
1	0	1	1	TM50 output[Note 3]				
Other than above				Setting prohibited				

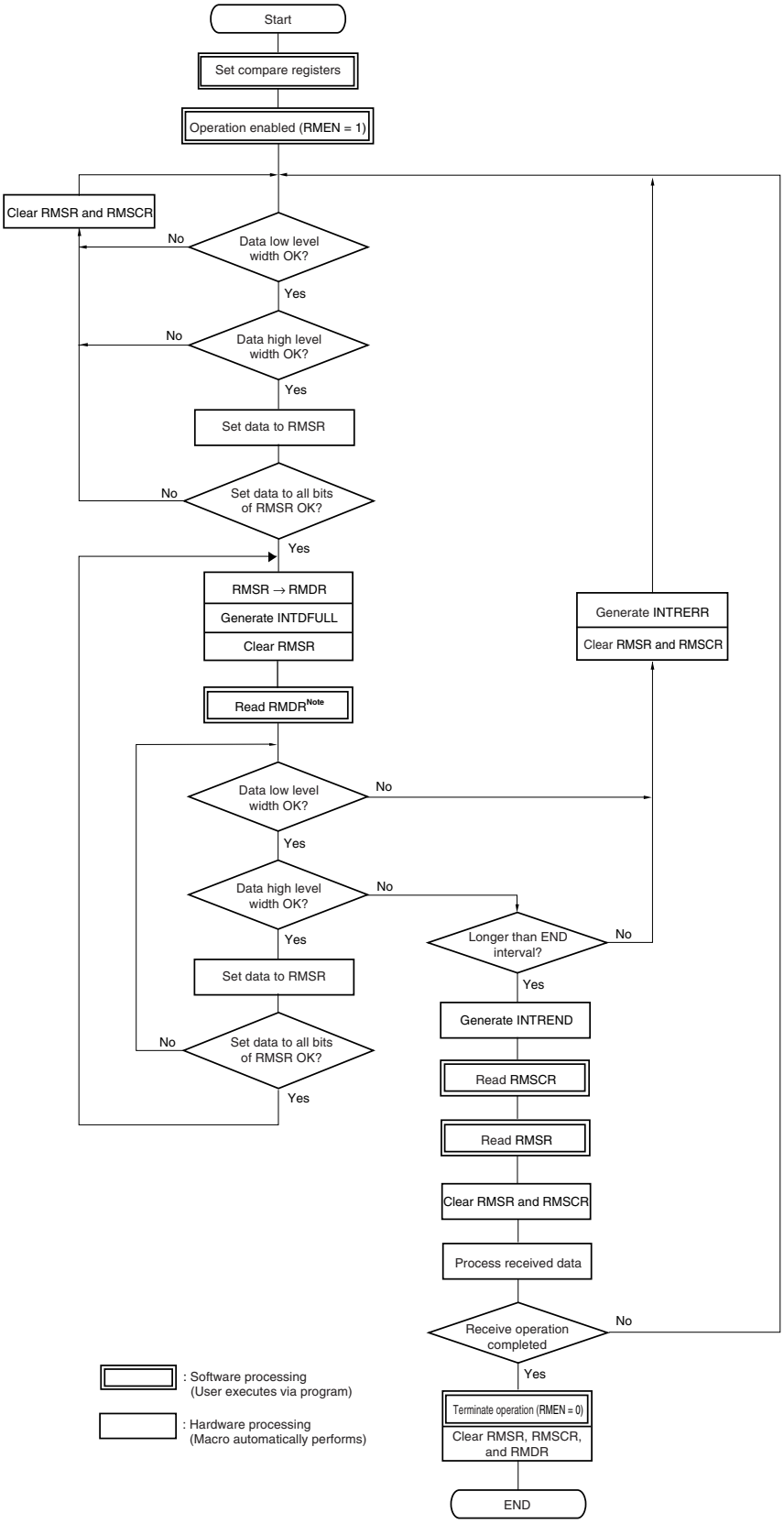
Notes 1. If the peripheral hardware clock (f_{PRS}) operates on the high-speed system clock (f_{XH}) (XSEL = 1), the f_{PRS} operating frequency varies depending on the supply voltage.
- V_{DD} = 2.7 to 5.5 V: $f_{PRS} \leq$ 10 MHz
- V_{DD} = 1.8 to 2.7 V: $f_{PRS} \leq$ 5 MHz

2. If the peripheral hardware clock (f_{PRS}) operates on the internal high-speed oscillation clock (f_{RH}) (XSEL = 0), when 1.8 V $\leq V_{DD} <$ 2.7 V, the setting of TPS63 = TPS62 = TPS61 = TPS60 = 0 (base clock: f_{PRS}) is prohibited.

3. Note the following points when selecting the TM50 output as the base clock.
- Mode in which the count clock is cleared and started upon a match of TM50 and CR50 (TMC506 = 0)
  Start the operation of 8-bit timer/event counter 50 first and then enable the timer F/F inversion operation (TMC501 = 1).
- PWM mode (TMC506 = 1)
  Start the operation of 8-bit timer/event counter 50 first and then set the count clock to make the duty = 50%.

It is not necessary to enable (TOE50 = 1) TO50 output in any mode.

Figure 19-11. Operation Flow of Type C Reception Mode



Note Read RMDR before data has been set to all the bits of RMSR.

Table 22-3. Operating Statuses in STOP Mode

STOP Mode Setting / Item			When STOP Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on Internal High-Speed Oscillation Clock (f_{RH})	When CPU Is Operating on X1 Clock (f_X)	When CPU Is Operating on External Main System Clock (f_{EXCLK})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{RH}	Stopped		
		f_X			
		f_{EXCLK}	Input invalid		
	Subsystem clock	f_{XT}	Status before STOP mode was set is retained		
	f_{RL}		Status before STOP mode was set is retained		
CPU			Operation stopped		
Flash memory			Operation stopped		
RAM			Status before STOP mode was set is retained		
Port (latch)			Status before STOP mode was set is retained		
16-bit timer/event counter 00 Note 1			Operable only when TM52 output or TI000 is selected as the count clock		
8-bit timer/event counter		50 Note 1	Operable only when TI50 is selected as the count clock		
		51 Note 1	Operable only when TI51 is selected as the count clock		
		52 Note 1	Operable only when TI52 is selected as the count clock		
8-bit timer		H0	Operable only when TM50 output is selected as the count clock during 8-bit timer/event counter 50 operation		
		H1	Operable only when f_{RL}, $f_{RL}/2^7$, $f_{RL}/2^9$ is selected as the count clock		
		H2	Operation stopped		
Real-time counter			Operable only when subsystem clock is selected as the count clock		
Watchdog timer			Operable. Clock supply to watchdog timer stops when “internal low-speed oscillator can be stopped by software” is set by option byte.		
Buzzer output			Operation stopped		
10-bit successive approximation type A/D converter					
<R> 16-bit ΔΣ type A/D converter Note 2			Operable		
Serial interface	UART0		Operable only when TM50 output is selected as the serial clock during 8-bit timer/event counter 50 operation		
	UART6				
	CSI10 Note 1		Operable only when external clock is selected as the serial clock		
LCD controller/driver			Operable only when subsystem clock is selected as the count clock		
Manchester code generator			Operation stopped		
Remote controller receiver			Operable only when subsystem clock is selected as the count clock		
Power-on-clear function			Operable		
Low-voltage detection function					
External interrupt					

Notes 1. Do not start operation of these functions on the external clock input from peripheral hardware pins in the stop mode.

<R> **2.** Be sure to turn off the power (ADDPON = 0) of the 16-bit ΔΣ type A/D converter, when executing a STOP instruction upon selecting $f_{PRS}/4$, $f_{PRS}/8$, or $f_{PRS}/16$ for the sampling clock.

Remark f_{RH}: Internal high-speed oscillation clock
f_X: X1 clock
f_{EXCLK}: External main system clock
f_{XT}: XT1 clock
f_{RL}: Internal low-speed oscillation clock

CHAPTER 23 RESET FUNCTION

The following four operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of power-on-clear (POC) circuit
(4) Internal reset by comparison of supply voltage and detection voltage of low-power-supply detector (LVI)

External and internal resets have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H when the reset signal is generated.

A reset is applied when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, or by POC and LVI circuit voltage detection, and each item of hardware is set to the status shown in Tables 23-1 and 23-2. Each pin is high impedance during reset signal generation or during the oscillation stabilization time just after a reset release.

When a low level is input to the $\overline{\text{RESET}}$ pin, the device is reset. It is released from the reset status when a high level is input to the $\overline{\text{RESET}}$ pin and program execution is started with the internal high-speed oscillation clock after reset processing. A reset by the watchdog timer is automatically released, and program execution starts using the internal high-speed oscillation clock (see **Figures 23-2** to **23-4**) after reset processing. Reset by POC and LVI circuit power supply detection is automatically released when $V_{DD} \geq V_{POC}$ or $V_{DD} \geq V_{LVI}$ after the reset, and program execution starts using the internal high-speed oscillation clock (see **CHAPTER 24 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 25 LOW-VOLTAGE DETECTOR**) after reset processing.

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
2. During reset input, the X1 clock, XT1 clock, internal high-speed oscillation clock, and internal low-speed oscillation clock stop oscillating. External main system clock input becomes invalid.
3. When the STOP mode is released by a reset, the STOP mode contents are held during reset input. However, the port pins become high-impedance.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
16-bit data transfer	**MOVW**	rp, #word	3	6	–	rp ← word	
		saddrp, #word	4	8	10	(saddrp) ← word	
		sfrp, #word	4	–	10	sfrp ← word	
		AX, saddrp	2	6	8	AX ← (saddrp)	
		saddrp, AX	2	6	8	(saddrp) ← AX	
		AX, sfrp	2	–	8	AX ← sfrp	
		sfrp, AX	2	–	8	sfrp ← AX	
		AX, rp Note 3	1	4	–	AX ← rp	
		rp, AX Note 3	1	4	–	rp ← AX	
		AX, !addr16	3	10	12	AX ← (addr16)	
		!addr16, AX	3	10	12	(addr16) ← AX	
	XCHW	AX, rp Note 3	1	4	–	AX ↔ rp	
8-bit operation	**ADD**	A, #byte	2	4	–	A, CY ← A + byte	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte	× × ×
		A, r Note 4	2	4	–	A, CY ← A + r	× × ×
		r, A	2	4	–	r, CY ← r + A	× × ×
		A, saddr	2	4	5	A, CY ← A + (saddr)	× × ×
		A, !addr16	3	8	9	A, CY ← A + (addr16)	× × ×
		A, [HL]	1	4	5	A, CY ← A + (HL)	× × ×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte)	× × ×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B)	× × ×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C)	× × ×
	ADDC	A, #byte	2	4	–	A, CY ← A + byte + CY	× × ×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte + CY	× × ×
		A, r Note 4	2	4	–	A, CY ← A + r + CY	× × ×
		r, A	2	4	–	r, CY ← r + A + CY	× × ×
		A, saddr	2	4	5	A, CY ← A + (saddr) + CY	× × ×
		A, !addr16	3	8	9	A, CY ← A + (addr16) + C	× × ×
		A, [HL]	1	4	5	A, CY ← A + (HL) + CY	× × ×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte) + CY	× × ×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B) + CY	× × ×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C) + CY	× × ×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Only when rp = BC, DE or HL

4. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

(4) Call instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

Second Operand / First Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP