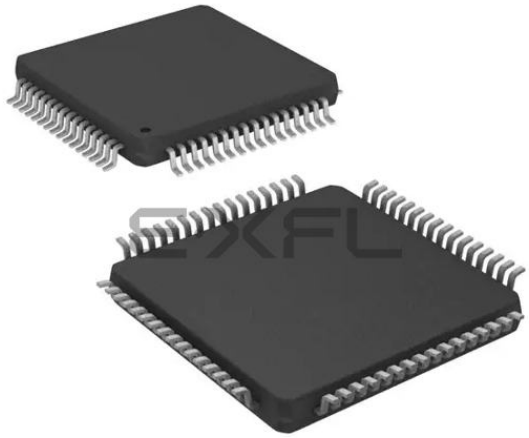
Renesas Electronics America Inc - UPD78F1144AGA-HAB-AX Datasheet



Welcome to E-XFL.COM

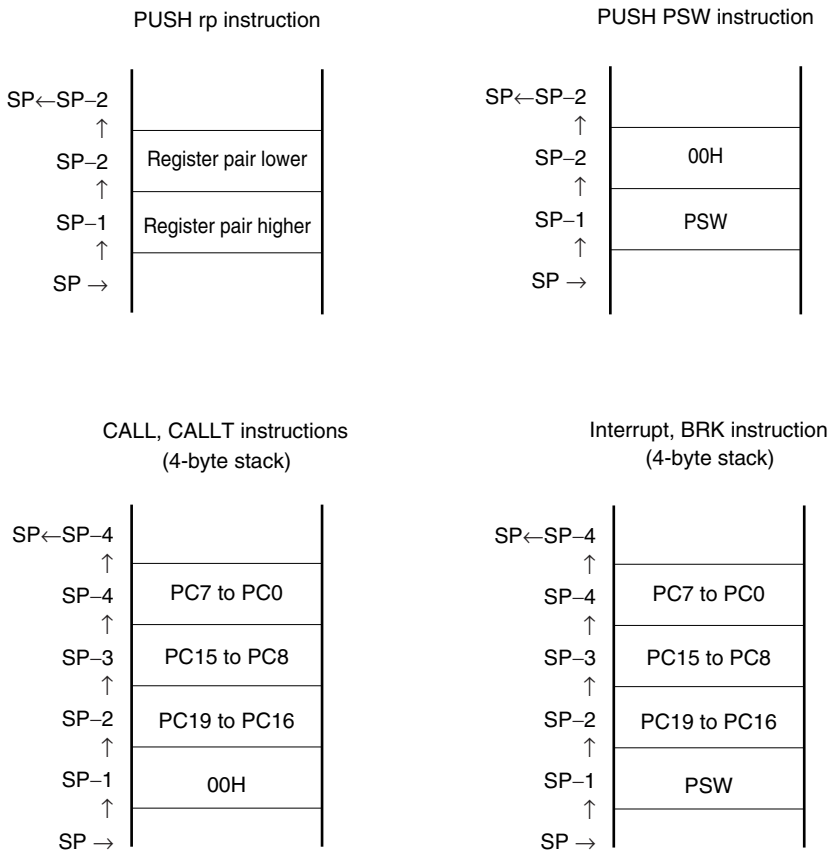
What is "Embedded - Microcontrollers"?

"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/0R
Core Size	16-Bit
Speed	20MHz
Connectivity	3-Wire SIO, I²C, LINbus, UART/USART
Peripherals	DMA, LVD, POR, PWM, WDT
Number of I/O	50
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd78f1144aga-hab-ax

Figure 3-15. Data to Be Saved to Stack Memory



3.2.2 General-purpose registers

General-purpose registers are mapped at particular addresses (FFEE0H to FFEFFH) of the data memory. The general-purpose registers consists of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

These registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Caution It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.

Remark f_X: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{EX}: External main system clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
<R> f_{MAINC}: Main system select clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{SUB}: Subsystem clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Internal low-speed oscillation clock frequency

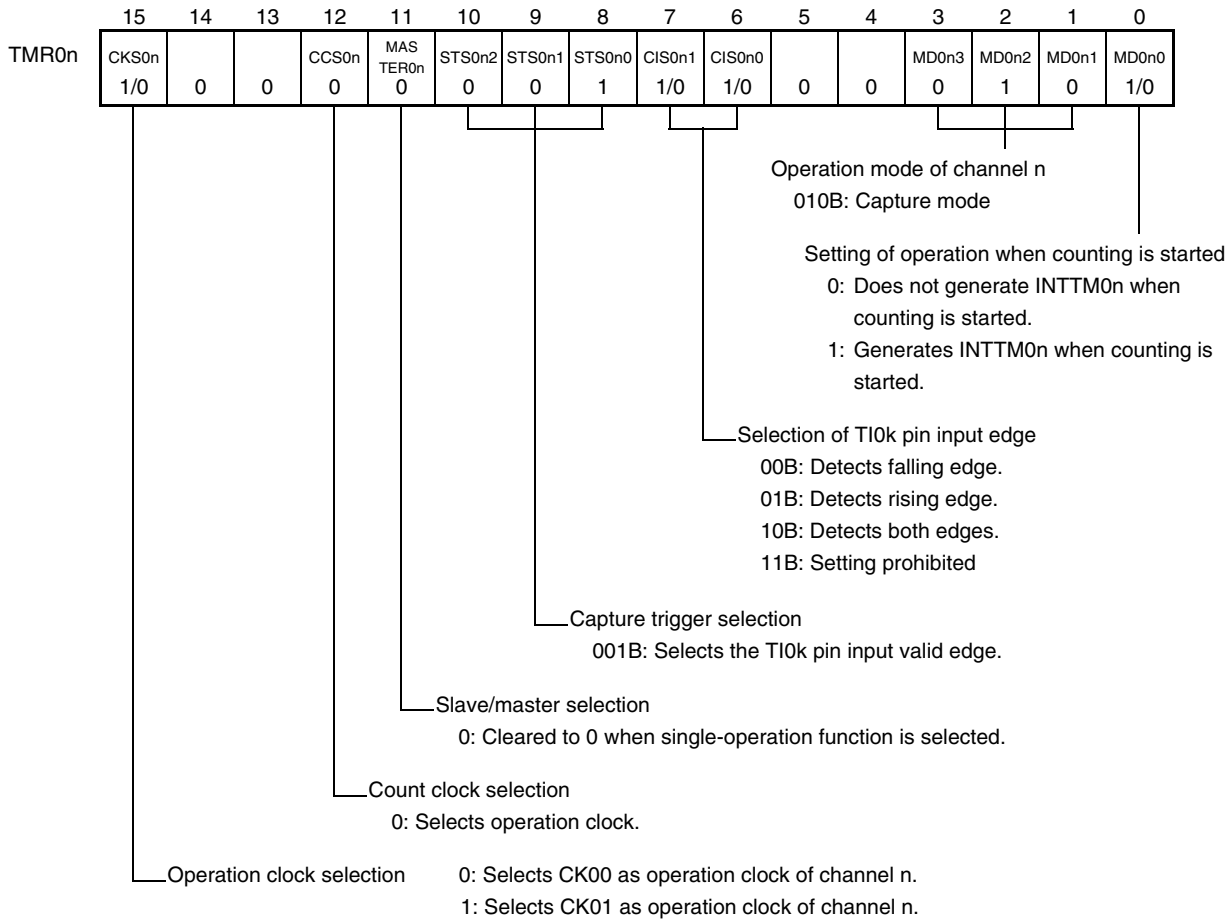
5.3 Registers Controlling Clock Generator

The following eight registers are used to control the clock generator.

- Clock operation mode control register (CMC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- System clock control register (CKC)
- Peripheral enable registers 0 (PER0)
- Operation speed mode control register (OSMC)
- Internal high-speed oscillator trimming register (HIOTRM)

Figure 6-49. Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit k	
TO0	TO0k 0	0: Outputs 0 from TO0k.

(c) Timer output enable register 0 (TOE0)

	Bit k	
TOE0	TOE0k 0	0: Stops TO0k output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit k	
TOL0	TOL0k 0	0: Cleared to 0 when TOM0k = 0 (toggle mode).

(e) Timer output mode register 0 (TOM0)

	Bit k	
TOM0	TOM0k 0	0: Sets toggle mode.

Remark n = 0 to 7, k = 0 to 6

8.4 Operation of Watchdog Timer

8.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (000C0H).

   - Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (000C0H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 22**).

WDTON	Watchdog Timer Counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

   - Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H) (for details, see **8.4.2** and **CHAPTER 22**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **8.4.3** and **CHAPTER 22**).

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

Cautions 1. When data is written to WDTE for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

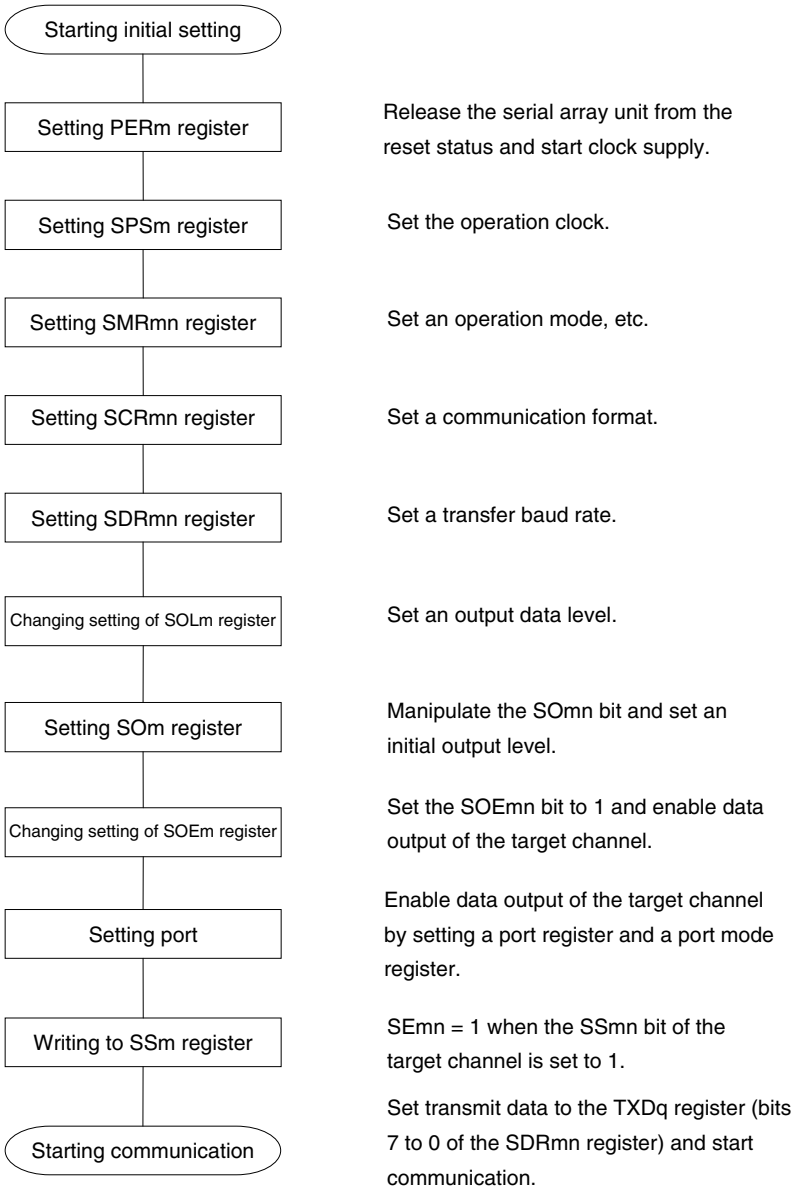
2. If the watchdog timer is cleared by writing “ACH” to WDTE, the actual overflow time may be different from the overflow time set by the option byte by up to $2/f_{IL}$ seconds.

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

<Example> When the overflow time is set to $2^{10}/f_{IL}$, writing “ACH” is valid up to count value 3FH.

(2) Operation procedure

Figure 11-72. Initial Setting Procedure for UART Transmission



Caution After setting the PER0 register to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Figure 11-79. Example of Contents of Registers for UART Reception of UART (UART0, UART1, UART3) (2/2)

(e) Serial mode register mr (SMRmr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmr	CKSmr	CCSmr						STSmr		SISmr0				MDmr2	MDmr1	MDmr0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0/1

Same setting value as CKSmn

Interrupt sources of channel r
0: Transfer end interrupt
1: Buffer empty interrupt

(f) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0		DLSmn2	DLSmn1	DLSmn0
	0	1	0	0	0	1	0/1	0/1	0/1	0	0	1	0	1	0/1	0/1

(g) Serial data register mn (SDRmn) (lower 8 bits: RXDq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Receive data register							

RXDq

Caution For the UART reception, be sure to set SMRmr of channel r that is to be paired with channel n.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 13,
r: Channel number (r = n − 1), q: UART number (q = 0, 1, 3)
□ : Setting is fixed in the UART reception mode, ▒ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 11-96. Flowchart of Address Field Transmission

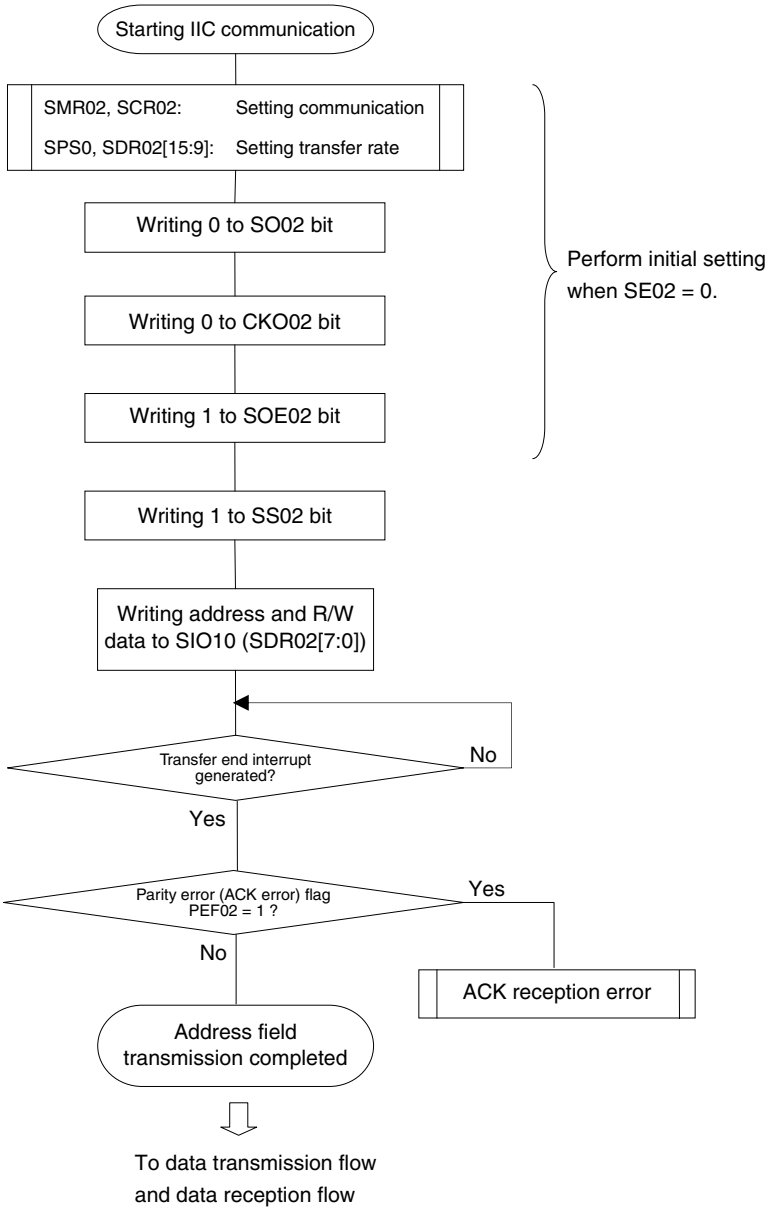
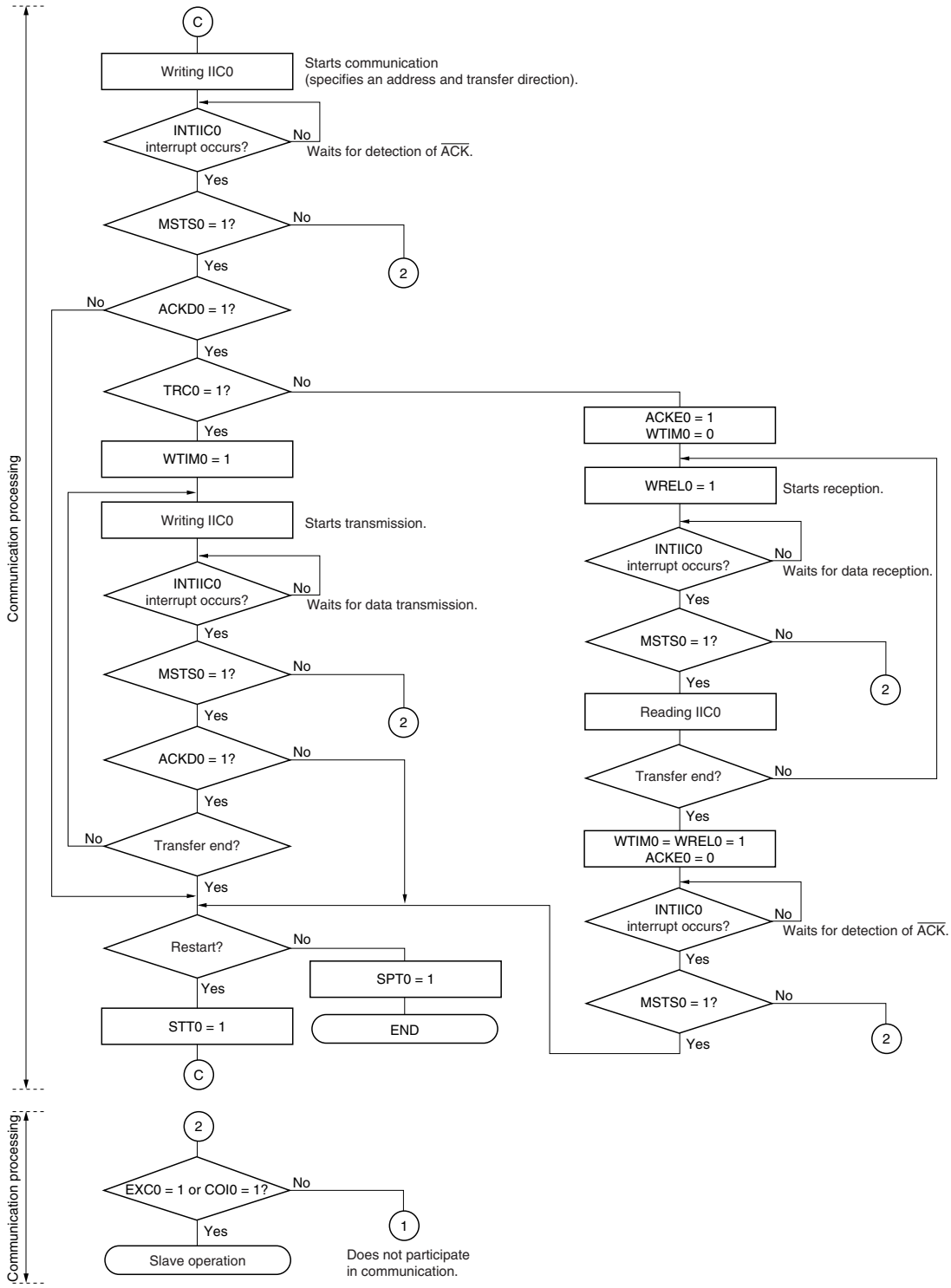


Figure 12-25. Master Operation in Multi-Master System (3/3)



Remarks 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

2. To use the device as a master in a multi-master system, read the MSTS0 bit each time interrupt INTIIC0 has occurred to check the arbitration result.

3. To use the device as a slave in a multi-master system, check the status by using the IICS0 and IICF0 registers each time interrupt INTIIC0 has occurred, and determine the processing to be performed next.

(2) DMA operation control register n (DRCn)

DRCn is a register that is used to enable or disable transfer of DMA channel n.

Rewriting bit 7 (DENn) of this register is prohibited during operation (when DSTn = 1).

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

Reset signal generation clears this register to 00H.

Figure 14-5. Format of DMA Operation Control Register n (DRCn)

Address: FFFBCH (DRC0), FFFBDH (DRC1) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
DRCn	DENn	0	0	0	0	0	0	DSTn

DENn	DMA operation enable flag
0	Disables operation of DMA channel n (stops operating cock of DMA).
1	Enables operation of DMA channel n.
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1).	

DSTn	DMA transfer mode flag
0	DMA transfer of DMA channel n is completed.
1	DMA transfer of DMA channel n is not completed (still under execution).
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1). When a software trigger (STGn) or the start source trigger set by IFCn3 to IFCn0 is input, DMA transfer is started. When DMA transfer is completed after that, this bit is automatically cleared to 0. Write 0 to this bit to forcibly terminate DMA transfer under execution.	

Caution The DSTn flag is automatically cleared to 0 when a DMA transfer is completed. Writing the DENn flag is enabled only when DSTn = 0. When a DMA transfer is terminated without waiting for generation of the interrupt (INTDMAn) of DMAn, therefore, set DSTn to 0 and then DENn to 0 (for details, refer to 14.5.7 Forcible termination by software).

Remark n: DMA channel number (n = 0, 1)

14.6 Cautions on Using DMA Controller

<R> **(1) Priority of DMA**

During DMA transfer, a request from the other DMA channel is held pending even if generated. The pending DMA transfer is started after the ongoing DMA transfer is completed. If two DMA requests are generated at the same time, however, DMA channel 0 takes priority over DMA channel 1.

If a DMA request and an interrupt request are generated at the same time, the DMA transfer takes precedence, and then interrupt servicing is executed.

CHAPTER 15 INTERRUPT FUNCTIONS

15.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into four priority groups by setting the priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H).

Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupt requests, each having the same priority, are simultaneously generated, then they are processed according to the priority of vectored interrupt servicing. For the priority order, see **Table 15-1**.

A standby release signal is generated and STOP and HALT modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

External: 13, internal: 25

(2) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

15.2 Interrupt Sources and Configuration

The 78K0R/KE3 has a total of 39 interrupt sources including maskable interrupts and software interrupts. In addition, they also have up to five reset sources (see **Table 15-1**). The vector codes that store the program start address when branching due to the generation of a reset or various interrupt requests are two bytes each, so interrupts jump to a 64 K address of 00000H to 0FFFFH.

16.3 Register Controlling Key Interrupt

(1) Key return mode register (KRM)

This register controls the KRM0 to KRM7 bits using the KR0 to KR7 signals, respectively.

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

Reset signal generation clears this register to 00H.

Figure 16-2. Format of Key Return Mode Register (KRM)

Address: FFF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	Key interrupt mode control
0	Does not detect key interrupt signal
1	Detects key interrupt signal

Cautions 1. If any of the KRM0 to KRM7 bits used is set to 1, set bits 0 to 7 (PU70 to PU77) of the corresponding pull-up resistor register 7 (PU7) to 1.

2. An interrupt will be generated if the target bit of the KRM register is set while a low level is being input to the key interrupt input pin. To ignore this interrupt, set the KRM register after disabling interrupt servicing by using the interrupt mask flag. Afterward, clear the interrupt request flag and enable interrupt servicing after waiting for the key interrupt input low-level width (250 ns or more).

3. The bits not used in the key interrupt mode can be used as normal ports.

Remark n = 0 to 7

<R> **(2) Port mode register 7 (PM7)**

This register sets the input or output of port 7 in 1-bit units.

When using the P70/KR0, P71/KR1, P72/KR2, P73/KR3, P74/KR4/INTP8, P75/KR5/INTP9, P76/KR6/INTP10, P77/KR7/INTP11 pins as the key interrupt function, set both PM70 to PM77 to 1. The output latches of P70 to P77 at this time may be 0 or 1.

PM7 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 16-3. Format of Port Mode Register 7 (PM7)

Address: FFF27H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	P7n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark n = 0 to 7

CHAPTER 25 BCD CORRECTION CIRCUIT

25.1 BCD Correction Circuit Function

The result of addition/subtraction of the BCD (binary-coded decimal) code and BCD code can be obtained as BCD code with this circuit.

The decimal correction operation result is obtained by performing addition/subtraction having the A register as the operand and then adding/ subtracting the BCDADJ register.

25.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

(1) BCD correction result register (BCDADJ)

The BCDADJ register stores correction values for obtaining the add/subtract result as BCD code through add/subtract instructions using the A register as the operand.

The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags.

BCDADJ is read by an 8-bit memory manipulation instruction.

Reset input sets this register to undefined.

Figure 25-1. Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH After reset: undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

(2) Subtraction: Calculating the result of subtracting a BCD code value from another BCD code value by using a BCD code value

<1> The BCD code value from which subtraction is performed is stored in the A register.

<2> By subtracting the value of the second operand (value of BCD code to be subtracted) from the A register as is in binary, the calculation result in binary is stored in the A register, and the correction value is stored in the BCDADJ register.

<3> Decimal correction is performed by subtracting the value of the BCDADJ register (correction value) from the A register (subtraction result in binary) in binary, and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Example: 91 − 52 = 39

Instruction	A Register	CY Flag	AC Flag	BCDADJ Register
`MOV A, #91H ; <1>`	91H	−	−	−
`SUB A, #52H ; <2>`	3FH	0	1	06H
`SUB A, !BCDADJ ; <3>`	39H	0	0	−

CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)

Target products μPD78F1142 A(A), 78F1143 A(A), 78F1144 A(A), 78F1145 A(A), 78F1146 A(A)

Caution The 78K0R/KE3 has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. NEC Electronics is not liable for problems occurring when the on-chip debug function is used.

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		−0.5 to +6.5	V
	EV_{DD}		−0.5 to +6.5	V
	V_{SS}		−0.5 to +0.3	V
	EV_{SS}		−0.5 to +0.3	V
	AV_{REF}		−0.5 to V_{DD} +0.3 Note 1	V
	AV_{SS}		−0.5 to +0.3	V
REGC pin input voltage	V_{IREGC}	REGC	−0.3 to +3.6 and −0.3 to V_{DD} +0.3 Note 2	V
Input voltage	V_{I1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P70 to P77, P120 to P124, P140, P141, EXCLK, $\overline{\text{RESET}}$, FLMD0	−0.3 to EV_{DD} +0.3 and −0.3 to V_{DD} +0.3 Note 1	V
	V_{I2}	P60 to P63 (N-ch open-drain)	−0.3 to +6.5	V
	V_{I3}	P20 to P27	−0.3 to AV_{REF} +0.3 and −0.3 to V_{DD} +0.3 Note 1	V
Output voltage	V_{O1}	P00 to P06, P10 to P17, P30, P31, P40 to P43, P50 to P55, P60 to P63, P70 to P77, P120, P130, P140, P141	−0.3 to EV_{DD} +0.3 Note 1	V
	V_{O2}	P20 to P27	−0.3 to AV_{REF} +0.3	V
Analog input voltage	V_{AN}	ANI0 to ANI7	−0.3 to AV_{REF} +0.3 Note 1 and −0.3 to V_{DD} +0.3 Note 1	V

Notes 1. Must be 6.5 V or lower.

2. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (9/10)

(T_A = −40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current	I_{DD2} Note 1	HALT mode	f_{SUB} = 32.768 kHz Note 2, T_A = −40 to +70 °C	V_{DD} = 5.0 V		2.2	14.0	μA
				V_{DD} = 3.0 V		2.2	14.0	μA
				V_{DD} = 2.0 V		2.1	13.8	μA
			f_{SUB} = 32.768 kHz Note 2, T_A = −40 to +85 °C	V_{DD} = 5.0 V		2.2	21.0	μA
				V_{DD} = 3.0 V		2.2	21.0	μA
				V_{DD} = 2.0 V		2.1	20.8	μA
	I_{DD3} Note 3	STOP mode	T_A = −40 to +70 °C			1.1	9.0	μA
			T_A = −40 to +85 °C			1.1	16.0	μA

Notes 1. Total current flowing into V_{DD}, EV_{DD}, and AV_{REF}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The maximum value include the peripheral operation current. However, not including the current flowing into the A/D converter, LVI circuit, I/O port, and on-chip pull-up/pull-down resistors. During HALT instruction execution by flash memory.

2. When internal high-speed oscillator and high-speed system clock are stopped. When watchdog timer is stopped.

3. Total current flowing into V_{DD}, EV_{DD}, and AV_{REF}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. When subsystem clock is stopped. When watchdog timer is stopped.

Remarks 1. f_{SUB} : Subsystem clock frequency (XT1 clock oscillation frequency)

<R> **2.** Temperature condition of the TYP. value is T_A = 25°C

(2) Serial interface: Serial array unit (9/18)

Remarks 1. R_b[Ω]:Communication line (TxD1) pull-up resistance,
C_b[F]: Communication line (TxD1) load capacitance, V_b[V]: Communication line voltage

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKS0n bit of the SMR0n register. n: Channel number (n = 2, 3))

3. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in UART mode.
$4.0\ V \le V_{DD} \le 5.5\ V$, $2.7\ V \le V_b \le 4.0\ V$: $V_{IH} = 2.2\ V$, $V_{IL} = 0.8\ V$
$2.7\ V \le V_{DD} \le 4.0\ V$, $2.3\ V \le V_b \le 2.7\ V$: $V_{IH} = 2.0\ V$, $V_{IL} = 0.5\ V$

4. UART0 and UART3 cannot communicate at different potential. Use UART1 for communication at different potential.

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 3	Soft	Memory space	Internal data memory space	It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.	p.56 ☐
				While using the self-programming function, the area of FFE20H to FFEFFH cannot be used as a stack memory. Furthermore, the areas of FCF00H to FD6FFH cannot be used with the μPD78F1146 and 78F1146A.	p.56 ☐
			SFR: Special function register area	Do not access addresses to which SFRs are not assigned.	pp.57, 68 ☐
			2nd SFR: Extended special function register	Do not access addresses to which extended SFR is not assigned.	pp.57, 74 ☐
		Processor registers	SP: Stack pointer	Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.	p.64 ☐
				The values of the stack pointer must be set to even numbers. If odd numbers are specified, the least significant bit is automatically cleared to 0.	p.64 ☐
				It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space as a stack area.	p.64 ☐
				While using the self-programming function, the area of FFE20H to FFEFFH cannot be used as a stack memory. Furthermore, the areas of FCF00H to FD6FFH cannot be used with the μPD78F1146 and 78F1146A.	p.64 ☐
			General-purpose registers	It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.	p.65 ☐
Chapter 4	Soft	Port functions	P01/TO00, P05/TI05/TO05, P06/TI06/TO06	To use P01/TO00, P05/TI05/TO05, or P06/TI06/TO06 as a general-purpose port, set bits 0, 5,and 6 (TO00, TO05, TO06) of timer output register 0 (TO0) and bits 0, 5, and 6 (TOE00, TOE05,TOE06) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.	p.96 ☐
			P02/SO10/TxD1, P03/SI10/RxD1/SDA10, P04/$\overline{\text{SCK10}}$/SCL10	To use P02/SO10/TxD1, P03/SI10/RxD1/SDA10, or P04/$\overline{\text{SCK10}}$/SCL10 as a general-purpose port, note the serial array unit 0 setting. For details, refer to the following tables. • Table 11-7 Relationship Between Register Settings and Pins (Channel 2 of Unit 0: CSI10, UART1 Transmission, IIC10) • Table 11-8 Relationship Between Register Settings and Pins (Channel 3 of Unit 0: UART1 Reception)	p.96 ☐
			P10/$\overline{\text{SCK00}}$/, P11/SI00/RxD0, P12/SO00/TxD0 P13/TxD3, P14/RxD3	To use P10/$\overline{\text{SCK00}}$, P11/SI00/RxD0, P12/SO00/TxD0, P13/TxD3, or P14/RxD3 as a general-purpose port, note the serial array unit setting. For details, refer to the following tables. • Table 11-5 Relationship Between Register Settings and Pins (Channel 0 of Unit 0: CSI00, UART0 Transmission) • Table 11-6 Relationship Between Register Settings and Pins (Channel 1 of Unit 0:, UART0 Reception) • Table 11-9 Relationship Between Register Settings and Pins (Channel 2 of Unit 1: UART3 Transmission) • Table 11-10 Relationship Between Register Settings and Pins (Channel 3 of Unit 1: UART3 Reception)	p.101 ☐

(3/33)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 4	Soft	Port functions	P16/TI01/TO01/INTP, P17/TI02/TO02	To use P16/TI01/TO01/INTP5 or P17/TI02/TO02 as a general-purpose port, set bits 1 and 2 (TO01, TO02) of timer output register 0 (TO0) and bits 1 and 2 (TOE01, TOE02) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.	p.101 ☐
			P15/RTCDIV/RTCCL	To use P15/RTCDIV/RTCCL as a general-purpose port, set bit 4 (RCLOE0) of real-time counter control register 0 (RTCC0) and bit 6 (RCLOE2) of real-time counter control register 2 (RTCC2) to “0”, which is the same as their default status settings.	p.101 ☐
	Hard		Port 2	See 2.2.12 AV_{REF} for the voltage to be applied to the AV_{REF} pin when using port 2 as a digital I/O.	p.107 ☐
	Soft		P31/TI03/TO03/INTP4	To use P31/TI03/TO03/INTP4 as a general-purpose port, set bit 3 (TO03) of timer output register 0 (TO0) and bit 3 (TOE03) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.	p.109 ☐
			P30/RTC1HZ/INTP3	To use P30/RTC1HZ/INTP3 as a general-purpose port, set bit 5 (RCLOE1) of real-time counter control register 0 (RTCC0) to “0”, which is the same as its default status setting.	p.109 ☐
			P40/TOOL0, P41/TOOL1	When a tool is connected, the P40 pin cannot be used as a port pin. When the on-chip debug function is used, P41 pin can be used as follows by the mode setting on the debugger. 1-line mode: can be used as a port (P41). 2-line mode: used as a TOOL1 pin and cannot be used as a port (P41).	p.110 ☐
			P42/TI04/TO04	To use P42/TI04/TO04 as a general-purpose port, set bit 4(TO04) of timer output register 0 (TO0) and bit 4(TOE04) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.	p.110 ☐
			P60/SCL0, P61/SDA0	When using P60/SCL0 or P61/SDA0 as a general-purpose port, stop the operation of serial interface IIC0.	p.117 ☐
			P121 to P124	The function setting on P121 to P124 is available only once after the reset release. The port once set for connection to an oscillator cannot be used as an input port unless the reset is performed.	p.120 ☐
			P140/PCLBUZ0/INTP6, P141/PCLBUZ1/INTP7	To use P140/PCLBUZ0/INTP6 or P141/PCLBUZ1/INTP7 as a general-purpose port, set bit 7 of clock output select registers 0 and 1 (CKS0, CKS1) to “0”, which is the same as their default status settings.	p.124 ☐
			PM0 to PM7, PM12 to PM14: Port mode registers	Be sure to set bit 7 of PM0, bits 2 to 7 of PM3, bits 4 to 7 of PM4, bits 6 and 7 of PM5, bits 4 to 7 of PM6, bits 1 to 7 of PM12, and bits 2 to 7 of PM14 to “1”.	p.127 ☐
			ADPC: A/D port configuration register	Set the channel used for A/D conversion to the input mode by using port mode registers 2 (PM2).	p.132 ☐
				Do not set the pin set by ADPC as digital I/O by analog input channel specification register (ADS).	p.132 ☐
				When using all ANI0/P20 to ANI7/P27 pins as digital I/O (D), the setting can be done by ADPC4 to ADPC0 = either 01000 or 10000.	p.132 ☐
				P20/ANI0 to P27/ANI7 are set as analog inputs in the order of P27/ANI7, …, P20/ANI0 by the A/D port configuration register (ADPC). When using P20/ANI0 to P27/ANI7 as analog inputs, start designing from P27/ANI7.	p.132 ☐

(6/33)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 5	Soft	Clock generator	HIOTRM: Internal-high-speed oscillator trimming register	The internal high-speed oscillation frequency becomes faster/slower by increasing/decreasing the HIOTRM value to a value larger/smaller than a certain value. A reversal, such as the frequency becoming slower/faster by increasing/decreasing the HIOTRM value does not occur.	p.155 ☐
	Hard	X1/XT1 oscillator	–	When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 5-10 and 5-11 to avoid an adverse effect from wiring capacitance. • Keep the wiring length as short as possible. • Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows. • Always make the ground point of the oscillator capacitor the same potential as V_{SS}. Do not ground the capacitor to a ground pattern through which a high current flows. • Do not fetch signals from the oscillator. Note that the XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption.	p.157 ☐
				When X2 and XT1 are wired in parallel, the crosstalk noise of X2 may increase with XT1, resulting in malfunctioning.	p.158 ☐
		Clock generator operation when power supply voltage is turned on	When LVI default start function stopped is set (option byte: LVIOFF = 1)	If the voltage rises with a slope of less than 0.5 V/ms (MIN.) from power application until the voltage reaches 1.8 V, input a low level to the $\overline{\text{RESET}}$ pin from power application until the voltage reaches 1.8 V, or set the LVI default start function stopped by using the option byte (LVIOFF = 0) (see Figure 5-14). By doing so, the CPU operates with the same timing as <2> and thereafter in Figure 5-13 after reset release by the $\overline{\text{RESET}}$ pin.	p.162 ☐
				It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.	p.162 ☐
			When LVI default start function enabled is set (option byte: LVIOFF = 0)	A voltage oscillation stabilization time is required after the supply voltage reaches 1.59 V (TYP.). If the supply voltage rises from 1.59 V (TYP.) to 2.07 V (TYP.) within the power supply oscillation stabilization time, the power supply oscillation stabilization time is automatically generated before reset processing.	p.163 ☐
				It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.	p.163 ☐
	Soft	Controlling high-speed system clock	X1/P121, X2/EXCLK/P122	The X1/P121 and X2/EXCLK/P122 pins are in the input port mode after a reset release.	p.164 ☐
			X1 clock	The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction. Therefore, it is necessary to also set the value of the OSCSELS bit at the same time. For OSCSELS bit, see 5.6.3 Example of controlling subsystem clock.	p.164 ☐
				Set the X1 clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 27 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS) and CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)).	p.164 ☐
			External main system clock	The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction. Therefore, it is necessary to also set the value of the OSCSELS bits at the same time. For OSCSELS bits, see 5.6.3 Example of controlling subsystem clock.	p.165 ☐