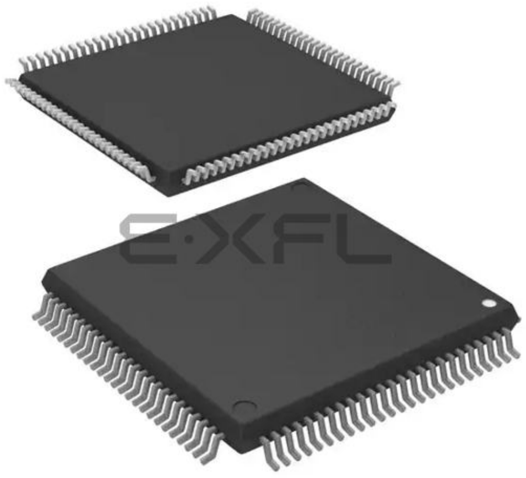


Welcome to E-XFL.COM

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	V850ES
Core Size	32-Bit Single-Core
Speed	32MHz
Connectivity	CSI, EBI/EMI, I²C, UART/USART
Peripherals	DMA, LVD, PWM, WDT
Number of I/O	84
Program Memory Size	768KB (768K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	60K x 8
Voltage - Supply (Vcc/Vdd)	2.85V ~ 3.6V
Data Converters	A/D 12x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd70f3741gc-ueu-ax

(2) NMI status saving registers (FEPC and FEPSW)

FEPC and FEPSW are used to save the status when a non-maskable interrupt (NMI) occurs.

If an NMI occurs, the contents of the program counter (PC) are saved to FEPC, and those of the program status word (PSW) are saved to FEPSW.

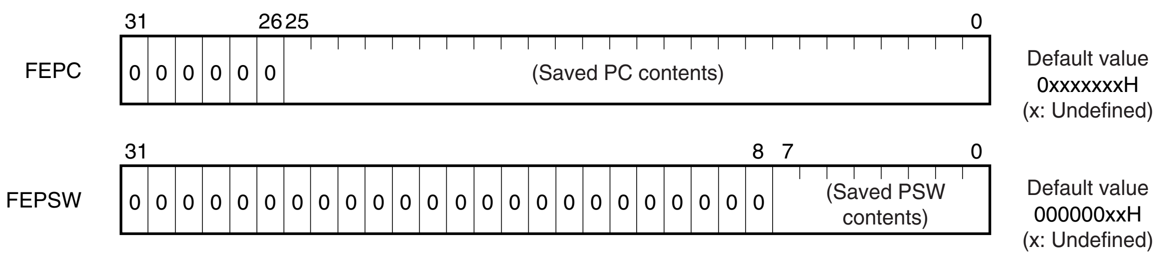
The address of the instruction next to the one of the instruction under execution, except some instructions, is saved to FEPC when an NMI occurs.

The current contents of the PSW are saved to FEPSW.

Because only one set of NMI status saving registers is available, the contents of these registers must be saved by program when multiple interrupts are enabled.

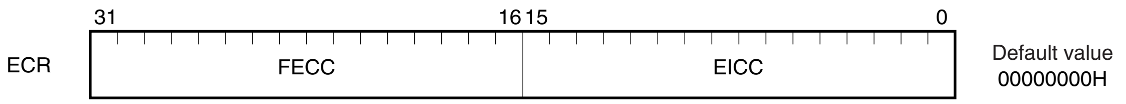
Bits 31 to 26 of FEPC and bits 31 to 8 of FEPSW are reserved for future function expansion (these bits are always fixed to 0).

The value of FEPC is restored to the PC and the value of FEPSW to the PSW by the RETI instruction.



(3) Interrupt source register (ECR)

The interrupt source register (ECR) holds the source of an exception or interrupt if an exception or interrupt occurs. This register holds the exception code of each interrupt source. Because this register is a read-only register, data cannot be written to this register using the LDSR instruction.



Bit position	Bit name	Meaning
31 to 16	FECC	Exception code of non-maskable interrupt (NMI)
15 to 0	EICC	Exception code of exception or maskable interrupt

(7) Port 9 function register (PF9)

After reset: 0000H R/W Address: PF3 FFFFFC72H,
PF9L FFFFFC72H, PF9H FFFFFC73H

	15	14	13	12	11	10	9	8
PF9 (PF9H)	PF915	PF914	PF913	PF912	PF911	PF910	PF99	PF98

	7	6	5	4	3	2	1	0
(PF9L)	PF97	PF96	PF95	PF94	PF93	PF92	PF91	PF90

PF9n	Control of normal output or N-ch open-drain output (n = 0 to 15)
0	Normal output (CMOS output)
1	N-ch open-drain output

Caution When an output pin is pulled up at EV_{DD} or higher, be sure to set the PF9n bit to 1.

Remarks 1. The PF9 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PF9 register as the PF9H register and the lower 8 bits as the PF9L register, PF9 can be read or written in 8-bit or 1-bit units.

2. To read/write bits 8 to 15 of the PF9 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PF9H register.

(3) Port CT mode control register (PMCCT)

After reset: 00H R/W Address: FFFFF04AH

	7	6	5	4	3	2	1	0
PMCCT	0	PMCCT6	0	PMCCT4	0	0	PMCCT1	PMCCT0

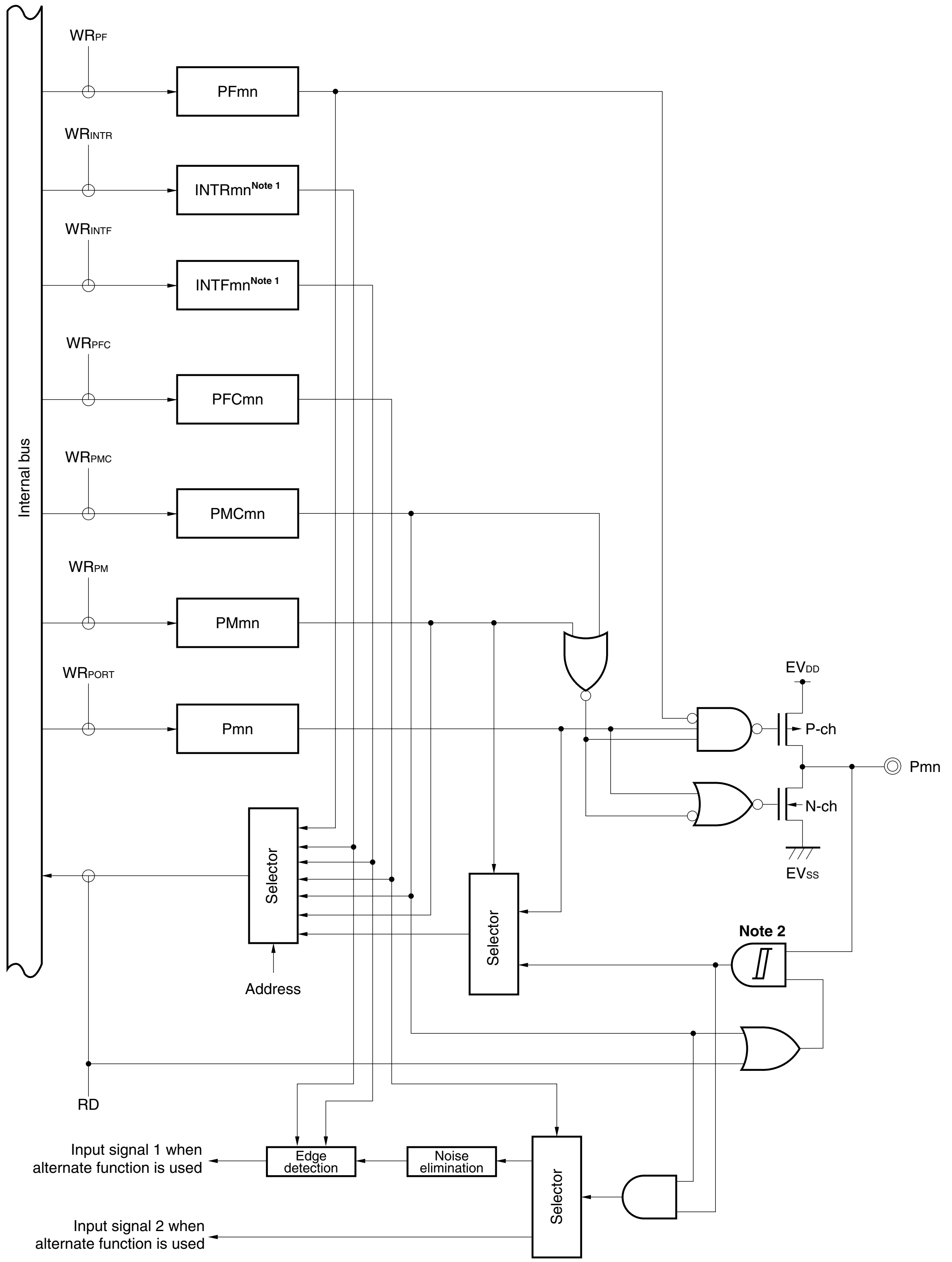
PMCCT6	Specification of PCT6 pin operation mode
0	I/O port
1	ASTB output

PMCCT4	Specification of PCT4 pin operation mode
0	I/O port
1	$\overline{\text{RD}}$ output

PMCCT1	Specification of PCT1 pin operation mode
0	I/O port
1	$\overline{\text{WR1}}$ output

PMCCT0	Specification of PCT0 pin operation mode
0	I/O port
1	$\overline{\text{WR0}}$ output

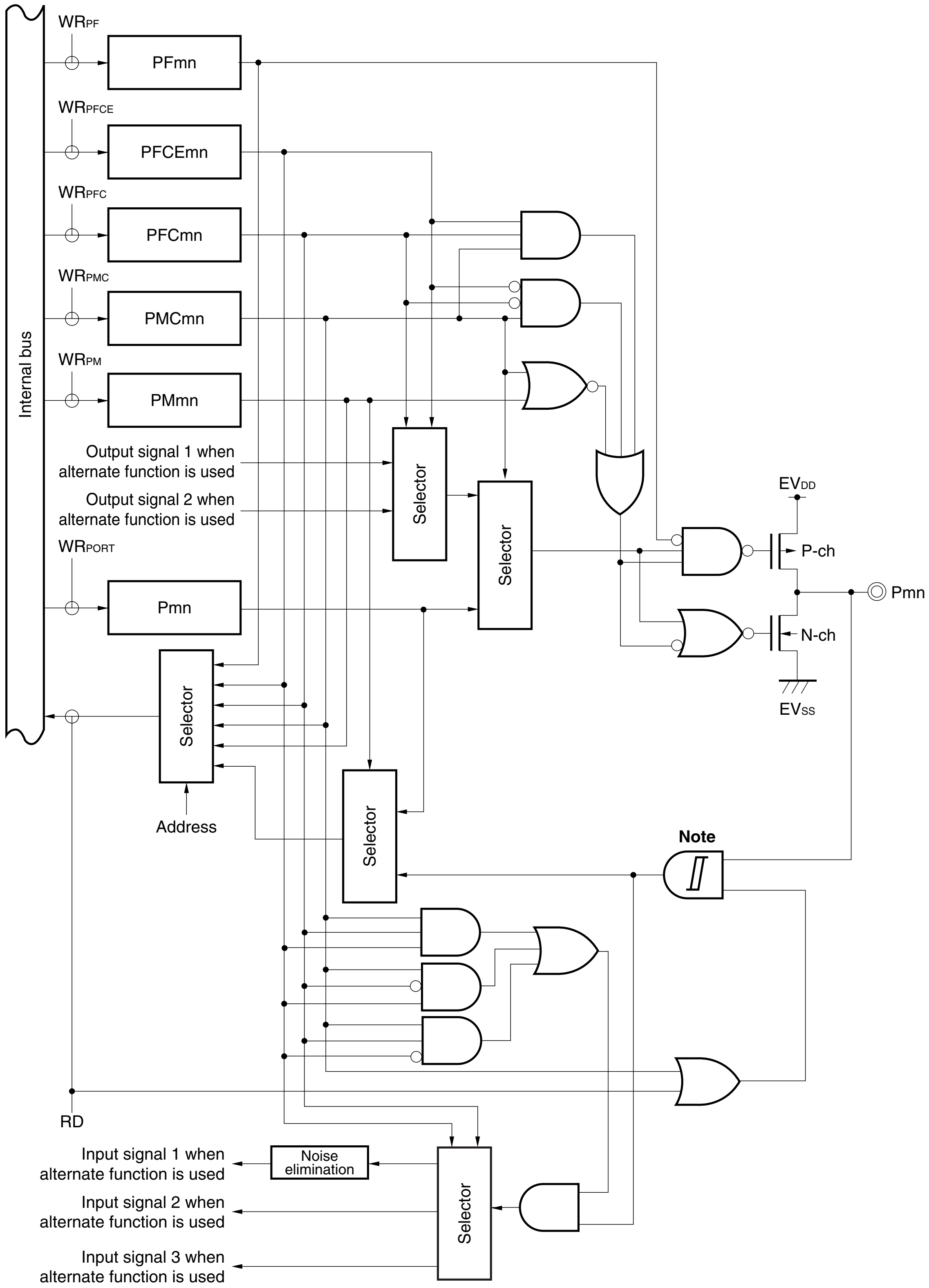
Figure 4-17. Block Diagram of Type N-1



Notes 1. See **19.6 External Interrupt Request Input Pins (NMI and INTP0 to INTP7)**.

2. Hysteresis characteristics are not available in port mode.

Figure 4-27. Block Diagram of Type U-11



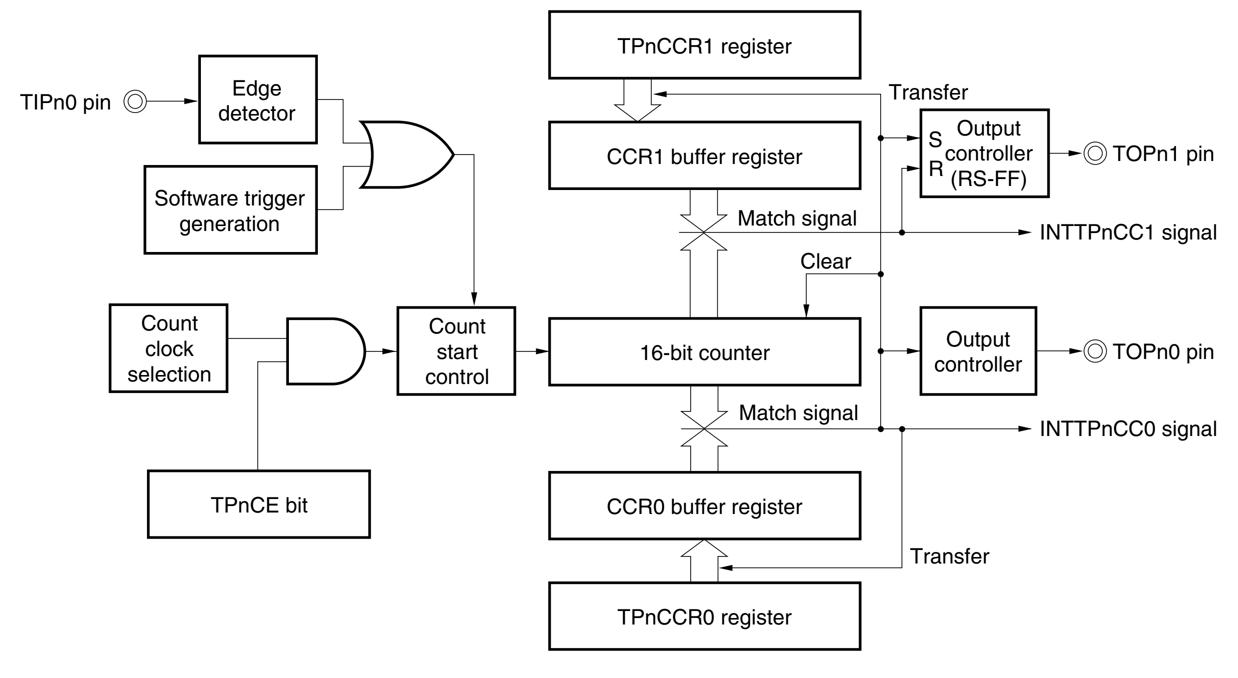
Note Hysteresis characteristics are not available in port mode.

7.5.3 External trigger pulse output mode (TPnMD2 to TPnMD0 bits = 010)

In the external trigger pulse output mode, 16-bit timer/event counter P waits for a trigger when the TPnCTL0.TPnCE bit is set to 1. When the valid edge of an external trigger input signal is detected, 16-bit timer/event counter P starts counting, and outputs a PWM waveform from the TOPn1 pin.

Pulses can also be output by generating a software trigger instead of using the external trigger. When using a software trigger, a square wave that has one cycle of the PWM waveform as half its cycle can also be output from the TOPn0 pin.

Figure 7-16. Configuration in External Trigger Pulse Output Mode



Remark n = 0 to 5

Figure 7-18. Register Setting for Operation in External Trigger Pulse Output Mode (2/2)

(d) TMPn I/O control register 2 (TPnIOC2)

							TPnEES1	TPnEES0	TPnETS1	TPnETS0
TPnIOC2	0	0	0	0	0	0	0/1	0/1

Select valid edge of external trigger input

(e) TMPn counter read buffer register (TPnCNT)

The value of the 16-bit counter can be read by reading the TPnCNT register.

(f) TMPn capture/compare registers 0 and 1 (TPnCCR0 and TPnCCR1)

If D_0 is set to the TPnCCR0 register and D_1 to the TPnCCR1 register, the cycle and active level of the PWM waveform are as follows.

Cycle = $(D_0 + 1) \times$ Count clock cycle

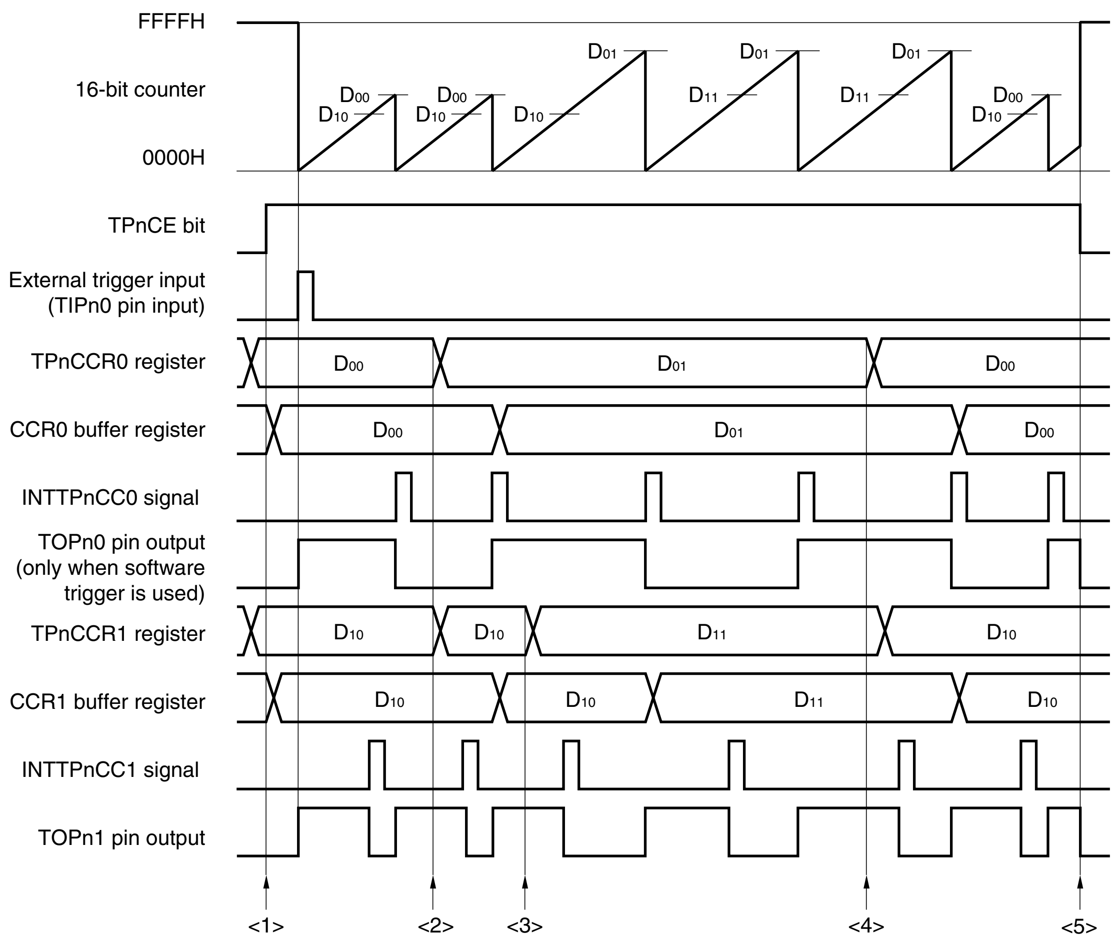
Active level width = $D_1 \times$ Count clock cycle

Remarks 1. TMPn I/O control register 1 (TPnIOC1) and TMPn option register 0 (TPnOPT0) are not used in the external trigger pulse output mode.

2. n = 0 to 5

(1) Operation flow in external trigger pulse output mode

Figure 7-19. Software Processing Flow in External Trigger Pulse Output Mode (1/2)



Remark n = 0 to 5

Figure 8-26. Register Setting for Operation in PWM Output Mode (3/3)

(f) TMQ0 capture/compare registers 0 to 3 (TQ0CCR0 to TQ0CCR3)

If D_0 is set to the TQ0CCR0 register and D_k to the TQ0CCR1 register, the cycle and active level of the PWM waveform are as follows.

Cycle = $(D_0 + 1) \times$ Count clock cycle

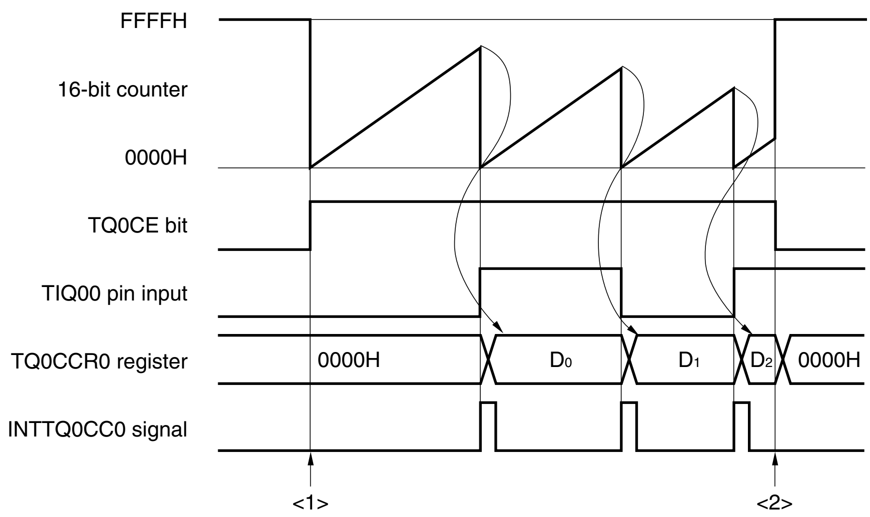
Active level width = $D_k \times$ Count clock cycle

Remarks 1. TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the PWM output mode.

2. Updating the TMQ0 capture/compare register 2 (TQ0CCR2) and TMQ0 capture/compare register 3 (TQ0CCR3) is validated by writing the TMQ0 capture/compare register 1 (TQ0CCR1).

(1) Operation flow in pulse width measurement mode

Figure 8-37. Software Processing Flow in Pulse Width Measurement Mode



<1> Count operation start flow

START

Register initial setting
TQ0CTL0 register
(TQ0CKS0 to TQ0CKS2 bits),
TQ0CTL1 register,
TQ0IOC1 register,
TQ0IOC2 register,
TQ0OPT0 register

Initial setting of these registers is performed before setting the TQ0CE bit to 1.

Set TQ0CTL0 register
(TQ0CE bit = 1)

The TQ0CKS0 to TQ0CKS2 bits can be set at the same time when counting has been started (TQ0CE bit = 1).

<2> Count operation stop flow

TQ0CE bit = 0

The counter is initialized and counting is stopped by clearing the TQ0CE bit to 0.

STOP

(2/3)

CBnDIR[Note]	Specification of transfer direction mode (MSB/LSB)
0	MSB-first transfer
1	LSB-first transfer

CBnTMS[Note]	Transfer mode specification
0	Single transfer mode
1	Continuous transfer mode

[In single transfer mode]
The reception complete interrupt request signal (INTCBnR) is generated.
Even if transmission is enabled (CBnTXE bit = 1), the transmission enable interrupt request signal (INTCBnT) is not generated.
If the next transmit data is written during communication (CBnSTR.CBnTSF bit = 1), it is ignored and the next communication is not started. Also, if reception-only communication is set (CBnTXE bit = 0, CBnRXE bit = 1), the next communication is not started even if the receive data is read during communication (CBnSTR. CBbTSF bit = 1).
[In continuous transfer mode]
The continuous transmission is enabled by writing the next transmit data during communication (CBnSTR.CBnTSF bit = 1). Writing the next transmission data is enabled after a transmission enable interrupt (INTCBnT) occurrence.
If reception-only communication is set (CBnTXE bit = 0, CBnRXE bit = 1) in the continuous transfer mode, the next reception is started continuously after a reception complete interrupt (INTCBnR) regardless of the read operation of the CBnRX register.
Therefore, read immediately the receive data from the CBnRX register. If this read operation is delayed, an overrun error (CBnOVE bit = 1) occurs.

Note These bits can only be rewritten when the CBnPWR bit = 0.
However, CBnPWR bit = 1 can also be set at the same time as rewriting these bits.

(3) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When WTIMn bit = 0 (after restart, extension code reception)

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP

▲1 ▲2 ▲3 ▲4 △5

▲1: IICSn register = 0010X010B

▲2: IICSn register = 0010X000B

▲3: IICSn register = 0010X010B

▲4: IICSn register = 0010X000B

△ 5: IICSn register = 00000001B

Remarks 1. ▲: Always generated
△: Generated only when SPIEn bit = 1
X: don't care
2. n = 0 to 2

<2> When WTIMn bit = 1 (after restart, extension code reception)

ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	ST	AD6 to AD0	R/$\overline{\text{W}}$	$\overline{\text{ACK}}$	D7 to D0	$\overline{\text{ACK}}$	SP

▲1 ▲2 ▲3 ▲4 ▲5 ▲6 △7

▲1: IICSn register = 0010X010B

▲2: IICSn register = 0010X110B

▲3: IICSn register = 0010XX00B

▲4: IICSn register = 0010X010B

▲5: IICSn register = 0010X110B

▲6: IICSn register = 0010XX00B

△ 7: IICSn register = 00000001B

Remarks 1. ▲: Always generated
△: Generated only when SPIEn bit = 1
X: don't care
2. n = 0 to 2

(8) When arbitration loss occurs due to low level of SDA0n pin when attempting to generate a stop condition

<1> When WTIMn bit = 0

IICCn.SPTn bit = 1
↓

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICSn register = 1000X110B

▲2: IICSn register = 1000X000B (WTIMn bit = 1)

▲3: IICSn register = 1000XX00B (WTIMn bit = 0)

▲4: IICSn register = 01000000B (Example: When ALDn bit is read during interrupt servicing)

Δ 5: IICSn register = 00000001B

Remarks 1. ▲: Always generated
Δ: Generated only when SPIEn bit = 1
X: don't care
2. n = 0 to 2

<2> When WTIMn bit = 1

IICCn.SPTn bit = 1
↓

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲1: IICSn register = 1000X110B

▲2: IICSn register = 1000XX00B

▲3: IICSn register = 01000000B (Example: When ALDn bit is read during interrupt servicing)

Δ 4: IICSn register = 00000001B

Remarks 1. ▲: Always generated
Δ: Generated only when SPIEn bit = 1
X: don't care
2. n = 0 to 2

19.3 Maskable Interrupts

Maskable interrupt request signals can be masked by interrupt control registers. The V850ES/JG3 has 55 maskable interrupt sources.

If two or more maskable interrupt request signals are generated at the same time, they are acknowledged according to the default priority. In addition to the default priority, eight levels of priorities can be specified by using the interrupt control registers (programmable priority control).

When an interrupt request signal has been acknowledged, the acknowledgment of other maskable interrupt request signals is disabled and the interrupt disabled (DI) status is set.

When the EI instruction is executed in an interrupt service routine, the interrupt enabled (EI) status is set, which enables servicing of interrupts having a higher priority than the interrupt request signal in progress (specified by the interrupt control register). Note that only interrupts with a higher priority will have this capability; interrupts with the same priority level cannot be nested.

To enable multiple interrupts, however, save EIPC and EIPSW to memory or general-purpose registers before executing the EI instruction, and execute the DI instruction before the RETI instruction to restore the original values of EIPC and EIPSW.

19.3.1 Operation

If a maskable interrupt occurs, the CPU performs the following processing, and transfers control to a handler routine.

<1> Saves the restored PC to EIPC.
<2> Saves the current PSW to EIPSW.
<3> Writes an exception code to the lower halfword of ECR (EICC).
<4> Sets the PSW. ID bit to 1 and clears the PSW. EP bit to 0.
<5> Sets the handler address corresponding to each interrupt to the PC, and transfers control.

The maskable interrupt request signal masked by INTC and the maskable interrupt request signal generated while another interrupt is being serviced (while the PSW.NP bit = 1 or the PSW.ID bit = 1) are held pending inside INTC. In this case, servicing a new maskable interrupt is started in accordance with the priority of the pending maskable interrupt request signal if either the maskable interrupt is unmasked or the NP and ID bits are cleared to 0 by using the RETI or LDSR instruction.

How maskable interrupts are serviced is illustrated below.

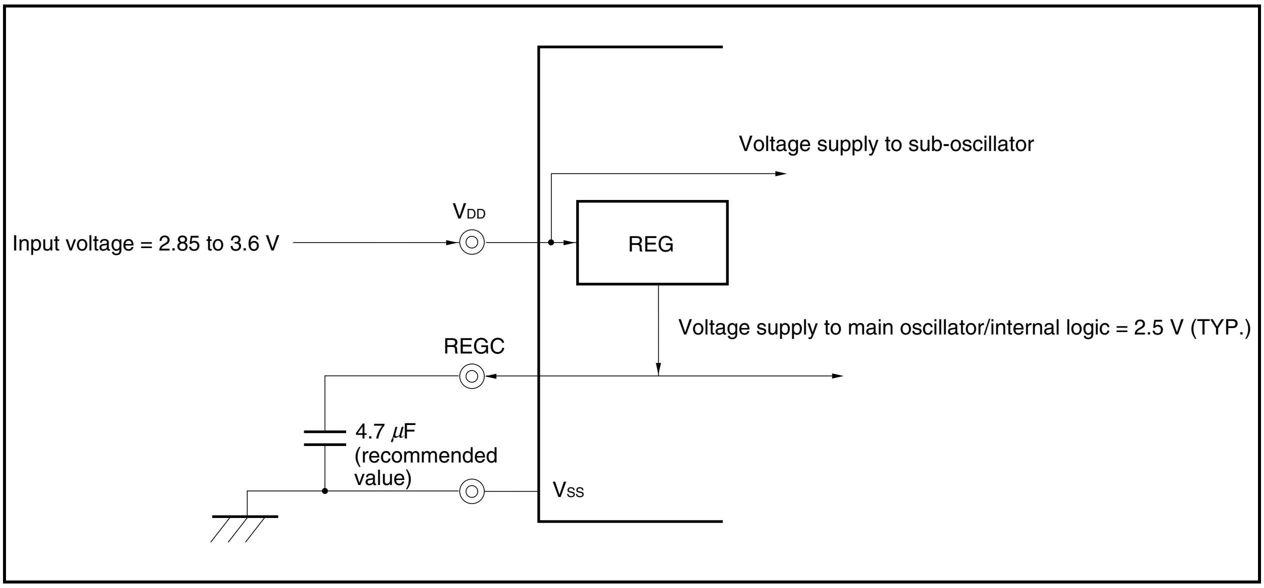
26.2 Operation

The regulator of this product always operates in any mode (normal operation mode, HALT mode, IDLE1 mode, IDLE2 mode, STOP mode, or during reset).

Be sure to connect a capacitor (4.7 μF (recommended value)) to the REGC pin to stabilize the regulator output.

A diagram of the regulator pin connection method is shown below.

Figure 26-2. REGC Pin Connection



(3) Bus hold

(a) CLKOUT asynchronous

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

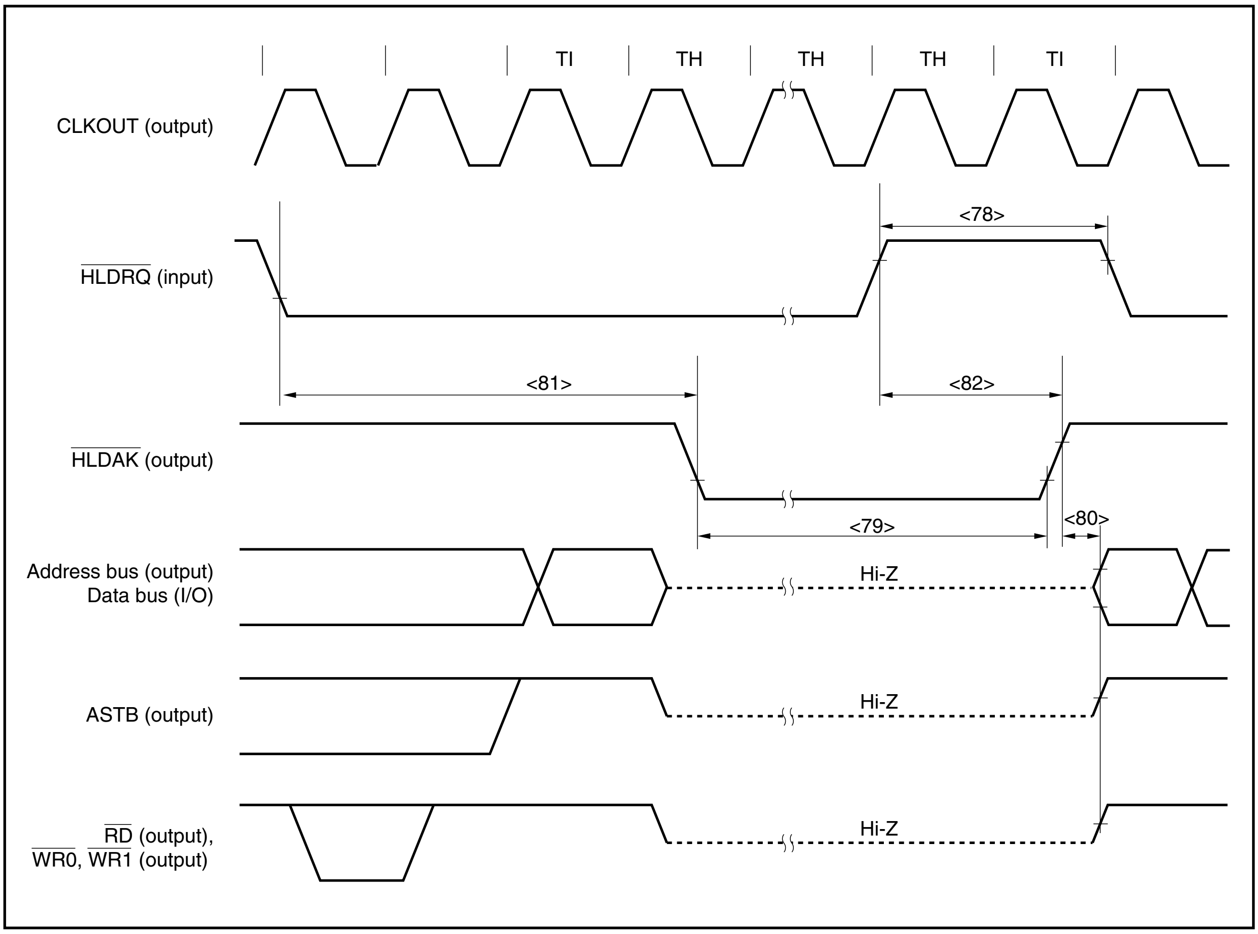
Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{HLDRQ}}$ high-level width	t_{WHQH}	<78>		T + 10		ns
$\overline{\text{HLDAK}}$ low-level width	t_{WHAL}	<79>		T – 15		ns
Delay time from $\overline{\text{HLDAK}}$↑ to bus output	t_{DHAC}	<80>		–3		ns
Delay time from $\overline{\text{HLDRQ}}$↓ to $\overline{\text{HLDAK}}$↓	t_{DHQHA1}	<81>			(2n + 7.5)T + 26	ns
Delay time from $\overline{\text{HLDRQ}}$↑ to $\overline{\text{HLDAK}}$↑	t_{DHQHA2}	<82>		0.5T	1.5T + 26	ns

Remarks 1. T = 1/f_{CPU} (f_{CPU}: CPU operating clock frequency)

2. n: Number of wait clocks inserted in the bus cycle
The sampling timing changes when a programmable wait is inserted.

3. The values in the above specifications are values for when clocks with a 1:1 duty ratio are input from X1.

Bus Hold (CLKOUT Asynchronous)



(2/10)

Symbol	Name	Unit	Page
CB0RX	CSIB0 receive data register	CSI	487
CB0RXL	CSIB0 receive data register L	CSI	487
CB0STR	CSIB0 status register	CSI	494
CB0TIC	Interrupt control register	INTC	651
CB0TX	CSIB0 transmit data register	CSI	487
CB0TXL	CSIB0 transmit data register L	CSI	487
CB1CTL0	CSIB1 control register 0	CSI	488
CB1CTL1	CSIB1 control register 1	CSI	491
CB1CTL2	CSIB1 control register 2	CSI	492
CB1RIC	Interrupt control register	INTC	651
CB1RX	CSIB1 receive data register	CSI	505
CB1RXL	CSIB1 receive data register L	CSI	505
CB1STR	CSIB1 status register	CSI	512
CB1TIC	Interrupt control register	INTC	651
CB1TX	CSIB1 transmit data register	CSI	487
CB1TXL	CSIB1 transmit data register L	CSI	487
CB2CTL0	CSIB2 control register 0	CSI	488
CB2CTL1	CSIB2 control register 1	CSI	491
CB2CTL2	CSIB2 control register 2	CSI	492
CB2RIC	Interrupt control register	INTC	651
CB2RX	CSIB2 receive data register	CSI	505
CB2RXL	CSIB2 receive data register L	CSI	505
CB2STR	CSIB2 status register	CSI	512
CB2TIC	Interrupt control register	INTC	651
CB2TX	CSIB2 transmit data register	CSI	487
CB2TXL	CSIB2 transmit data register L	CSI	487
CB3CTL0	CSIB3 control register 0	CSI	488
CB3CTL1	CSIB3 control register 1	CSI	491
CB3CTL2	CSIB3 control register 2	CSI	492
CB3RIC	Interrupt control register	INTC	651
CB3RX	CSIB3 receive data register	CSI	505
CB3RXL	CSIB3 receive data register L	CSI	505
CB3STR	CSIB3 status register	CSI	512
CB3TIC	Interrupt control register	INTC	651
CB3TX	CSIB3 transmit data register	CSI	487
CB3TXL	CSIB3 transmit data register L	CSI	487
CB4CTL0	CSIB4 control register 0	CSI	488
CB4CTL1	CSIB4 control register 1	CSI	491
CB4CTL2	CSIB4 control register 2	CSI	492
CB4RIC	Interrupt control register	INTC	651
CB4RX	CSIB4 receive data register	CSI	505
CB4RXL	CSIB4 receive data register L	CSI	505
CB4STR	CSIB4 status register	CSI	512
CB4TIC	Interrupt control register	INTC	651

(2/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
DBTRAP		1111100001000000	DBPC←PC+2 (restored PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	3	3	3					
DI		0000011111100000 0000000101100000	PSW.ID←1	1	1	1					
DISPOSE	imm5,list12	0000011001iiiiiL LLLLLLLLLLL00000	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	imm5,list12,[reg1]	0000011001iiiiiL LLLLLLLLLLLRRRRR **Note 5**	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded PC←GR[reg1]	n+3 Note 4	n+3 Note 4	n+3 Note 4					
DIV	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000000	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVH	reg1,reg2	rrrrr000010RRRRR	GR[reg2]←GR[reg2]÷GR[reg1] Note 6	35	35	35		×	×	×	
	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000000	GR[reg2]←GR[reg2]÷GR[reg1] Note 6 GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVHU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000010	GR[reg2]←GR[reg2]÷GR[reg1] Note 6 GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
DIVU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000010	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
EI		1000011111100000 0000000101100000	PSW.ID←0	1	1	1					
HALT		0000011111100000 0000000100100000	Stop	1	1	1					
HSW	reg2,reg3	rrrrr11111100000 wwwww01101000100	GR[reg3]←GR[reg2](15 : 0) ll GR[reg2] (31 : 16)	1	1	1	×	0	×	×	
JARL	disp22,reg2	rrrrr11110dddddd ddddddddddddddd0 **Note 7**	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	2	2	2					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]	3	3	3					
JR	disp22	0000011110dddddd ddddddddddddddd0 **Note 7**	PC←PC+sign-extend(disp22)	2	2	2					
LD.B	disp16[reg1],reg2	rrrrr111000RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 11					
LD.BU	disp16[reg1],reg2	rrrrr11110bRRRRR ddddddddddddddd1 **Notes 8, 10**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 11					

(18/36)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 14	Hard	D/A converter	D/A converter	DAC0 and DAC1 share the AV_{REF1} pin.	p. 442 ☐
				DAC0 and DAC1 share the AV_{SS} pin. The AV_{SS} pin is also shared by the A/D converter.	p. 442 ☐
			DA0M register	The output trigger in the real-time output mode (DA0MDn bit = 1) is as follows. • When n = 0: INTTP2CC0 signal (see **CHAPTER 7 16-BIT TIMER/EVENT COUNTER P (TMP)**) • When n = 1: INTTP3CC0 signal (see **CHAPTER 7 16-BIT TIMER/EVENT COUNTER P (TMP)**)	p. 443 ☐
	Soft		DA0CS0, DA0CS1 registers	In the real-time output mode (DA0M.DA0MDn bit = 1), set the DA0CSn register before the INTTP2CC0/INTTP3CC0 signals are generated. D/A conversion starts when the INTTP2CC0/INTTP3CC0 signals are generated.	p. 444 ☐
			Cautions	Do not change the set value of the DA0CSn register while the trigger signal is being issued in the real-time output mode.	p. 446 ☐
				Before changing the operation mode, be sure to clear the DA0M.DA0CEn bit to 0.	p. 446 ☐
				When using one of the P10/AN00 and P11/AN01 pins as an I/O port and the other as a D/A output pin, do so in an application where the port I/O level does not change during D/A output.	p. 446 ☐
	Hard			Make sure that AV_{REF0} = V_{DD} = AV_{REF1} = 3.0 to 3.6 V. If this range is exceeded, the operation is not guaranteed.	p. 446 ☐
				Apply power to AV_{REF1} at the same timing as AV_{REF0}.	p. 446 ☐
				No current can be output from the ANOn pin (n = 0, 1) because the output impedance of the D/A converter is high. When connecting a resistor of 2 MΩ or less, insert a JFET input operational amplifier between the resistor and the ANOn pin.	p. 446 ☐
	Soft			Because the D/A converter stops operation in the STOP mode, the ANO0 and ANO1 pins go into a highimpedance state, and the power consumption can be reduced. In the IDLE1, IDLE2, or subclock operation mode, however, the operation continues. To lower the power consumption, therefore, clear the DA0M.DA0CEn bit to 0.	p. 446 ☐
Chapter 15	Soft	Asynchro-nous serial interface A (UARTA)	CSIB4 and UARTA0 mode switching	The transmit/receive operation of CSIB4 and UARTA0 is not guaranteed if these functions are switched during transmission or reception. Be sure to disable the one that is not used.	p. 447 ☐
			UARTA2 and I^2C00 mode switching	The transmit/receive operation of UARTA2 and I^2C00 is not guaranteed if these functions are switched during transmission or reception. Be sure to disable the one that is not used.	p. 448 ☐
			UARTA1 and I^2C02 mode switching	The transmit/receive operation of UARTA1 and I^2C02 is not guaranteed if these functions are switched during transmission or reception. Be sure to disable the one that is not used.	p. 449 ☐
			UAnOPT0 register	Do not set the UAnSRT and UAnSTT bits (to 1) during SBF reception (UAnSRF bit = 1).	p. 455 ☐
			SBF reception	If SBF is transmitted during a data reception, a framing error occurs.	p. 465 ☐
				Do not set the SBF reception trigger bit (UAnSRT) and SBF transmission trigger bit (UAnSTT) to 1 during an SBF reception (UAnSRF = 1).	p. 465 ☐
			Continuous transmission	When initializing transmissions during the execution of continuous transmissions, make sure that the UAnSTR.UAnTSF bit is 0, then perform the initialization. Transmit data that is initialized when the UAnTSF bit is 1 cannot be guaranteed.	p. 466 ☐
			UART reception	Be sure to read the UAnRX register even when a reception error occurs. If the UAnRX register is not read, an overrun error occurs during reception of the next data, and reception errors continue occurring indefinitely.	p. 470 ☐