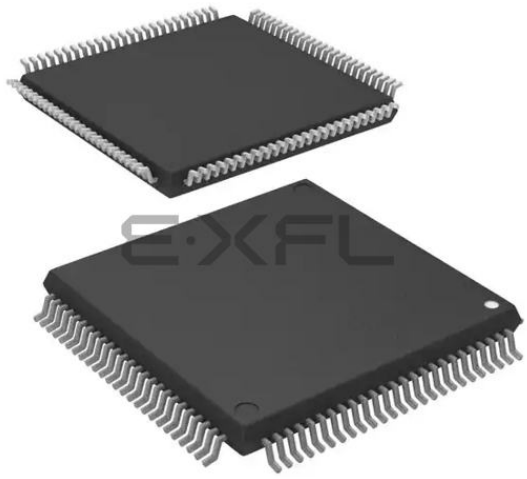


Welcome to E-XFL.COM

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	RL78
Core Size	16-Bit
Speed	24MHz
Connectivity	CSI, I²C, IrDA, LINbus, UART/USART
Peripherals	LCD, LVD, POR, PWM, WDT
Number of I/O	68
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	1.7V ~ 5.5V
Data Converters	A/D 6x10b, 4x24b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LFQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/r5f10npgdfb-50

5.3 Initial Setting of Flash Operation Modes

The option byte (000C2H) is used to set the initial state of flash operation mode and the high-speed on-chip oscillator after a reset is released.

Set an appropriate flash operation mode according to the V_{DD} voltage and the high-speed on-chip oscillator frequency at a reset release.

<R> When a reset is released, the value of CMODE1 and CMODE0 is updated in MODE1 and MODE0 in the flash operation mode select register (FLMODE) and the value of FRQSEL2 to FRQSEL0 is updated in the high-speed on-chip oscillator frequency select register (HOCODIV).

Figure 5 - 5 Format of User option byte (000C2H)

Address: 000C2H

Symbol	7	6	5	4	3	2	1	0
	CMODE1	CMODE0	1	0	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE1	CMODE0	Selection of flash operation mode after reset release
0	0	LV (low-voltage main) mode
1	0	LS (low-speed main) mode
1	1	HS (high-speed main) mode
Other than above		Setting prohibited

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	High-speed on-chip oscillator frequency
0	0	0	0	24 MHz
1	0	0	1	16 MHz
0	0	0	1	12 MHz
1	0	1	0	8 MHz
0	0	1	0	6 MHz
1	0	1	1	4 MHz
0	0	1	1	3 MHz
1	1	0	0	2 MHz
0	1	0	0	1.5 MHz
1	1	0	1	1 MHz
Other than above				Setting prohibited

Figure 8-2. Internal Block Diagram of Channel 0 of Timer Array Unit

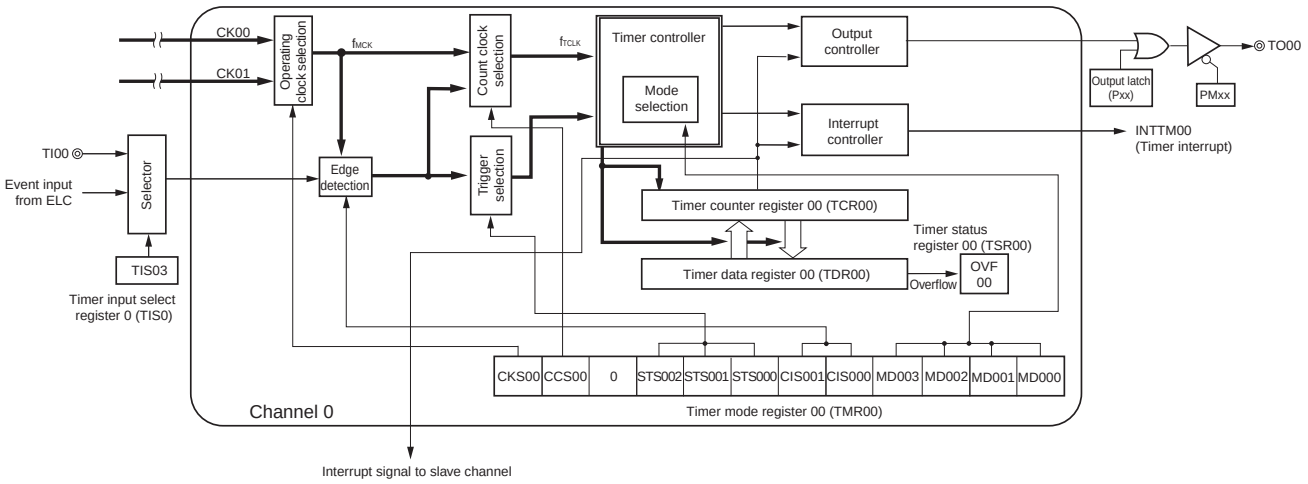
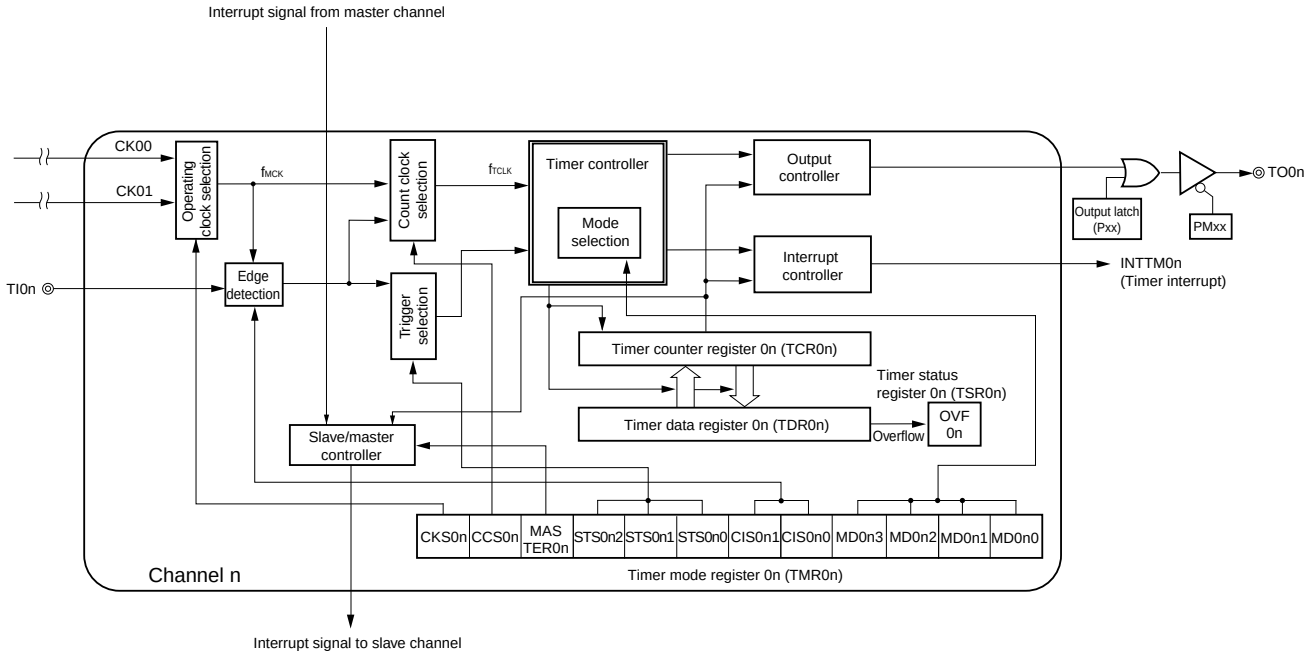


Figure 8-3. Internal Block Diagram of Channels 2, 4 of Timer Array Unit



Remark n = 2, 4

9.3.5 30-second adjustment procedure

Figure 9 - 58 shows how to execute 30-second adjustment.

Figure 9 - 58 30-Second Adjustment Procedure

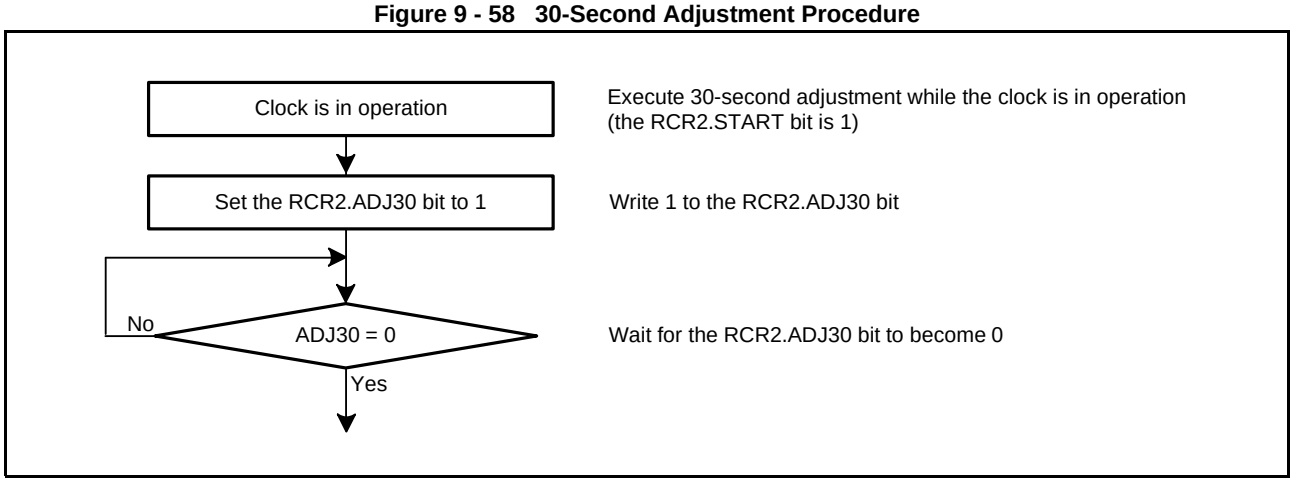


Figure 17-10. INTDSADZCn Interrupt Generation Timing
(Pulse output: DSADZCMDn = 0, DSADZCEGNn = 0, DSADZCEGPn = 1)

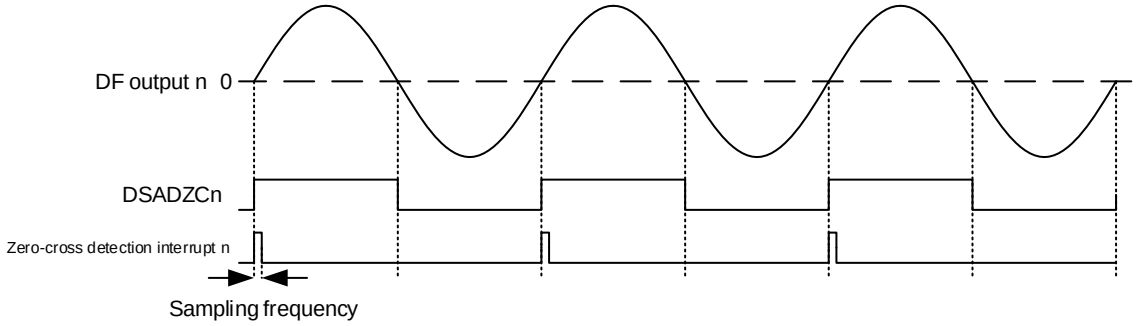
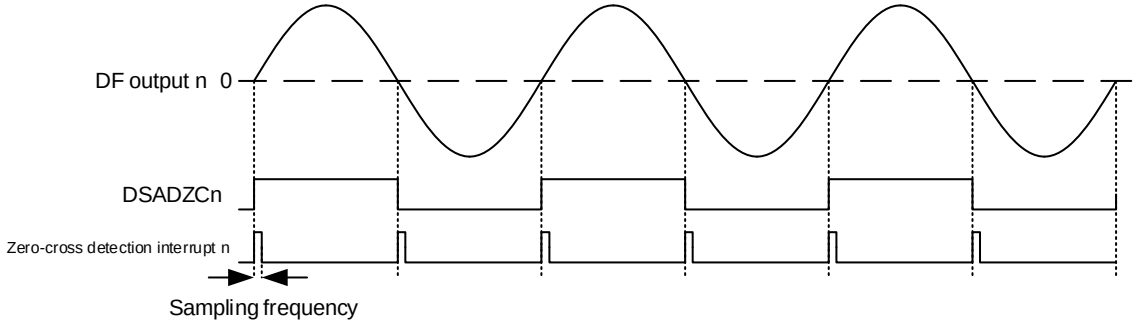


Figure 17-11. INTDSADZCn Interrupt Generation Timing
(Pulse output: DSADZCMDn = 0, DSADZCEGNn = 1, DSADZCEGPn = 1)



DSADZCMDn bit (n = 0, 1)

This bit is used to select the output type of zero-cross detection interrupt.

Interrupts may be generated for several times in the proximity of a zero cross when the DF output includes harmonic signals in the case of setting “0” (pulse output mode). Set “1” (level output mode) to prevent the interrupt from generating for several times.

It is necessary to clear the assertion by software when the zero-cross detection interrupt is asserted once in the case of setting a level output mode.

For more information on the operation, see **Figures 17-23** and **17-25**, in **17.3.3.1 Zero-cross detection interrupt operation**.

DSADZCCTLn bit (n = 0, 1)

This bit is used to select a target channel for detecting a zero-cross.

17.4 Notes on Using 24-Bit ΔΣ A/D Converter

17.4.1 External pins

The V_{DD} and VBAT pins are the analog power supply pin of the ΔΣ A/D converter.

The AV_{SS} pin is the ground power supply pin of the ΔΣ A/D converter. Always keep the voltage on this pin the same as that on the V_{SS} pin even when the ΔΣ A/D converter is not used.

17.4.2 SFR access

(1) Read the DSADCRn register by ΔΣ A/D conversion end interrupt (INTDSAD) servicing. If the DSADCRn register is read before a ΔΣ A/D conversion end interrupt is generated, an illegal value may be read because of a conflict between storing the conversion value in the DSADCRn register and reading the register.
The period of the INTDSAD processing during which the DSADCRn register is read is 192 μs (when DSADFR is set to 0) or 384 μs (when DSADFR is set to 1), so complete reading of the register within this time.
Reading the DSADCRnL, DSADCRnM, and DSADCRnH registers are performed in the same conditions as those described above.

(2) After powering on the ΔΣ A/D converter (DSADPONn in the DSADMR register = 1), internal setup time is necessary. Consequently, the data of the first 80 conversions is invalid.

(3) Setup time is also necessary when the ΔΣ A/D converter has been temporarily stopped for initialization (by clearing the DSADCEn bit in the DSADMR register to 0 with DSADPONn = 1) and then restarted. In this case, since stabilization time is necessary for the converter, wait for one INTDSAD to be generated as the setup time. To initialize the ΔΣ A/D converter, make sure that DSADCEn remains 0 for at least 1.4 μs.

(4) The time required for the correct data to be output after the conversion operation has been enabled (by setting the DSADCEn bit to 1) differs depending on the analog input status at that time. This is because the stabilization time of the high-pass filter changes depending on the analog input status.

(5) Set the sampling frequency (DSADFR bit in the DSADMR register) while the DSADPONn bit in the DSADMR register is 0.
Be sure to set the DSADGCR1 and DSADGCR0 registers, DSADCOF[1:0] bits in the DSADHPFCR register, DSADZCCTL1 and DSADZCCTL0 bits in the DSADICR register, and DSADPHCRn register while the ΔΣ A/D converter is stopped (DSADCEn = 0).

(6) Since the DSADCRn register is initialized when the DSADCEn bit is 0, read the DSADCRn register when the DSADCEn bit is 1.

(7) Clear the DSADPONn bit in the DSADMR register to 0 before shifting to software STOP mode. If software STOP mode is entered with the DSADPONn bit set to 1, a current will flow.

(8) Latency is required to allow 3 times of ΔΣ A/D conversion completion interrupt generation in order to reflect new setting to the internal logic when the DSADICR register is rewrote. Rewriting the DSADICR register is prohibited during the latency for reflection.
It is required to wait for a total of 3 times of ΔΣ A/D conversion completion interrupt generation before the conversion stop and after the restart of the A/D conversion because the reflection into the internal logic from the DSADICR register stops if the A/D conversion in all channels stops (DSADCEn = 0).

(9) Latency is required to allow 3 times of ΔΣ A/D conversion completion interrupt generation until the zero-cross detection interrupt is cleared when 1 is written to the DSADICL0 or DSADICL1 bit in the DSADICLR register. Writing 1 to the same bit described above during the latency of clear is invalid.
A total of 3 times of ΔΣ A/D conversion completion interrupts are generated before the conversion stop and after the restart of the A/D conversion because the reflection into the internal logic from the DSADICL0 or DSADICL1 bit stops if the A/D conversion in all channels stops (DSADCEn = 0).

18.3.2 Serial clock select register m (SPSm)

The SPSm register is a 16-bit register that is used to select two types of operation clocks (CKm0, CKm1) that are commonly supplied to each channel. CKm1 is selected by bits 7 to 4 of the SPSm register , and CKm0 is selected by bits 3 to 0.

Rewriting the SPSm register is prohibited when the register is in operation (when SEmn = 1).

The SPSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SPSm register can be set with an 8-bit memory manipulation instruction with SPSmL.

Reset signal generation clears the SPSm register to 0000H.

Figure 18-6. Format of Serial Clock Select Register m (SPSm)

Address: F0126H, F0127H (SPS0), F0166H, F0167H (SPS1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Section of operation clock (CKmk)[Note]					
					f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 12 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	0	0	f_{CLK}	4 MHz	8 MHz	12 MHz	20 MHz	24 MHz
0	0	0	1	$f_{CLK}/2$	2 MHz	4 MHz	6 MHz	10 MHz	12 MHz
0	0	1	0	$f_{CLK}/2^2$	1 MHz	2 MHz	3 MHz	5 MHz	6 MHz
0	0	1	1	$f_{CLK}/2^3$	500 kHz	1 MHz	1.5 MHz	2.5 MHz	3 MHz
0	1	0	0	$f_{CLK}/2^4$	250 kHz	500 kHz	750 kHz	1.25 MHz	1.5 MHz
0	1	0	1	$f_{CLK}/2^5$	125 kHz	250 kHz	375 kHz	625 kHz	750 KHz
0	1	1	0	$f_{CLK}/2^6$	62.5 kHz	125 kHz	187.5 kHz	313 kHz	375 kHz
0	1	1	1	$f_{CLK}/2^7$	31.25 kHz	62.5 kHz	93.8 kHz	156 kHz	187.5 kHz
1	0	0	0	$f_{CLK}/2^8$	15.62 kHz	31.25 kHz	46.9 kHz	78.1 kHz	93.8 kHz
1	0	0	1	$f_{CLK}/2^9$	7.81 kHz	15.62 kHz	23.4 kHz	39.1 kHz	46.9 kHz
1	0	1	0	$f_{CLK}/2^{10}$	3.91 kHz	7.81 kHz	11.7 kHz	19.5 kHz	23.4 kHz
1	0	1	1	$f_{CLK}/2^{11}$	1.95 kHz	3.91 kHz	5.86 kHz	9.77 kHz	11.7 kHz
1	1	0	0	$f_{CLK}/2^{12}$	976 Hz	1.95 kHz	2.93 kHz	4.88 kHz	5.86 kHz
1	1	0	1	$f_{CLK}/2^{13}$	488 Hz	976 Hz	1.46 kHz	2.44 kHz	2.93 kHz
1	1	1	0	$f_{CLK}/2^{14}$	244 Hz	488 Hz	732 Hz	1.22 kHz	1.46 kHz
1	1	1	1	$f_{CLK}/2^{15}$	122 Hz	244 Hz	366 Hz	610 Hz	732 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

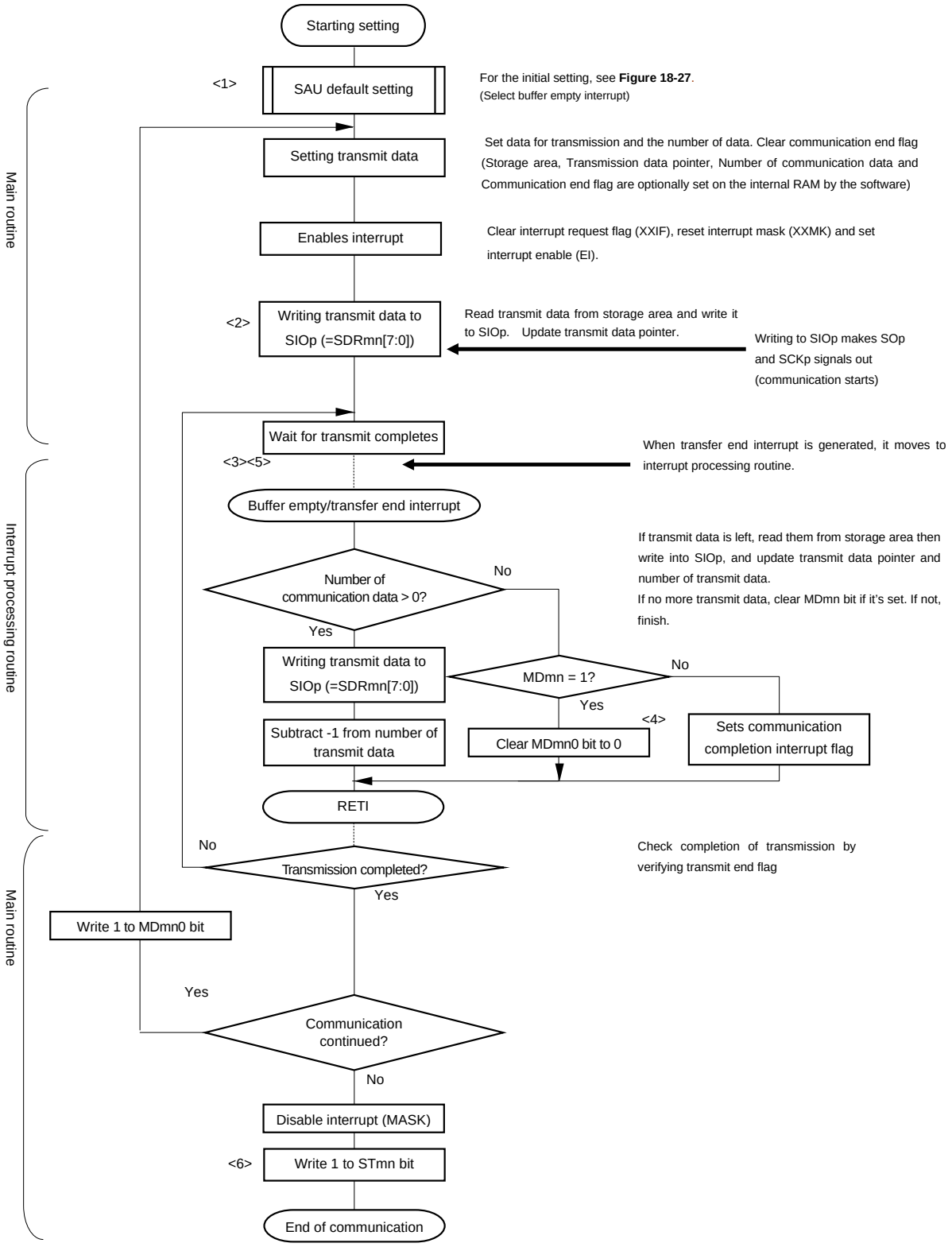
Caution Be sure to clear bits 15 to 8 to "0".

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency

2. m: Unit number (m = 0, 1)

3. k = 0, 1

Figure 18-33. Flowchart of Master Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 18-32 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

Figure 18-39. Flowchart of Master Reception (in Single-reception Mode)

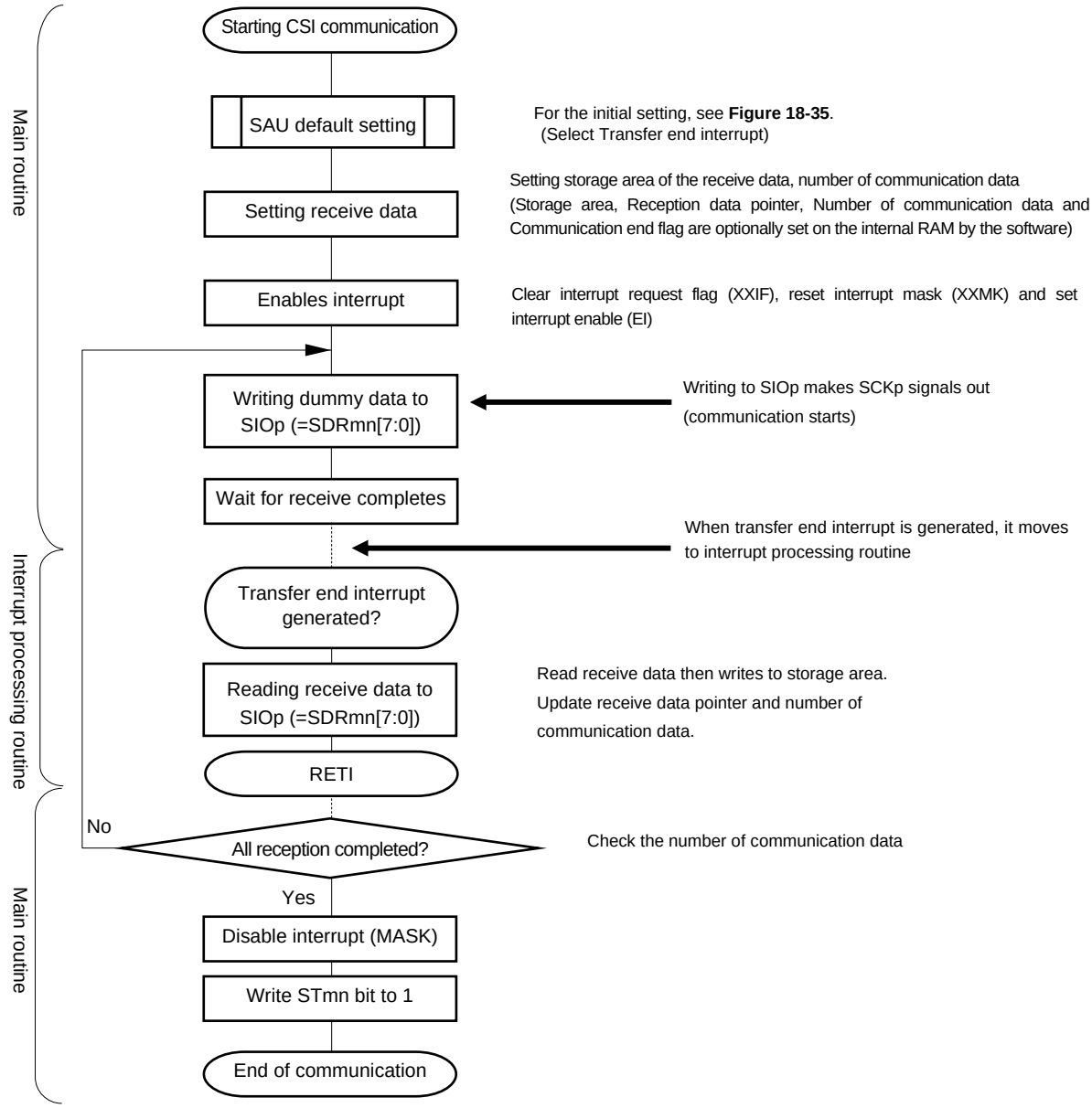


Table 18-3. Baud Rate Setting for UART Reception in SNOOZE Mode

High-speed On-chip Oscillator (f_{IH})	Baud Rate for UART Reception in SNOOZE Mode			
	Baud Rate of 4800 bps			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Maximum Permissible Value	Minimum Permissible Value
24 MHz ± 1.0% Note	$f_{CLK}/2^5$	79	1.60%	-2.18%
16 MHz ± 1.0% Note	$f_{CLK}/2^4$	105	2.27%	-1.53%
12 MHz ± 1.0% Note	$f_{CLK}/2^4$	79	1.60%	-2.19%
8 MHz ± 1.0% Note	$f_{CLK}/2^3$	105	2.27%	-1.53%
6 MHz ± 1.0% Note	$f_{CLK}/2^3$	79	1.60%	-2.19%
4 MHz ± 1.0% Note	$f_{CLK}/2^2$	105	2.27%	-1.53%
3 MHz ± 1.0% Note	$f_{CLK}/2^2$	79	1.60%	-2.19%
2 MHz ± 1.0% Note	$f_{CLK}/2$	105	2.27%	-1.54%
1 MHz ± 1.0% Note	f_{CLK}	105	2.27%	-1.57%

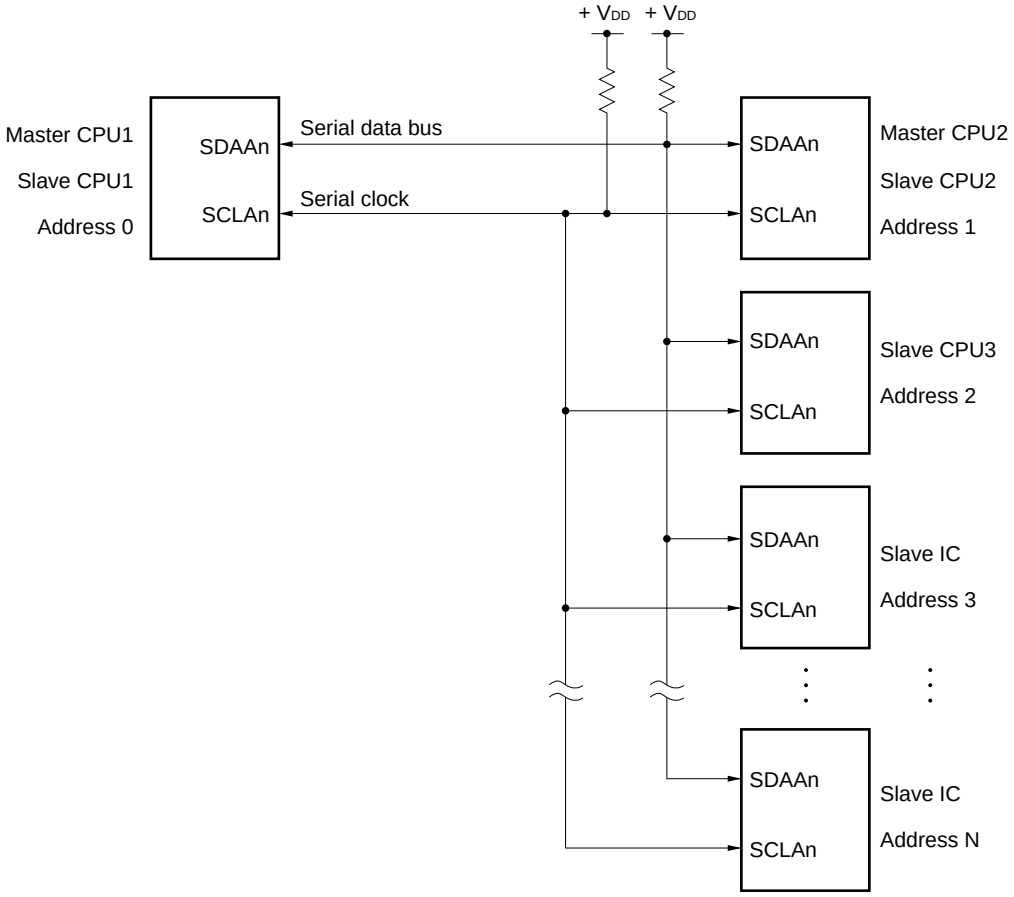
Note When the accuracy of the clock frequency of the high-speed on-chip oscillator is ±1.5%, the permissible range becomes smaller as shown below.

- In the case of f_{IH} ± 1.5%, perform (Maximum permissible value − 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.

Remark The maximum permissible value and minimum permissible value are permissible values for the baud rate in UART reception. The baud rate on the transmitting side should be set to fall inside this range.

Figure 19-2 shows a serial bus configuration example.

Figure 19-2. Serial Bus Configuration Example Using I²C Bus



Remark n = 0

Table 21-2. Maximum Number of Pixels (2/3)

(b) 80-pin products

Drive Waveform for LCD Driver	LCD Driver Voltage Generator	Bias Mode	Number of Time Slices	Maximum Number of Pixels
Waveform A	External resistance division	–	Static	34 (34 segment signals, 1 common signal)
		1/2	2	68 (34 segment signals, 2 common signals)
			3	102 (34 segment signals, 3 common signals)
		1/3	3	
			4	136 (34 segment signals, 4 common signals)
			6	192 (32 segment signals, 6 common signals)
			8	240 (30 segment signals, 8 common signals)
		1/4	8	
	Internal voltage boosting	1/3	3	102 (34 segment signals, 3 common signals)
			4	136 (34 segment signals, 4 common signals)
			6	192 (32 segment signals, 6 common signals)
			8	240 (30 segment signals, 8 common signals)
		1/4	6	192 (32 segment signals, 6 common signals)
			8	240 (30 segment signals, 8 common signals)
	Capacitor split	1/3	3	102 (34 segment signals, 3 common signals)
			4	136 (34 segment signals, 4 common signals)
			6	192 (32 segment signals, 6 common signals)
			8	240 (30 segment signals, 8 common signals)
Waveform B	External resistance division, internal voltage boosting	1/3	3	102 (34 segment signals, 3 common signals)
			4	136 (34 segment signals, 4 common signals)
			6	192 (32 segment signals, 6 common signals)
			8	240 (30 segment signals, 8 common signals)
		1/4	8	
	Capacitor split	1/3	3	102 (34 segment signals, 3 common signals)
			4	136 (34 segment signals, 4 common signals)
			6	192 (32 segment signals, 6 common signals)
			8	240 (30 segment signals, 8 common signals)

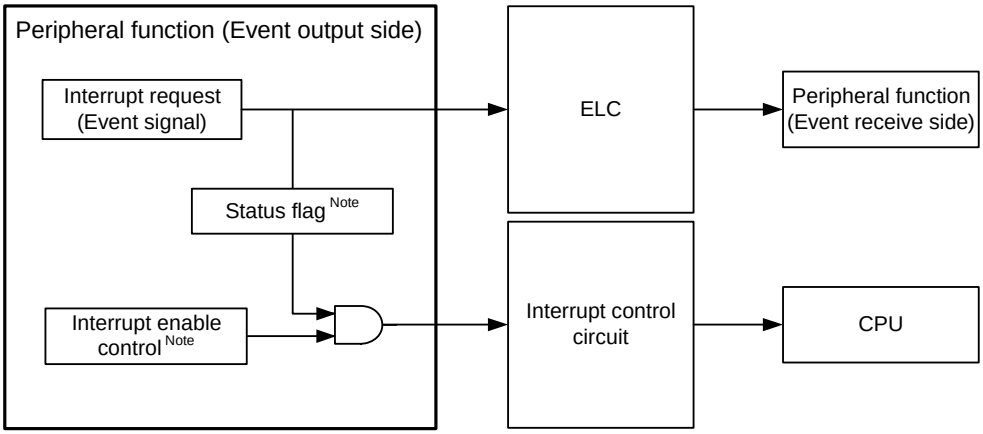
23.4 ELC Operation

The path for using an event signal generated by a peripheral function as an interrupt request to the interrupt control circuit is independent from the path for using it as an ELC event. Therefore, each event signal can be used as an event signal for operation of an event-receiving peripheral function, regardless of interrupt control.

Figure 23 - 5 shows the Relationship Between Interrupt Handling and ELC. The figure show an example of an interrupt request status flag and a peripheral function possessing the enable bits that control enabling/disabling of such interrupts.

A peripheral function which receives an event from the ELC will perform the operation corresponding to the event-receiving peripheral function after reception of an event (See **Table 23 - 3 Correspondence Between Values Set to ELSELRn (n = 00 to 21) Registers and Operation of Link Destination Peripheral Functions at Reception**).

Figure 23 - 5 Relationship Between Interrupt Handling and ELC



Note Not available depending on the peripheral function.

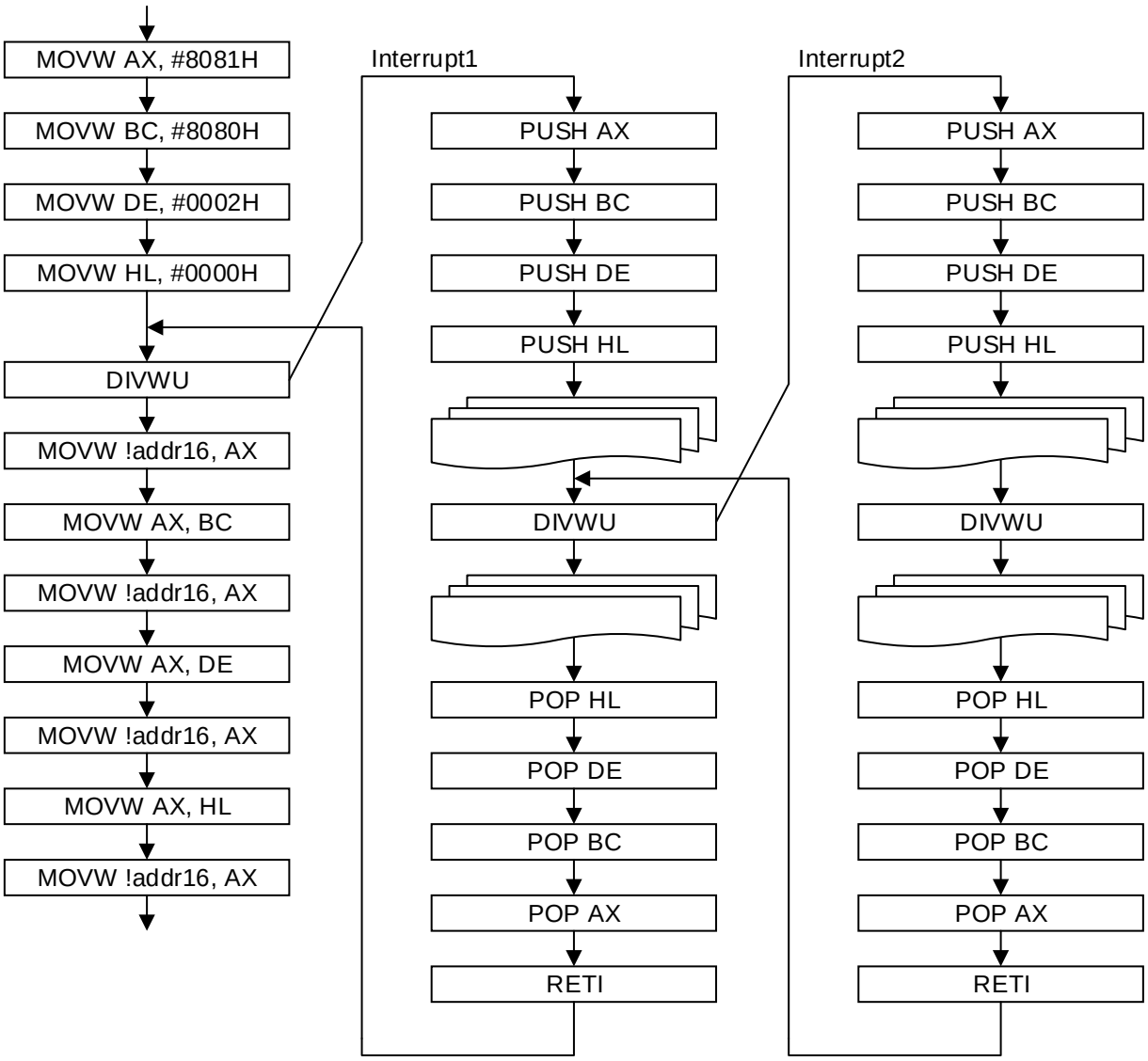
<R> 24.4.4 Interrupt servicing during division instruction

The RL78/I1C handles interrupts during the DIVHU/DIVWU instruction in order to enhance the interrupt response when a division instruction is executed.

- When an interrupt is generated while the DIVHU/DIVWU instruction is executed, the instruction is suspended
- After the instruction is suspended, the PC indicates the next instruction after DIVHU/DIVWU
- An interrupt is generated by the next instruction
- PC-3 is stacked to execute the DIVHU/DIVWU instruction again

Normal interrupt	Interrupts while Executing DIVHU/DIVWU Instruction
(SP-1) ← PSW	(SP-1) ← PSW
(SP-2) ← (PC)S	(SP-2) ← (PC-3)S
(SP-3) ← (PC)H	(SP-3) ← (PC-3)H
(SP-4) ← (PC)L	(SP-4) ← (PC-3)L
PCS ← 0000	PCS ← 0000
PCH ← (Vector)	PCH ← (Vector)
PCL ← (Vector)	PCL ← (Vector)
SP ← SP-4	SP ← SP-4
IE ← 0	IE ← 0

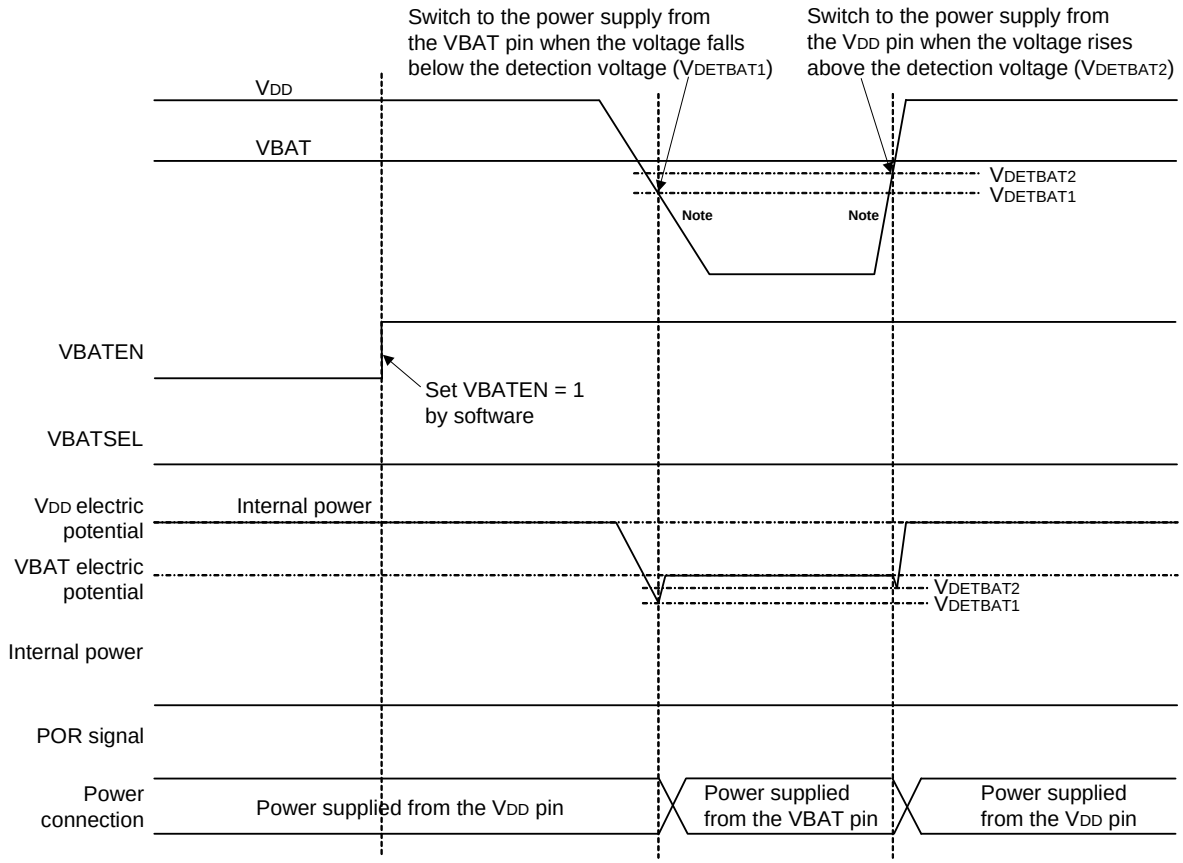
The AX, BC, DE, and HL registers are used for DIVHU/DIVWU. Use these registers by stacking them for interrupt servicing.



Caution Disable interrupts when executing the DIVHU or DIVWU instruction in an interrupt servicing routine. Alternatively, unless they are executed in the RAM area, note that execution of a DIVHU or DIVWU instruction is possible even with interrupts enabled as long as a NOP instruction is added immediately after the DIVHU or DIVWU instruction in the assembly language source code. The following compilers automatically add a NOP instruction immediately after any DIVHU or DIVWU instruction output during the build process.

- **V. 1.71 and later versions of the CA78K0R (Renesas Electronics compiler), for both C and assembly language source code**
- **Service pack 1.40.6 and later versions of the EWRL78 (IAR compiler), for C language source code**
- **GNURL78 (KPIT compiler), for C language source code**

Figure 30-5. Battery Backup Operation (1) with VBATEN = 1 and VBATSEL = 0



Note For details about the power rising and falling slopes, see **CHAPTER 41 ELECTRICAL SPECIFICATIONS**.

<3> Stopping (only independent power supply RTC operates) in battery backup mode

Figure 30-14 shows the operation when only the independent power supply RTC operates and the CPU maintains the STOP state in battery backup mode.

Power failure

Perform power failure detection by using a voltage detection interrupt (INTLVDVDD) of the V_{DD} pin, and transition to STOP mode before power supply switching by the battery backup circuit occurs.

In battery backup mode (VBAT pin supply)

Put the system in the STOP state. Clearing of STOP due to any cause other than a voltage detection interrupt (INTLVDVDD) of the V_{DD} pin is prohibited.

Power recovery

Perform power recovery detection by using a voltage detection interrupt (INTLVDVDD) of the V_{DD} pin, and clear STOP.

Caution During power recovery, make sure to keep the power supply rising slope at maximum of 0.06 V/ms.

Figure 30-14. Operation When the CPU can Maintain the STOP State in Battery Backup Mode

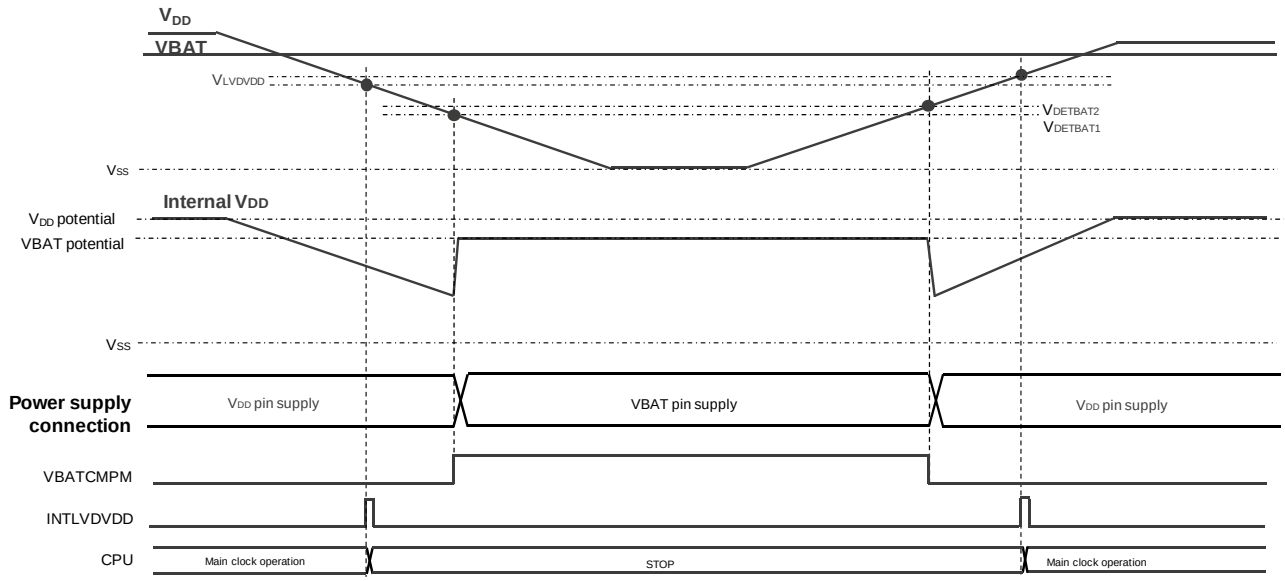
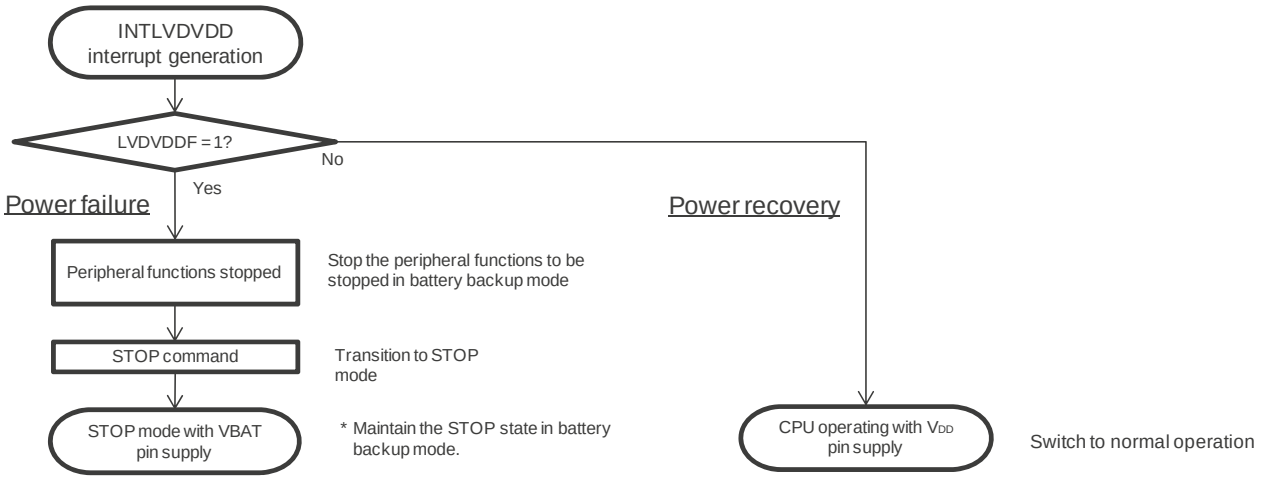


Figure 30-15. Setting Procedure When the CPU can Maintain the STOP State in Battery Backup Mode



32.3.4 RAM guard function

In order to guarantee safety during operation, the IEC61508 standard mandates that important data stored in the RAM be protected, even if the CPU freezes.

This RAM guard function is used to protect data in the specified memory space.

If the RAM guard function is specified, writing to the specified RAM space is disabled, but reading from the space can be carried out as usual.

32.3.4.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GRAM1 and GRAM0 bits are used in RAM guard function.

The IAWCTL register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 32-9. Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	RAM guard space[Note]
0	0	Disabled. RAM can be written to.
0	1	The 128 bytes of space starting at the start address in the RAM
1	0	The 256 bytes of space starting at the start address in the RAM
1	1	The 512 bytes of space starting at the start address in the RAM

Note The RAM start address differs depending on the size of the RAM provided with the product.

39.2.1 Multiplication data register A (MUL32UH, MUL32UL, MUL32SH, MUL32SL, MAC32UH, MAC32UL, MAC32SH, MAC32SL)

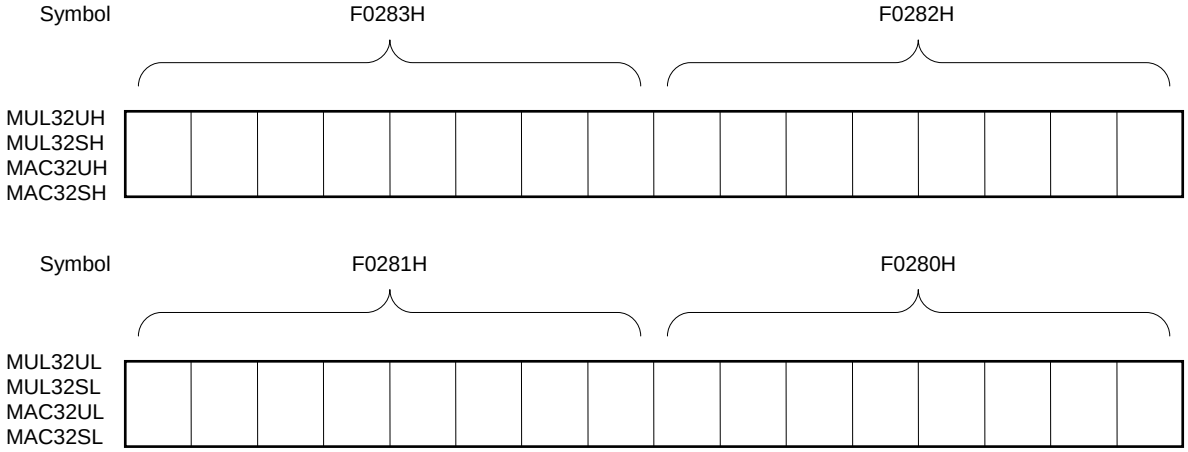
Multiplication data register A specifies the multiplicand used for multiplication and multiply-accumulation.

Multiplication data register A can be set by a 16-bit manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 39-2. Format of Multiplication Data Register A
(MUL32UH, MUL32UL, MUL32SH, MUL32SL, MAC32UH, MAC32UL, MAC32SH, MAC32SL)

Address: F0282H (MUL32UH), F0286H (MUL32SH), F028AH (MAC32UH), F028EH (MAC32SH)
F0280H (MUL32UL), F0284H (MUL32SL), F0288H (MAC32UL), F028CH (MAC32SL)
After reset: 0000H R/W



Caution Do not rewrite the values of the multiplication data register A (MUL32UH, MUL32UL, MUL32SH, MUL32SL, MAC32UH, MAC32UL, MAC32SH, MAC32SL) during the operation processing (when the bit 0 (MULST) of the multiplication control register (MULC) = 1). If this is done, the operation result will be an undefined value.
Multiplication data register A stores the written value. The MACMODE and MULSM bits of the MULC register are also rewritten with the values of the supported operation mode.

For the multiplication data register A, the operation mode can be switched by the register that specifies the multiplicand since the different register name and register address are set for each operation mode. The MACMODE and MULSM bits of the MULC register are also rewritten with the values of the supported operation mode.

For the MUL32UL, MUL32UH, MUL32SL, MUL32SH, MAC32UL, MAC32UH, MAC32SL, and MAC32SH registers, all the register values are rewritten when rewriting one register value since a common register is used for these registers.

The following table shows the relationship between the operation mode and register name.

Table 39-2. Relationship between Operation Mode and Register Name

Operation Mode	Register Name of Multiplication Data Register A	
	Bits 31 to 16 (MULAH)	Bits 15 to 0 (MULAL)
Multiplication mode (unsigned)	MUL32UH	MUL32UL
Multiplication mode (signed)	MUL32SH	MUL32SL
Multiply-accumulation mode (unsigned)	MAC32UH	MAC32UL
Multiply-accumulation mode (signed)	MAC32SH	MAC32SL

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. C is the load capacitance of the SOp output lines.

4. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

5. Either V_{DD} or VBAT is selected by the battery backup function.

Caution Select the normal input buffer for the SIp pin and SCKp pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remarks 1. p: CSI number (p = 00, 10, 30), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM number (g = 0, 1, 8)

2. f_{MCK}: Serial array unit operation clock frequency
(Operating clock that is set with the serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00, 10, 30))