

## **RX261 Group**

### Microcontroller Migration Guide for Migration from H8/3048 (H8/300H Series) to RX261

---

#### **Introduction**

This application note describes precautions to observe when replacing the H8/3048 device with the RX261 device and provides other information such as differences between the H8/3048 and RX261 devices. For details on individual functions, refer to the latest hardware manual.

#### **Target Device**

RX261

## Contents

1. CPU Architecture .....	4
1.1 System Registers .....	4
1.1.1 General-Purpose Registers .....	4
1.1.2 Control Registers .....	5
1.2 Option-Setting Memory .....	8
1.2.1 Outline of Option-Setting Memory .....	8
1.2.2 Endian Settings .....	9
1.3 Reset Function .....	10
1.3.1 Reset Sources .....	10
1.3.2 Reset Sources and Initialization Scope .....	11
1.4 Clock Settings .....	13
1.4.1 Clock Sources .....	13
1.4.2 RX261 Clock Generation Circuit .....	14
1.5 Operating Modes .....	16
1.5.1 Comparison of Operating Modes .....	16
1.5.2 Comparison of the Memory Map Between the H8/3048 and RX261 Devices .....	17
1.5.3 Operating Mode Settings .....	18
1.6 Processor Modes .....	19
1.7 Exception Handling .....	20
1.7.1 Comparison of Exception Sources .....	20
1.7.2 Priority of Exception Sources in Exception Handling .....	20
1.7.3 Basic Processing Sequence of Exception Handling .....	21
1.7.4 Vector Configuration .....	22
1.7.5 Interrupt Masking by CCR (PSW) .....	23
1.8 Interrupt Handling .....	24
1.8.1 Interrupt Controller .....	24
1.8.2 Management of Interrupt Flags .....	27
1.8.3 Control of Fast Interrupts .....	29
1.8.4 Digital Filter .....	30
1.8.5 Multiple Interrupts .....	31
2. On-Chip Functions (Peripheral Modules) .....	32
2.1 List of On-Chip Functions .....	32
2.2 I/O Ports .....	33
2.2.1 Comparison of I/O Port Specifications .....	33
2.2.2 I/O Settings .....	33
2.3 Buses .....	40
2.3.1 Bus Configuration .....	40
2.4 Interrupt Controller .....	43
2.4.1 Usage Example for IRQ Pins .....	43

2.5	DMA Controller .....	44
2.5.1	Comparison of Specifications .....	44
2.5.2	DMAC Block Diagram .....	46
2.5.3	List of Registers .....	48
2.5.4	DMAC Activation Sources .....	49
2.5.5	Transfer Modes .....	50
2.5.6	Module Stop .....	52
2.6	Serial Communications Interface .....	53
2.6.1	Comparison of Specifications .....	53
2.6.2	SCI-related Settings Required During Migration .....	58
2.6.3	Module Stop .....	58
2.7	High-Function Timers (ITU, MTU2a, and TPUa).....	59
2.7.1	Comparison of Specifications .....	59
2.7.2	Notes on Replacing the Timer .....	61
2.7.3	List of Registers .....	61
2.7.4	Module Stop .....	62
2.8	Timer-based Pulse Output Function .....	63
2.8.1	Comparison of Specifications .....	63
2.9	Watchdog Timers .....	66
2.9.1	Comparison of Specifications .....	66
2.10	A/D Converter .....	67
2.10.1	Comparison of Specifications .....	67
2.10.2	Operating Modes .....	68
2.10.3	List of Registers .....	69
2.10.4	Module Stop .....	70
2.11	D/A Converter .....	71
2.11.1	Comparison of Specifications .....	71
2.11.2	Comparison of Registers .....	71
2.11.3	Module Stop .....	71
2.12	Flash Memory .....	72
2.12.1	Comparison of Specifications .....	72
2.13	Low-Power Modes and Module Stop Function .....	73
2.13.1	Comparison of Low-Power Mode Specifications.....	73
2.13.2	Mode Transitions .....	76
2.13.3	Module Stop .....	77
3.	Reference Documents .....	78
	Revision History .....	79

## 1. CPU Architecture

### 1.1 System Registers

The following sections show the differences between the system registers of the H8/3048 and the RX261 devices.

#### 1.1.1 General-Purpose Registers

The H8/3048 and RX261 devices have 32-bit general-purpose registers. With the H8/3048 device, eight\*1 32-bit general-purpose registers are available. With the RX261 device, sixteen 32-bit general-purpose registers are available. The register used as the stack pointer (SP) on the H8/3048 device is ER7, and that on the RX261 device is R0.

Note: 1. On the H8/3048 device, eight 32-bit ER general-purpose registers can be divided into two (E and R) groups of eight 16-bit registers, and the eight 16-bit registers of group R can further be divided into two groups of eight 8-bit registers. On the RX261 device, general-purpose registers cannot be divided.

Figure 1.1 shows the differences between the general-purpose registers of the H8/3048 and RX261 devices.

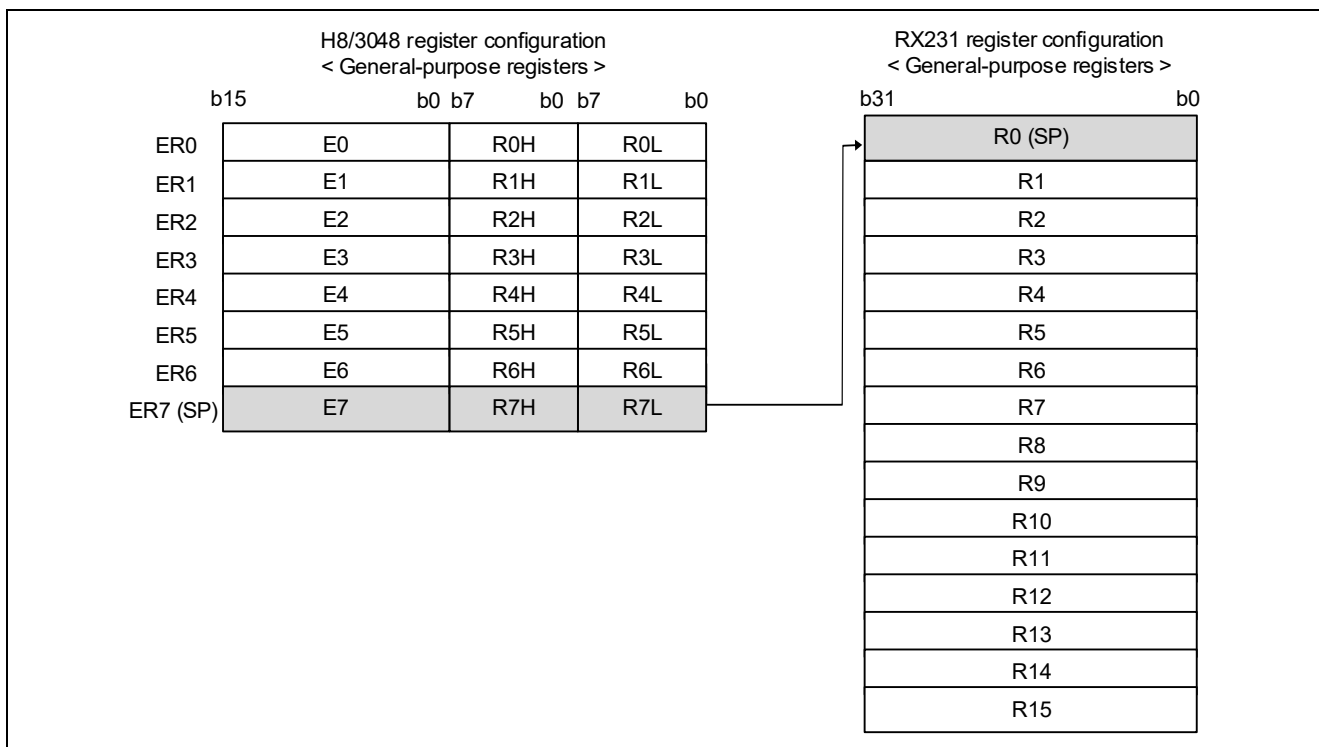
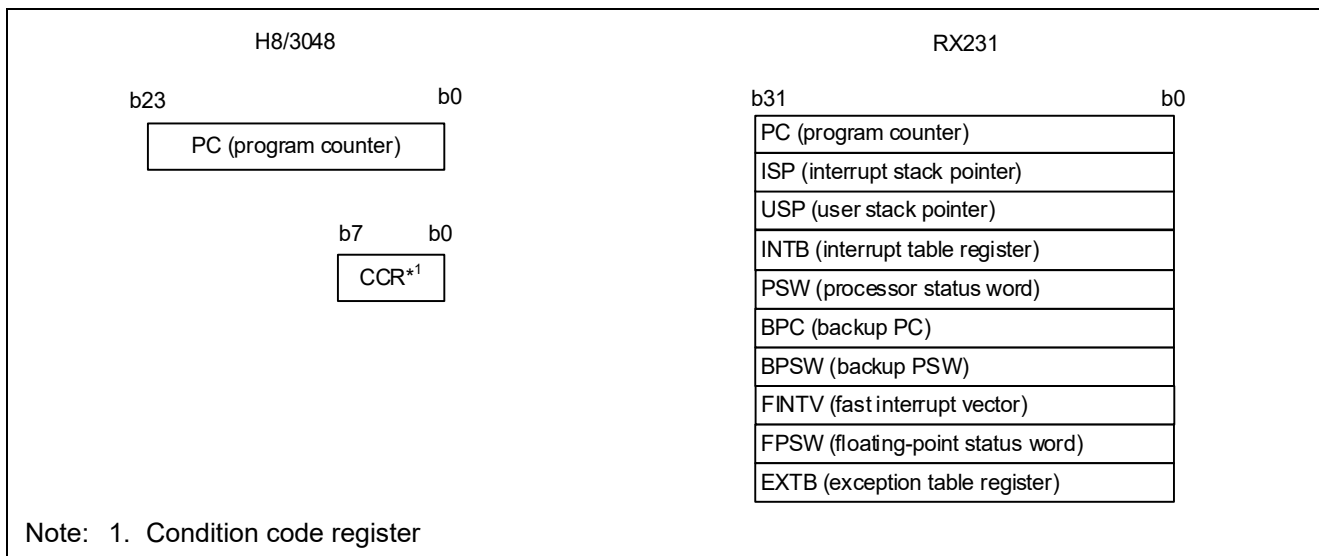


Figure 1.1 Differences of General-Purpose Registers

### 1.1.2 Control Registers

Figure 1.2 shows the differences between the control registers of the H8/3048 and RX261 devices.

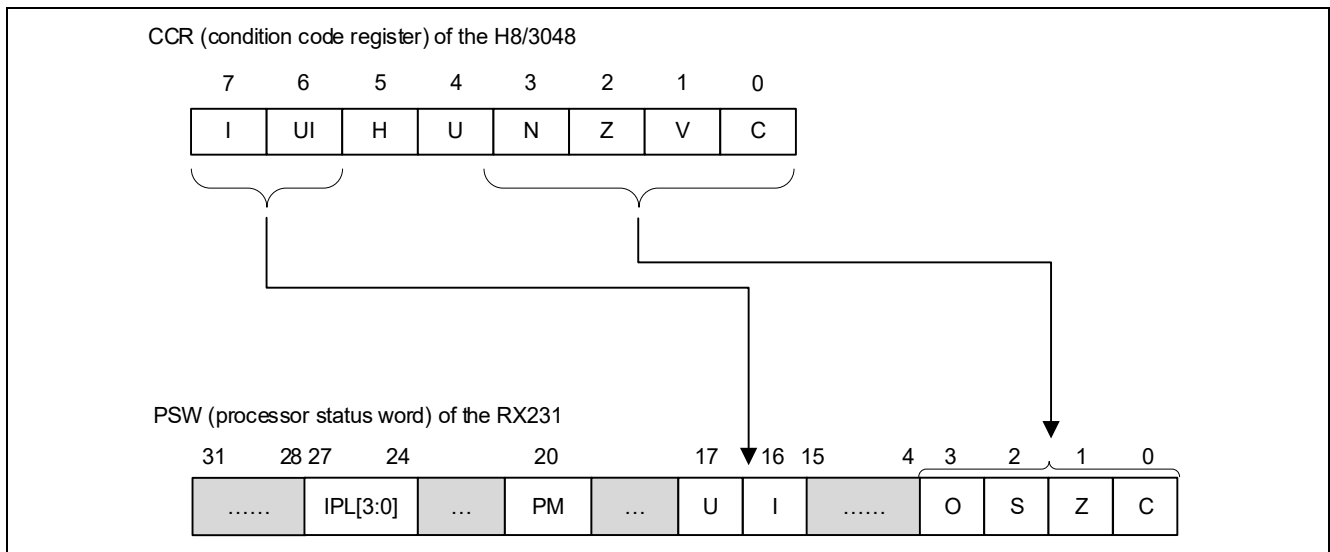


**Figure 1.2 Differences of Control Registers**

The following outlines the RX261-specific control registers that are not present on the H8/3048 device.

- Interrupt stack pointer (ISP) / user stack pointer (USP)  
There are two types of stack pointers (SP): the interrupt stack pointer (ISP) and user stack pointer (USP). The stack pointer (ISP or USP) to be used can be selected by using the stack pointer specification bit (U) of the processor status word (PSW).
- Interrupt table register (INTB)  
This register is used to specify the start address of the interrupt vector table.
- Backup PC (BPC) / backup PSW (BPSW)  
Ordinary interrupts and fast interrupts are supported. For fast interrupts, the data of the program counter (PC) and processor status word (PSW) is saved to the dedicated registers (BPC and BPSW), thereby reducing the processing time needed to save the register data. Note that the BPC and BPSW registers do not support multiple interrupts.
- Fast interrupt vector register (FINTV)  
This register is used to specify the jump destination when a fast interrupt occurs.
- Floating-point status word (FPSW)  
There are floating-point status words for indicating floating-point arithmetic results.
- Exception table register (EXTB)  
This register is used to set the first address of the exception vector table.
- Differences of the condition code register  
On the RX261 device, the flags equivalent to the half-carry flag (H) and user bit (U) of the CCR control register on the H8/3048 device are unavailable. However, for the other bits of the CCR register, the corresponding bits are provided by the PSW register.

Figure 1.3 and Table 1.1 show the differences between the CCR register on the H8/3048 device and the PSW register on the RX261 device.



**Figure 1.3 Differences Between the CCR Register on the H8/3048 and the PSW Register on the RX261**

**Table 1.1 Differences Between the CCR Register on the H8/3048 and the PSW Register on the RX261**

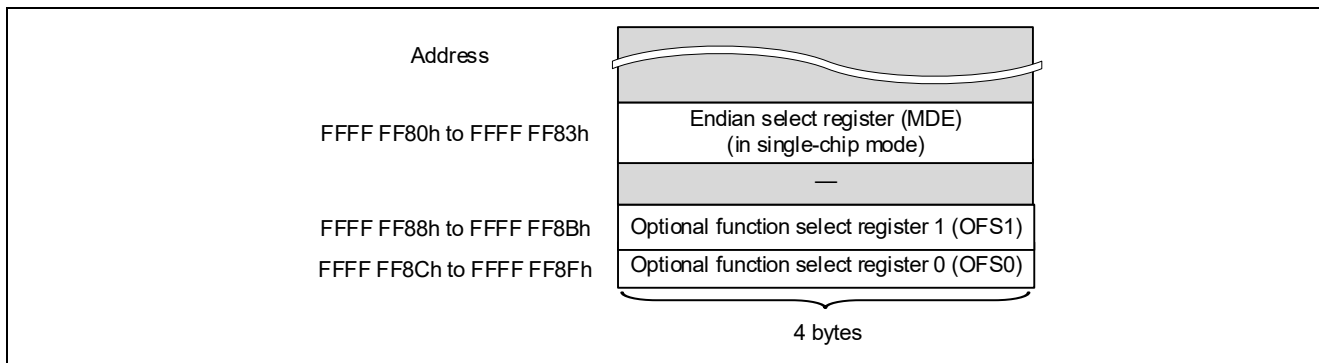
CCR Bit Name	PSW Bit Name	Description
C (carry flag)	C	The calculation result indicated by the C bit in CCR on the H8/3048 is indicated by the C bit in PSW on the RX261. This flag indicates that a carry, borrow, or shift-out occurred in the calculation result.
Z (zero flag)	Z	The calculation result indicated by the Z bit in CCR on the H8/3048 is indicated by the Z bit in PSW on the RX261. This flag indicates that the calculation result was 0.
N (negative flag)	S	The calculation result indicated by the N bit in CCR on the H8/3048 is indicated by the S bit in PSW on the RX261. This flag indicates that the calculation result was negative.
V (overflow flag)	O	The calculation result indicated by the V bit in CCR on the H8/3048 is indicated by the O bit in PSW on the RX261. This flag indicates that an overflow occurred in the calculation result.
I (interrupt mask bit)	I	The I and UI bits in CCR on the H8/3048 correspond to the I bit in PSW on the RX261. 0: Interrupts are disabled. 1: Interrupts are enabled.  This bit is used to enable reception of interrupt requests on the RX261. The initial state is 0, so it is necessary to set this bit to 1 to accept interrupts. Also, this bit is cleared to 0 when an exception is accepted, and no interrupts are accepted while its value remains 0. Note that the interrupt status flag of the interrupt controller is reset when an interrupt request occurs, regardless of the setting of this bit.
UI (user bit/interrupt mask bit)		
—	U	This bit specifies the stack pointer used by the RX261. 0: Interrupt stack pointer (ISP) 1: User stack pointer (USP) This bit is cleared to 0 when an exception is accepted.
—	PM	This bit specifies the processor mode of the RX261. 0: Supervisor mode 1: User mode This bit is cleared to 0 when an exception is accepted.
—	IPL[3:0]	These bits specify the interrupt priority level. The RX261 supports level settings from 0 (lowest) to 15 (highest). Only interrupts with a priority level higher than this setting are accepted.
U (user bit)	—	There is no corresponding bit on the RX261.
H (half-carry flag)	—	There is no corresponding bit on the RX261.

## 1.2 Option-Setting Memory

On the RX261, it is necessary to set up the option-setting memory. Use the option-setting memory to configure the settings such as the operation that follows a CPU reset and the endian order of data. The option-setting memory is allocated in the ROM, and cannot be overwritten by a software program. When programming the ROM, it is necessary to also set up the option-setting memory.

### 1.2.1 Outline of Option-Setting Memory

Figure 1.4 shows the areas in the option-setting memory.



**Figure 1.4 Areas in the Option-Setting Memory**

An outline of the registers is provided below:

- Endian select register (MDE)  
Register for configuring the CPU endian settings
- Optional function select register 1 (OFS1)  
The OFS1 register is used to configure the following settings:
  - Whether to enable or disable the function that resets voltage monitor 0 after a reset, and the level settings for voltage detector 0
  - The setting for the high-speed on-chip oscillator (HOCO) after a reset
  - Time before startup when power is turned on (normal/short)
- Optional function select register 0 (OFS0)  
The OFS0 register is used to determine how the independent watchdog timer (IWDT) and watchdog timer (WDT) will operate after a reset.

Figure 1.5 shows sample option-setting memory settings.

```

/* Settings for single-chip mode and big-endian */
#pragma section C EXCEPTVECT
void (*const Except_Vectors[])(void) = {
  //;0xfffff80 MDE register
  #ifdef __BIG
    (void (*)(void))0xfffff8, // big
  #else
    (void (*)(void))0xfffffff, // little
  #endif
}
    
```

**Figure 1.5 Example of Endian Settings**

Figure 1.6 shows an example of the settings for OFS0 and OFS1.

```
#pragma section C EXCEPTVECT
void (*const Except_Vectors[])(void) = {
...
//;0xfffff88 OFS1 register
(void (*)(void))0xfffffff, // OFS1
//;0xfffff8c OFS0 register
(void (*)(void))0xfffffff, // OFS0
```

Figure 1.6 Example of the Settings for OFS0 and OFS1

### 1.2.2 Endian Settings

The H8/3048 device supports only big-endian order. On the RX261 device, only little-endian is supported for instructions, and little-endian or big-endian can be selected for data. The endian settings are specified by means of the endian select bits (MDE[2:0]) of the MDE register in the option-setting memory.

If the big-endian order is to be used on the RX261 device after migration from the H8/3048 device, big-endian can be specified in the option settings of the genuine Renesas compiler. This allows migration without the need to consider endianness in the user program.

In actuality, code such as that shown in Figure 1.5 is generated automatically according to the compiler option settings. Figure 1.7 shows an overview of specifying the endian settings by using compiler options.

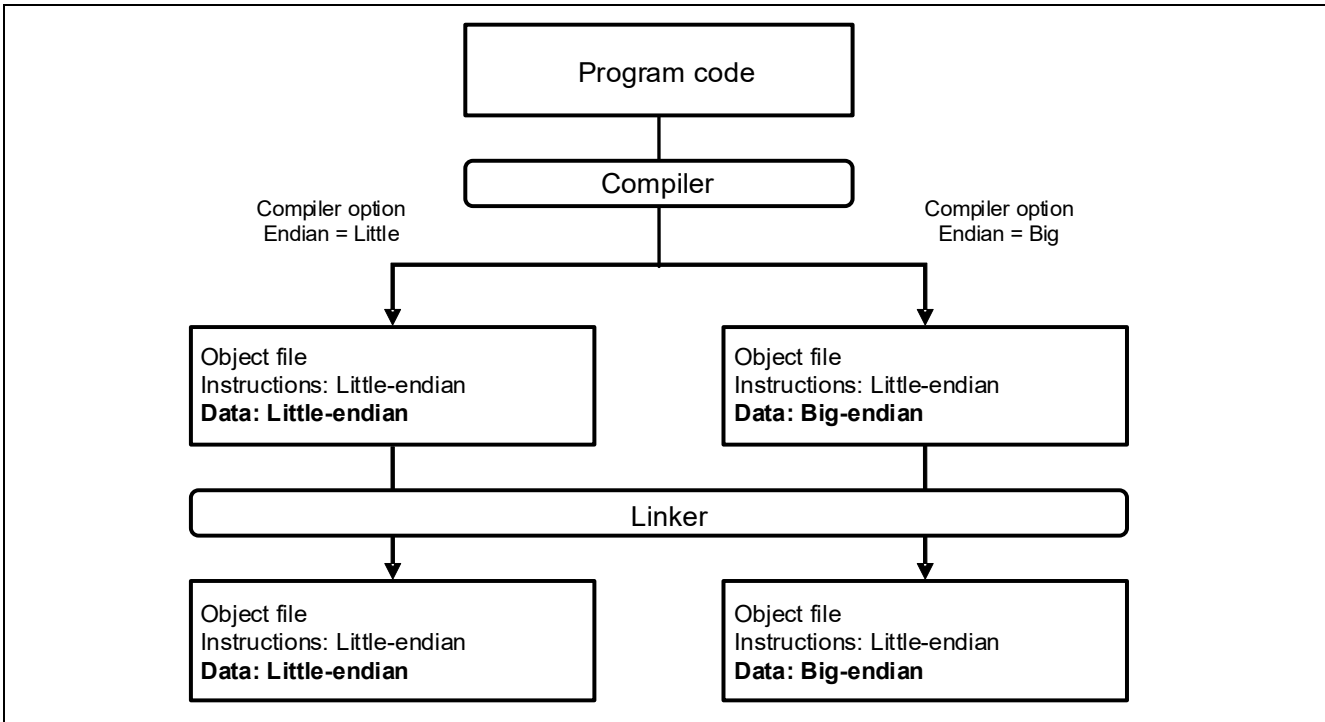


Figure 1.7 Specifying Endian Settings by Using Compiler Options

### 1.3 Reset Function

This section describes the differences in reset types between the H8/3048 and RX261 devices.

#### 1.3.1 Reset Sources

Table 1.2 lists the reset sources of the H8/3048 and RX261 devices.

**Table 1.2 Reset Sources**

H8/3048	RX261
<ul style="list-style-type: none"> <li>Reset</li> </ul>	<ul style="list-style-type: none"> <li>RES# pin reset</li> <li>Power-on reset</li> <li>Voltage monitor 0 reset</li> <li>Voltage monitor 1 reset</li> <li>Voltage monitor 2 reset</li> <li>Independent watchdog timer reset</li> <li>Watchdog timer reset</li> <li>Software reset</li> </ul>

#### (1) Reset vector configuration

The reset vector of the H8/3048 device is assigned to vector number 0 (vector address: H'0000 to H'0003).

The RX261 device has a single reset vector for multiple reset sources. The reset source is identified by using reset status registers 0 to 2 during reset processing, and the processing appropriate for the identified source is performed.

### 1.3.2 Reset Sources and Initialization Scope

On the H8/3048 device, a reset caused by the RES# pin initializes the CPU and all the internal peripheral modules. On the RX261 device, there is a flag related to reset detection for each reset source. After recovery from a reset, the reset source can be identified by checking the flags related to reset detection. Table 1.3 lists the reset types and their initialization scope on the RX261.

**Table 1.3 Reset Initialization Scope for the RX261 Device**

Target to be Initialized	Reset Source							
	RES # Pin Reset	Power-On Reset	Voltage Monitoring 0 Reset	Independent Watchdog Timer Reset	Watchdog Timer Reset	Voltage Monitoring 1 Reset	Voltage Monitoring 2 Reset	Software Reset
The power-on reset detect flag (RSTSR0.PORF)	✓	—	—	—	—	—	—	—
Register related to the cold start/warm start determination flag (RSTSR1.CWSF)	—*1	✓	—	—	—	—	—	—
Voltage monitoring 0 reset detect flag (RSTSR0.LVD0RF)	✓	✓	—	—	—	—	—	—
The independent watchdog timer reset detect flag (RSTSR2.IWDTRF)	✓	✓	✓	—	—	—	—	—
Registers related to the independent watchdog timer (IWDTRR, IWDTCR, IWDTSR, IWDTRCR, IWDTCSTPR, ILOCOCR)	✓	✓	✓	—	—	—	—	—
Watchdog timer reset detect flag (RSTSR2.WDTRF)	✓	✓	✓	✓	—	—	—	—
Registers related to the watchdog timer (WDTRR, WDTCR, WDTSR, WDTRCR)	✓	✓	✓	✓	—	—	—	—
The voltage monitoring 1 reset detect flag (RSTSR0.LVD1RF)	✓	✓	✓	✓	✓	—	—	—
Registers related to voltage monitor function 1 (LVD1CR0, LVCMPCR.LVD1E,	✓	✓	✓	✓	✓	—	—	—

LVDLVL.R.LVD1LVL [3:0])								
(LVD1CR1, LVD1SR)	✓	✓	✓	✓	✓	—	—	—
The voltage monitoring 2 reset detect flag (RSTSR0.LVD2RF)	✓	✓	✓	✓	✓	✓	—	—
Registers related to voltage monitor function 2 (LVD2CR0, LVCMP.R.EXVCCI NP2, LVD2E, LVDLVL.R.LVD2LVL [1:0])	✓	✓	✓	✓	✓	✓	—	—
(LVD2CR1, LVD2SR)	✓	✓	✓	✓	✓	✓	—	—
The software reset detect flag (RSTSR2.SWRF)	✓	✓	✓	✓	✓	✓	✓	—
Register related to the realtime clock*2	—	—	—	—	—	—	—	—
Register related to the sub-clock oscillator (SOSCCR, SOMCR)	—	✓	—	—	—	—	—	—
Operating mode*3	✓	✓	—	—	—	—	—	—
Registers other than the above, CPU, and internal state	✓	✓	✓	✓	✓	✓	✓	✓

✓: Initialization occurs.

—: No change occurs.

Note 1. Initialized at a power-on.

Note 2. Some control bits are initialized by all types of reset or a power-on reset. For details on the target bits, refer to section 26, Realtime Clock (RTCBa).

Note 3. The operating mode is determined by the level of the mode setting pin when the reset is released. For details, refer to section 3, Operating Modes.

## 1.4 Clock Settings

### 1.4.1 Clock Sources

This section describes the types of clock sources. On the H8/3048 device, clocks are not controlled by software. On an RX device, however, each clock must be configured. Therefore, the clock source and clock frequency can be switched. Table 1.4 lists the clock sources of the H8/3048 and RX261 devices.

**Table 1.4 List of Clock Sources of the H8/3048 and RX261 Devices**

H8/3048	RX261
<ul style="list-style-type: none"> <li>• System clock</li> </ul>	<ul style="list-style-type: none"> <li>• Main clock oscillator</li> <li>• Subclock oscillator</li> <li>• PLL/PLL2 circuit</li> <li>• High-speed on-chip oscillator (HOCO)</li> <li>• Low-speed on-chip oscillator (LOCO)</li> <li>• IWDT-dedicated on-chip oscillator</li> </ul>

### 1.4.2 RX261 Clock Generation Circuit

On the RX261 device, the LOCO is selected as the system clock after a reset. During system initialization, oscillate necessary ones of the clock sources shown in Figure 1.8 to determine the system clock. When switching the selection of the clock oscillators and system clocks, take into account the oscillation sequence and the clock oscillation stabilization time.

For details on the setting procedure, refer to the following application note:

- RX261 Group Initial Setting (R01AN7292EJ)

Figure 1.8 shows the clock generation circuit of the RX261 device.

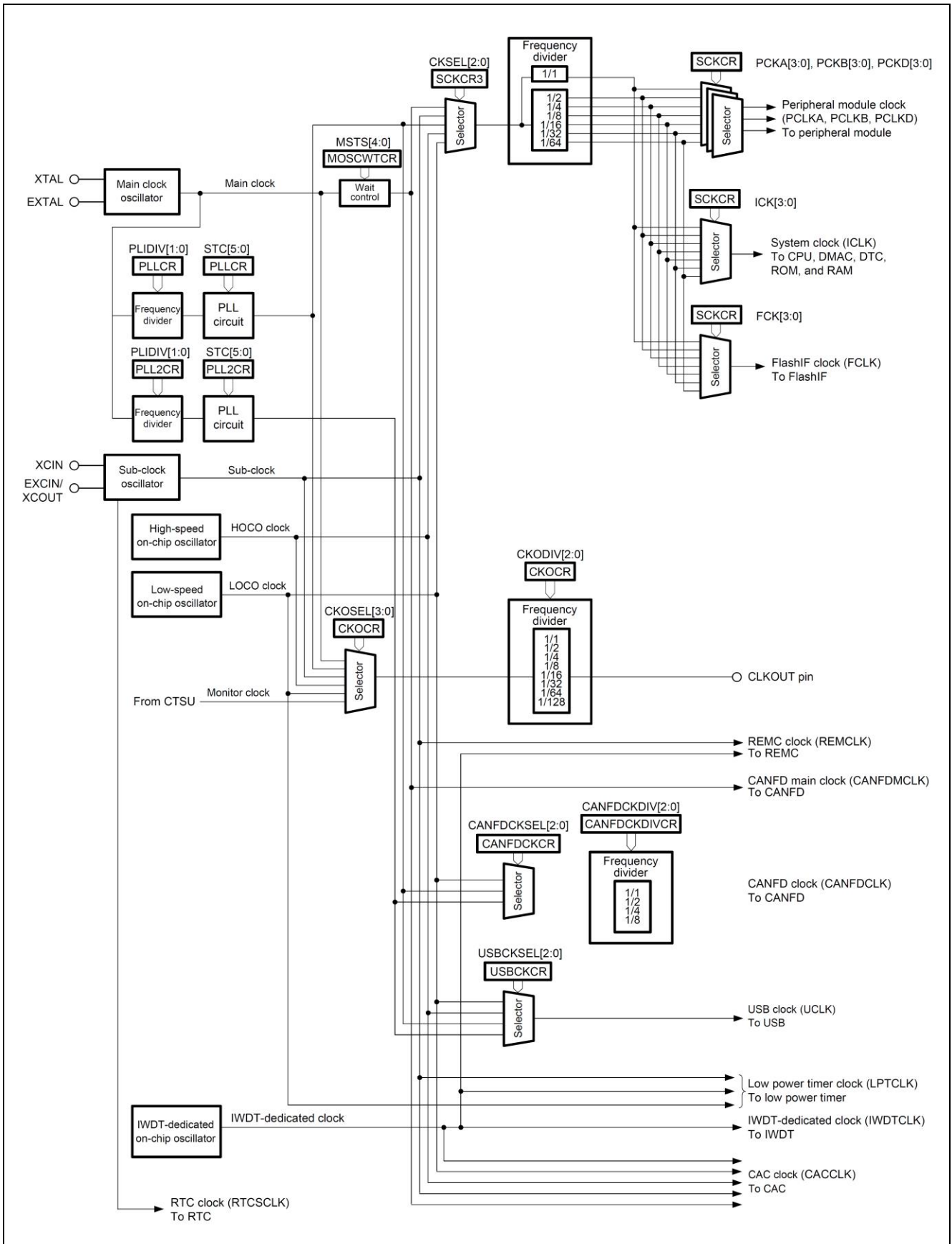


Figure 1.8 Clock Generation Circuit of the RX261 Device

## 1.5 Operating Modes

### 1.5.1 Comparison of Operating Modes

The RX261 uses only the mode pin to set the mode. This is similar to the H8/3048. For details on operating mode settings, refer to section 1.5.3, “Operating Mode Settings”, and the “RX260 Group, RX261 Group User’s Manual: Hardware”. Table 1.5 lists the operating modes of the RX261 device.

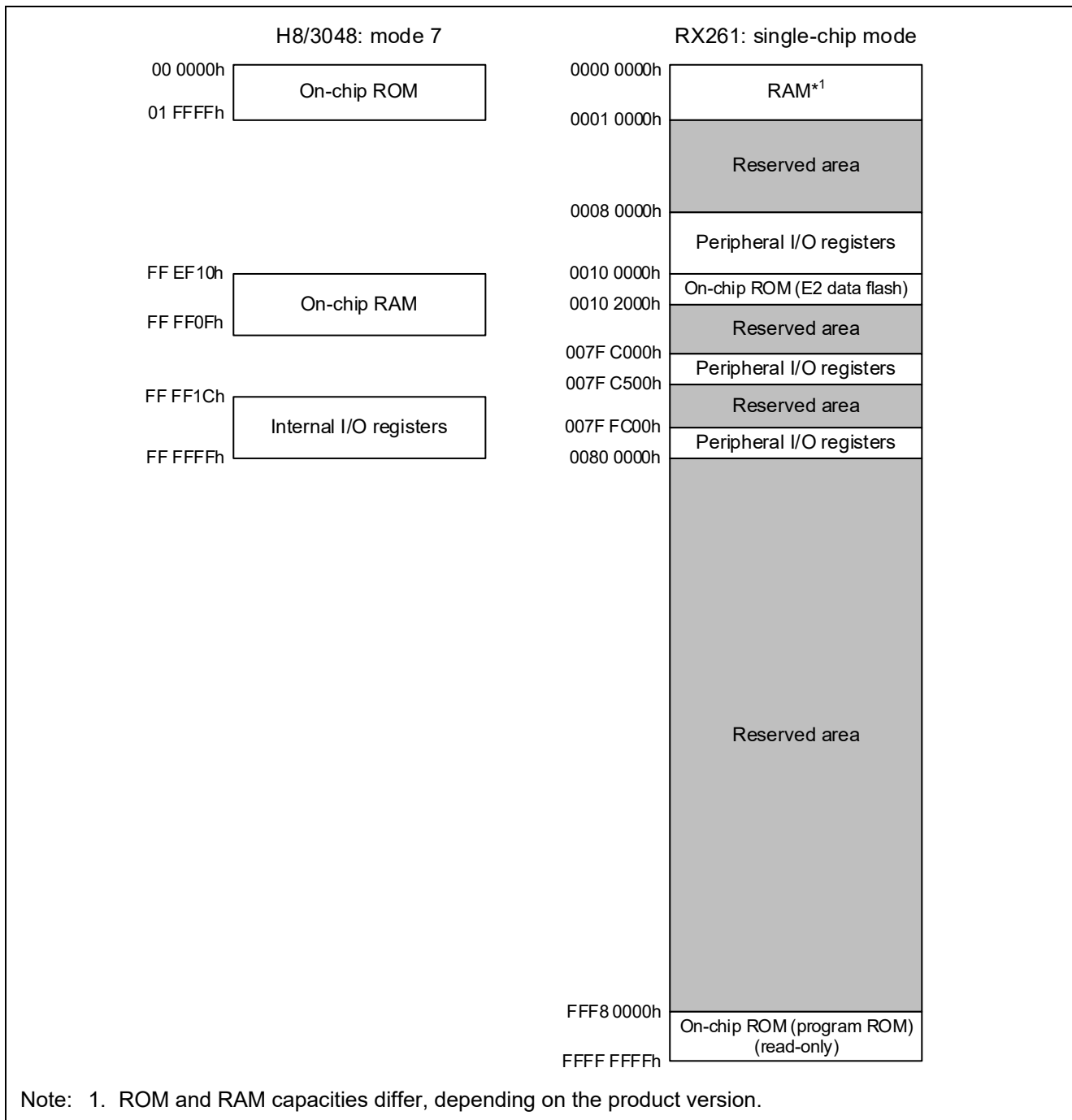
**Table 1.5 Operating Modes of the RX261 Device**

Operating Modes of the RX261 Device	Mode Description
Single-chip mode*1	Operating mode in which the on-chip ROM is enabled
Boot mode	Operating mode in which the on-chip flash memory rewrite program (boot program), which is stored in a dedicated area inside the microcontroller, runs. The on-chip ROM can be programmed by a device outside of the microcontroller by using the USB or asynchronous serial interface (SCI1).

Note: 1. The on-chip ROM can be programmed without resetting the system by means of the ROM rewrite routine in the user program.

### 1.5.2 Comparison of the Memory Map Between the H8/3048 and RX261 Devices

Figure 1.9 compares the memory maps between the H8/3048 device in mode 7 and the RX261 device in single chip mode.



**Figure 1.9 Comparison of the Memory Map Between the H8/3048 Device and the RX261 Device (in Single-chip Mode)**

### 1.5.3 Operating Mode Settings

The operating mode of the RX261 can be selected based on the level of the mode pins when the reset is released.

Table 1.6 shows the operating modes determined by the mode pins.

**Table 1.6 Pin Settings and Operating Modes of the RX261 Device**

Pin		Mode Name
MD <sup>*1</sup>	UB <sup>*2</sup>	
High	—	Single mode
Low	Low	Boot mode (SCI interface)
	High	Boot mode (USB interface)
Low → High <sup>*3</sup>	Low	Boot mode (FINE interface)

Note 1. Transition between operating modes is in progress during the wait time after RES# cancellation or power-on reset time. Accordingly, do not change the input level on the MD pin during these times. For details on the wait time after RES# cancellation and power-on reset time, see section 47, Electrical Characteristics. In boot mode, an input pull-up resistor is in the enabled state during the period from the resetting in progress to the completion of the operating mode transition. In single-chip mode, an input pull-up resistor is in the enabled state during the period from the resetting in progress to the setting of disabling the input pull-up resistor in the pull-up control register.

Note 2. The UB pin function is multiplexed on the same pin as the PC7 general-purpose port pin function and input and output pin functions for peripheral modules, so the pin can also be used in those roles.

Note 3. Place the low level on this pin when the reset is released, and change it to the high level within 20 to 100 msec

## 1.6 Processor Modes

The RX261 device supports two processing modes: supervisor mode and user mode. These two processor modes can be used to provide a hierarchical protection mechanism for CPU resources.

During migration from the H8/3048 device, using the RX261 device in only supervisor mode allows for software replacement without consideration of processor mode.

**Table 1.7 Processor Modes**

Processor Modes	Migration Conditions	Brief Description
Supervisor mode	<ul style="list-style-type: none"> <li>Reset cancellation</li> <li>Exception occurrence (The PSW.PM bit is cleared to "0".)</li> </ul>	<p>All CPU resources are accessible, and all instructions can be executed (no limitations).</p> <p>This is the mode in which the OS and other system programs ordinarily operate.</p>
User mode	<p>The PSW.PM bit is set to "1".</p> <p>Note that, in this case, the following conditions must also be satisfied: After the PSW.PM bit saved in a stack is set to 1, the RTE instruction is executed. Alternatively, after the PSW.PM bit saved in the BPSW register is set to 1, the RTFI instruction is executed.</p>	<p>For some CPU resources, write access is restricted and privileged instructions cannot be used. These resources include the BPC register, BPSW register, and some bits in the PSW register.</p> <p>This is the mode in which user programs such as application programs ordinarily operate.</p>

Switchover from supervisor mode to user mode:

The mode switches to user mode by using either of the following procedures:

- 1) Set the PSW.PM bit saved in a stack to "1", and then execute the RTE instruction.
- 2) Set the PSW.PM bit saved in the backup PSW (BPSW) register to "1", and then execute the RTFI instruction.

When the mode changes to user mode, the stack pointer specification bit (U) of the PSW register is set to "1".

Switchover from user mode to supervisor mode:

When an exception occurs, the PSW.PM bit is set to "0" and the CPU is placed in supervisor mode. Hardware preprocessing is performed in supervisor mode. The processor mode in which the CPU was placed immediately before the exception occurs is held by the PSW.PM bit that is saved.

## 1.7 Exception Handling

This section describes the differences between the H8/3048 and RX261 devices in regard to handling of general exceptions (including interrupts).

### 1.7.1 Comparison of Exception Sources

Table 1.8 compares the exception sources in the H8/3048 and RX261 devices.

**Table 1.8 Comparison of Exception Sources**


Source		H8/3048	RX261
Reset		Implemented	Implemented
Interrupt	NMI	Implemented	Implemented
	Internal	Implemented	Implemented* <sup>1</sup> , * <sup>2</sup>
	External	Implemented	Implemented* <sup>1</sup> , * <sup>2</sup>
Undefined instruction exception		Not implemented	Implemented* <sup>3</sup>
Privileged instruction exception		Not implemented	Implemented* <sup>4</sup>
Access exception		Not implemented	Implemented* <sup>5</sup>
Floating-point exception		Not implemented	Implemented* <sup>6</sup>
Trap		Implemented (TRAPA instruction)	Implemented (INT or BRK instruction)

- Notes: 1. Fast interrupts are supported. (Fast interrupts have priority level 15, the highest level.)  
 2. Figure 1.10 shows interrupt exception handling.  
 3. This exception source occurs upon detection of an attempt to execute an undefined instruction (an unimplemented instruction).  
 4. A privileged instruction exception occurs when a privileged instruction is used while the CPU is operating in user mode.  
 5. This exception source occurs upon detection of an error occurring in memory access from the CPU.  
 6. This exception source occurs upon detection of an unimplemented process and five exception events (overflow, underflow, abnormal accuracy, division by zero, and invalid operation).

### 1.7.2 Priority of Exception Sources in Exception Handling

Table 1.9 shows the priority of exception sources in the H8/3048 and RX261 devices.

**Table 1.9 Priority of Exception Sources**

Priority	H8/3048	RX261
High  Low	Reset	Reset
	External interrupt (NMI)	Non-maskable interrupt
	— (Not applicable)	Fast interrupt
	Interrupt (internal/external)	Interrupt (internal/external)* <sup>1</sup>
	— (Not applicable)	Instruction access exception
	— (Not applicable)	Undefined instruction exception or privileged instruction exception
	Trap instruction	Unconditional trap
	— (Not applicable)	Operand access exception
	— (Not applicable)	Single-precision floating-point exception

Note: 1. The priority of these interrupts is determined by the interrupt controller.

### 1.7.3 Basic Processing Sequence of Exception Handling

Figure 1.10 shows the processing sequence of interrupt exception handling (internal/external) on the H8/3048 and RX261 devices.

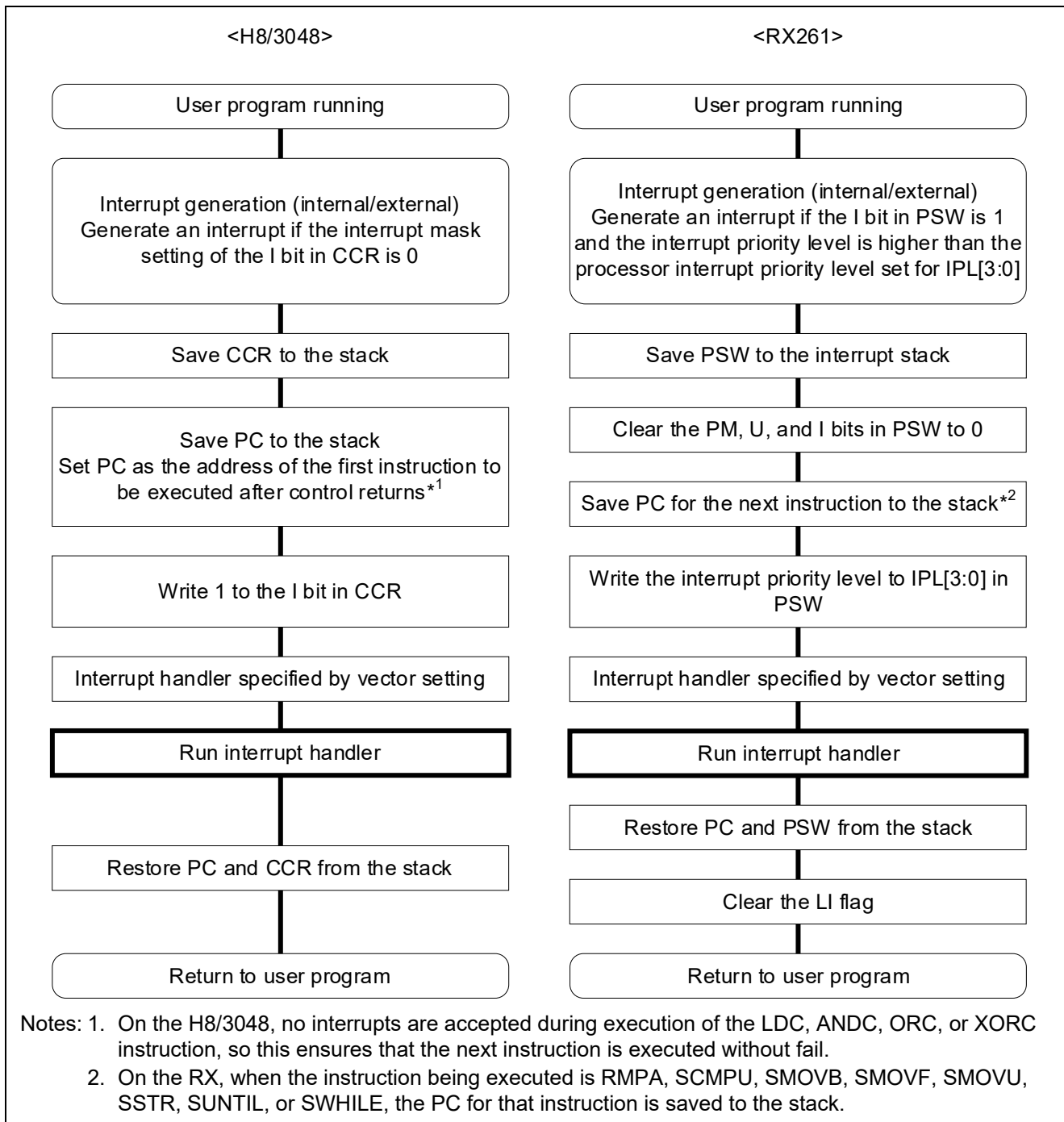


Figure 1.10 Processing Sequence of Interrupt Exception Handling (Internal/External)

### 1.7.4 Vector Configuration

On the H8/3048 device, only fixed vectors are used. On the RX261 device, however, the vectors are divided into two vector tables: exception vector table and interrupt vector table. On the RX261 device, the interrupt vector table is set before interrupts are enabled. The interrupt vector table is set by setting the start address of the interrupt vector table in the INTB register. Exception vectors cannot be reallocated because they are assigned to resets and other system exceptions. For information on memory relocation of vector addresses, refer to the “RX260 Group, RX261 Group User’s Manual: Hardware”.

Figure 1.11 shows the configuration of the interrupt vector table in the RX261 device.

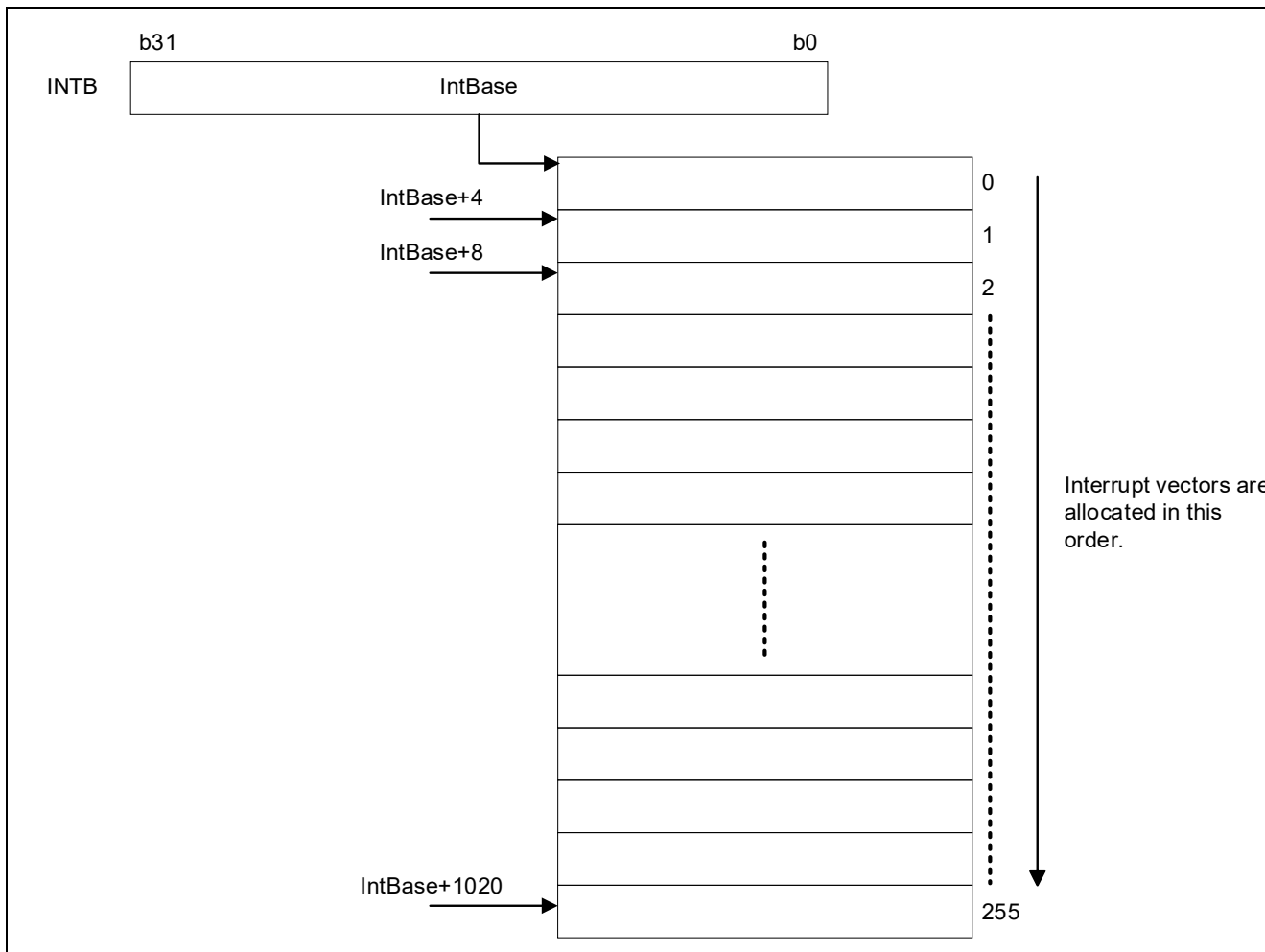


Figure 1.11 Configuration of the Interrupt Vector Table in the RX261 Device

### 1.7.5 Interrupt Masking by CCR (PSW)

On the H8/3048 device, a maximum of three interrupt mask levels can be set by using a combination of the I and UI bits in CCR when the UE bit in SYSCR is set to 0. On the RX261 device, a maximum of 16 interrupt mask levels can be set by using a combination of the IPL and I bits in PSW.

Table 1.10 lists the interrupt-related bits in PSW.

**Table 1.10 Interrupt-related Bits in PSW**

Bits in PSW on the RX261	Description
IPL[3:0]	Interrupt mask level (priority level) set by MPU Value that can be set: 0 to Fh (level 0 to 15) When an interrupt request occurs, the priority level of the interrupt source is compared with the mask level set by these bits. If the priority level is higher than the mask level, the interrupt is accepted.
I	Value set by the I bit (interrupt enable bit): 0: Interrupts are disabled. 1: Interrupts are enabled. After a system reset occurs, no interrupts are accepted unless this bit is set to "1". This bit is cleared to "0" when an interrupt is accepted. While the value of this bit is "0", no interrupts are accepted. However, "1" is set for the interrupt status flag in the interrupt controller when an interrupt is generated.

## 1.8 Interrupt Handling

This section compares the interrupt controller functions of the H8/3048 and RX261 devices.

### 1.8.1 Interrupt Controller

Table 1.11 shows the differences in the interrupt controller specifications.

**Table 1.11 Differences in the Interrupt Controller Between the Devices**

Item		H8/3048	RX261
Interrupts	Peripheral function interrupts	<ul style="list-style-type: none"> <li>Interrupts from peripheral modules</li> <li>Interrupt detection: Edge/level*1</li> </ul>	<ul style="list-style-type: none"> <li>Interrupts from peripheral modules</li> <li>Interrupt detection: Edge/level*1</li> </ul>
	External function interrupts	<ul style="list-style-type: none"> <li>IRQ0 to IRQ5 pins</li> <li>Sources: 6</li> <li>Interrupt detection: Low level or falling edge can be specified for each source.</li> </ul>	<ul style="list-style-type: none"> <li>IRQ0 to IRQ7 pins</li> <li>Sources: 8</li> <li>Interrupt detection: Low level, falling edge, rising edge, or both edges can be specified for each source.</li> <li>Digital filter function supported</li> </ul>
	Software interrupts	—	Number of interrupt sources due to a write to a register: 1
	Event link interrupts	—	ELSR8I, ELSR18I, and ELSR19I interrupts are generated by ELC events.
	Interrupt priority	Priority/non-priority levels can be specified for each source or each module by configuring register settings.	Priority is set by registers.
	Fast interrupts	—	Fast interrupts supported Source:1
	DTC/DMAC control	Activation supported*2	Activation supported*2
Non-maskable interrupts	NMI pin interrupts	<ul style="list-style-type: none"> <li>Interrupt from the NMI pin</li> <li>Interrupt detection: falling edge/rising edge</li> </ul>	<ul style="list-style-type: none"> <li>Interrupt from the NMI pin</li> <li>Interrupt detection: falling edge/rising edge</li> <li>Digital filter function supported</li> </ul>
	Other sources	<ul style="list-style-type: none"> <li>Trap instruction</li> </ul>	<ul style="list-style-type: none"> <li>Interrupt on detection of oscillation stop</li> <li>WDT underflow or refresh error</li> <li>IWDT underflow or refresh error</li> <li>Voltage monitor 1 interrupt</li> <li>Voltage monitor 2 interrupt</li> <li>RAM error interrupt</li> </ul>
Exiting the low-power mode			<ul style="list-style-type: none"> <li>Sleep mode/deep sleep mode: Return is initiated by any non-maskable interrupt or any interrupt.</li> <li>Software standby mode: Return is initiated by the non-maskable interrupt, external pin interrupts (IRQ0 to IRQ7), peripheral interrupts (voltage monitoring 1, voltage monitoring 2, RTC alarm/periodic, REMC, USB0 resume), or ELSR8I</li> </ul>

		<p>interrupt (LPT dedicated interrupt).</p> <ul style="list-style-type: none"> <li>• Snooze mode:</li> <li>• Return is initiated by the non-maskable interrupt, external pin interrupts (IRQ0 to IRQ7), peripheral interrupts (voltage monitoring 1, voltage monitoring 2, RTC alarm/periodic, REMC, USB0 resume), or SNZI interrupt (snooze release interrupt).</li> </ul>
--	--	---

- Notes: 1. The detection method is fixed for fixed-connection peripheral modules.  
 2. On the H8/3048 device, activation sources are set in the DMAC. On the RX261 device, they are set in the interrupt controller.

The interrupt controller of the H8/3048 device controls IRQ interrupt flags only, while peripheral module interrupt flags are controlled by the peripheral modules. On the RX261 device, the interrupt request registers of the interrupt controller control the interrupt flags for IRQ and all peripheral modules. The interrupt controller also controls DMAC activation source settings. On the H8/3048 device, the DMAC provides a function that disables transfer when an NMI interrupt occurs. On the RX261 device, however, no such function is available.

Table 1.12 compares the interrupt controllers of the devices.

**Table 1.12 Differences in Interrupt Controller Registers Between the Devices**

H8/3048	RX261
Interrupt controller	
System control register (SYSCR)	
Software standby (SSBY)	Low power consumption Standby control register (SBYCR)
Standby timer select 2 to 0 (STS)	— (There is no corresponding register.)
User bit enable (UE)	— (There is no corresponding register.)
RAM enable (RAME)	Operating mode System control register 1 (SYSCR1)
NMI edge select (NMIEG)	Interrupt controller (ICUb) NMI pin interrupt control register (NMICR)
Interrupt controller	Interrupt controller (ICUb)
Interrupt priority register A/B (IPRA/IPRB)	Interrupt source priority register n (IPRn)
Interrupt controller	Interrupt controller (ICUb)
IRQ Status Register (ISR)	Interrupt request register n (IRn)
Peripheral module	
Interrupt request flag register	
Interrupt controller	Interrupt controller (ICUb)
IRQ enable register (IER)	Interrupt request enable register m (IERm)
Peripheral module	
Interrupt request enable register	
Interrupt controller	Interrupt controller (ICUb)
IRQ sense control register (ISCR)	IRQ control register i (IRQCRI)
DMA controller	Interrupt controller (ICUb)
Data transfer control register (DTCR)	DMAC activation source select register m (DMRSRm)
— (There is no corresponding register.)	Interrupt controller (ICUb) Fast interrupt setting register (FIR) Software interrupt generation register (SWINTR) DTC transfer request enable register n (DTCERn) IRQ pin digital filter enable register 0 (IRQFLTE0) IRQ pin digital filter setting register 0 (IRQFLTC0) Non-maskable interrupt status register (NMISR) Non-maskable interrupt enable register (NMIER) Non-maskable interrupt status clearing register (NMICLR) NMI pin digital filter enable register (NMIFLTE) NMI pin digital filter setting register (NMIFLTC)

### 1.8.2 Management of Interrupt Flags

When a peripheral module of the H8/3048 device generates an interrupt on detection of an edge, the corresponding interrupt flag (interrupt source flag) is cleared (the flag is cleared and a dummy read is performed) in the interrupt handler. This is done because the interrupt will be generated again if the flag is not cleared in the handler. On the RX261 device, the interrupt flags (interrupt status flags) are managed internally by the interrupt controller. The interrupt controller has a function that sends an interrupt request to the CPU, DMAC, or DTC and automatically clears the corresponding interrupt status flag when receiving acknowledgment. Therefore, it is unnecessary to perform flag clearing and dummy-read operations that are performed on the H8/3048 device. Note that, in the case of interrupts generated by level detection, the source flags residing in the peripheral modules must be cleared. For details, refer to the “RX260 Group, RX261 Group User’s Manual: Hardware”.

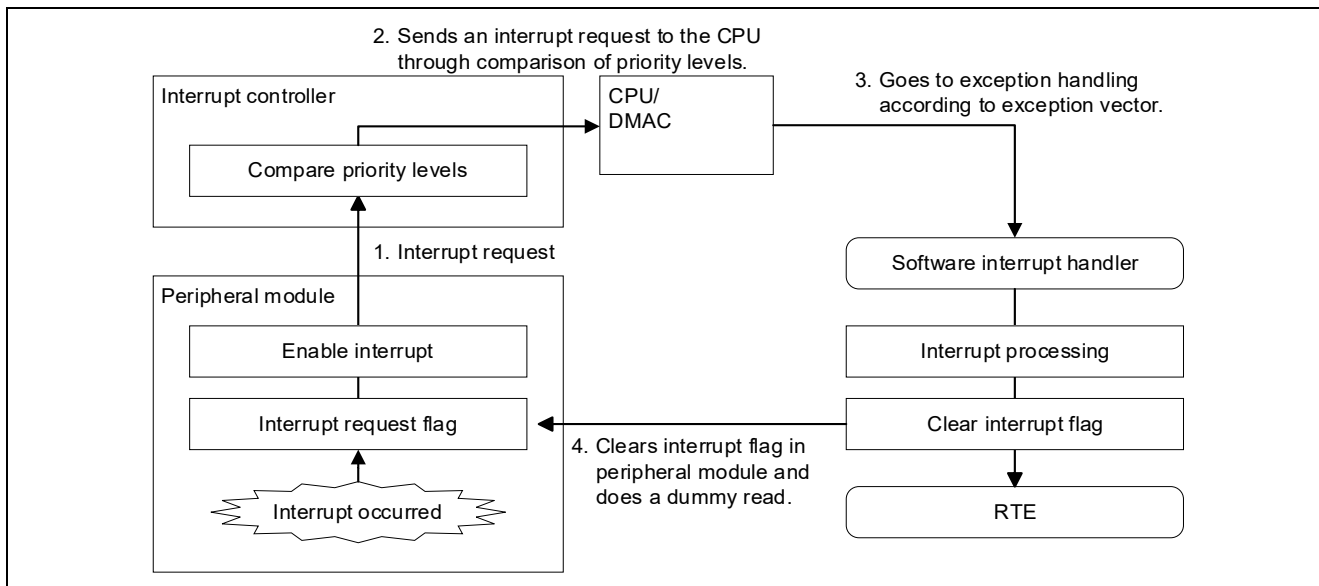
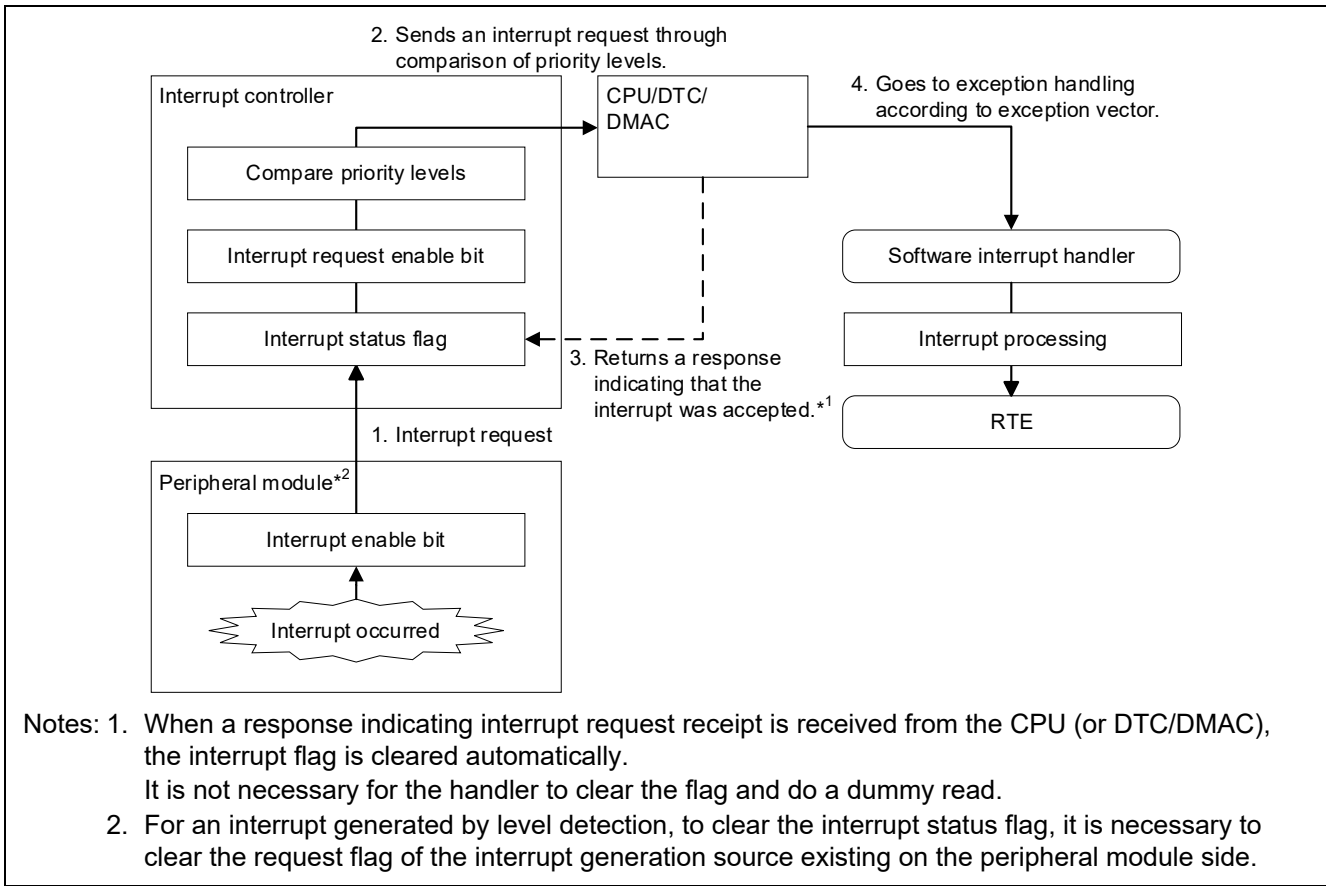


Figure 1.12 Peripheral Module Interrupt (Generated on Edge Detection) for the H8/3048 Device



**Figure 1.13 Peripheral Module Interrupt (Generated on Edge Detection) for the RX261 Device**

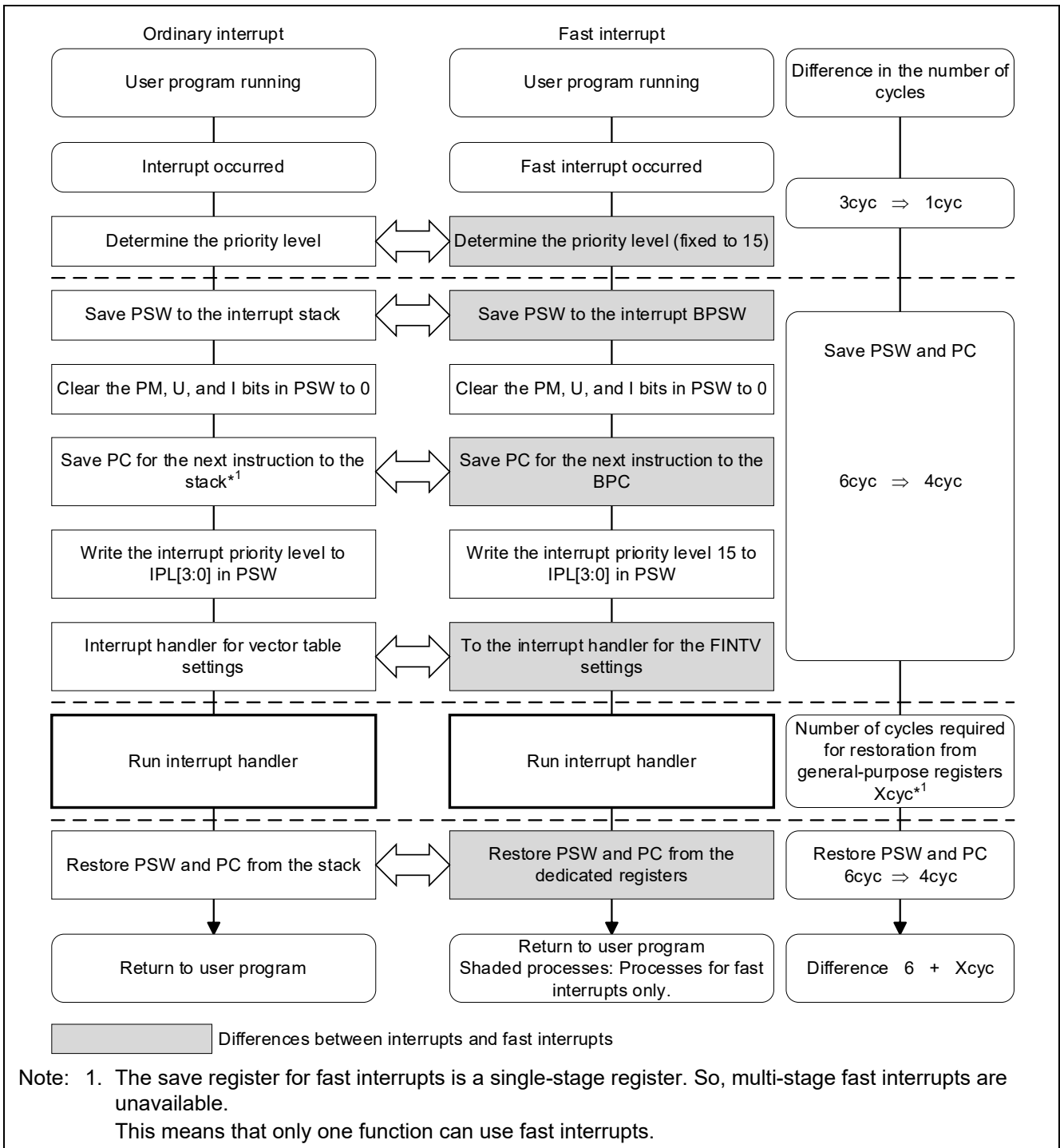
### 1.8.3 Control of Fast Interrupts

The RX261 device supports fast interrupts in addition to ordinary interrupts.

Ordinary interrupt: After determining the interrupt priority, it is necessary to save the contents of the control registers and general-purpose registers to the internal or external RAM by software.

Fast interrupt: Operation gives the interrupt the highest priority. When a fast interrupt occurs, the contents of the control registers are saved to dedicated registers, thus enabling faster interrupt-triggered activation than an ordinary interrupt.

Some general-purpose registers can be dedicated for interrupts by setting a compiler option. Use of interrupt-dedicated registers makes it unnecessary to perform save and restore operations for general-purpose registers, thus further speeding up fast interrupts.



**Figure 1.14 Differences Between Ordinary Interrupts and Fast Interrupts on the RX261 Device**

### 1.8.4 Digital Filter

The RX261 device is provided with a digital filter function for the IRQ and NMI level signals. The sampling clock for the digital filter can be set. Interrupt signals that do not last for at least three cycles of the sampling clock base are not accepted.

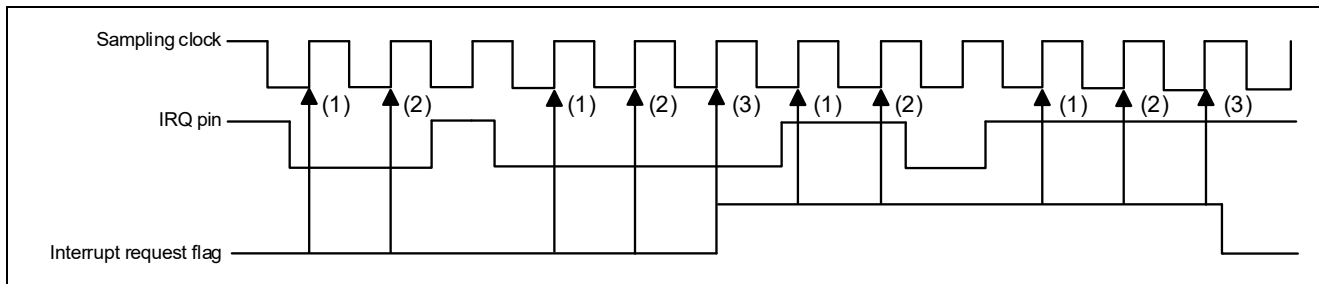


Figure 1.15 Digital Filter Operation Example

### 1.8.5 Multiple Interrupts

The RX261 device supports multiple interrupts. To handle multiple interrupts on the RX261 device, it is necessary to set the PSW(I) bit to 1 (enable interrupts) in the interrupt handler. Figure 1.16 shows the sequence of operations performed when multiple interrupts are prohibited. Figure 1.17 shows the sequence of operations performed when multiple interrupts occur.

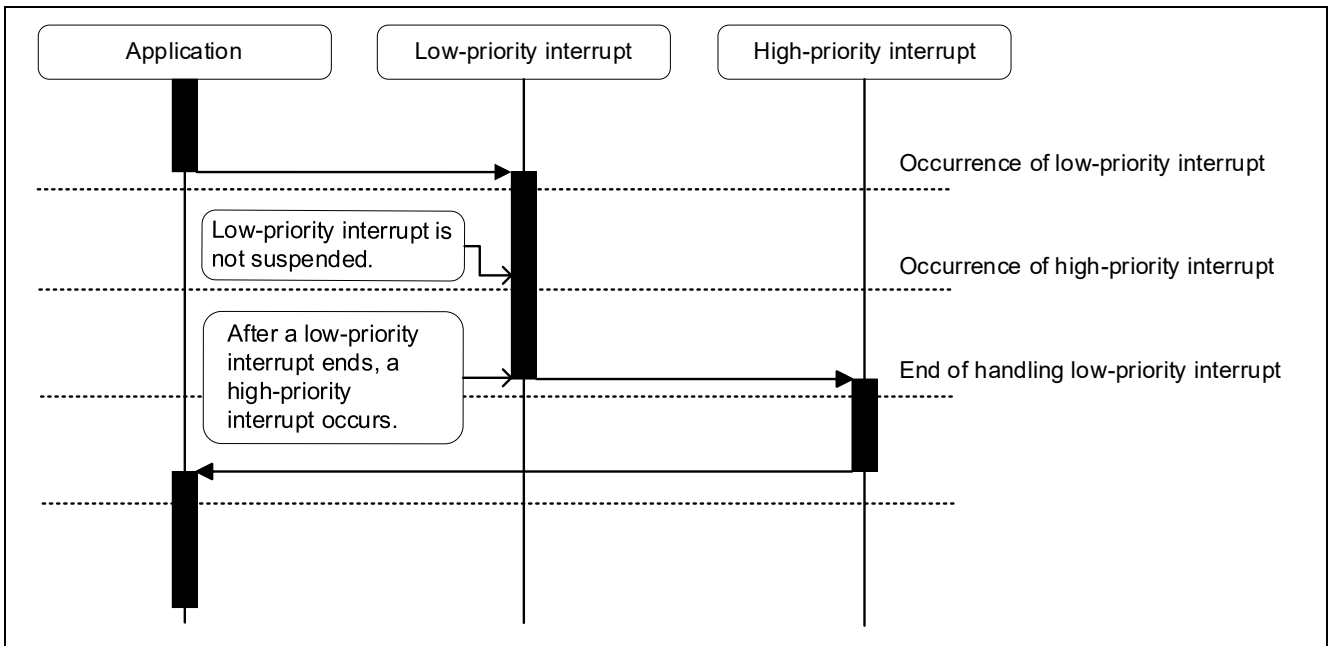


Figure 1.16 Interrupt Handling Sequence on the RX261 Device (when the PSW(I) bit is cleared)

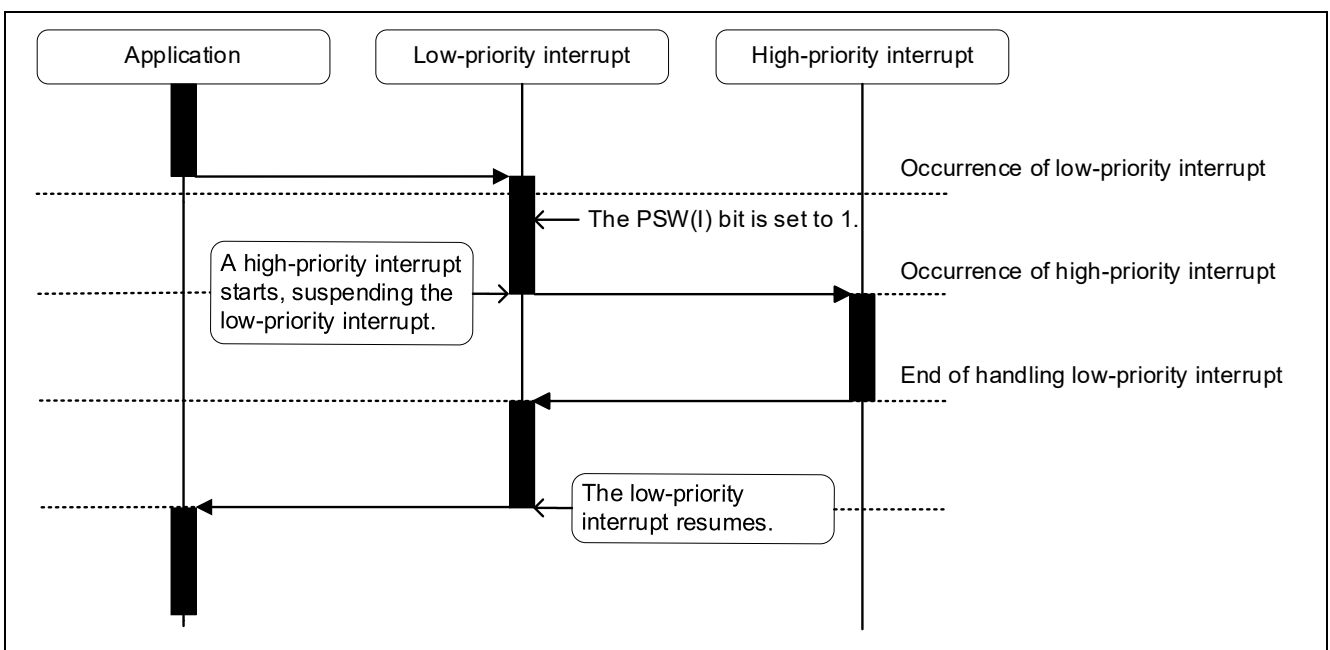


Figure 1.17 Interrupt Handling Sequence on the RX261 Device (when the PSW(I) bit is set)

## 2. On-Chip Functions (Peripheral Modules)

### 2.1 List of On-Chip Functions

**Table 2.1 List of On-Chip Functions**

H8/3048	RX261
Bus controller	-(Does not has external Bus controller)
DMA controller	DMA controller (DMACA)
I/O Ports	I/O Ports
16-bit integrated timer unit (ITU)	General PWM Timer (GPTWa)
Programmable timing pattern controller (TPC)	—
Watchdog timer	Watchdog timer (WDTA) Independent watchdog timer (IWDTa)
Serial communications interface	Serial communications interface (SCIh/SCIh/RSCI)
Smartcard interface	
A/D converter	12-bit A/D converter (S12ADE)
D/A converter	8-bit D/A converter (DAa)
4 KB RAM	128 KB (max.) RAM
ROM Mask ROM of 128 KB, max., or flash memory of 128 KB	ROM User area of 512 KB, max. Data area of 8 KB Extra area for storing the startup area information, access window information, and unique IDs
Interrupt controller	Interrupt controller (ICUb)
Low-power mode	Low power consumption
—	Function for lower operating power consumption
—	Data transfer controller (DTCb)
—	Multi-function pin controller (MPC)
—	Event link controller (ELC)
—	CRC calculator (CRC)
—	Serial peripheral interface (RSPIC)
—	I <sup>2</sup> C-bus controller (RIICa)
—	GPTW Port Output Enable (POEGc)
—	Compare match timer (CMT)
—	Realtime clock (RTCba)
—	Data operation circuit (DOC)
—	Clock frequency accuracy measurement circuit (CAC)
—	Capacitive touch sensing unit (CTSU2SLa)
—	Memory protection unit (MPU)
—	8-bit timer (TMRa)
—	CAN FD module (CAN FD)
—	Low-power timer (LPT)
—	USB 2.0 host/function module (USBe)
—	Remote Control Signal Receiver (REMCa)
—	Renesas Secure IP (RSIP-E11A)
—	Temperature sensor (TEMPSA)
—	Comparator B (CMPBa)

## 2.2 I/O Ports

### 2.2.1 Comparison of I/O Port Specifications

Table 2.2 Number of I/O Ports

Item	H8/3048		RX261		
	100-pin	100-pin	80-pin	64-pin	48-pin
I/O pins	70	89	69	53	39
Input pull-up MOS	20	—	—	—	—
Input pull-up resistor	—	89	69	53	39
Open-drain output	—	63	47	35	27
5 V tolerant	—	4	4	2	2
Schmitt-trigger input	15	—	—	—	—
Input pins	8	3	3	3	1
Total number of ports	78	92	72	56	40

### 2.2.2 I/O Settings

On the H8/3048 and RX261 devices, the ports are shared by multiple functions, such as peripheral module I/O pins and interrupt input pins. Therefore, the functions of the general-purpose input/output ports, peripheral function input/output pins, interrupt input pins, and bus control pins must be selected by configuring the settings of the I/O port registers and on-chip peripheral modules.

On the H8/3048 device, to select the pin for a specific function, a combination of settings for the operating mode and function registers must be specified, and this requires complex operations. In contrast, on the RX261 device, control of pin function selection is centralized in the multi-function pin controller (MPC).

In addition, on the RX261 device, a variety of additional settings can be configured. Figure 2.1 shows an overview of I/O settings on the RX261 device.

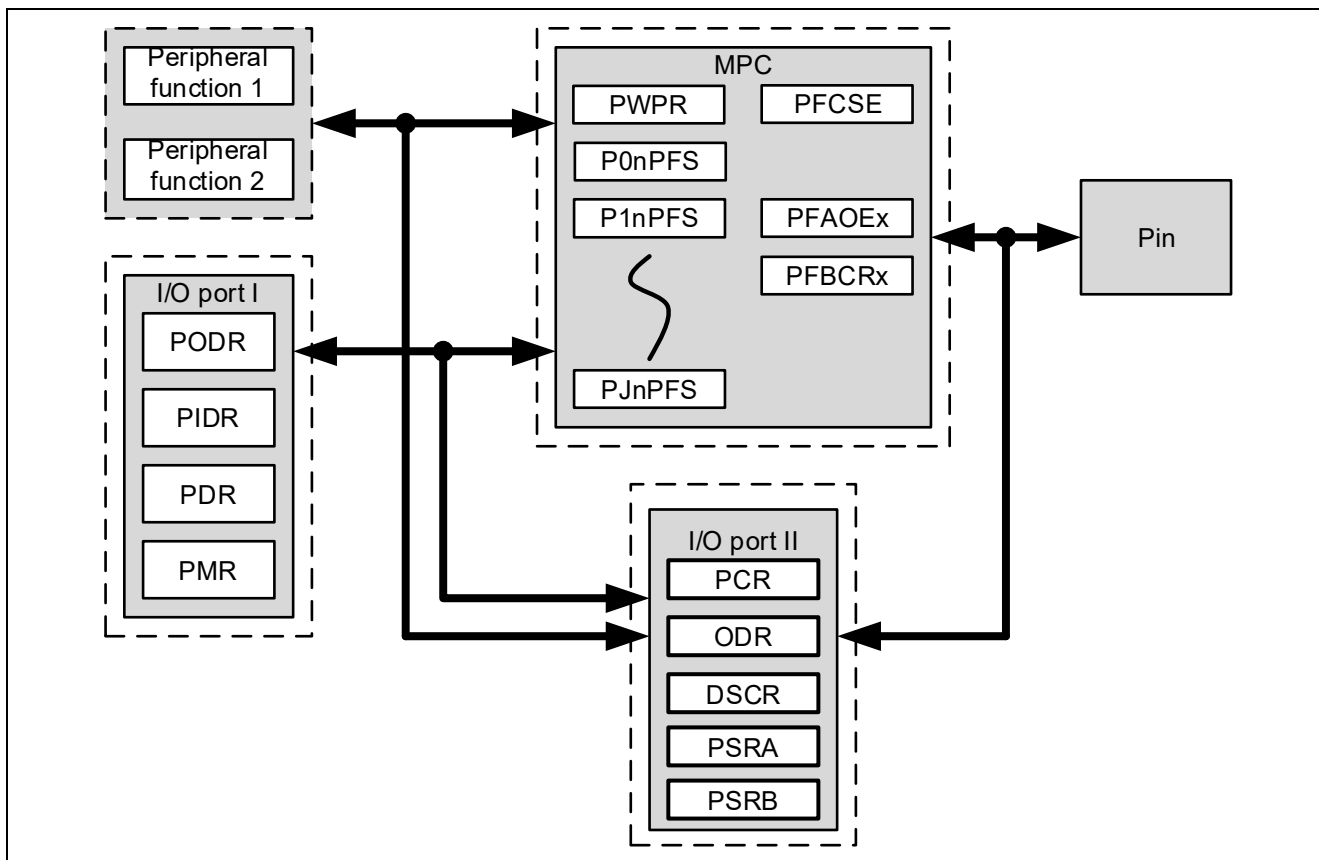


Figure 2.1 Overview of I/O Settings on the RX261 Device

Each I/O port on the RX261 device consists of multiplexed pins. Therefore, the pins of an I/O port must be assigned to the desired port type (a general-purpose I/O port or on-chip module function).

- To use a port as a general-purpose port, assign the port to I/O Port n.\*1
  - Table 2.3 lists the registers that must be configured when an I/O port is to be used as a general-purpose port.
  - Figure 2.2 shows an example of the settings for using a port as a general-purpose port.
- To use a port for a peripheral function, assign the port to I/O Port n and configure the multi-function pin controller (MPC).
  - Table 2.3 to Table 2.6 list the registers that must be configured when an I/O port is to be used for a peripheral function.
  - Figure 2.3 shows the peripheral module initialization sequence.

Note: 1. n indicates a port number that can be 0 to 5, A to E, H, or J.

**Table 2.3 Registers for I/O Ports on the RX261 Device**

Register	Function Name	Description
PDR	Port direction register	If a general-purpose I/O port is selected, this register is used to specify whether the port is to be used for input or output.
PODR	Port output register	A register used for a general-purpose output port. This register stores the output data of the pins to be used for the general-purpose output port.
PIDR	Port input register	A register used for a general-purpose input port. The pin status of the port is reflected to this register. The pin status can be read from this register regardless of the PDR and PMR values.
PMR	Port mode register	This register is used to set the pin functions of the port. Whether the pin is used for a general-purpose I/O port or a peripheral function is set for each pin.
ODR0	Open drain control register 0	Selects the port output format from among the following: <ul style="list-style-type: none"> <li>• CMOS output</li> <li>• N-channel open drain</li> <li>• P-channel open drain</li> <li>• Hi-Z</li> </ul>
ODR1	Open drain control register 1	Selects the port output format from among the following: <ul style="list-style-type: none"> <li>• CMOS output</li> <li>• N-channel open drain</li> </ul>
PCR	Pull-up control register	This register is used to specify whether to enable or disable the input pull-up resistor of the port.
PSRA	Port switching register A	This register is used to select which of the following is to be used:*1 <ul style="list-style-type: none"> <li>• General-purpose I/O function for the PB6 and PB7 pins</li> <li>• General-purpose I/O function for the PC0 and PC1 pins</li> </ul>
PSRB	Port switching register B	This register is used to select which of the following is to be used:*2 <ul style="list-style-type: none"> <li>• General-purpose I/O function for the PB5, PB3, PB1, and PB0 pins</li> <li>• General-purpose I/O function for the PC3, PC2, PC1, and PC0 pins</li> </ul>
PRWCNTR	Port Reading Wait Control Register	<ul style="list-style-type: none"> <li>• This register specifies the number of wait cycles to read the PIDR register</li> </ul>

Notes: 1. The PSRA register is present on only a 80 and 64-pin package product.

2. The PSRB register is present on only a 48-pin package product.

To assign an I/O port for a peripheral function, configure the registers for the I/O port registers listed in Table 2.3 and the registers listed in Table 2.4 to Table 2.6.

**Table 2.4 Register for the Register Write-Protection Function on the RX261 Device**

Register	Function Name	Description
PRCR	Protect register	<p>This register is used to control write protection that prevents an accidental rewrite of important registers. This register controls write protection on the following registers: the registers related to the clock generation circuit, the registers related to the operating mode, the registers related to the low power consumption function, the software reset register, the registers related to the low-power timer, the LVD-related registers.</p> <p>In addition to the above registers, this register also controls write protection on this register itself.</p>

**Table 2.5 Registers Related to the Module Stop Function on the RX261 Device**

Register	Function Name	Description
SBYCR	Standby control register	This register is used to control the standby mode.
MSTPCRA	Module stop control register A	This register is used to disable the following peripheral functions or to control the operating status: TMR, GPTW, CMT, S12AD, DA, DMAC, and DTC
MSTPCRB	Module stop control register B	This register is used to disable the following peripheral functions or to control the operating status: SCI12, DOC, ELC, CMPB, RSPI, USB0, RIIC0, CRC, S CI5,6, 1RSCI0
MSTPCRC	Module stop control register C	This register is used to disable the following peripheral functions or to control the operating status: RAM0, CAC, RSCI8/9, REMC0 and Deep sleep mode
MSTPCRD	Module stop control register D	This register is used to disable the following peripheral functions or to control the operating status: CAN FD, CTSU and RSIP
OPCCR	Operating power consumption control register	<p>This register can be used to set either of the following operating modes:</p> <ul style="list-style-type: none"> <li>• High-speed operating mode</li> <li>• Middle-speed operating mode</li> <li>• Middle-speed operating mode2</li> </ul>
SOPCCR	Secondary operating power consumption control register	<p>This register can be used to set either of the following operating modes:</p> <ul style="list-style-type: none"> <li>• High-speed or middle-speed operating mode or Middle-speed operating mode2</li> <li>• Low-speed operating mode</li> </ul>
RSTCKCR	Register for switching the clock source for exiting sleep mode	<p>This register can be used to select the clock source that triggers exiting sleep mode from the following clock sources:</p> <ul style="list-style-type: none"> <li>• LOCO</li> <li>• HOCO</li> <li>• Main clock oscillator</li> </ul>
SNZCR	Snooze Control Register	This register is used to specify the conditions for transitions from the software standby mode to the snooze mode and for release from the snooze mode.
SNZCR2	Snooze Control Register2	This register is used to select the interrupt sources for release from the snooze mode.

RPSCR	RAM Power Saving Control Register	This register is used to set the RAM area where the power supply is shut off in the software standby mode.
-------	-----------------------------------	--

Note: Rewrite this register after setting the PRCR.PRC1 bit to "1" (write enabled).

\*1 Depends on the setting of OPCCR.OPCM[2:0].

Table 2.6 Multi-Function Pin Controller Registers on the RX261 Device

Register	Function Name	Description
PWPR	Write protection register	Function that protects writes to PFS registers
P0nPFS	P0n pin function control register	Register for selecting the pin function (for port 0)
P1nPFS	P1n pin function control register	Register for selecting the pin function (for port 1)
P2nPFS	P2n pin function control register	Register for selecting the pin function (for port 2)
P3nPFS	P3n pin function control register	Register for selecting the pin function (for port 3)
P4nPFS	P4n pin function control register	Register for selecting the pin function (for port 4)
P5nPFS	P5n pin function control register	Register for selecting the pin function (for port 5)
PAnPFS	PAn pin function control register	Register for selecting the pin function (for port A)
PBnPFS	PBn pin function control register	Register for selecting the pin function (for port B)
PCnPFS	PCn pin function control register	Register for selecting the pin function (for port C)
PDnPFS	PDn pin function control register	Register for selecting the pin function (for port D)
PEnPFS	PEn pin function control register	Register for selecting the pin function (for port E)
PHnPFS	PHn pin function control register	Register for selecting the pin function (for port H)
PJnPFS	PJn pin function control register	Register for selecting the pin function (for port J)

The following figure shows the sequence of initialization operations needed to use an I/O port on the RX261 device as a general-purpose I/O port.

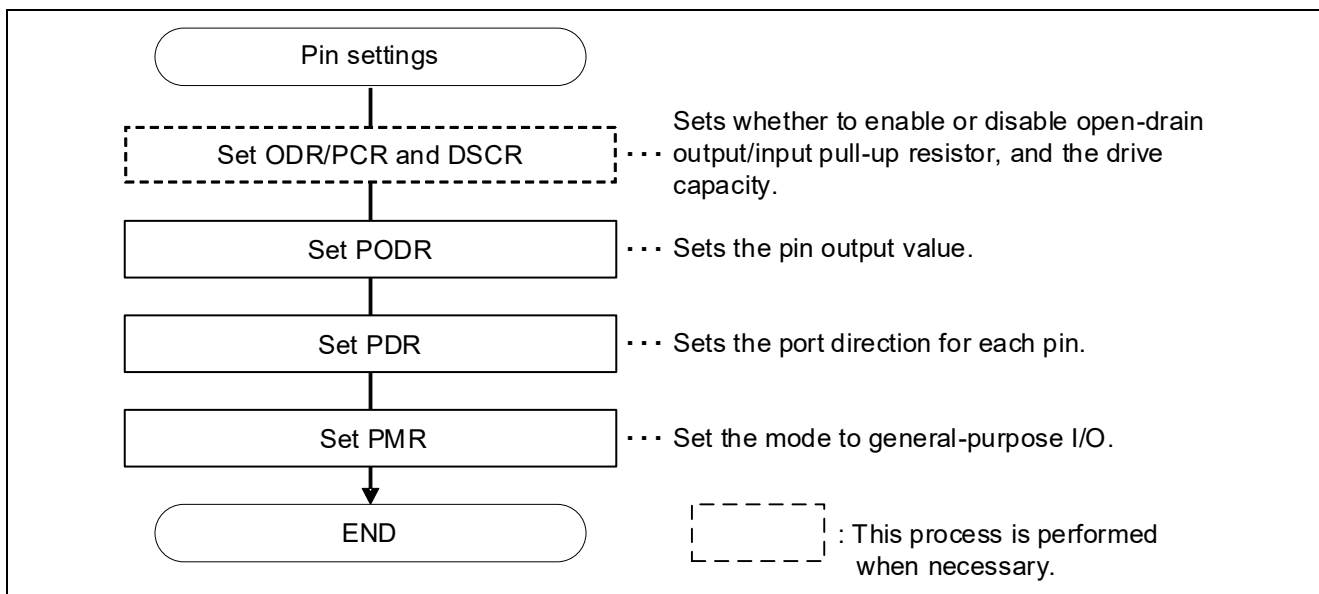
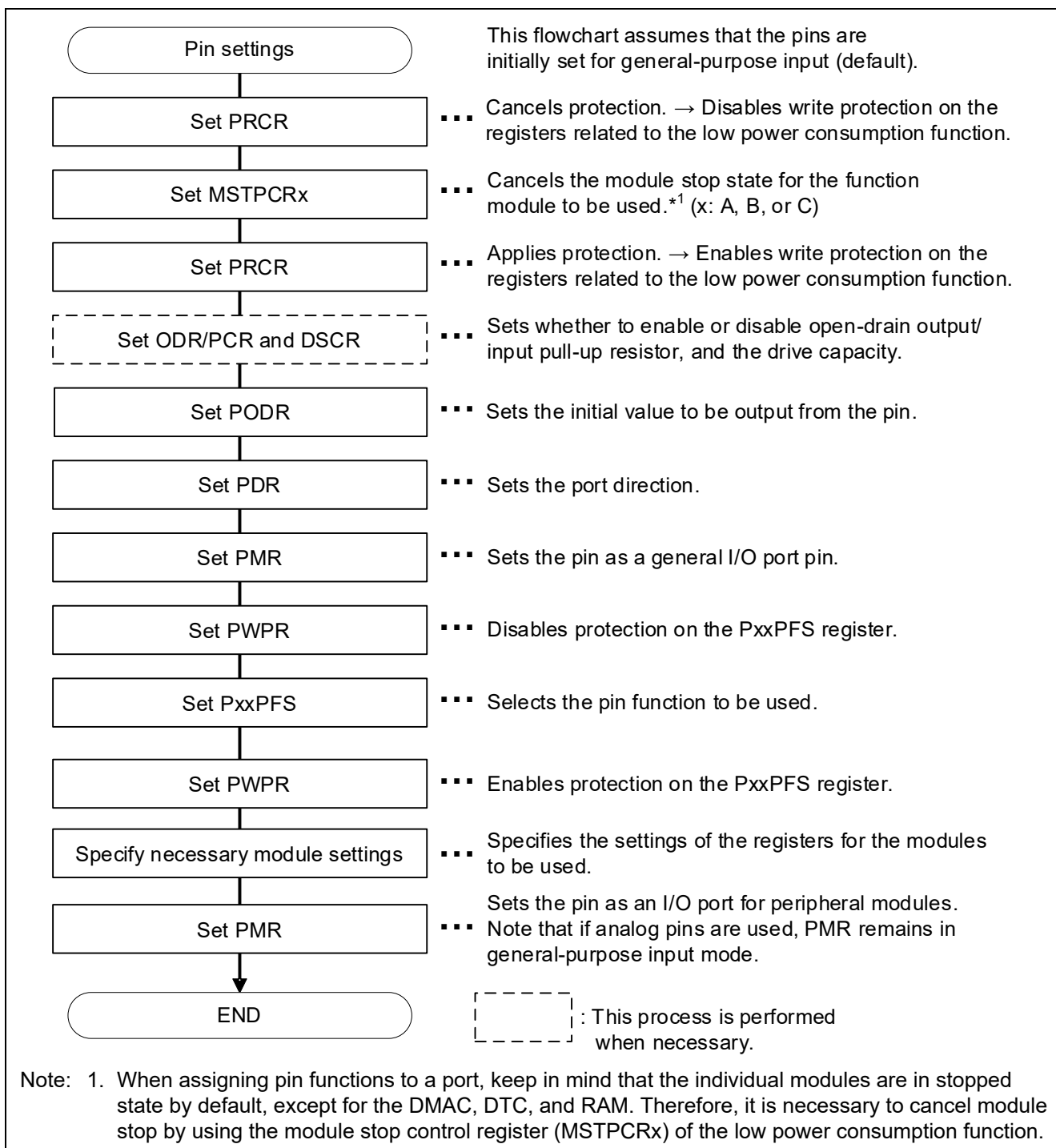


Figure 2.2 Sequence of Operations for Using a General-Purpose I/O Port on the RX261 Device

Figure 2.3 shows the sequence of initialization operations needed to use an I/O port on the RX261 device for pin functions. For example settings for using peripheral functions, including general-purpose input/output, refer to the relevant chapters.



**Figure 2.3 Sequence of Operations for Configuring Pin Functions on the RX261 Device**

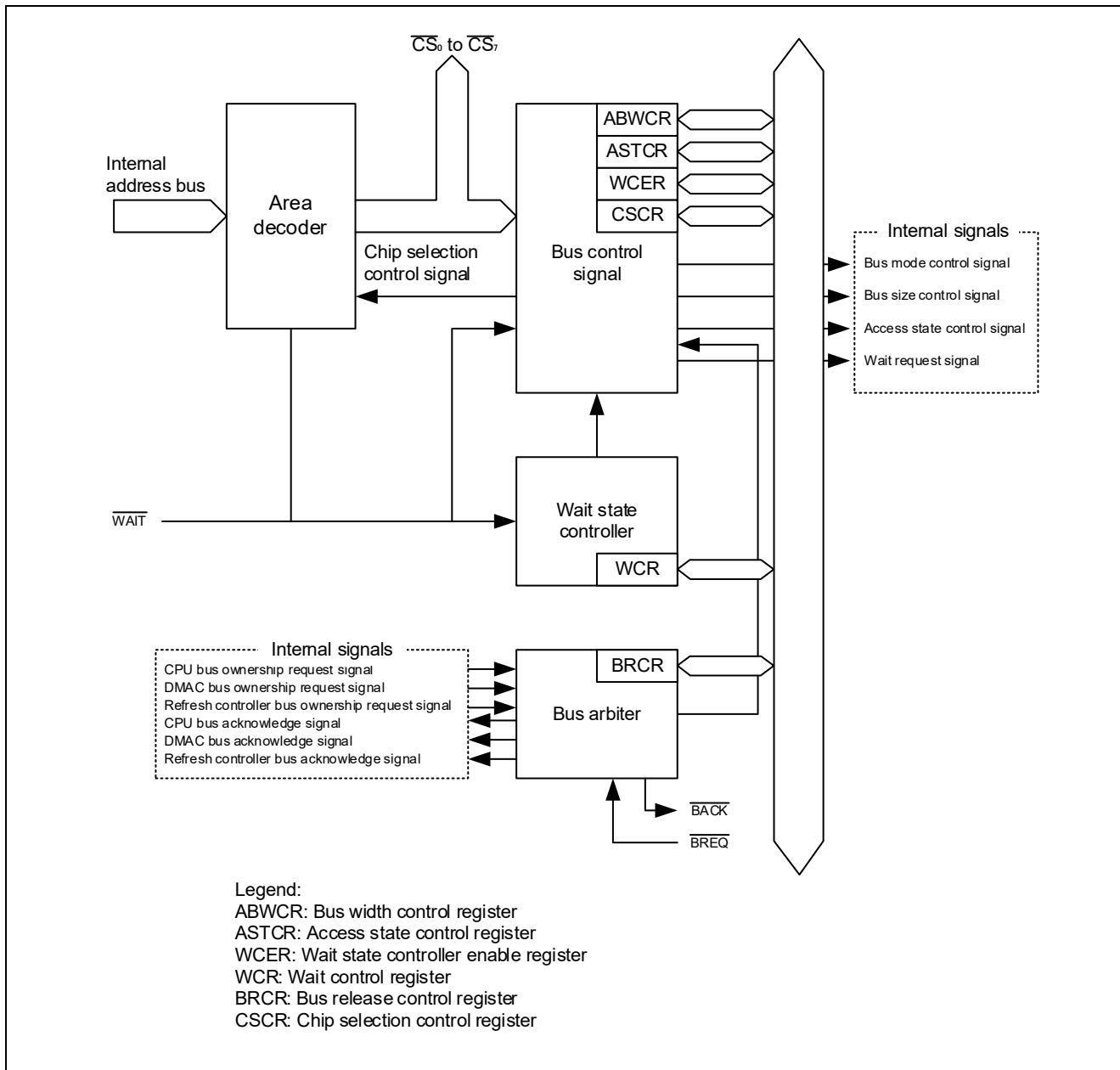
Note that the MSTPCR<sub>x</sub> register is write-protected by the register write protection function when canceling module stop. Therefore, use the protect register (PRCR) to cancel write protection before rewriting the MSTPCR<sub>x</sub> register. DMAC, DTC, and RAM are operational by default.

## 2.3 Buses

### 2.3.1 Bus Configuration

Figure 2.4 compares the bus configurations of the H8/3048 and RX261 devices. Furthermore, RX261 has no external bus controller.

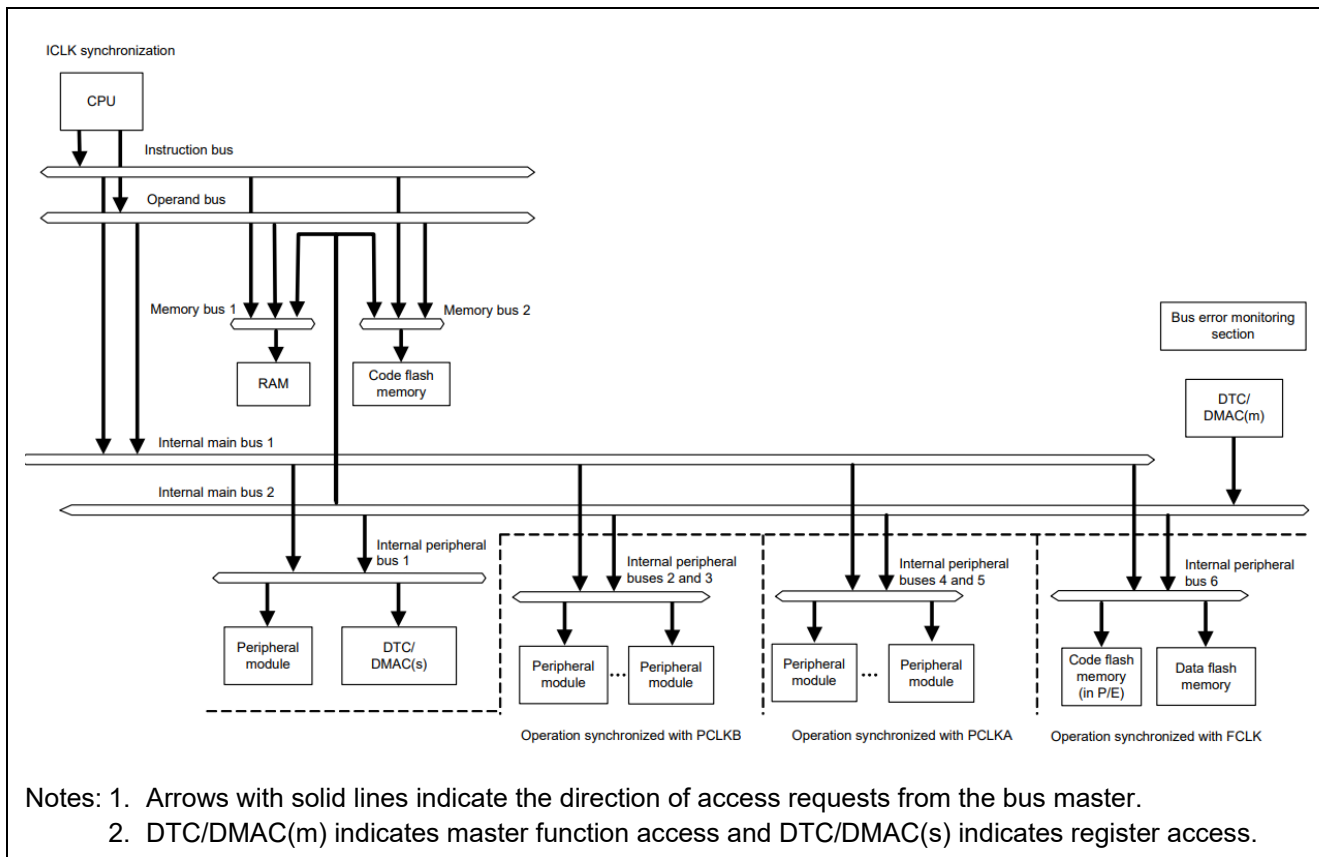
The following figure shows the configuration of the bus controller of the H8/3048 device.



**Figure 2.4 Configuration of the Bus Controller of the H8/3048 Device**

The H8/3048 device has only internal data buses that are synchronized with the system clock.

The following figure shows the configuration of the buses of the RX261 device.



**Figure 2.5 Configuration of the Buses of the RX261 Device**

The RX261 device has memory buses, internal buses, and peripheral buses, all of which are in a multistage configuration. This configuration allows modules to operate in parallel for each of the CPU, DMAC/DTC, and peripheral buses.

The following lists the buses of the RX261 device. On the RX261 device, as shown in Table 2.7, the clock with which the bus is synchronized differs depending on the bus.

**Table 2.7 Buses of the RX261 Device**

Bus	Connected modules	Clock*1
CPU buses (instruction bus and operand bus)	Instruction bus: CPU (instructions) and on-chip memory Operand bus: CPU (operands) and on-chip memory	ICLK
Memory bus 1	On-chip RAM	ICLK
Memory bus 2	On-chip ROM	ICLK
Internal main bus 1	CPU	ICLK
Internal main bus 2	DTC, DMAC, and on-chip memory (RAM and ROM)	ICLK
Internal peripheral bus 1	DTC, DMAC, interrupt controller, and bus error monitoring block	ICLK
Internal peripheral bus 2	Peripheral functions other than internal peripheral buses 1, 3, 4 and 5	PCLKB
Internal peripheral bus 3	USB0, CAN FD, CTSU, REMD and RSCI	PCLKB
Internal peripheral bus 4	GPTW	PCLKA
Internal peripheral bus 5	Can FD(message buffer RAM)	PCLKA
Internal peripheral bus 6	Flash control module and E2 data flash	FCLK

Note: 1. ICLK: System clock

PCLKA: Peripheral module clock A

PCLKB: Peripheral module clock B

FCLK: FlashIF clock

BCLK: External bus clock

## 2.4 Interrupt Controller

### 2.4.1 Usage Example for IRQ Pins

The following shows an example of settings for using the IRQ3 pin. On the H8/3048 device, P83 is used for the IRQ3 input pin. On the RX261 device, P33 is used for the IRQ3 input pin.

**Table 2.8 Interrupt Initial Setting Example (IRQ3 Settings)**

Procedure		H8/3048	RX261
1	Configure the I/O port settings (pin I/O and pin function settings)	P83DDR = 0	PORT3.PDR.B3 = 0 (Assigns P33 for input.) PORT3.PMR.B3 = 0 (Assigns P33 for a function pin.) MPC.PWPR.B0WI = 0 MPC.PWPR.PFSWE = 1 (Enables writes to PFS.) MPC.P33PFS.ISEL = 1 (Assigns IRQ3 for the interrupt function.) MPC.PWPR.PFSWE = 0 (Disables writes to PFS.) MPC.PWPR.B0WI = 1
2	Configure the interrupt controller	ISCR.IRQ3SC = 1 (IRQ detection: Detects a falling edge.) IER.IRQ3E = 1 (Enables IRQ3.)	IRQCR3.IRQMD[1:0] = 0x3 (IRQ detection: Detects a falling edge.) IRQFLTE0.FLTEN3 = 1 (Enables a digital filter for IRQ3.) IRQFLTC0.FCLKSEL3[1:0] = 0x3 (Selects PCLK/64 as the sampling clock.) IR067 = 0 (Clears the interrupt status flag.) IER08.IEN3 = 1 (Enables interrupt requests for IRQ3.) IPR067 = 15 (Sets the interrupt priority level to 15.)

## 2.5 DMA Controller

### 2.5.1 Comparison of Specifications

This section compares the DMA controller specifications of the H8/3048 and RX261 devices.

**Table 2.9 Differences in the DMAC Specifications Between the H8/3048 and RX261 Devices (1/2)**

Item	H8/3048		RX261
	DMAC		DMACA
Number of channels	4 channels (short address mode) or 2 channels (full address mode)		4 channels (DMACm (m = 0 to 3))
Transfer space	The 16-MB address space can be directly accessed.		512 MB (areas within the range from 00000000h to 0FFFFFFFh and the range from F0000000h to FFFFFFFFh, excluding reserved areas)
Maximum number of data units that can be transferred	16 M data units (= Maximum number of data units that can be transferred in block transfer mode (255) × Maximum transfer count (65,536))		1 M data units (= Maximum number of data units that can be transferred in block transfer mode (1,024) × Number of blocks (1,024))
DMA activation sources	<ul style="list-style-type: none"> <li>Internal interrupt</li> <li>External request</li> <li>Auto request</li> </ul>		The activation source can be selected for each channel. <ul style="list-style-type: none"> <li>Software trigger</li> <li>Interrupt request from a peripheral module</li> <li>Trigger input to an external interrupt input pin</li> </ul>
Channel priority	<ul style="list-style-type: none"> <li>Short address: Channel 0A &gt; channel 0B &gt; channel 1A &gt; channel 1B (Channel 0A has the highest priority.)</li> <li>Full address: Channel 0 &gt; channel 1 (Channel 0 has the highest priority.)</li> </ul>		<ul style="list-style-type: none"> <li>Channel 0 &gt; channel 1 &gt; channel 2 &gt; channel 3 (Channel 0 has the highest priority.)</li> </ul>
Transfer data	1 data unit	Bit length: 8 or 16 bits	Bit length: 8, 16, or 32 bits
	Block size	1 to 255 bytes	Number of data units: 1 to 1,024
Transfer mode	<ul style="list-style-type: none"> <li>Normal transfer mode</li> <li>Repeat transfer mode*<sup>1</sup></li> <li>Block transfer mode</li> <li>I/O mode*<sup>1</sup></li> <li>Idle mode*<sup>1</sup></li> </ul>		<ul style="list-style-type: none"> <li>Normal transfer mode</li> <li>Repeat transfer mode</li> <li>Block transfer mode</li> </ul>

**Table 2.9 Differences in the DMAC Specifications Between the H8/3048 and RX261 Devices (2/2)**

Item	H8/3048	RX261
	DMAC	DMACA
DMA interrupt request	<ul style="list-style-type: none"> <li>• Issued when as many data units as specified by the transfer counter are transferred.</li> </ul>	<ul style="list-style-type: none"> <li>• Transfer end interrupt                             <ul style="list-style-type: none"> <li>— Generated when as many data units as specified by the transfer counter are transferred.</li> </ul> </li> <li>• Transfer escape end interrupt                             <ul style="list-style-type: none"> <li>— Generated when as much data as the repeat size is transferred.</li> <li>— Generated when the extended repeat area overflows.</li> </ul> </li> </ul>
Low power consumption	The modules can be placed in standby status.	The modules can be placed in stopped state.
Other items	—	<ul style="list-style-type: none"> <li>• Extended repeat area function</li> <li>• Event link function</li> </ul>

Note: 1. This is a transfer mode available in short address mode.

### 2.5.2 DMAC Block Diagram

The following figure shows the block diagram for the DMAC of the H8/3048 device.

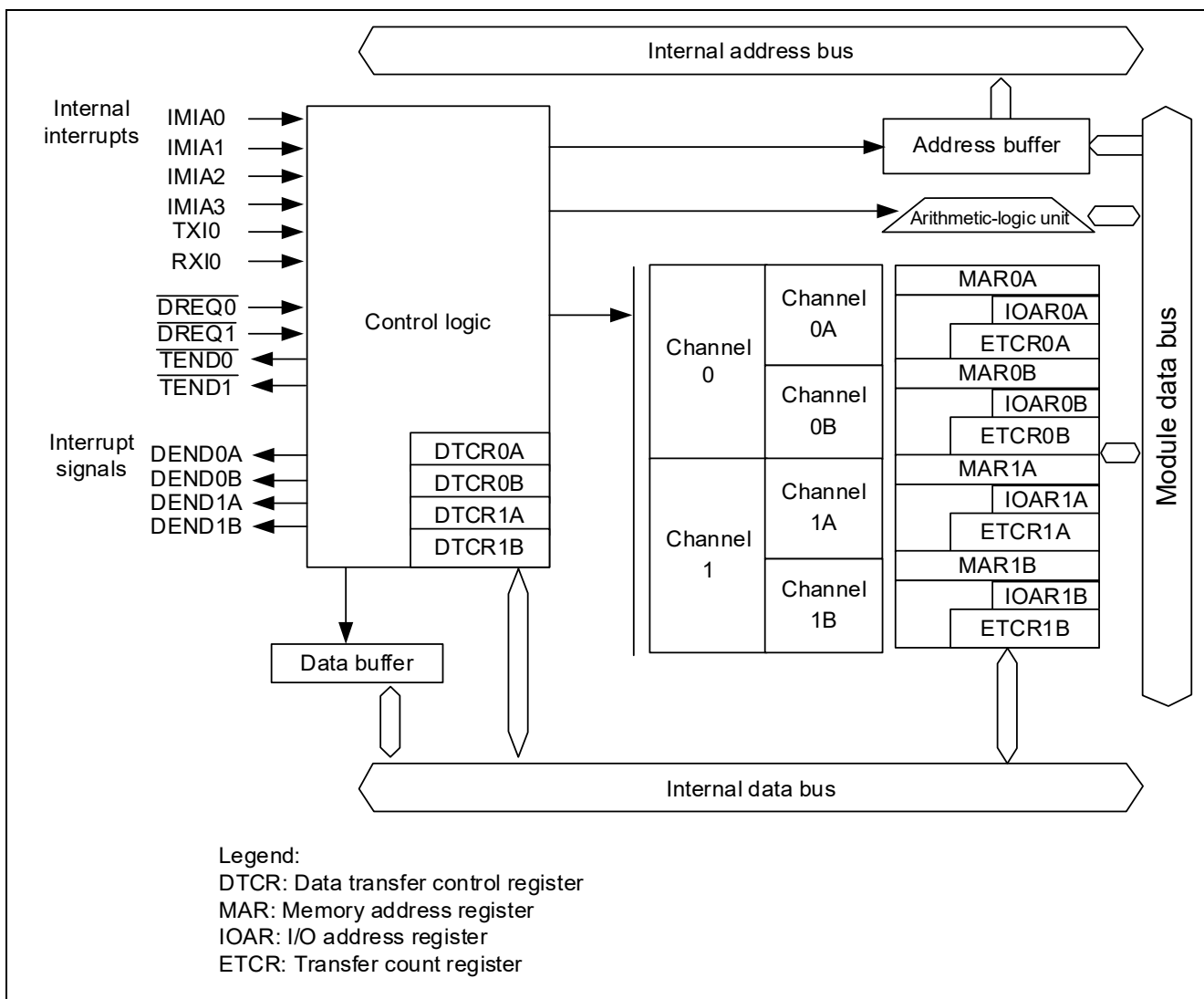


Figure 2.6 Block Diagram for the DMAC of the H8/3048 Device

The following figure shows the block diagram for the DMACA of the RX261 device.

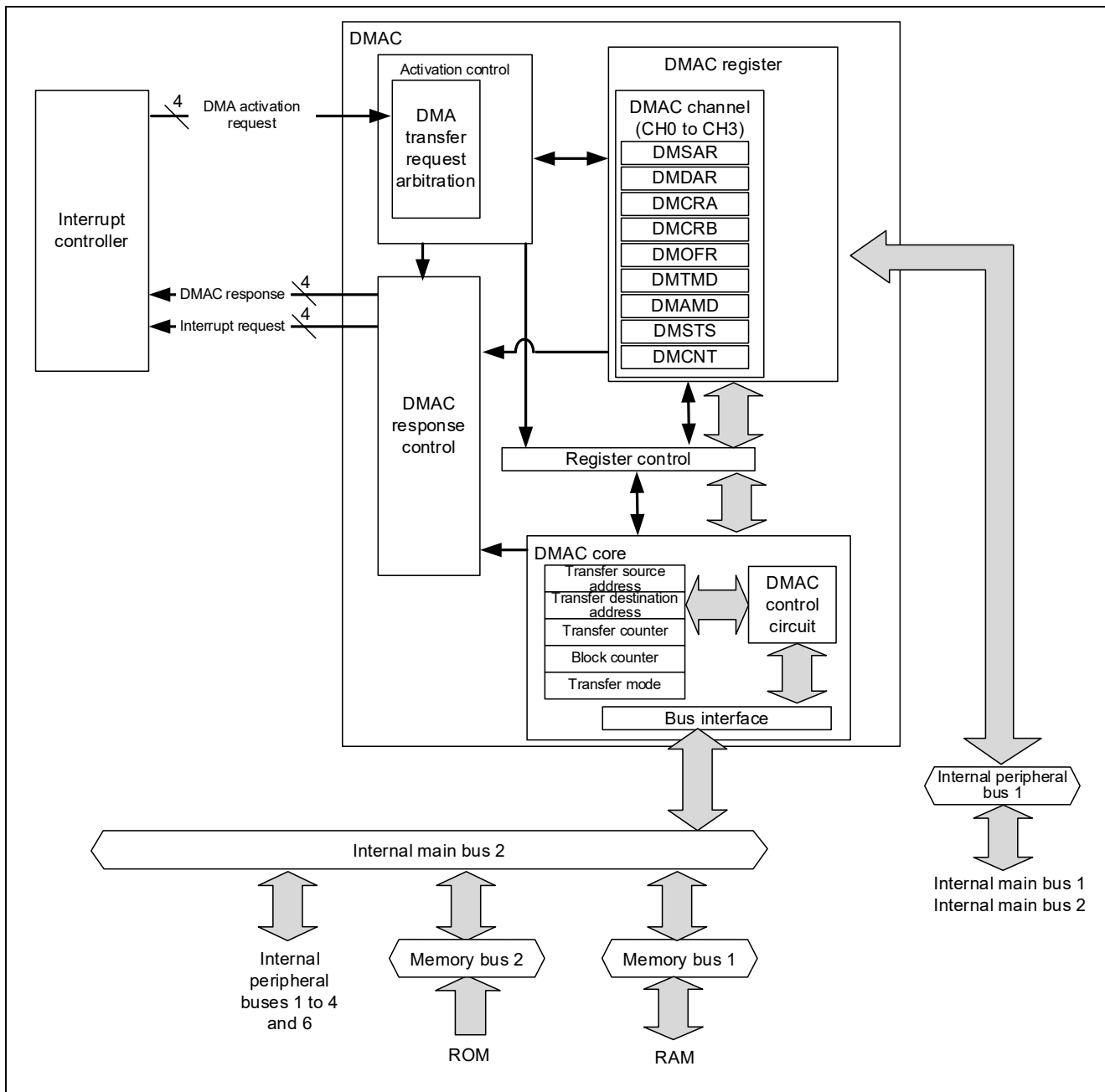


Figure 2.7 Block Diagram for the DMACA of the RX261 Device

### 2.5.3 List of Registers

The following table lists the registers for the DMA controllers of the H8/3048 and RX261 devices.

**Table 2.10 List of Registers for the DMA Controller**

H8/3048	RX261
<b>DMAC</b>	<b>DMACA</b>
Memory address register	DMA module start register (DMAST)
Short address mode: MAR <sub>k</sub> m	DMA transfer source register n (DMACn.DMSAR)
Full address mode: MAR <sub>k</sub> m and MAR <sub>k</sub> m	DMA transfer destination register n (DMACn.DMDAR)
I/O address register	DMA transfer count register n (DMACn.DMCRA)
Short address mode: IOAR <sub>k</sub> m	DMA block transfer count register n (DMACn.DMCRB)
Full address mode: None	DMA transfer mode register n (DMACn.DMTMD)
Transfer count register	DMA interrupt setting register n (DMACn.DMINT)
Short address mode: ETCR <sub>k</sub> m	DMA address mode register n (DMACn.DMAMD)
Full address mode: ETCR <sub>k</sub> m and ETCR <sub>k</sub> m	DMA offset register (DMACn.DMOFR)
Data transfer control register	DMA transfer enable register n (DMACn.DMCNT)
Short address mode: DTCR <sub>k</sub> m	DMA software start register n (DMACn.DMREQ)
Full address mode: DTCR <sub>k</sub> m and DTCR <sub>k</sub> m	DMA status register n (DMACn.DMSTS)
—	DMA activation source flag control register n (DMACn.DMCSL)

Note: k = 0 or 1, m = A or B, n = 0 to 3

### 2.5.4 DMAC Activation Sources

Table 2.11 compares the DMA controller activation sources of the H8/3048 and RX261 devices.

**Table 2.11 Differences of DMA Controller Activation Sources Between the Devices**

Activation Sources				H8/3048 DMAC				RX261 DMACA	
				Short address mode		Full address mode		All transfer modes	
				Channel 0A, 1A	Channel 0B, 1B	Normal	Block		
Internal interrupt	High-function timer	ITU compare match/ input capture interrupt (H8/3048)	IMIA0 to IMIA3	○		×	○	/	
		GPTW compare match/input capture interrupt(RX261)	GTCIn0 to GTCIn7 n= A to F	/	/	/	/		○
		GTCNT over/undwe flow(RX261)	GTCIVn/ GTCIU <sub>n</sub>	/	/	/	/		○
	SCI0		TXI0	○		×		○	
	Transmit data empty interrupt		RXI0	○		×		○	
Receive data full interrupt									
Interrupts from peripheral modules other than the above	On the RX261 device, interrupts other than the above can also be set as DMACA activation sources. These interrupts include IRQ, TXI1 to TXI12, and RXI1 to RXI12.  For details, refer to the interrupt vector table in section 14.3.1 in the hardware manual for the RX261 Group.			×		×		○	
External request	Falling edge of the DREQ pin			×	○	○	○	×*1	
	Low-level input on the DREQ pin			×	○	○	×		
Auto-request				×		○	×	○	

○ : Available as an activation source

× : Unavailable as an activation source

Note: 1. Request signals from an external device cannot be used to initiate transfers directly using the DMACA of the RX261 device. To use a request signal from an external device to active the DMAC, set IRQ as the DMAC activation source and input the request signal to the IRQ pin.

- Activation source setting method:
  - On the H8/3048 device, set the DMAC activation source in the DTCR register.
  - On the RX261 device, set the DMACA activation source in DMRSRm\*1 for the interrupt controller.

Note: 1. m: DMACA channel number (0 to 3)

## 2.5.5 Transfer Modes

Table 2.12 shows the transfer mode specifications of the DMA controllers of the H8/3048 and RX261 devices.

**Table 2.12 Transfer Mode Specifications of the DMA Controller**

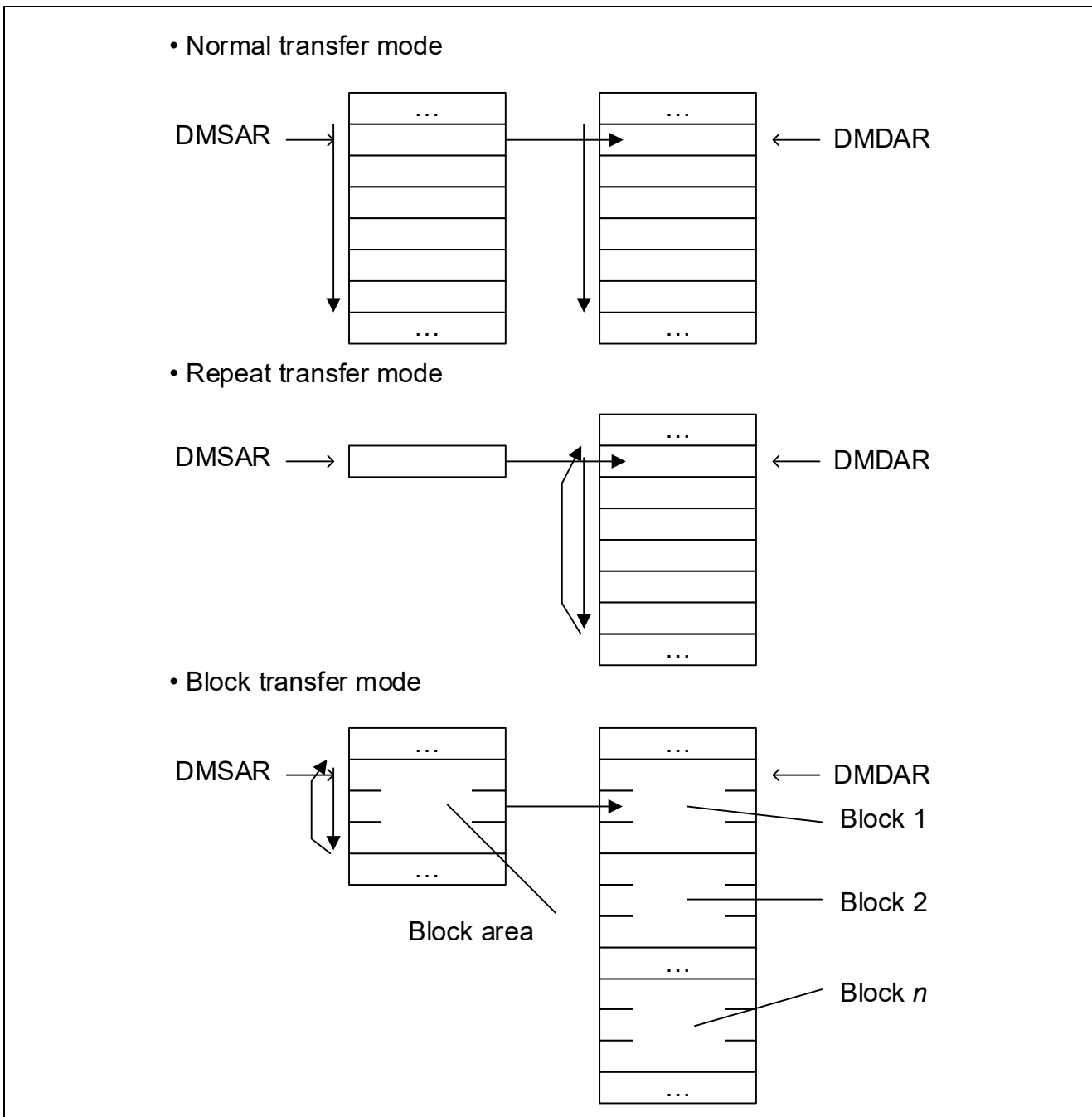
	Transfer Mode		Transfer Size	Transfer Count	How the Transfer Source and Destination Addresses Are Updated after a Requested Transfer Is Complete
H8/3048 DMAC	Short address mode	I/O mode	1 byte 1 word	65,536	<ul style="list-style-type: none"> <li>IOAR registers: Non-change</li> <li>MAR registers: Increment, decrement, or non-change can be selected.</li> </ul>
		Idle mode		65,536	IOAR and MAR registers: Non-change
		Repeat mode		256	<ul style="list-style-type: none"> <li>IOAR registers: Non-change</li> <li>MAR registers: Increment, decrement, or non-change can be selected.</li> </ul>
	Full address mode	Normal mode		65,536	<ul style="list-style-type: none"> <li>MARA and MARB registers: Increment, decrement, or non-change can be selected.</li> </ul>
		Block transfer mode (Maximum number of data units per block: 255)		65,536	
	RX261 DMACA	Normal transfer mode		1 byte 1 word 1 longword	65,535
Repeat transfer mode (Maximum repeat size: 1024 data units)		1,024			
Block transfer mode (Maximum number of data units per block: 1024)		1,024			

Note: Supplementary information regarding the repeat mode on the H8/3048 device and the repeat transfer mode on the RX261 device:

In repeat mode on the H8/3048 device, the data in the range set by the transfer count is transferred repeatedly until the CPU clears the DTE bit to 0.

In repeat transfer mode on the RX261 device, a DMA transfer ends when data of the repeat size is transferred the number of times specified by the repeat count. The DMA transfer can be resumed by writing "1" to the DMACm.DMCNT.DTE bit in the handler of the repeat size end interrupt.

Figure 2.8 shows an overview of the DMACA operation on the RX261 device in each transfer mode.



**Figure 2.8 Example Operations of the RX261 Device in Transfer Modes**

- Normal transfer mode**  
 In normal transfer mode, a single unit of data is transferred per transfer request. This transfer can be performed a maximum of 65,535 times. The address update (increment, decrement, non-change, or increment-by-offset) performed after a transfer can be controlled using DMSAR and DMDAR registers separately. In the operation example shown in Figure 2.8, the values of the DMSAR and DMDAR registers are incremented.

- Repeat transfer mode

In repeat transfer mode, a single unit of data is transferred per transfer request. In this mode, a maximum of 1,024 can be set for the repeat size and repeat count. The address set in the address register (DMSAR or DMDAR) assigned for the repeat area is reset to the initial address when data of the repeat size is transferred. When this processing is repeated the number of times specified by the repeat count, a DMA transfer ends. The DMA transfer can be resumed by writing “1” to the DMACm.DMCNT.DTE bit in the handler of the repeat size end interrupt. In the operation example shown in Figure 2.8, destination address register is assigned for the repeat area.

- Block transfer mode

In block transfer mode, a single block (a maximum of 1 KB) of data is transferred per transfer request. This transfer can be performed a maximum of 1,024 times. The block size is set in the DMCRA register. The DTS bit of the MRB register is used to set whether the source address register or destination address register is assigned for the block area. In the operation example in Figure 2.8, the source address register is assigned for the block area.

### 2.5.6 Module Stop

On the RX261 device, after recovery from a reset, modules other than the DMAC, DTC, and RAM are placed in stopped state (clock supply is stopped) to reduce power consumption. The DMAC can also be stopped for further reduction of power consumption by using the module stop control register. In this case, however, the DTC module is also stopped.

## 2.6 Serial Communications Interface

### 2.6.1 Comparison of Specifications

Whereas the H8/3048 device provides only one serial communications interface (SCI), the RX261 device provides two: SCIk/SCIh and RSCI. SCIk supports the smartcard (IC card) interface as an extended function of the asynchronous mode, in addition to conventional transfer modes, asynchronous mode and clock-synchronous mode. SCIk also supports the simplified I<sup>2</sup>C bus interface (single master operation) and simplified SPI bus interface. SCIh has all functions of SCIk and extended serial interface functions.

RSCI support communication with Manchester code, expansion serial communication. For details on the transfer modes not available on the H8/3048 device, refer to the “RX260 Group, RX261 Group User's Manual: Hardware”.

**Table 2.13 Differences in the Serial Communications Interface Between the Devices**

Item	H8/3048	RX261
Number of channels	2 (SCI0 and SCI1)	7ch (SCIk : SCI1,5,6, SCIh : SCI12, RSCI : RSCI0,8,9)
Serial communication modes	Asynchronous Clock-synchronous Smartcard interface (SCI0 only)	Asynchronous Clock-synchronous Smartcard interface Simplified I <sup>2</sup> C bus Simplified SPI bus Manchester code(only RSCI9)
Transfer rate	The desired bit rate can be selected using the on-chip baud rate generator.	
Full duplex communication	Transmit block: A dual-buffer configuration allows continuous transmission. Receive block: A dual-buffer configuration allows continuous reception.	
Data transfer	Only LSB-first is supported.	LSB-first or MSB-first can be selected. (The simplified I <sup>2</sup> C bus supports only MSB-first.)
Interrupt source	Transmit data empty Transmit end Receive data full Receive error	SCIk/SCIh: Transmit end Transmit data empty Receive data full Receive error Data match Completion of generation of a start condition Restart condition, or stop condition (for simple I <sup>2</sup> C mode) RSCI: Transmit data empty Transmit end Receive data full Receive error Receive data ready Receive data match Break Field detection/transmission Bus collision detection Active edge detection Completion of generation of a start condition, restart condition, or stop condition

Low power consumption		The module standby state can be set for each channel.	The module stop state can be set for each channel.
Asynchronous mode	Data length	7 or 8 bits	7, 8, or 9 bits
	Stop bit	1 or 2 bits	
	Parity	Even, odd, or none	
	Receive error detection	Parity, overrun, and framing errors	
	Hardware flow control	Not implemented	Implemented (controllable using the CTSn and RTSn pins)
	Start bit detection	Falling edge	Low level or falling edge can be selected.
	Break detection	A break can be detected by directly reading the level of the RxDn pin when a framing error occurs.	SCIk,SCIh: A break can be detected by directly reading the level of the RxDn pin when a framing error occurs or reading the SPTR.RXDMON flag.  RSCI: A break can be detected by directly reading the level of the RxDn pin when a framing error occurs.
Asynchronous mode	Clock source	The internal or external clock can be selected.	The internal or external clock can be selected. The transfer rate clock can be input from TMR (SCI5 and SIC6).
	Double-speed mode	Not implemented	The baud rate generator double-speed mode can be selected.
	Multiprocessor communication	Implemented	
	Digital noise filter	Not implemented	Internally implemented on the RXDn pin input route
Clock synchronous mode	Data length	8 bits	
	Receive error detection	Overrun errors	
	Hardware flow control	Not implemented	Implemented (controllable using the CTSn and RTSn pins)
Smartcard interface		Implemented	
Simplified I2C mode		Not implemented	Implemented
Simplified SPI mode		Not implemented	Implemented
Extended serial mode		Not implemented	Implemented SCI12 and RSCI9
Event link function		Not implemented	Available with SCI5 only.

The following table compares the on-chip SCI registers of the devices and shows the differences in these registers.

**Table 2.14 List of SCI Registers (1/2)**

H8/3048	RX261		Difference
	SCIk,SCIh	RSCI	
Transmit data register (TDR)	Transmit data register (TDR)		None
Transmit shift register (TSR)	Transmit shift register (TSR)		
Receive data register (RDR)	Receive data register (RDR)		
Receive shift register (RSR)	Receive shift register (RSR)		
Serial mode register (SMR)	Serial mode register (SMR)	- (Supported SCR2)	
Serial control register (SCR)	Serial control register (SCR)	Control register (Supported SCR0 to 4)	
Serial status register (SSR)	Serial status register (SSR)		
Bit rate register (BRR)	Bit rate register (BRR)	- (Supported SCR2 and SCR3)	
Smart card mode register (SCMR)	Smart card mode register (SCMR)	- (Supported SCR3)	

**Table 2.14 List of SCI Registers (2/2)**

H8/3048	RX261	Difference
— (There is no corresponding register.)	Control Field 0 compare enable register (CF0CR)	New register added on the RX261
	Control Field 0 data register(CF0DR)	
	Control Field 0 receive data register (CF0RR)	
	Control Field 1 compare enable register(CF1CR)	
	Control Field 1 receive data register (CF1RR)	
	I2C status register(SISR)	
	I2C mode register1/2/3(SIMR1/2/3)	
	SPI mode register(SPMR)	
	Control register 0/1/2/3(CR0/1/2/3)	
	Serial port register(SPTR)	
	Serial extended mode register (SEMR)	
	Status clear register (STCR)	
	Status register(STR)	
	Secondary control Field 1 data register (SCF1DR)	
	Timer count register (TCNT)	
	Timer control register(TCR)	
	Timer prescaler register (TPRE)	
	Timer mode register(TMR)	
	Data comparison control register (DCCR)	
	Transmit data register H(TDRH)	
	Transmit data register HL(TDRHL)	
	Transmit data register L(TDRL)	
	Noize filter setting register(SNFR)	
	Primary Control Field 1 data register (PCF1DR)	
	Port control register (PCR)	
	Modulation duty register(MDDR)	
	Receive data register H(RDRH)	
	Receive data register HL(RDRHL)	
	Receive data register L(RDRL)	
	Extended serial module enable register (ESMER)	
	Interrupt control register (ICR)	
	Transmit/Receive timing delect register (TMGR)	
	Comparison data register (CDR)	
	Comparison data register H(CDR.H)	
	Comparison data register L(CDR.L)	
	DE signal control register (DECR)	
	HBS support mode control register (HBSCR)	
	I2C status clear register (SISCR)	
	I2C status register (SISR)	
	I2C mode register(SIMR)	
Status clear register(SSCR)		

	Manchester mode status clear register (MMSCR)	
	Manchester mode status register(MMSR)	
	Manchester mode control register(MMCR)	
	Extended serial mode status clear register (XSCR)	
	Extended serial mode status register0/1(XSR0/1)	
	Extended serial mode control register 0/1/2(XCR0/1/2)	

### 2.6.2 SCI-related Settings Required During Migration

For the processing that performed serial communication in asynchronous mode or clock-synchronous mode by using the SCI on the H8/3048 device, the following settings must be configured on the RX261 device:

- Determining the one-bit period and selecting the clock source  
To perform communication in asynchronous mode, use the serial extended mode register (SEMR) to select the external clock input or TM clock input (channel SCI5, SCI6, or SCI12 only) as the clock source for determining the one-bit period. Also select 8 or 16 as the number of base clock cycles per one-bit period.
- Digital noise filter  
Whether to enable or disable the digital noise filter is set by using the serial extended mode register (SEMR). If the noise filter is enabled, use the noise filter setting register (SNFR) to select the clock for the noise filter.

### 2.6.3 Module Stop

On the RX261 device, after recovery from a reset, modules other than the DMAC, DTC, and RAM are placed in stopped state (clock supply is stopped) to reduce power consumption. Before configuring the modules, make sure that they are not in stopped state.

## 2.7 High-Function Timers (ITU, GPTWa)

### 2.7.1 Comparison of Specifications

The comparison of specifications between the RX261 and the H8/3048 is shown.

**Table 2.15 Comparison of Specifications (Functions) of the ITU on the H8/3048 Device, and the GPTWa on the RX261 Device**

Item		H8/3048	RX261
		ITU	GPTWa
Input clock	Internal clock	Internal clock: $\phi$ , $\phi/2$ , $\phi/4$ , and $\phi/8$	PCLKA, PCLKA/2, PCLKA/4, PCLKA/8, PCLKA/16, PCLKA/32, PCLKA/64, PCLKA/128, PCLKA/256, PCLKA/512, PCLKA/1024,
	External clock	TCLKA, TCLKB, TCLKC, and TCLKD	GTETRGA, GTETRGB, GTETRGC, GTETRGD
Number of channels		5 (channels 0 to 4)	8 (GPTW0 to 7)
Buffer register		Supported on channels 3 and 4 only	Supported on all channel
Counter clearing source		Compare match and input capture	Compare match for the GTPR register, input capture, ELC input, the state of input pins, external trigger input, GTCCR register compare match, and Other channel's counter clear sources
Compare match output		0, 1, or Toggle (Toggle is not supported on channel 2.)	0, 1, or Toggle
Synchronization		Channels 0 to 4	Supported on all channel
PWM mode 1		Channels 0 to 4	-
Sawtooth-wave PWM mode 1,2		-	Mode1:Supported on all channel Mode2:Supported on GPTW0 to 2
Complementary PWM mode		Channels 3 and 4	-
Complementary PWM mode 1/2/3/4		-	Supported on GPTW0 to 2
Reset-synchronized PWM mode		Channels 3 and 4	Supported by Complementary PWM mode 1/2/3/4 on GPTW0 to 2
Phase counting mode		Channel 2	Supported on GPTW3 to 5
DMAC activation		Channels 0 to 3	Supported on all channel
Number of interrupt sources		15	8
Low power consumption		The modules can be placed in standby state.	The modules can be placed in standby state.
AC synchronous motor drive mode		—	Supported on all channel
DTC activation		—	Supported on all channel
A/D conversion start trigger		—	Supported on GPTW0 to 2
Event link function (output)		—	11 sources Supported on all channel
Event link function (input)		—	Supported on all channel Countstart

		Count start/stop/clear Up count/down count Input Capture A, B Operation
A/D Conversion Start Request Delayed Function	—	Supported on all channel
Interrupt Skipping Function	—	Supported on GPTW0 to 2

**Table 2.16 Comparison of Specifications (Pins) of the ITU on the H8/3048 Device, and the MTU2a and TPUa on the RX261 Device**

Item	H8/3048	RX261
	ITU	GPTWa
I/O pins	10 pins	14 pins
Output pins	2 pins (channel 4)	7 pins
Input pins	—	4 pins

### 2.7.2 Notes on Replacing the Timer

GPTWa has synchronus operation by software and hardware.

Those operaton different ITU.write any data to a GTCNT, does not write any data to other GTCNT at same time.

See RX260,RX261 user's manual hardware for more detail.

### 2.7.3 List of Registers

**Table 2.17 List of Registers for the ITU, GPTWa**

H8/3048	RX261
ITU	GPTWa
TSTR	GTSTR GTSTP
TSNC	—
TMDR	GTCR
TFCR	GTCR
TOER	GTIOR
TCNT0 to TCNT4	GTCNT
GRA0 to GRA4 and GRB0 to GRB4	GTCCRm(m= A to F)
BRA3, BRA4, BRB3, and BRB4	GTCCRm(m= C to F)
TOCR	GTIOR
TCR0 to TCR4	GTCR GTCSR
TIOR0 to TIOR4	GTIOR
TSR0 to TSR4	GTST
TIER0 to TIER4	GTST
—	GTWP
	GTCLR
	GTSSR
	GTPSR
	GTUPSR
	GTDNSR
	GTICASR
	GTICBSR
	GTUDDTYC
	GTBER
	GTITC
	GTPR
	GTPBR
	GTPDBR
	GTADTRA

	GTADTBRA
	GTADTDBRA
	GTADTRB
	GTADTBRB
	GTADTDBRB
	GTDTCR
	GTDVU
	GTSECR
	GTICCR
	OPSCR

#### 2.7.4 Module Stop

On the RX261 device, after recovery from a reset, modules other than the DMAC, DTC, and RAM are placed in stopped state (clock supply is stopped) to reduce power consumption. Before configuring the modules, make sure that they are not in stopped state.

## 2.8 Timer-based Pulse Output Function

### 2.8.1 Comparison of Specifications

The RX261 device does not have a module equivalent to the programmable timing pattern controller (TPC) on the H8/3048 device. This section shows a method for achieving the same functionality as the TPC on the RX261 device. Note that the RX651 or RX65N device has a programmable pulse generator (PPG), which can substitute for the TPC on the H8/3048 device. For details, refer to the “RX65N Group, RX651 Group User’s Manual: Hardware”.

#### (1) Overview of the TPC on the H8/3048 Device

The basic function of the TPC is to output pulses from pins TP0 to TP15 by using a 16-bit integrated timer unit (ITU) as the time base. The 16-bit TPC consists of four 4-bit output groups (0 to 3). These groups can operate concurrently or independently.

Figure 2.9 shows the block diagram of the TPC.

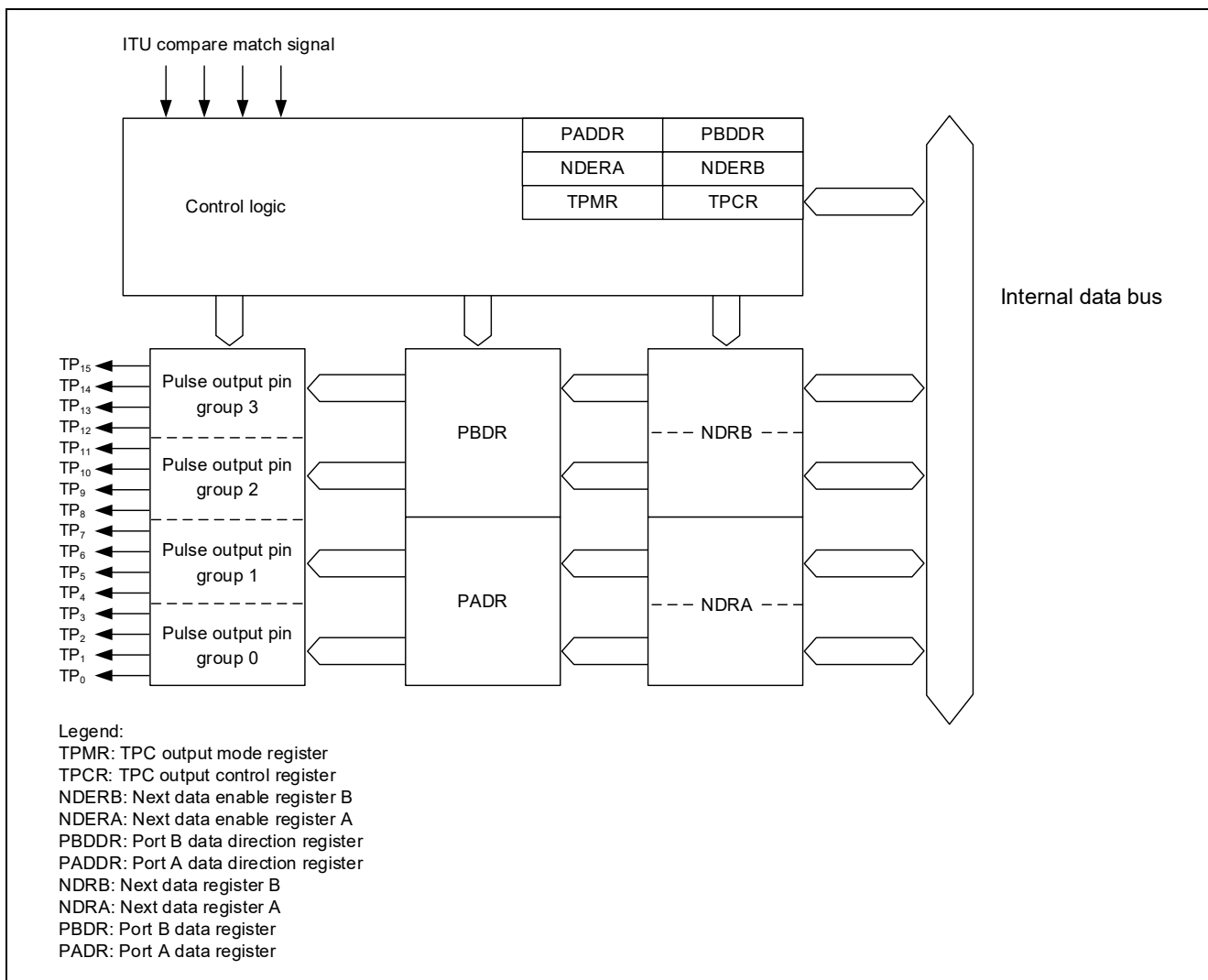
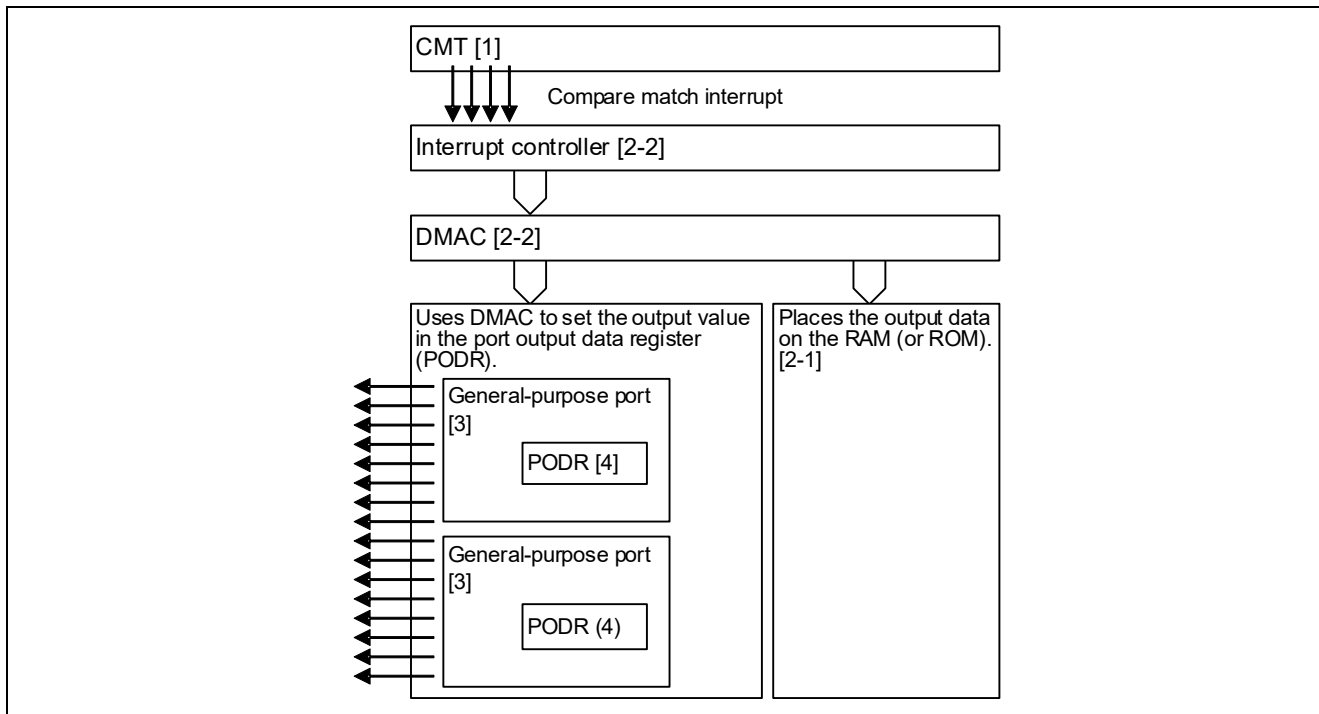


Figure 2.9 Block Diagram of the TPC on the H8/3048 Device

**(2) Example settings for achieving a TPC-equivalent function on the RX261 device**

Figure 2.10 shows an example of the settings for achieving a TPC-equivalent function on the RX261 device. In this example, general-purpose ports, the DMAC, and the CMT are used to achieve the function.



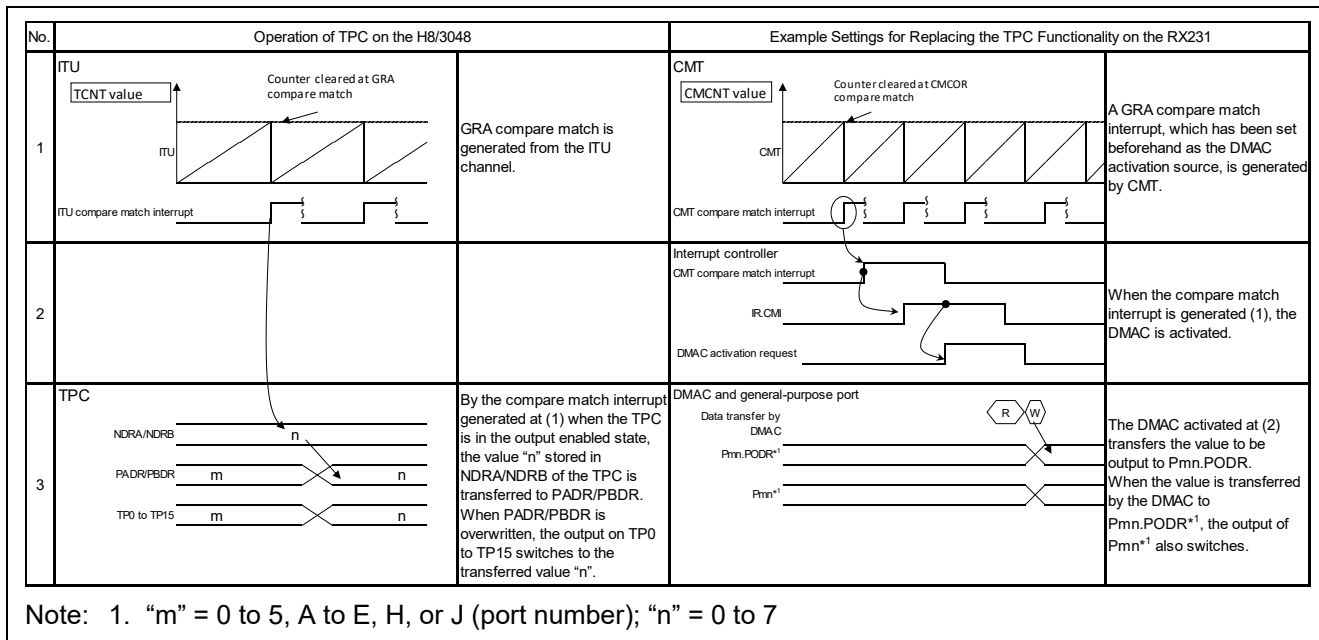
**Figure 2.10 Block Diagram of a TPC-Equivalent Function on the RX261 Device**

Description of the substitutive modules

- [1] Substitute for the TPC time base on the H8/3048 device  
The CMT is used as the time base and a compare match interrupt of the CMT is used as a trigger to activate the DMAC.
- [2] Substitute for the TPC NDRA/NDRB on the H8/3048 device  
[2-1]: A variable that stores the output pattern is prepared in the memory.  
[2-2]: Upon receiving the compare match, the interrupt controller activates the DMAC.  
The DMAC transfers the content of the variable (output pattern) to Pmn.PODR.
- [3] Substitute for the TPC output pins (TP0 to TP15) on the H8/3048 device  
Pmn is assigned for general-purpose output so that it can be used for pulse output.
- [4] Substitute for TPC PADR/PBDR on the H8/3048 device  
Use Pmn.PODR as the substitute. Output data is stored to Pmn.PODR and is output from Pmn.  
(m = 0 to 9, A to F, H, J to L; n = 0 to 7)

**(3) Description of the operation of the example TPC-equivalent function**

Figure 2.11 describes the operation of the TPC on the H8/3048 device and the operation of the example TPC-equivalent function on the RX261 device.



Note: 1. "m" = 0 to 5, A to E, H, or J (port number); "n" = 0 to 7

**Figure 2.11 Description of the Operation of the Example TPC-Equivalent Function**

## 2.9 Watchdog Timers

### 2.9.1 Comparison of Specifications

The H8/3048 device has a watchdog timer module called “WDT”. The RX261 device has two watchdog timer modules, WDTA and IWDTA. IWDTA operates based on an independent dedicated clock. Table 2.18 compares the specifications of these modules.

**Table 2.18 Comparison of the Specifications of the Watchdog Timer Modules on the H8/3048 and RX261 Devices (WDT, WDTA, and IWDTA)**

Item	H8/3048	RX261	
	WDT	WDTA	IWDTA
Count source	System clock	Peripheral module clock B (PCLKB)	IWDTCLK (IWDT-dedicated clock)
Clock division ratio	$\phi/2$ , $\phi/32$ , $\phi/64$ , $\phi/128$ , $\phi/256$ , $\phi/512$ , $\phi/2048$ , or $\phi/4096$	$1/4$ , $1/64$ , $1/128$ , $1/512$ , $1/2048$ , or $1/8192$	$1/1$ , $1/16$ , $1/32$ , $1/64$ , $1/128$ , or $1/256$
Behavior of counting	Up-counting using an 8-bit up-counter	Down-counting using a 14-bit down-counter	
Interrupt source	<ul style="list-style-type: none"> <li>Overflow of the up-counter</li> </ul>	<ul style="list-style-type: none"> <li>Underflow of the down-counter</li> <li>Refresh performed outside the period in which a refresh is permitted (refresh error)</li> </ul>	
Behavior on underflow	—	Reset output or NMI interrupt	
Other items	<ul style="list-style-type: none"> <li>Use as an interval timer is possible.</li> </ul>	<ul style="list-style-type: none"> <li>Use as an interval timer is not possible.</li> <li>Window function</li> <li>Auto-start mode</li> <li>Register start mode</li> </ul>	

## 2.10 A/D Converter

### 2.10.1 Comparison of Specifications

Table 2.19 compares the specifications of the A/D converters on the H8/3048 and RX261 device.

**Table 2.19 Comparison of Specifications of the A/D Converters on the H8/3048 and RX261 Devices**

Item	H8/3048	RX261
	A/D converter	12-bit A/D converter (S12ADE)
Resolution	10 bits	12 bits
A/D conversion method	Successive approximation	Successive approximation
Conversion time	7.45 $\mu$ s (min.) per channel (during operation at 18 MHz)	0.7 $\mu$ s per channel(ADCCR.CCS bit=0) 0.5 $\mu$ s per channel(ADCCR.CCS bit=1) (ADCLK is 64 MHz)
Operating mode	Single mode	Single-scan mode
	Scan mode	Continuous-scan mode
	—	Group-scan mode with group-A priority control enabled or disabled
A/D conversion start condition	<ul style="list-style-type: none"> <li>Software</li> <li>External trigger signal (<math>\_ADTRG</math>)</li> </ul>	<ul style="list-style-type: none"> <li>Software trigger</li> <li>Synchronous trigger (trigger signal from MTU, ELC, or TPU)</li> <li>Asynchronous trigger (ADTRG0# pin)</li> </ul>
Other functions	Sample-and-hold function	<ul style="list-style-type: none"> <li>Sample-and-hold function</li> <li>Variable number of sampling states</li> <li>Self-diagnosis for 12-bit A/D converter</li> <li>Addition mode or averaging mode can be selected for A/D conversion values</li> <li>Analog input disconnection detection assist function</li> <li>Double trigger mode</li> <li>A/D data register automatic clearing</li> <li>Compare (windows A and B)</li> <li>16 ring buffers available when the compare function is in use</li> <li>Event link function</li> </ul>
Interrupt source	(A/D conversion end interrupt request)	<ul style="list-style-type: none"> <li>The scan end interrupt request (S12ADI0) is generated when a scan ends in modes other than double trigger mode and group-scan mode.</li> <li>In double trigger mode, the scan end interrupt request (S12ADI0) is generated when two scans end.</li> <li>In group-scan mode, the scan end interrupt request (S12ADI0) is generated when a scan for group A ends. When a scan for group B ends, GBADI, which is the scan end interrupt request dedicated to group B, is generated.</li> <li>In group-scan mode with double trigger mode selected, the scan end interrupt request (S12ADI0) is generated when two scans for group A end. When a scan for group B ends, GBADI, which is the scan end interrupt request dedicated to group B, is generated.</li> </ul>
Low power consumption	The modules can be placed in standby state.	The modules can be placed in stopped state.
Conversion target	Analog input pins 0 to 7	Analog input pins 0 to 7 and 16 to 31 Internal reference voltage Temperature sensor

## 2.10.2 Operating Modes

The A/D converter on the H8/3048 device can operate in two modes: single mode and scan mode. Table 2. shows the correspondence between the conversion modes of the H8/3048 and RX261 devices.

**Table 2.20 Correspondence of A/D Converter Operating Modes**

No.	H8/3048	RX261
1	Single mode	Single-scan mode (conversion on only one channel)
2	Scan mode	Continuous-scan mode
3	—	Group-scan mode

The following table provides an overview of the operating modes.

**Table 2.20 Overview of the A/D Converter Operating Modes**

Microcontroller	Operating Mode	Operational Overview
H8/3048	Single mode	(1) A/D conversion is performed once on the single specified channel only. (2) If interrupts are enabled, an ADI interrupt is generated. (3) The value of the ADST bit in ADCSR is held at 1 (A/D converter started) while A/D conversion is in progress and automatically cleared to 0 when A/D conversion finishes, ending A/D converter operation.
	Scan mode	(1) Conversion is performed successively on analog input from the specified channels (or channel), starting from the lowest numbered channel. (2) When conversion of all the specified channels finishes, an ADI interrupt is generated. (3) Steps 1 and 2 are repeated for as long as the ADST bit in ADCSR remains set to 1 (A/D converter started). A/D conversion ends when the ADST bit is cleared to 0.
RX261	Single-scan mode	(1) Conversion is performed successively on analog input from the specified channels (or channel), starting from the lowest numbered channel. (2) When conversion of all the specified channels finishes, an S12ADI0 interrupt is generated. (3) The value of the ADST bit in ADCSR is held at 1 (start of A/D conversion) while A/D conversion is in progress, and is automatically cleared to 0 when A/D conversion of the selected channels is completed.
	Continuous-scan mode	The operation in single-scan mode shown above is repeated multiple times.
	Group-scan mode	(1) Scanning of group A and group B starts when the selected trigger corresponding to the group occurs, the selected channels in group A and group B are scanned once each, and operation ends. (2) If interrupts are enabled for group A, an S12ADI0 interrupt is generated when scanning of group A finishes. If interrupts are enabled for group B, a GBADI interrupt is generated when scanning of group B finishes.

### 2.10.3 List of Registers

Table 2.21 compares the functions of the registers for the A/D converter on the H8/3048 device and the 12-bit A/D converter on the RX261 device.

**Table 2.21 Comparison of the Functions of the Registers for the AD Converters (1/2)**

H8/3048	RX261	Difference
<b>A/D converter</b>	<b>12-bit A/D converter</b>	
A/D data registers A to D (ADDRA to ADDR D)	A/D data register y (ADDRy) (y = 0 to 7 or 16 to 31)	Present*1
A/D control/status register (ADCSR) A/D control register (ADCR)	A/D control register (ADCSR) A/D channel select register A0 (ADANSA0) A/D channel select register A1 (ADANSA1) A/D channel select register B0 (ADANSB0) A/D channel select register B1 (ADANSB1) A/D sampling state register n (ADSSTRn) (n = 0 to 7, L, T, or O) A/D conversion start trigger select register (ADSTRGR)	Present
— (There is no corresponding register.)	A/D data duplication register (ADDBLDR) A/D temperature sensor data register (ADTSDR) A/D internal reference voltage data register (ADOCDR) A/D self-diagnosis data register (ADRD) A/D-converted value addition/averaging function channel select register n (ADADSn) (n = 0 or 1) A/D-converted value addition/averaging count select register (ADADC) A/D control extended register (ADCER) A/D conversion extended input control register (ADEXICR) A/D disconnection detection control register (ADDISCR) A/D sampling state register n (ADSSTRn) (n = 0 to 7, L, T, or O) A/D disconnection detection control register (ADDISCR) A/D event linkage control register (ADELCCR) A/D group-scan priority control register (ADGSPCR) A/D compare function control register (ADCMPCR) A/D compare function window-A channel select register n (ADCMPANSRn) (n = 0 or 1) A/D compare function window-A extended input select register (ADCMPANSER) A/D compare function window-A comparison condition setting register n (ADCMPPLRn) (n = 0 or 1)	New register added on the RX261

**Table 2.21 Comparison of the Functions of the Registers for the AD Converters (2/2)**

H8/3048	RX261	
A/D converter	12-bit A/D converter	Difference
— (There is no corresponding register.)	A/D compare function window-A extended input comparison condition setting register (ADCMPLER)	New register added on the RX261
	A/D compare function window-A lower-bit level setting register (ADCMPDR0)	
	A/D compare function window-A upper-bit level setting register (ADCMPDR1)	
	A/D compare function window-A channel status register n (ADCMPSRn) (n = 0 or 1)	
	A/D compare function window-A extended input channel status register (ADCMPSER)	
	A/D high-potential/low-potential reference voltage control register (ADHVREFCNT)	
	A/D compare function window-A/B status monitor register (ADWINMON)	
	A/D compare function window-B channel select register (ADCMPBNSR)	
	A/D compare function window-B lower-bit level setting register (ADWINLLB)	
	A/D compare function window-B upper-bit level setting register (ADWINULB)	
	A/D compare function window-B channel status register (ADCMPBSR)	
	A/D data storage buffer register n (ADBUFn) (n = 0 to 15)	
	A/D data storage buffer enable register (ADBUFEN)	
	A/D data storage buffer pointer register (ADBUFPTR)	

Note: 1. On the H8/3048 and RX261 devices, these registers are 16-bit registers. Note that, on the H8/3048 device, the 10 active bits are left-aligned, whereas on the RX261 device, the 12 active bits are right-aligned in the initial (default) state. (Left-aligned format can be selected by setting the ADRFMT bit in ADCER to 1.)

### 2.10.4 Module Stop

On the RX261 device, after recovery from a reset, modules other than the DMAC, DTC, and RAM are placed in stopped state (clock supply is stopped) to reduce power consumption. Before configuring the modules, make sure that they are not in stopped state.

## 2.11 D/A Converter

### 2.11.1 Comparison of Specifications

**Table 2.22 Comparison of Specifications of the D/A Converters on the H8/3048 and RX261 Devices**

Item	H8/3048	RX261
	D/A converter	8-bit D/A converter (DAa)
Resolution	8 bits	8 bits
Number of input channels	2	2
Conversion speed	10 $\mu$ s (at a load capacitance of 20 pF)	30 $\mu$ s, max.
Output voltage	0V to $255/256 \times V_{REF}$	0V to AVCC0
D/A output hold function in software standby mode	Implemented	Implemented
Low power consumption	—	The modules can be placed in stopped state.*1

Notes: 1. Do not place the 12-bit A/D converter module in stopped state if the DAADST bit in the DAADSCR register is 1 (measure against interference between D/A conversion and A/D conversion is enabled). In this case, stopping the A/D converter may also cause the D/A converter to stop.

### 2.11.2 Comparison of Registers

This section shows the differences in the registers between the H8/3048 and RX261 devices.

**Table 2.23 Comparison of Registers on the H8/3048 and RX261 Devices**

H8/3048	RX261	Difference
D/A data registers 0 and 1 (DADR0 and DADR1)	D/A data registers 0 and 1 (DADR0 and DADR1)	Present*1
D/A control register (DACR)	D/A control register (DACR)	Present*1
D/A standby control register (DASTCR)	—*2	—
—	DADRm format select register (DADPR)	—
	D/A and A/D conversions synchronization start control register (DAADSCR)	—
	D/A VREF control register (DAVREFCR)	—

Notes: 1. Bit assignments are different.

- The RX261 device does not have a register equivalent to the D/A standby control register (DASTCR) of the H8/3048 device.

### 2.11.3 Module Stop

On the RX261 device, after recovery from a reset, modules other than the DMAC, DTC, and RAM are placed in stopped state (clock supply is stopped) to reduce power consumption. Before configuring the modules, make sure that they are not in stopped state.

## 2.12 Flash Memory

### 2.12.1 Comparison of Specifications

**Table 2.24 Comparison of Specifications of Flash Memory on the H8/3048F and RX261 Devices**

Item		H8/3048F	RX261
Size		128 KB	User area: 512 KB, max. Data area: 8 KB Extra area: Size enough to store the startup area information, access window information, and unique IDs
Block size × Number of blocks		<ul style="list-style-type: none"> <li>16 KB × 7 (112 KB)</li> <li>12 KB × 1 (12 KB)</li> <li>512 bytes × 8 (4 KB)</li> </ul>	<ul style="list-style-type: none"> <li>User area: 2 KB × 256 (512 KB)</li> <li>2 KB × 192 (384KB)</li> <li>2 KB × 128 (256KB)</li> <li>Data area: 256B × 32 (8 KB)</li> </ul>
Operating mode		<ul style="list-style-type: none"> <li>Program mode</li> <li>Program-and-verify mode</li> <li>Erase mode</li> <li>Erase-and-verify mode</li> <li>Pre-write-and-verify mode</li> </ul>	<ul style="list-style-type: none"> <li>E2 data flash access prohibition mode</li> <li>ROM/E2 data flash read mode</li> <li>ROM P/E mode</li> <li>E2 data flash P/E mode</li> </ul>
ROM	Unit of data to write/erase	Write: In bytes Erase: In blocks	Write: In 8 bytes Erase: In blocks
	Write/erase time	System clock: 1 to 16 MHz <ul style="list-style-type: none"> <li>Write: 50 μs or 1000 μs (per byte)*1</li> <li>Erase: 1 s or 30 s (per 128 KB (entire block))*1</li> </ul>	FCLK: 64 MHz <ul style="list-style-type: none"> <li>Write: 45.0 μs or 445 μs (per 8 bytes)*1</li> <li>Erase: 65.4 ms or 1494.3 ms (per 512 KB (entire block))*1</li> </ul>
	Maximum rewrite count	100	1000
E2 data flash	Unit of data to write/erase	—	Write: In bytes Erase: In blocks
	Write/erase time	—	FCLK: 64 MHz <ul style="list-style-type: none"> <li>Write: 34.6 μs or 377.7 μs (per byte)*1</li> <li>Erase: 12.7 ms or 375.2 ms (per 8 KB (entire block))*1</li> </ul>
	Maximum rewrite count	—	1,000,000 / 100,000*2
Programming mode		<ul style="list-style-type: none"> <li>On-board programming                             <ul style="list-style-type: none"> <li>Boot mode</li> <li>User program mode</li> </ul> </li> <li>PROM mode</li> </ul>	<ul style="list-style-type: none"> <li>On-board programming                             <ul style="list-style-type: none"> <li>Boot mode                                     <ul style="list-style-type: none"> <li>— FINE interface</li> <li>— USB interface</li> <li>— SCI interface</li> </ul> </li> <li>Self-programming (single-chip mode)</li> </ul> </li> <li>Off-board programming                             <ul style="list-style-type: none"> <li>The user area and data area can be rewritten by using a flash programmer (serial or parallel programmer).</li> </ul> </li> </ul>
Other items		Automatic bit rate adjustment Protect mode	Automatic bit rate adjustment Protection

Notes: 1. The values before and after a slash (/) respectively indicate the typical and maximum values.

2. The values before and after a slash (/) respectively indicate the typical and minimum values.

## 2.13 Low-Power Modes and Module Stop Function

### 2.13.1 Comparison of Low-Power Mode Specifications

The H8/3048 device provides three low-power modes. The RX261 device also provides four low-power modes. On the H8/3048 and RX261 devices, in a low-power mode, the CPU stops and the operating states of the internal modules change according to the selected low-power mode. The following table describes the low-power modes.

#### (1) Low-power modes of the H8/3048 device

Table 2.25 shows the operating states of the internal modules in each low-power mode of the H8/3048 device.

**Table 2.25 Operating States of Internal Modules in Low-Power Modes of the H8/3048 Device**

Module	Sleep mode	Software standby mode	Hardware standby mode
Clock oscillator	Operating	Stopped	Stopped
CPU	Stopped	Stopped	Stopped
	Registers	Settings retained	Settings undetermined
Refresh controller	Operating	Stopped/ Settings retained*1	Stopped/Reset
Other peripherals	Operating	Stopped/Reset	Stopped/Reset
RAM	Settings retained	Settings retained	Settings retained
$\phi$ clock output	$\phi$ output	High-level output	High-impedance
I/O ports	Settings retained	Settings retained	High-impedance

Note: 1. Bits 7 and 6 in the RTCNT and RTMCSR registers are reset.

**(2) Low-power modes of the RX261 device**

Table 2.26 shows the operating states of the internal modules in each low-power mode of the RX261 device.

**Table 2.26 Operating States of Internal Modules in Low-Power Modes of the RX261 Device**

Operating Conditions	Sleep Mode	Deep Sleep Mode	Software Standby Mode	Snooze Mode*1
Main clock oscillator	Operating possible	Operating possible	Stopped	Operating possible
Sub-clock oscillator	Operating possible	Operating possible	Operating possible	Operating possible
High-speed on-chip oscillator (HOCO)	Operating possible	Operating possible	Stopped	Operating possible
Low-speed on-chip oscillator (LOCO)	Operating possible	Operating possible	Operating possible*5	Operating possible
IWDT-dedicated on-chip oscillator (ILOCO)	Operating possible*6	Operating possible*6	Operating possible*6	Operating possible*6
PLL	Operating possible	Operating possible	Stopped	Operating possible
PLL2	Operating possible	Operating possible	Stopped	Operating possible
CLKOUT	Operating possible	Operating possible	Operating possible*7	Operating possible
CPU	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
RAM0 (0000 0000h to 0001 FFFFh)	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)	Operating possible (Retained)*8
Flash memory	Operating	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
DMAC	Operating possible*9	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
DTC	Operating possible*10	Stopped (Retained)	Stopped (Retained)	Operating possible*8, *10
Watchdog timer (WDT)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
Independent watchdog timer (IWDT)	Operating possible*6	Operating possible*6	Operating possible*6	Operating possible*6
Remote control signal receiver (REMC)	Operating possible	Operating possible	Operating possible*11	Operating possible
Realtime clock (RTC)	Operating possible	Operating possible	Operating possible	Operating possible
RTCCOUT	Operating possible	Operating possible	Operating possible	Operating possible
Voltage detection circuit (LVD)	Operating possible	Operating possible	Operating possible	Operating possible
Power-on reset circuit	Operating	Operating	Operating	Operating

Comparator B	Operating possible	Operating possible	Operating possible* <sup>12</sup>	Operating possible* <sup>12</sup>
Low power timer (LPT)	Operating possible	Operating possible	Operating possible	Operating possible
Peripheral modules	Operating possible	Operating possible	Stopped (Retained)* <sup>13</sup>	Operating possible
I/O ports	Operating	Operating	Retained* <sup>14</sup>	Operating

“Operating possible” means that operating or stopped can be controlled by the register setting.

“Stopped (Retained)” means that internal register values are retained and internal operations are suspended.

- Note 1. The operating clock is supplied to the on-chip peripheral modules for which supply of the clock signal is enabled (set to 0) by the MSTPCRm.MSTPmi bits (m = A to D, i = 31 to 0) once they have been placed in the snooze mode, and their operation is restarted. Accordingly, if operation of any on-chip modules is not required in the snooze mode, set the corresponding MSTPmi bit to 1 to place the modules in the module-stop state before placing them in the software standby mode.
- Note 2. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ7) or any of peripheral interrupts (interrupts on the RTC alarm, RTC periodic, IWDTC, voltage monitoring, USB, REMC, ELC (LPT dedicated interrupt)).
- Note 3. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ7), any of peripheral interrupts (interrupts on the RTC alarm, RTC periodic, IWDTC, voltage monitoring, USB, REMC), an interrupt for release from the snooze mode, or a non-maskable RAM error interrupt.
- Note 4. This does not include a RES# pin reset, power-on reset, voltage monitoring reset, or independent watchdog-timer reset. One of these reset sources initiate transition to reset state.
- Note 5. In this state, operation or stopping is selectable on the low-speed on-chip oscillator forced oscillation bit (LOFCR.LOFOXIN) in the low-speed on-chip oscillator forced oscillation control register.
- Note 6. Operating or stopping is selected by setting the IWDTC sleep mode count stop control bit (IWDTC.SLCSTP) in option function select register 0 (OFS0) in IWDTC auto-start mode. In any mode other than IWDTC auto-start mode, operating or stopping is selected by the setting of the sleep mode count stop control bit (SLCSTP) in the IWDTC count stop control register (IWDTC.CSTPR).
- Note 7. The clock signal is not output when the clock output select bits in the CLKOUT output control register (CKOCR.CKOSEL[3:0]) are set to a value other than 0011b (sub clock oscillator). When the LOCO forced oscillation is enabled in software standby mode, do not set the CKOCR.CKOSEL[3:0] bits to 0000b (LOCO clock).
- Note 8. In this state, operation or stopping is selectable by setting the DTC enable in the snooze mode bit in the snooze control register (SNZCR.SNZDTCE).
- Note 9. Writing to the system control registers in the sleep mode is prohibited. For the system control registers, see Table 5.1, List of I/O Registers (Address Order), in which they are indicated as “SYSTEM” in the Module System column.
- Note 10. Writing to the system control registers in the sleep mode or snooze mode is prohibited. For the system control registers, see Table 5.1, List of I/O Registers (Address Order), in which they are indicated as “SYSTEM” in the Module System column.
- Note 11. Operation is possible when the clock source for the operating clock is the sub clock or IWDTC-dedicated on-chip oscillator clock. Note 12. Using the digital filter function is prohibited. Operation for outputting the comparison result to the CMPOBn pin is possible.
- Note 13. The peripheral logic states are retained.
- Note 14. When the remote control signal receiver (REMC) is operated, related pins continue operating.

### 2.13.2 Mode Transitions

Figure 2.12 shows the transitions of low-power modes on the RX261 device.

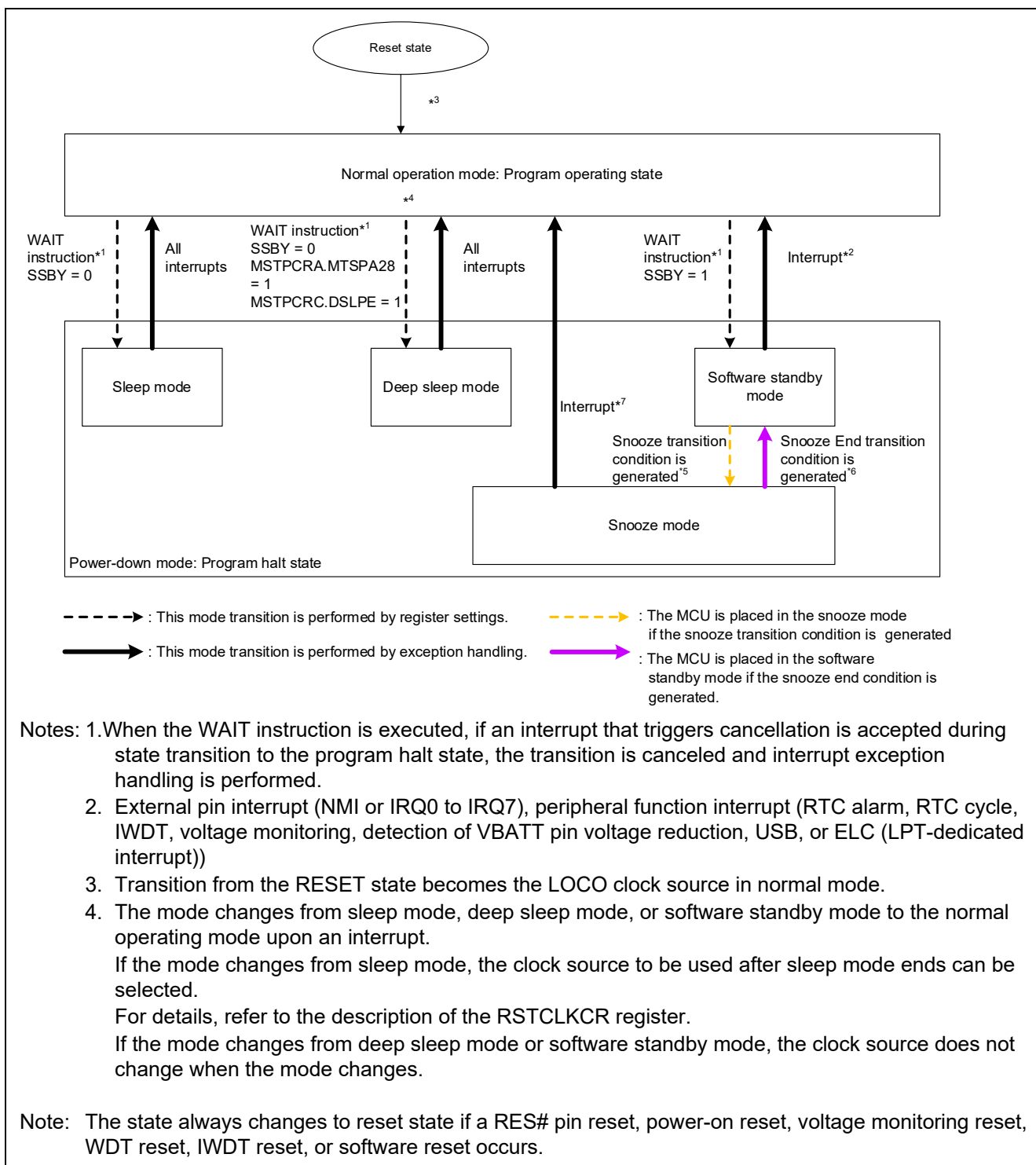


Figure 2.12 Transitions of Modes on the RX261 Device

### 2.13.3 Module Stop

The H8/3048 device provides the module standby function, which can stop the peripheral modules independently of the low-power mode. When the operating state changes to the module standby state, the states of the internal modules are initialized. On the H8/3048 device, after a reset is performed, the peripheral modules are operating. The RX261 device also provides a similar function, the module stop function, which can stop the peripheral modules independently of the low-power mode. When the operating state changes to the module stop state, the states of the internal modules are retained. On the RX261 device, after recovery from a reset, all peripheral modules except the DMAC, DTC, and RAM are stopped. To start the peripheral modules, cancel module stop. The operating state (operating or stopped) of each peripheral module is controlled by configuring the registers shown in Table 2.27.

**Table 2.27 List of Modules That Can Be Stopped**

H8/3048	RX261
<b>Module standby function</b>	<b>Module stop function</b>
MSTCR: Module standby control register	SYSTEM.MSTPCRA: Module stop control register A
ITU	TMR
SCI	GPTW
DMA	CMT
Refresh controller	S12AD
A/D converter	DA
—	DMAC and DTC
	SYSTEM.MSTPCRB: Module stop control register B
	SCI1,5,6,12 and RSCI0
	DOC
	ELC
	Comparator B
	RSPI0
	USB0
	RIIC0
	CRC
	SYSTEM.MSTPCRC: Module stop control register C
	RAM0
	CAC
	RSCI8,9
	REMC
	SYSTEM.MSTPCRD: Module stop control register D
	CAN FD
	CTSU
	RSIP

### 3. Reference Documents

This chapter lists the documents referenced in the preparation of this application note. When referring to the documents listed below, substitute the latest version if a newer version is available. The latest versions of these documents can be confirmed and downloaded from the Renesas Electronics website.

**Table 3.1 Reference Documents**

Reference Documents
H8/3048 Group, H8/3048 F-ZTAT™ Hardware Manual (REJ09B0259)
RX260 Group, RX261 Group User's Manual: Hardware (R01UH1045EJ)
RX261 Group Initial Setting (R01AN7292EJ)
RX65N Group, RX651 Group User's Manual: Hardware (R01UH0590EJ)

## Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Mar. 25, 2025	—	First edition issued

# General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

## 1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity.

Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

## 2. Processing at power-on

The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

## 3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

## 4. Handling of unused pins

Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

## 5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

## 6. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.).

## 7. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

## 8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

## Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
  - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
  - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

## Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,  
Koto-ku, Tokyo 135-0061, Japan  
[www.renesas.com](http://www.renesas.com)

## Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

## Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:  
[www.renesas.com/contact/](http://www.renesas.com/contact/).