RX231 Group

Microcontroller Migration Guide for Migration from H8/3048 (H8/300H Series) to RX231

Introduction

This application note describes precautions to observe when replacing the H8/3048 device with the RX231 device and provides other information such as differences between the H8/3048 and RX231 devices. For details on individual functions, refer to the latest hardware manual.

Target Device

RX231

Contents

1. CPU Architecture

1.1 System Registers

The following sections show the differences between the system registers of the H8/3048 and the RX231 devices.

1.1.1 General-Purpose Registers

The H8/3048 and RX231 devices have 32-bit general-purpose registers. With the H8/3048 device, eight*1 32-bit general-purpose registers are available. With the RX231 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 RX231 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 RX231 device, general-purpose registers cannot be divided.

Figure 1.1 shows the differences between the general-purpose registers of the H8/3048 and RX231 devices.

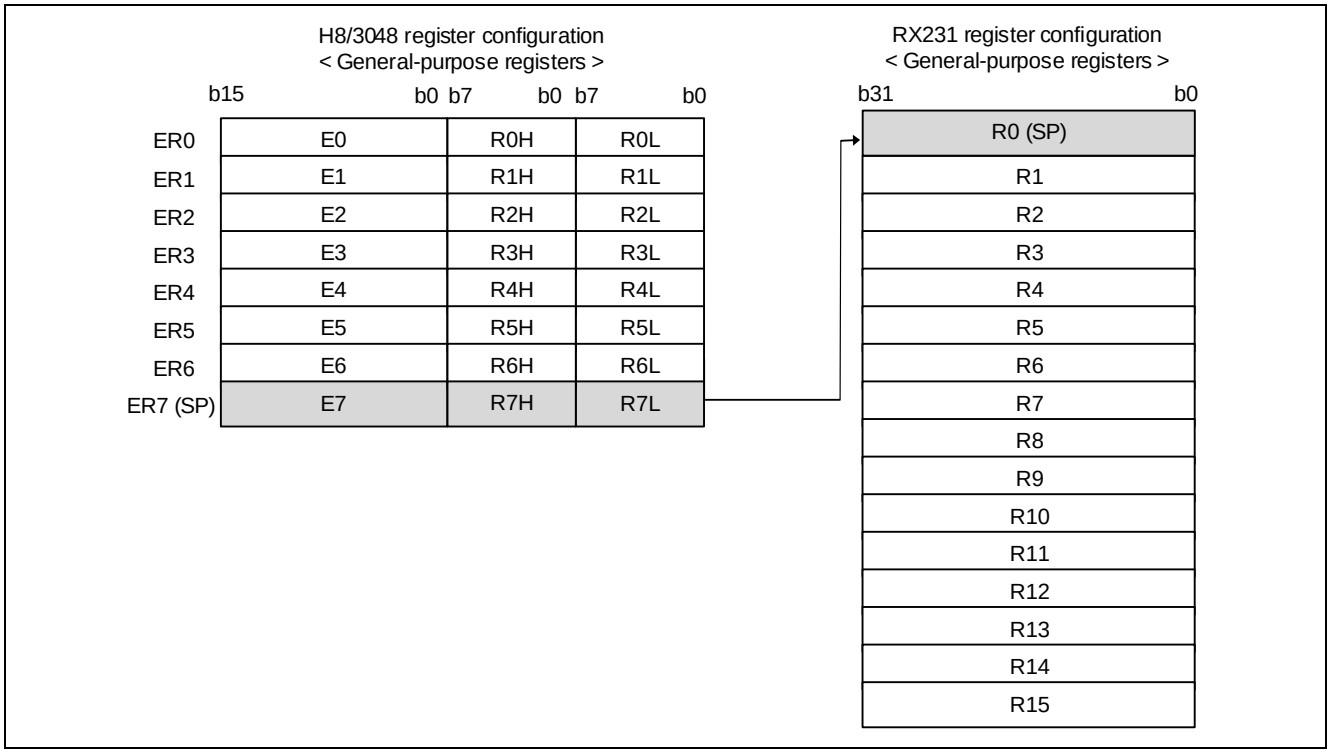
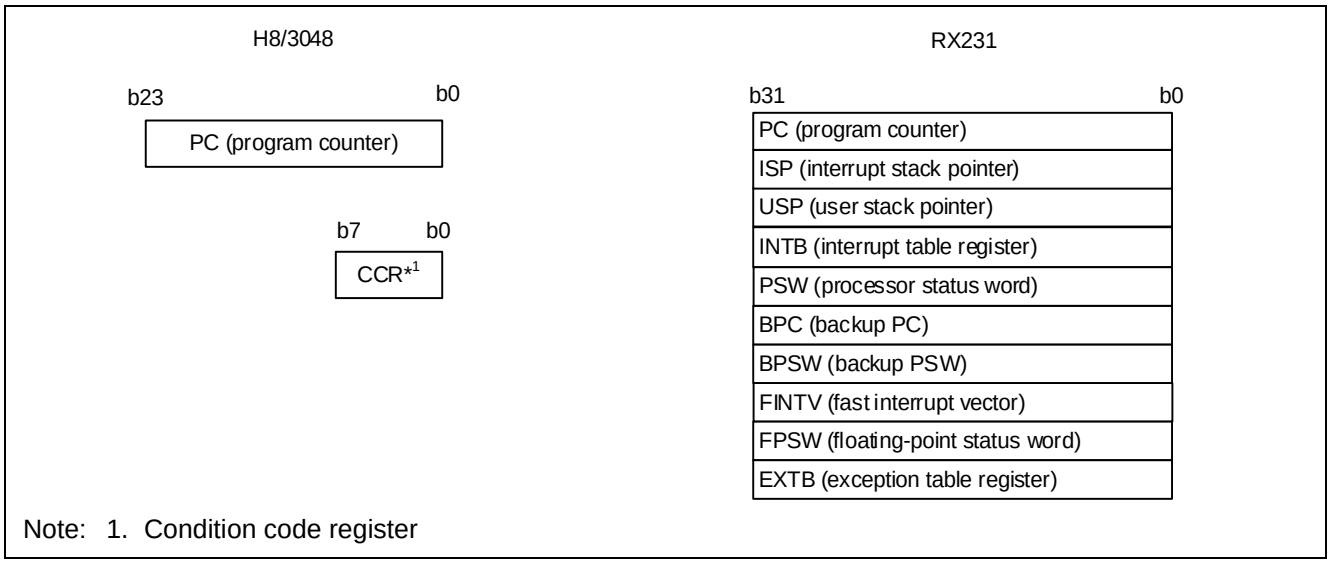


Figure 1.1 Differences of General-Purpose Registers

1.1.2 Control Registers

Figure 1.2 shows the differences between the control registers of the H8/3048 and RX231 devices.



Note: 1. Condition code register

Figure 1.2 Differences of Control Registers

The following outlines the RX231-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 RX231 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 RX231 device.

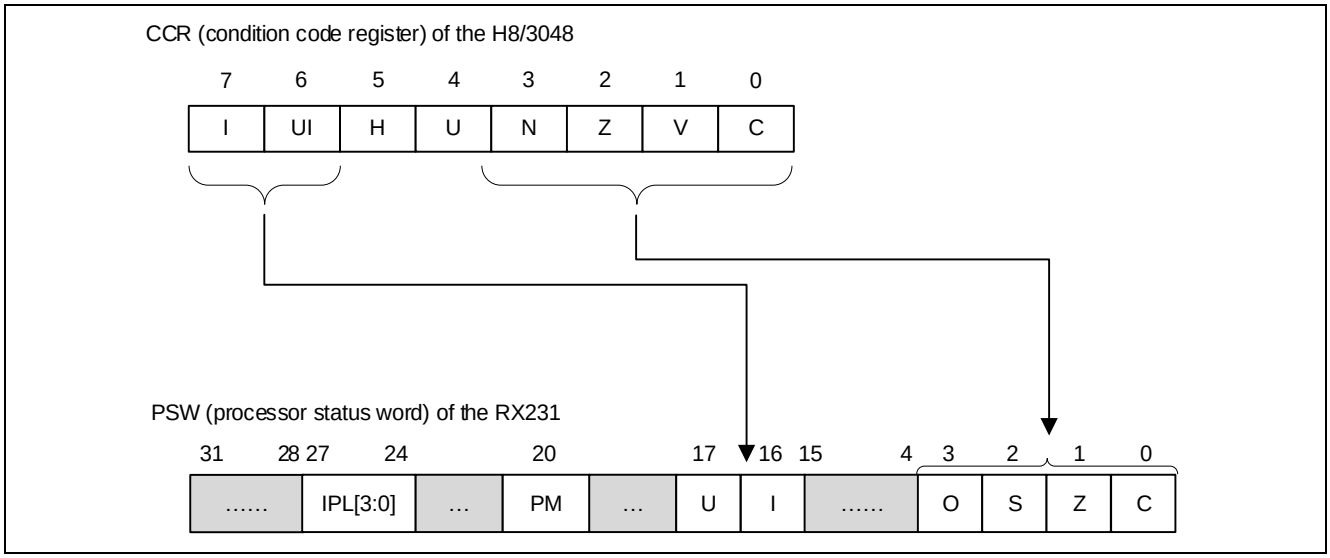


Figure 1.3 Differences Between the CCR Register on the H8/3048 and the PSW Register on the RX231

Table 1.1 Differences Between the CCR Register on the H8/3048 and the PSW Register on the RX231

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 RX231. 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 RX231. 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 RX231. 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 RX231. 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 RX231. 0: Interrupts are disabled. 1: Interrupts are enabled. This bit is used to enable reception of interrupt requests on the RX231. 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 RX231. 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 RX231. 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 RX231 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 RX231.
H (half-carry flag)	—	There is no corresponding bit on the RX231.

1.2 Option-Setting Memory

On the RX231, 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.

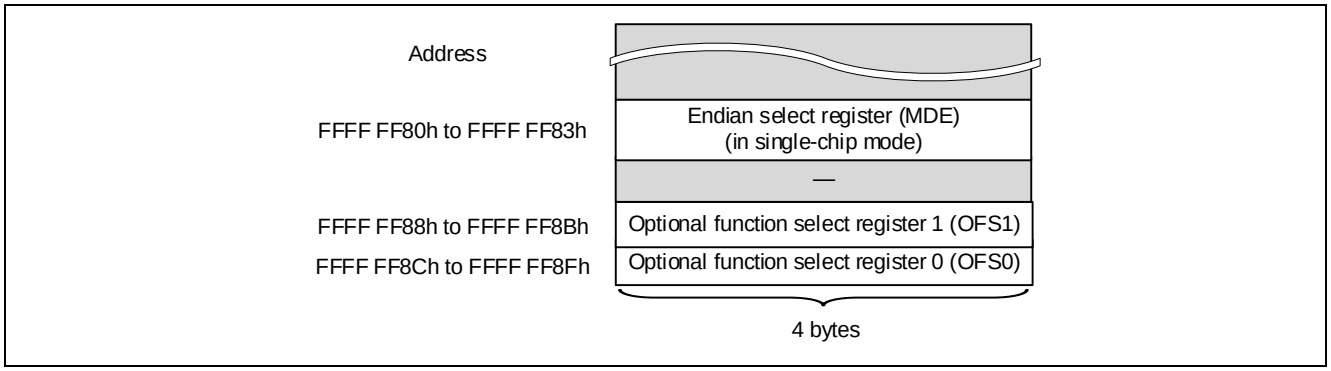


Figure 1.4 Areas in the Option-Setting Memory

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
  - Whether to enable or disable the function that activates 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) = {
//;0xffffff80  MDE register
#ifdef __BIG
   (void (*)(void))0xfffffff8, // big
#else
   (void (*)(void))0xffffffff, // 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) = {
...
//;0xffffff88  OFS1 register
    (void (*)(void))0xffffffff, // OFS1
//;0xffffff8c  OFS0 register
    (void (*)(void))0xffffffff, // 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 RX231 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 RX231 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.

The endian settings can be switched for each CS area in the external address space. However, instruction code cannot be allocated to an external space with endian settings that differ from those of the chip. When allocating instruction code to an external space, allocate the code to an area with the same endian settings as the chip. (For details, refer to the “RX230 Group, RX231 Group User’s Manual: Hardware”.)

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.

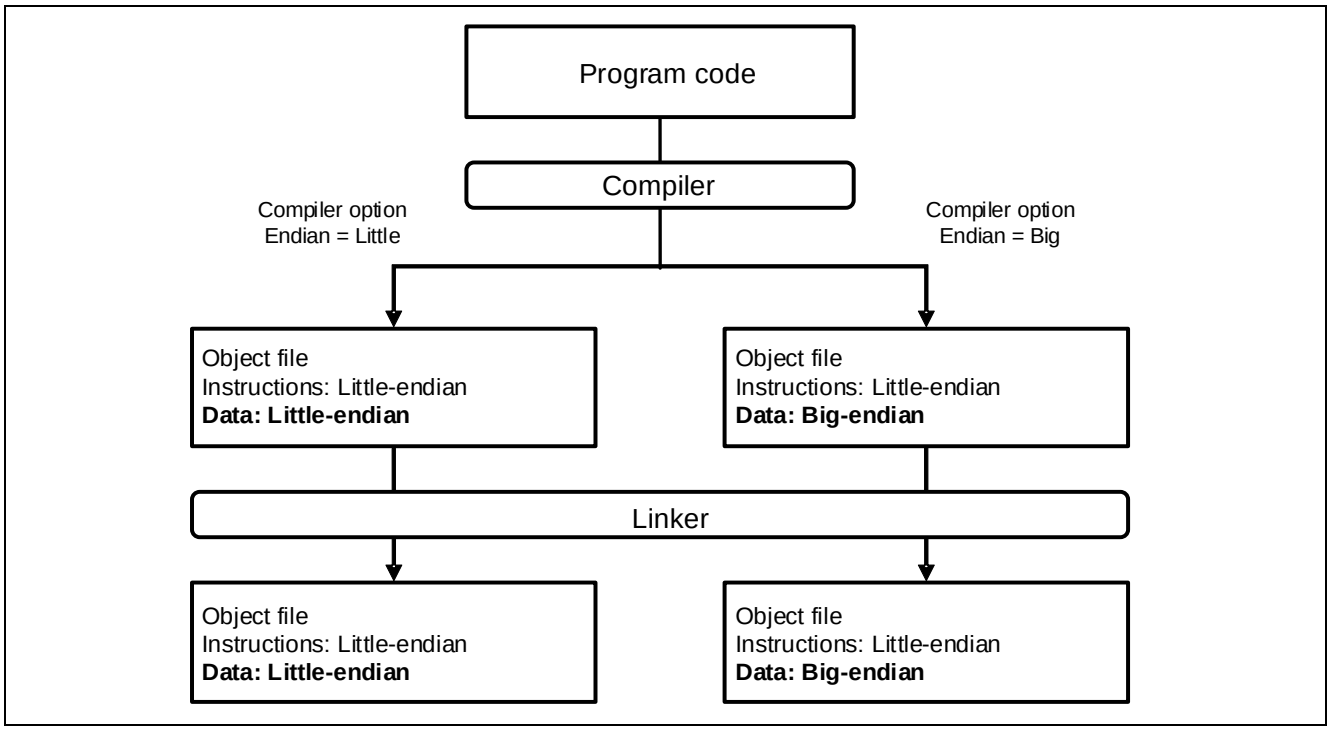


Figure 1.7 Specifying Endian Settings by Using Compiler Options

1.3 Reset Function

This section describes the differences in reset types between the H8/3048 and RX231 devices.

1.3.1 Reset Sources

Table 1.2 lists the reset sources of the H8/3048 and RX231 devices.

Table 1.2 Reset Sources

H8/3048	RX231
• Reset	• RES# pin reset • Power-on reset • Voltage monitor 0 reset • Voltage monitor 1 reset • Voltage monitor 2 reset • Independent watchdog timer reset • Watchdog timer reset • Software reset

(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 RX231 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 RX231 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 RX231.

Table 1.3 Reset Initialization Scope for the RX231 Device

	Reset Sources							
Reset Target	**RES# pin reset**	**Power-on reset**	**Voltage monitor 0 reset**	**Independent watchdog timer reset**	**Watchdog timer reset**	**Voltage monitor 1 reset**	**Voltage monitor 2 reset**	**Software reset**
Power-on reset detection flag	○	—	—	—	—	—	—	—
Cold start/warm start determination flag	— *1	○	—	—	—	—	—	—
Voltage monitor 0 reset detection flag/ battery back-up function registers	○	○	—	—	—	—	—	—
Independent watchdog timer reset detection flag/ independent watchdog timer registers	○	○	○	—	—	—	—	—
Watchdog timer reset detection flag/ watchdog timer registers	○	○	○	○	—	—	—	—
Voltage monitor 1 reset detection flag/ voltage monitor function 1 registers	○	○	○	○	○	—	—	—
Voltage monitor 2 reset detection flag/ voltage monitor function 2 registers	○	○	○	○	○	○	—	—
Software reset detection flag	○	○	○	○	○	○	○	—
Realtime clock registers*2	—	—	—	—	—	—	—	—
Registers other than the above, CPU, and internal state	○	○	○	○	○	○	○	○

○: Initialization occurs.

—: No change occurs.

Notes: 1. Initialization occurs when power is turned on.

2. Some control bits are initialized by any types of resets.

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 RX231 devices.

Table 1.4 List of Clock Sources of the H8/3048 and RX231 Devices

H8/3048	RX231
• System clock	• Main clock oscillator • Subclock oscillator • PLL circuit • USB-dedicated PLL circuit • High-speed on-chip oscillator (HOCO) • Low-speed on-chip oscillator (LOCO) • IWDT-dedicated on-chip oscillator

1.4.2 RX231 Clock Generation Circuit

On the RX231 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:

- RX231 Group Initial Setting (R01AN2185EJ)

Figure 1.8 shows the clock generation circuit of the RX231 device.

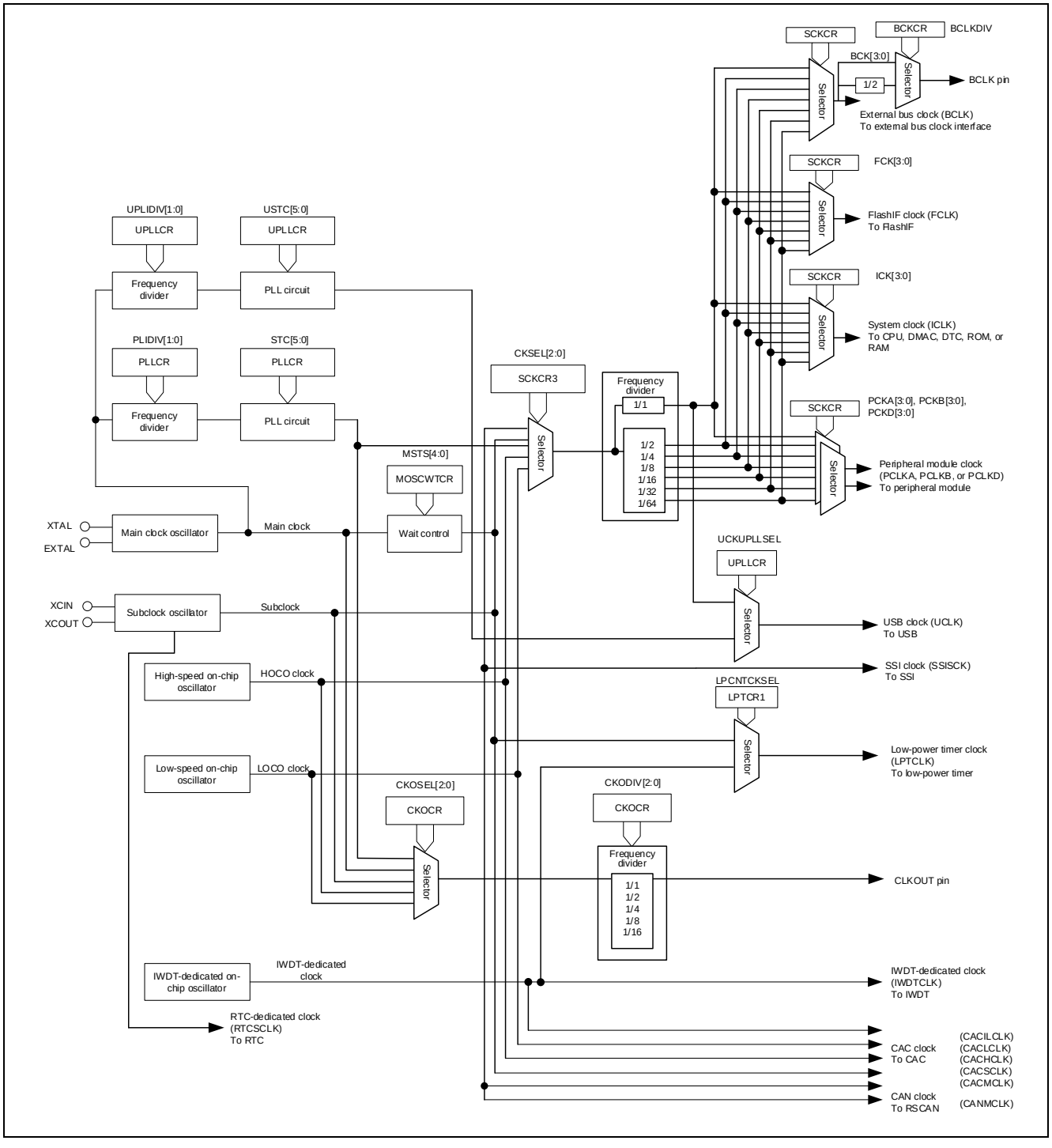


Figure 1.8 Clock Generation Circuit of the RX231 Device

1.5 Operating Modes

1.5.1 Comparison of Operating Modes

On the H8/3048 device, mode settings are configured by means of pins alone. On the RX231 device, however, system registers must also be configured in addition to pin configuration. For details on operating mode settings, refer to section 1.5.3, “Operating Mode Settings”, and the “RX230 Group, RX231 Group User’s Manual: Hardware”. Table 1.5 lists the operating modes of the RX231 device.

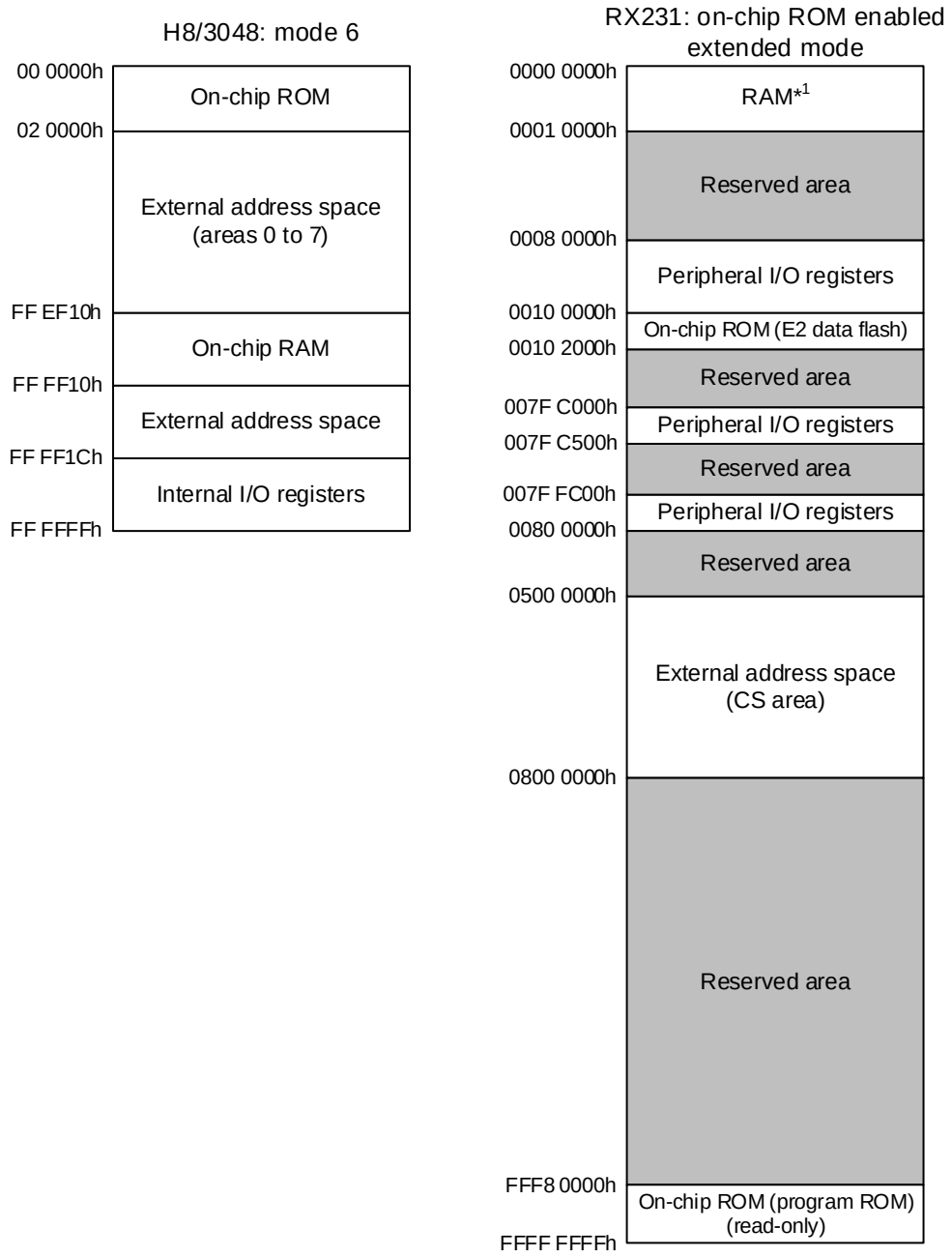
Table 1.5 Operating Modes of the RX231 Device

Operating Modes of the RX231 Device	Mode Description
Single-chip mode*1	Operating mode in which the on-chip ROM is enabled and the external address space is disabled
On-chip ROM enabled extended mode*1	Operating mode in which the on-chip ROM is enabled and the external address space is enabled
On-chip ROM disabled extended mode	Operating mode in which the on-chip ROM is disabled and the external address space 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 RX231 Devices

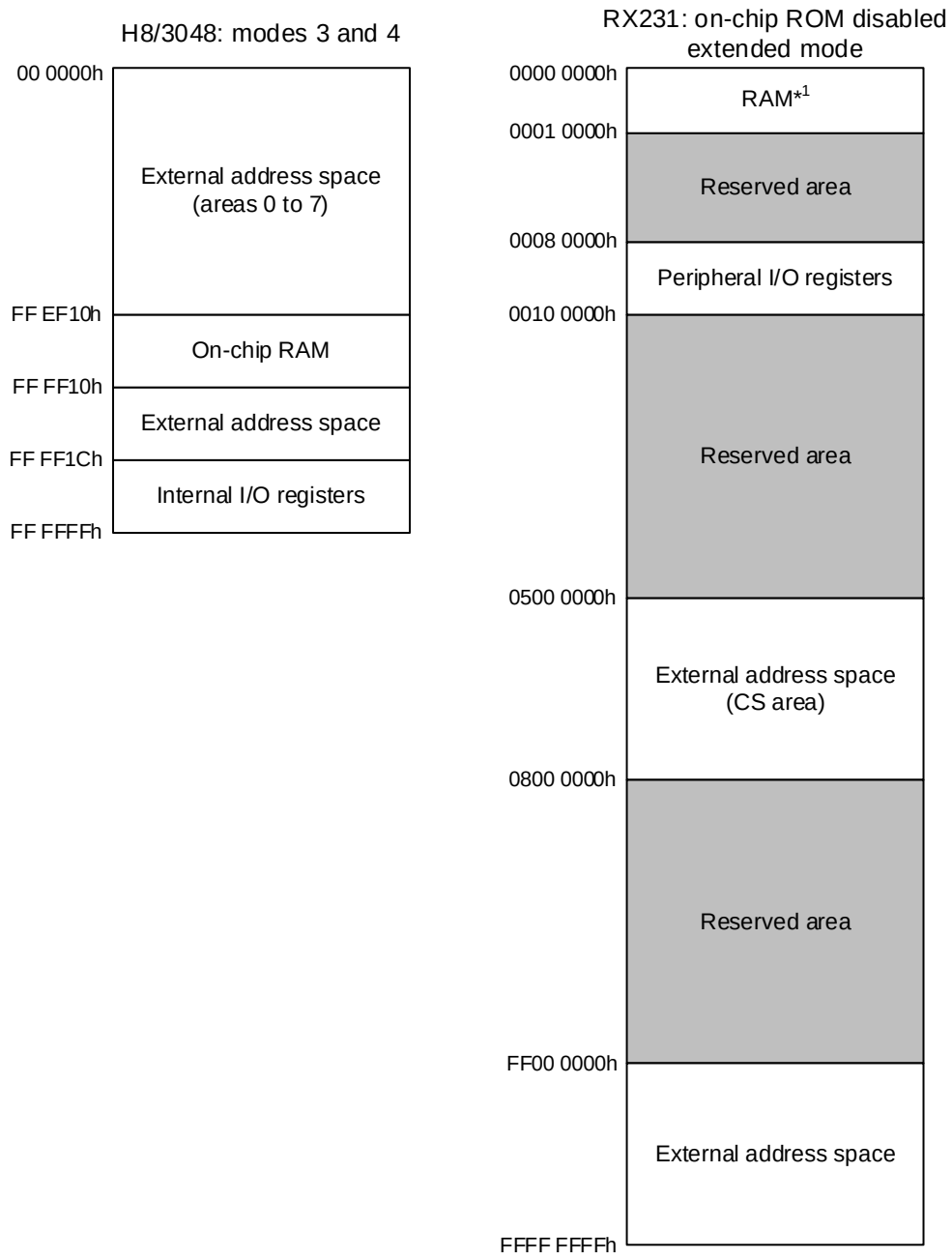
Figure 1.9 compares the memory maps between the H8/3048 device in mode 6 and the RX231 device in on-chip ROM enabled extended mode.



Note: 1. ROM and RAM capacities differ, depending on the product version.

Figure 1.9 Comparison of the Memory Map Between the H8/3048 Device and the RX231 Device (in On-chip ROM Enabled Extended Mode)

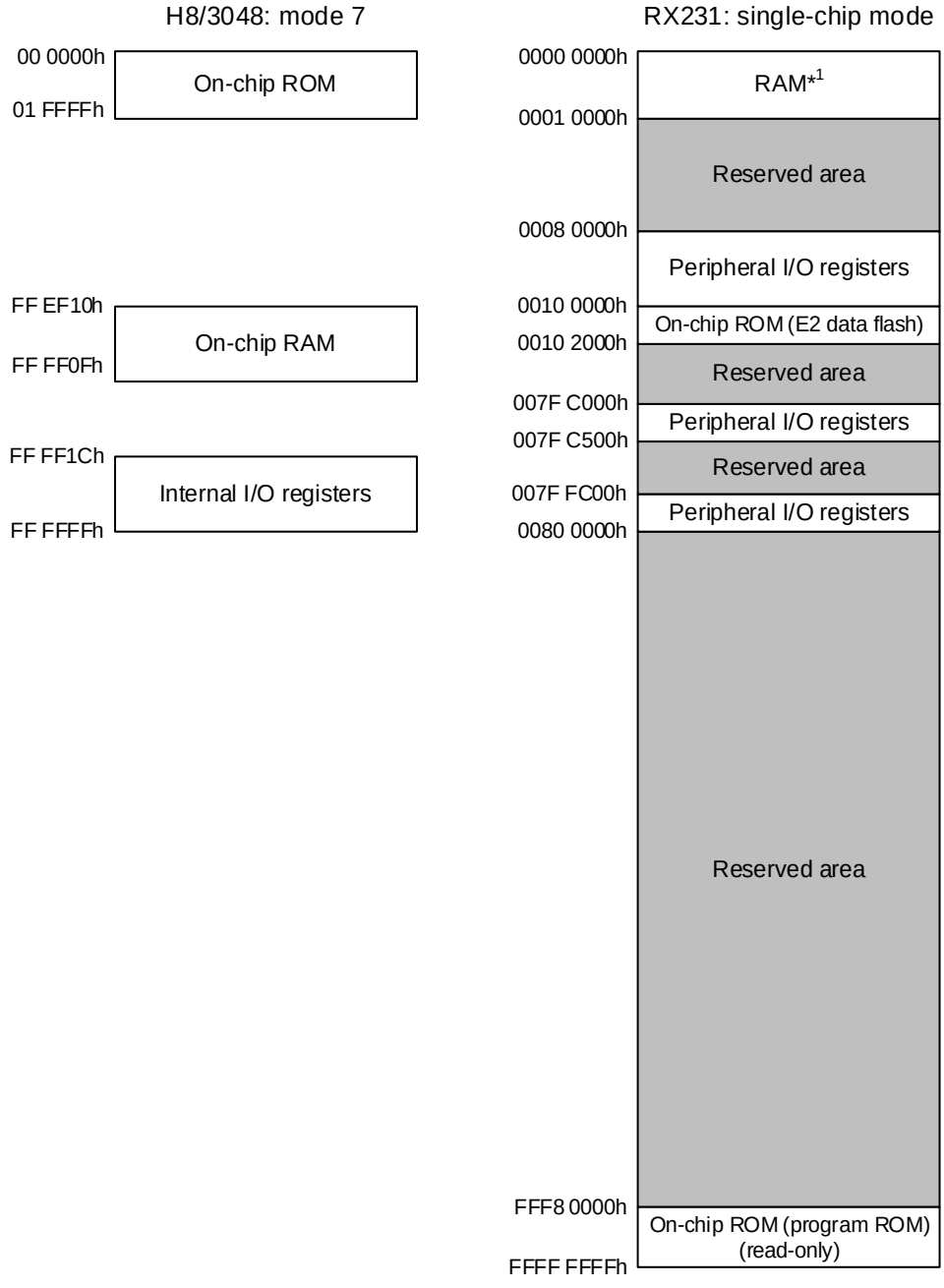
Figure 1.10 compares the memory maps of the H8/3048 device in mode 3/4 and the RX231 device in on-chip ROM disabled extended mode.



Note: 1. ROM and RAM capacities differ, depending on the product version.

Figure 1.10 Comparison of the Memory Map Between the H8/3048 Device and the RX231 Device (in On-chip ROM Disabled Extended Mode)

Figure 1.11 compares the memory maps of the H8/3048 device in mode 7 and the RX231 device in single-chip mode.



Note: 1. ROM and RAM capacities differ, depending on the product version.

Figure 1.11 Comparison of the Memory Map Between the H8/3048 Device and the RX231 Device (in Single-chip Mode)

1.5.3 Operating Mode Settings

The operating modes of the RX231 device can be categorized into two types: modes that can be selected by pin levels when a reset is canceled, and modes that can be selected by software after a reset is canceled.

Table 1.6 lists the operating modes that are determined by mode pin settings. Table 1.7 lists the operating modes that are determined by software after a reset is canceled. The software sets whether to enable or disable the on-chip ROM and whether to enable or disable the external bus based on the settings of the SYSCR0 register. For more detailed specifications, refer to the “RX230 Group, RX231 Group User’s Manual: Hardware”.

Table 1.6 Pin Settings and Operating Modes of the RX231 Device

Pin		
MD	**UB**	**Mode Name**
1	—	Single mode
0	0	Boot mode (SCI interface)
	1	Boot mode (USB interface)

Table 1.7 SYSCR0 Register Settings and Operating Modes of the RX231 Device

SYSCR0 Register		
ROME*1	**EXBE**	**Mode Name**
0 (Disable the on-chip ROM)	0 (Disable the external bus)	Single-chip mode
1 (Enable the on-chip ROM)	0 (Disable the external bus)	
0 (Disable the on-chip ROM)	1 (Enable the external bus)	On-chip ROM disabled extended mode
1 (Enable the on-chip ROM)	1 (Enable the external bus)	On-chip ROM enabled extended mode

Note: 1. Once the ROME bit is cleared to 0, it can no longer be set back to 1.

1.6 Processor Modes

The RX231 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 RX231 device in only supervisor mode allows for software replacement without consideration of processor mode.

Table 1.8 Processor Modes

Processor Modes	Migration Conditions	Brief Description
Supervisor mode	• Reset cancellation • Exception occurrence (The PSW.PM bit is cleared to “0”.)	All CPU resources are accessible, and all instructions can be executed (no limitations). This is the mode in which the OS and other system programs ordinarily operate.
User mode	The PSW.PM bit is set to “1”. 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.	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. This is the mode in which user programs such as application programs ordinarily operate.

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 RX231 devices in regard to handling of general exceptions (including interrupts).

1.7.1 Comparison of Exception Sources

Table 1.9 compares the exception sources in the H8/3048 and RX231 devices.

Table 1.9 Comparison of Exception Sources

Source		H8/3048	RX231
Reset		Implemented	Implemented
Interrupt	NMI	Implemented	Implemented
	Internal	Implemented	Implemented*1, *2
	External	Implemented	Implemented*1, *2
Undefined instruction exception		Not implemented	Implemented*3
Privileged instruction exception		Not implemented	Implemented*4
Access exception		Not implemented	Implemented*5
Floating-point exception		Not implemented	Implemented*6
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 1.12 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 1.10 shows the priority of exception sources in the H8/3048 and RX231 devices.

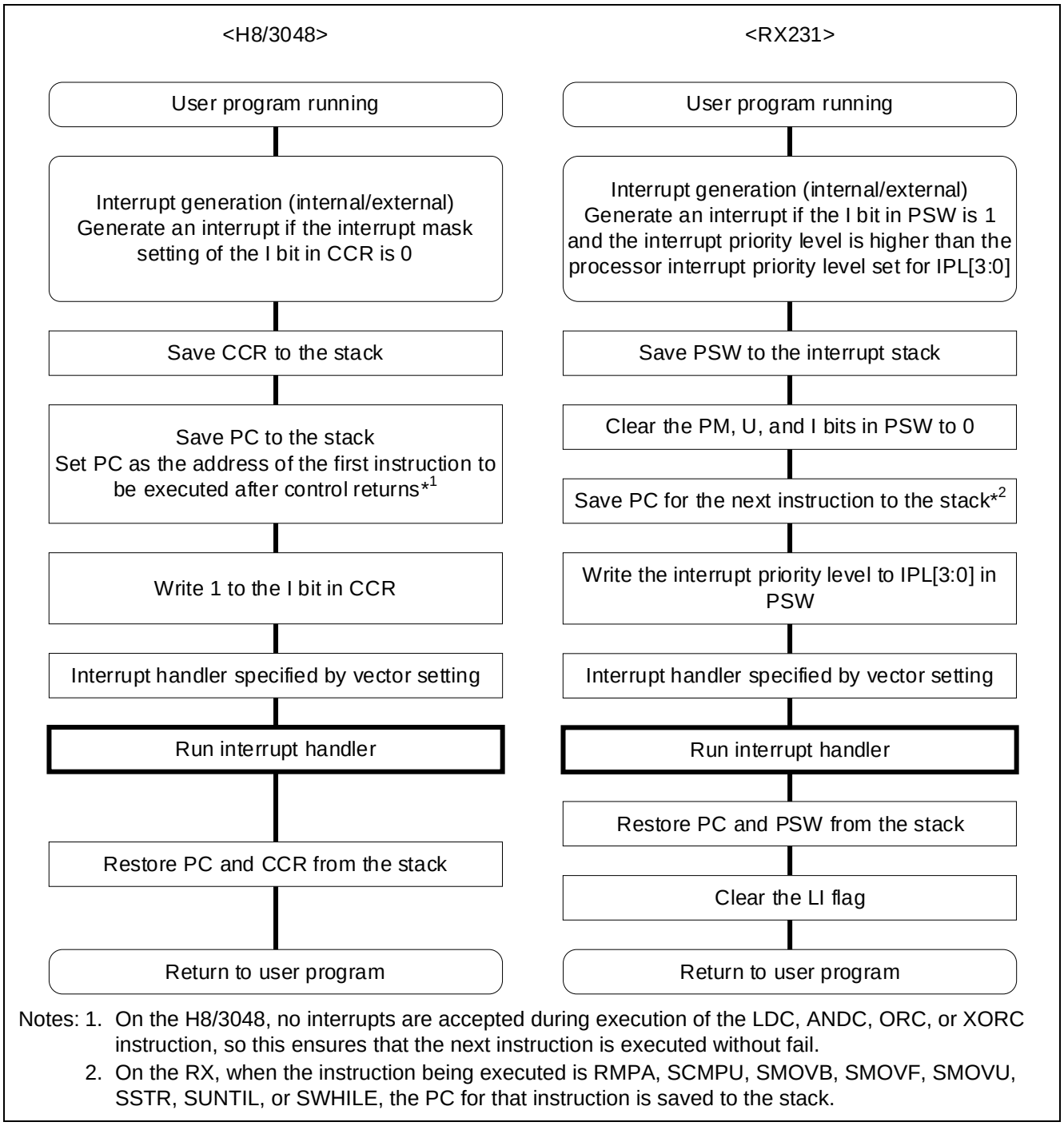
Table 1.10 Priority of Exception Sources

Priority	H8/3048	RX231
High	Reset	Reset
▲	External interrupt (NMI)	Non-maskable interrupt
	— (Not applicable)	Fast interrupt
	Interrupt (internal/external)	Interrupt (internal/external)*1
	— (Not applicable)	Instruction access exception
	— (Not applicable)	Undefined instruction exception or privileged instruction exception
	Trap instruction	Unconditional trap
	— (Not applicable)	Operand access exception
Low	— (Not applicable)	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 1.12 shows the processing sequence of interrupt exception handling (internal/external) on the H8/3048 and RX231 devices.



Notes: 1. On the H8/3048, no interrupts are accepted during execution of the LDC, ANDC, ORC, or XORC instruction, so this ensures that the next instruction is executed without fail.
2. On the RX, when the instruction being executed is RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE, the PC for that instruction is saved to the stack.

Figure 1.12 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 RX231 device, however, the vectors are divided into two vector tables: exception vector table and interrupt vector table. On the RX231 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 reallocation of vector addresses, refer to the “RX230 Group, RX231 Group User’s Manual: Hardware”.

Figure 1.13 shows the configuration of the interrupt vector table in the RX231 device.

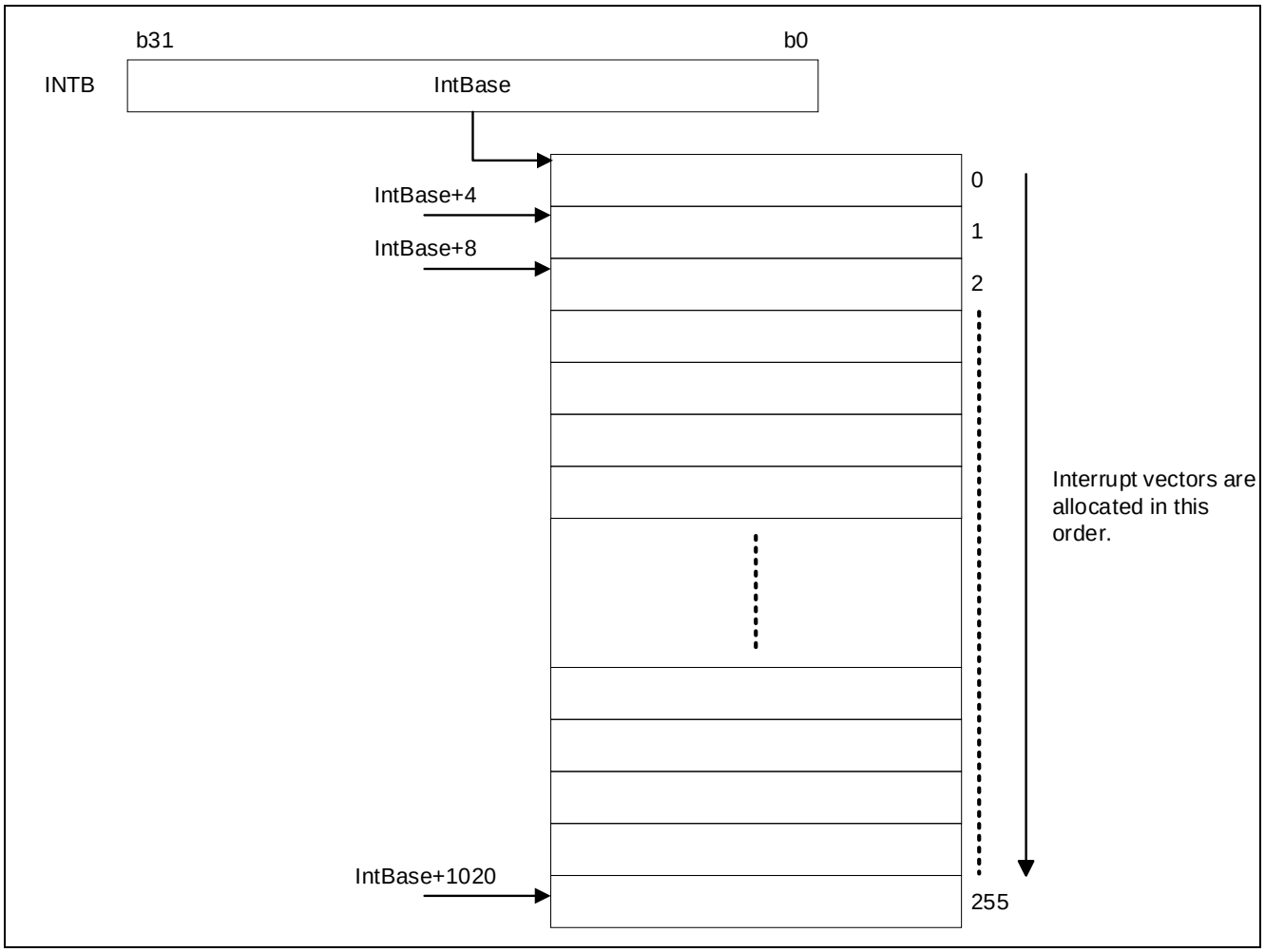


Figure 1.13 Configuration of the Interrupt Vector Table in the RX231 Device

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 RX231 device, a maximum of 16 interrupt mask levels can be set by using a combination of the IPL and I bits in PSW.

Table 1.11 lists the interrupt-related bits in PSW.

Table 1.11 Interrupt-related Bits in PSW

Bits in PSW on the RX231	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 RX231 devices.

1.8.1 Interrupt Controller

Table 1.12 shows the differences in the interrupt controller specifications.

Table 1.12 Differences in the Interrupt Controller Between the Devices

Item		H8/3048	RX231
Interrupts	Peripheral function interrupts	• Interrupts from peripheral modules • Interrupt detection: Edge/level*1	• Interrupts from peripheral modules • Interrupt detection: Edge/level*1
	External function interrupts	• IRQ0 to IRQ5 pins • Sources: 6 • Interrupt detection: Low level or falling edge can be specified for each source.	• IRQ0 to IRQ7 pins • Sources: 8 • Interrupt detection: Low level, falling edge, rising edge, or both edges can be specified for each source. • Digital filter function supported
	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.	A level from 0 to Fh can be specified for each source by configuring register settings.
	Fast interrupts	—	Fast interrupts supported
	DTC/DMAC control	Activation supported*2	Activation supported*2
Non-maskable interrupts	NMI pin interrupts	• Interrupt from the NMI pin • Interrupt detection: falling edge/rising edge	• Interrupt from the NMI pin • Interrupt detection: falling edge/rising edge • Digital filter function supported
	Other sources	• Trap instruction	• Interrupt on detection of oscillation stop • WDT underflow or refresh error • IWDT underflow or refresh error • Voltage monitor 1 interrupt • Voltage monitor 2 interrupt • VBATT voltage monitor interrupt • Exiting the low-power mode
Exiting the low-power mode			• Sleep mode/deep sleep mode: Non-maskable interrupt and all interrupt sources can trigger exiting the mode. • Software standby mode: Non-maskable interrupts, IRQ0 to IRQ7 interrupts, RTC alarm/ cycle interrupts can trigger exiting the mode.

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 RX231 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 RX231 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 RX231 device, however, no such function is available.

Table 1.13 compares the interrupt controllers of the devices.

Table 1.13 Differences in Interrupt Controller Registers Between the Devices

H8/3048	**RX231**
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 RX231 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 “RX230 Group, RX231 Group User’s Manual: Hardware”.

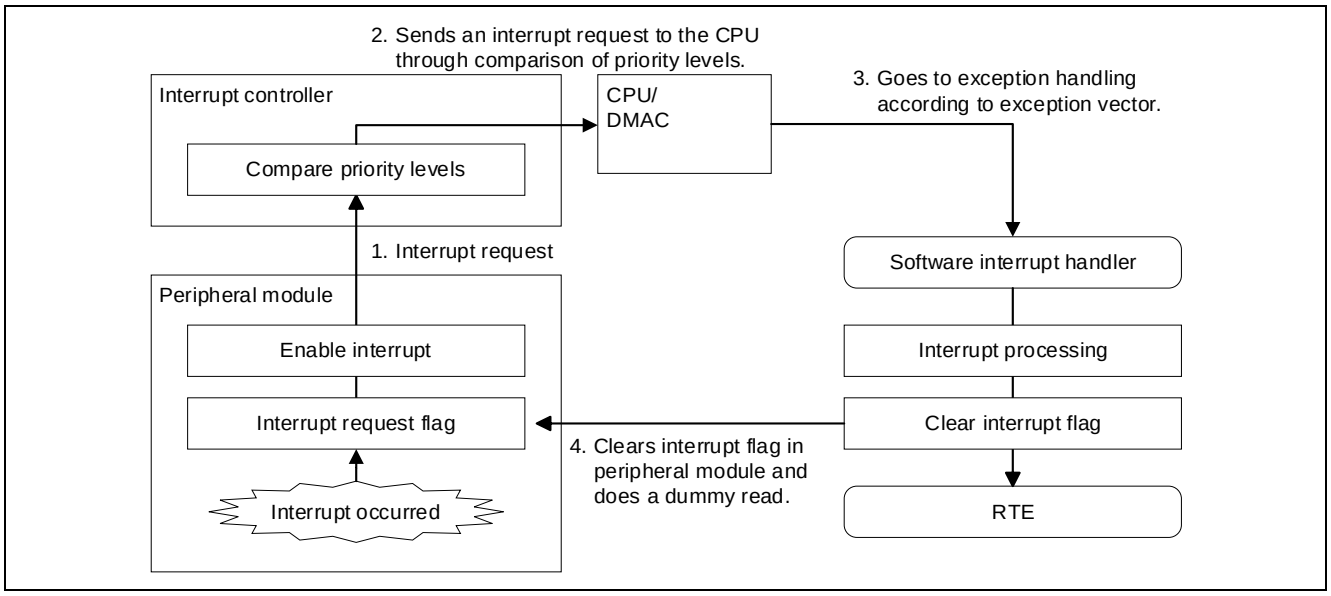
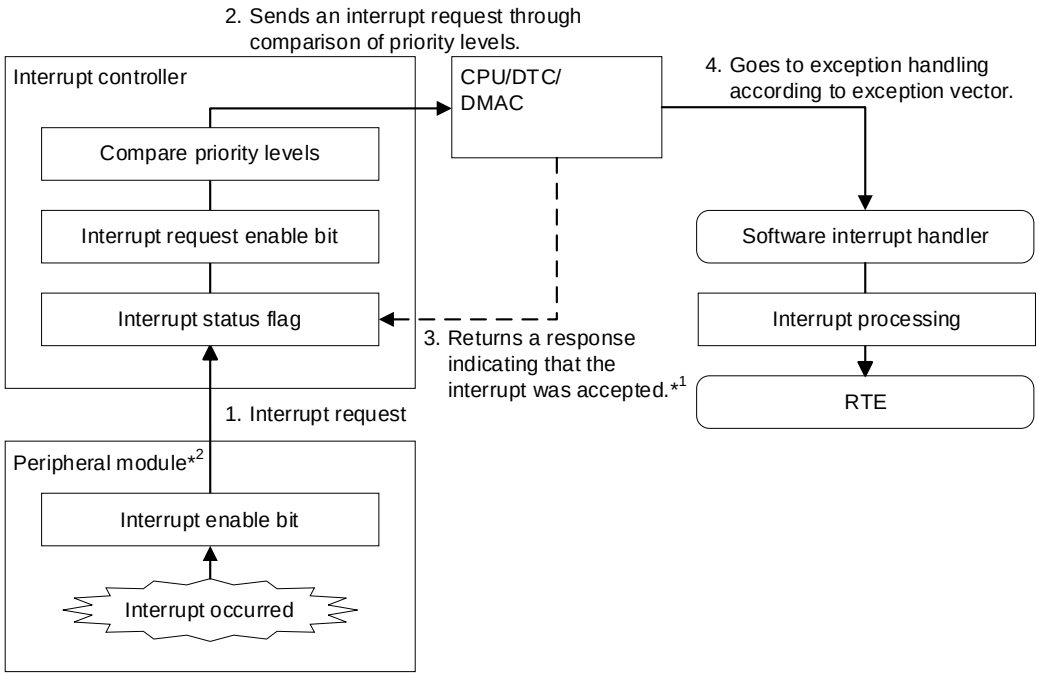


Figure 1.14 Peripheral Module Interrupt (Generated on Edge Detection) for the H8/3048 Device



Notes: 1. When a response indicating interrupt request receipt is received from the CPU (or DTC/DMAC), the interrupt flag is cleared automatically.
It is not necessary for the handler to clear the flag and do a dummy read.
2. For an interrupt generated by level detection, to clear the interrupt status flag, it is necessary to clear the request flag of the interrupt generation source existing on the peripheral module side.

Figure 1.15 Peripheral Module Interrupt (Generated on Edge Detection) for the RX231 Device

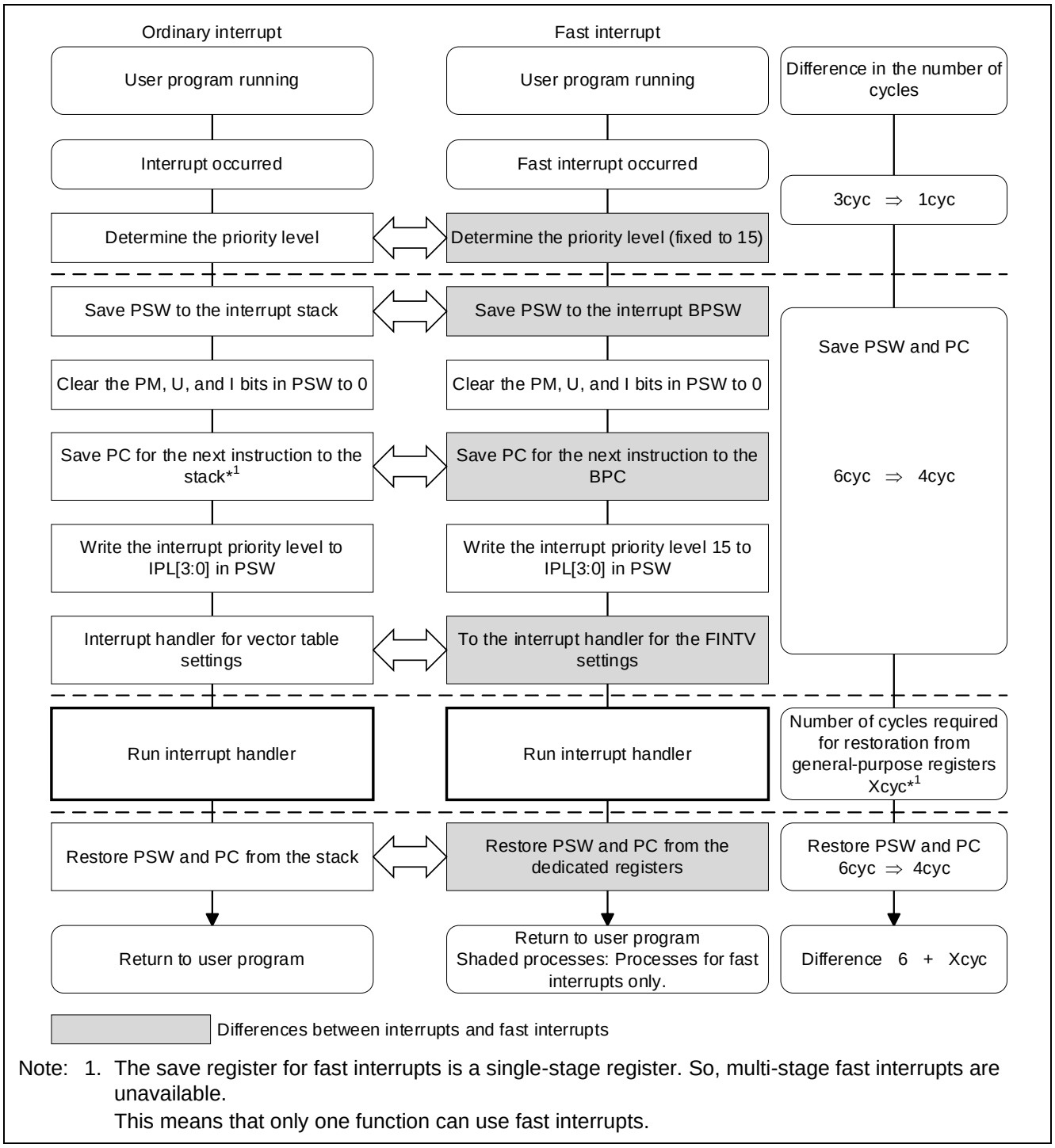
1.8.3 Control of Fast Interrupts

The RX231 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.



Note: 1. The save register for fast interrupts is a single-stage register. So, multi-stage fast interrupts are unavailable.
This means that only one function can use fast interrupts.

Figure 1.16 Differences Between Ordinary Interrupts and Fast Interrupts on the RX231 Device

1.8.4 Digital Filter

The RX231 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.

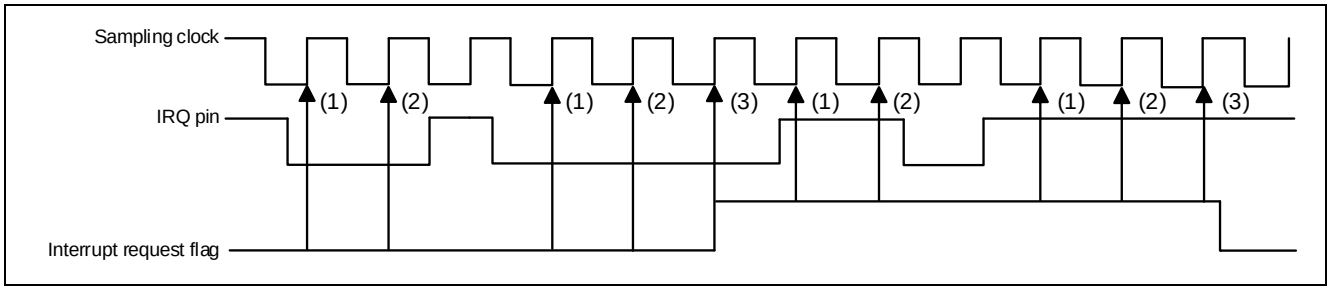


Figure 1.17 Digital Filter Operation Example

1.8.5 Multiple Interrupts

The RX231 device supports multiple interrupts. To handle multiple interrupts on the RX231 device, it is necessary to set the PSW(I) bit to 1 (enable interrupts) in the interrupt handler. Figure 1.18 shows the sequence of operations performed when multiple interrupts are prohibited. Figure 1.19 shows the sequence of operations performed when multiple interrupts occur.

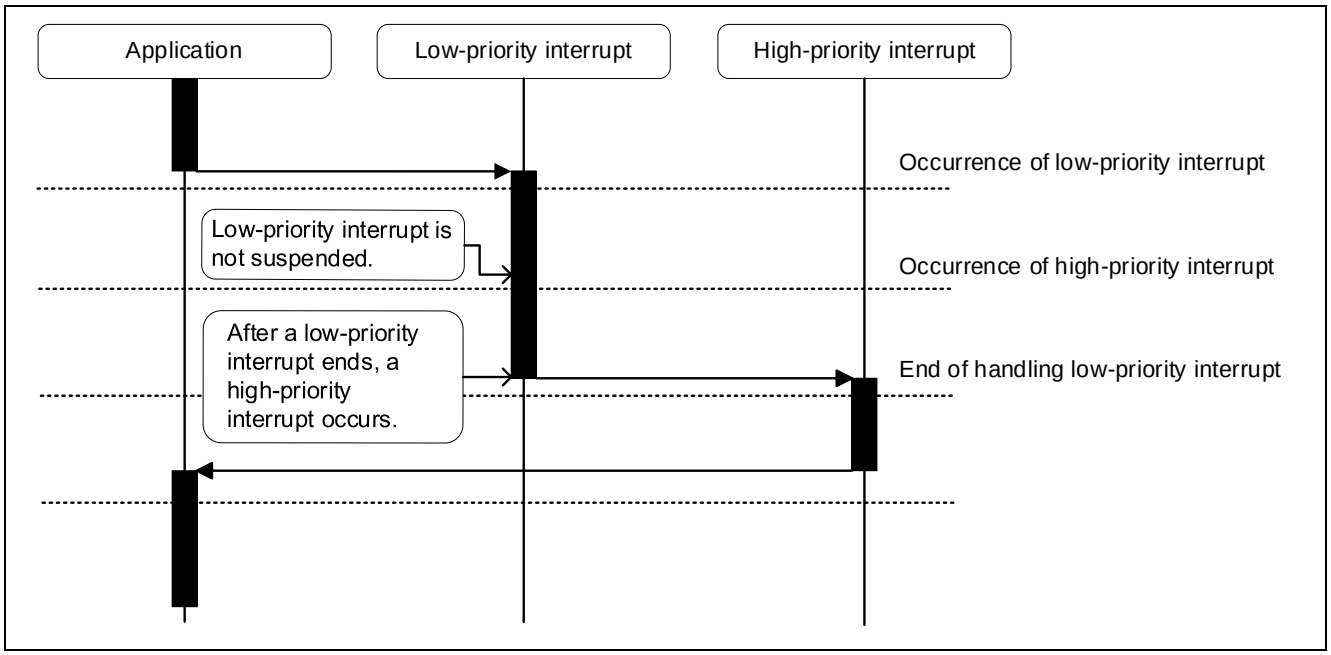


Figure 1.18 Interrupt Handling Sequence on the RX231 Device (when the PSW(I) bit is cleared)

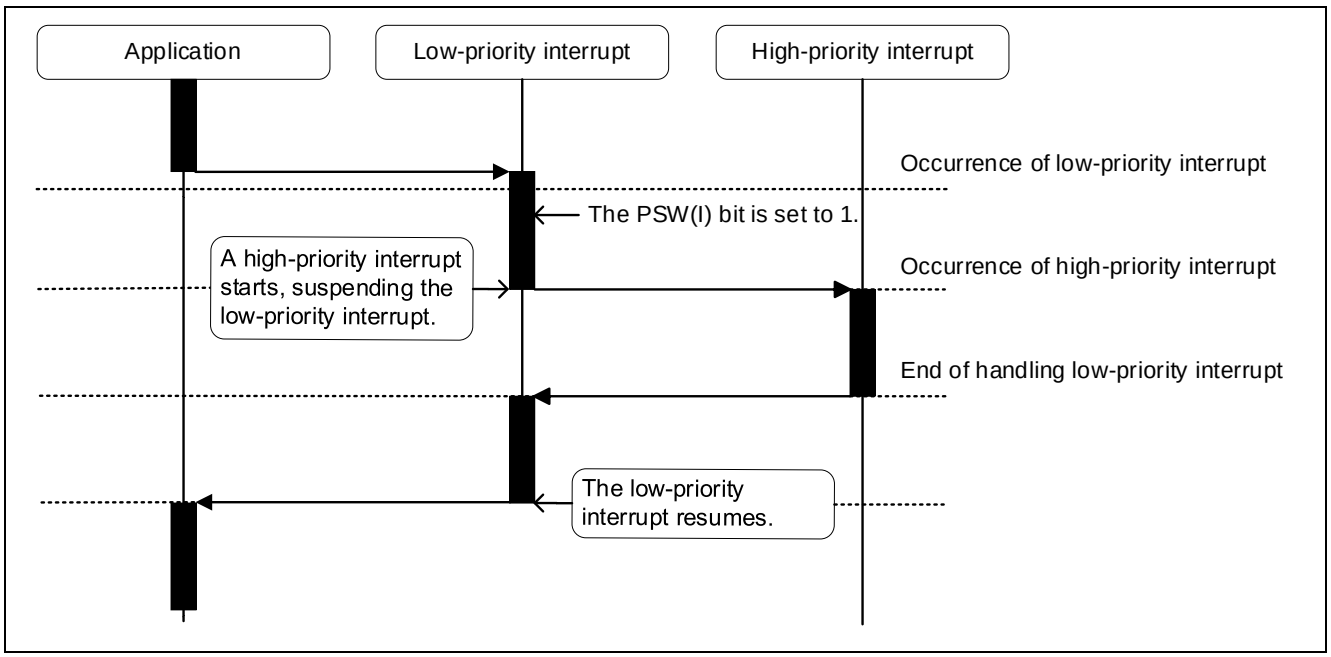


Figure 1.19 Interrupt Handling Sequence on the RX231 Device (when the PSW(I) bit is set)

2. On-Chip Functions (Peripheral Modules)

2.1 List of On-Chip Functions

Table 2.1 List of On-Chip Functions

H8/3048	RX231
Bus controller	Bus controller
DMA controller	DMA controller (DMACA)
I/O Ports	I/O Ports
16-bit integrated timer unit (ITU)	Multifunction timer pulse unit 2 (MTU2a) 16-bit timer pulse unit (TPUa)
Programmable timing pattern controller (TPC)	—
Watchdog timer	Watchdog timer (WDTA) Independent watchdog timer (IWDTa)
Serial communications interface	Serial communications interface (SCIg/SCIh)
Smartcard interface	
A/D converter	12-bit A/D converter (S12ADE)
D/A converter	12-bit D/A converter (R12DAA)
4 KB RAM	64 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 (DTCa)
	Multi-function pin controller (MPC)
	Event link controller (ELC)
	CRC calculator (CRC)
	Serial peripheral interface (RSPIa)
	Serial sound interface (SSI)
	I^2C-bus controller (RIICa)
	Port output enable 2 (POE2a)
	Compare match timer (CMT)
	Realtime clock (RTCe)
	Data operation circuit (DOC)
	Clock frequency accuracy measurement circuit (CAC)
	Capacitive touch sensing unit (CTSU)
	SD host interface (SDHIa)
	Memory protection unit (MPU)
	8-bt timer (TMR)
	CAN module (RSCAN)
	Low-power timer (LPT)
	USB 2.0 host/function module (USBd)
	IrDA interface
	Trusted Secure IP (TSIP-Lite)
	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	RX231		
Number of pins		100-pin	100-pin	64-pin	48-pin
I/O pins		70	83	47	34
	Input pull-up MOS	20	—	—	—
	Input pull-up resistor	—	83	47	34
	Open-drain output	—	58	35	26
	5 V tolerant	—	8	5	5
	Schmitt-trigger input	15	—	—	—
Input pins		8	1	1	1
Total number of ports		78	84	48	34

2.2.2 I/O Settings

On the H8/3048 and RX231 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 RX231 device, control of pin function selection is centralized in the multi-function pin controller (MPC).

In addition, on the RX231 device, a variety of additional settings can be configured. Figure 2.1 shows an overview of I/O settings on the RX231 device.

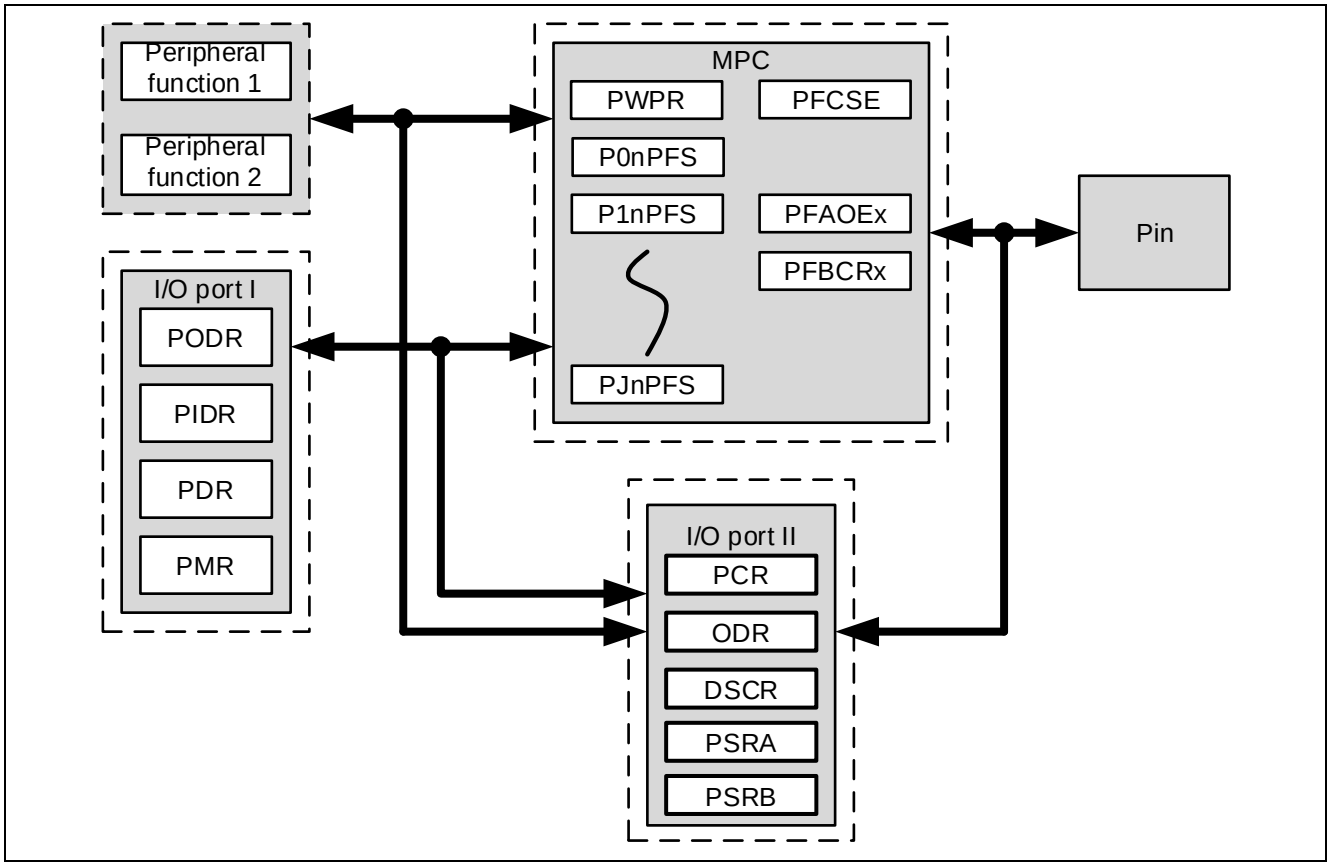


Figure 2.1 Overview of I/O Settings on the RX231 Device

Each I/O port on the RX231 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 RX231 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: • CMOS output • N-channel open drain • P-channel open drain • Hi-Z
ODR1	Open drain control register 1	Selects the port output format from among the following: • CMOS output • N-channel open drain
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 • General-purpose I/O function for the PB6 and PB7 pins • General-purpose I/O function for the PC0 and PC1 pins
PSRB	Port switching register B	This register is used to select which of the following is to be used:*2 • General-purpose I/O function for the PB5, PB3, PB1, and PB0 pins • General-purpose I/O function for the PC3, PC2, PC1, and PC0 pins
DSCR	Drive capacity control register	This register is used to specify the drive capacity. • Normal drive output • High drive output

Notes: 1. The PSRA register is present on only a 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 RX231 Device

Register	Function Name	Description
PRCR	Protect register	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, and the registers related to the battery back-up function. In addition to the above registers, this register also controls write protection on this register itself.

Table 2.5 Registers Related to the Module Stop Function on the RX231 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, MTU, TPU, 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: RSCAN, SCIh, DOC, ELC, comparator B, RSPI0, USB0, RIIC0, CRC, and SCI0/1/5/6
MSTPCRC	Module stop control register C	This register is used to disable the following peripheral functions or to control the operating status: RAM, CAC, IRDA, SCI8/9, 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: CTSU, SSI, SDHI, and Trusted Secure IP
OPCCR	Operating power consumption control register	This register can be used to set either of the following operating modes: • High-speed operating mode • Middle-speed operating mode
SOPCCR	Secondary operating power consumption control register	This register can be used to set either of the following operating modes: • High-speed or middle-speed operating mode • Low-speed operating mode
RSTCKCR	Register for switching the clock source for exiting sleep mode	This register can be used to select the clock source that triggers exiting sleep mode from the following clock sources: • LOCO • HOCO • Main clock oscillator

Table 2.6 Multi-Function Pin Controller Registers on the RX231 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)
PFCSE	CS output enable register	This register is used to enable or disable output on CSn# (n: 0 to 3). Note: This is a register for selecting whether the corresponding pins are to be used for an I/O port or as CSn# output pins.
PFAOE0	Address output enable register 0	This register is used to select whether to enable or disable output on the A8 to A15 buses.
PFAOE1	Address output enable register 1	This register is used to select whether to enable or disable output on the A16 to A23 buses.
PFBCR0	External bus control register 0	This register is used to control the I/O pins of external buses (A0 to A7, D8 to D15, and WR1#/BC1#).
PFBCR1	External bus control register 1	This register is used to control the I/O pins of external buses (WAIT# and ALE).

The following figure shows the sequence of initialization operations needed to use an I/O port on the RX231 device as a general-purpose I/O port.

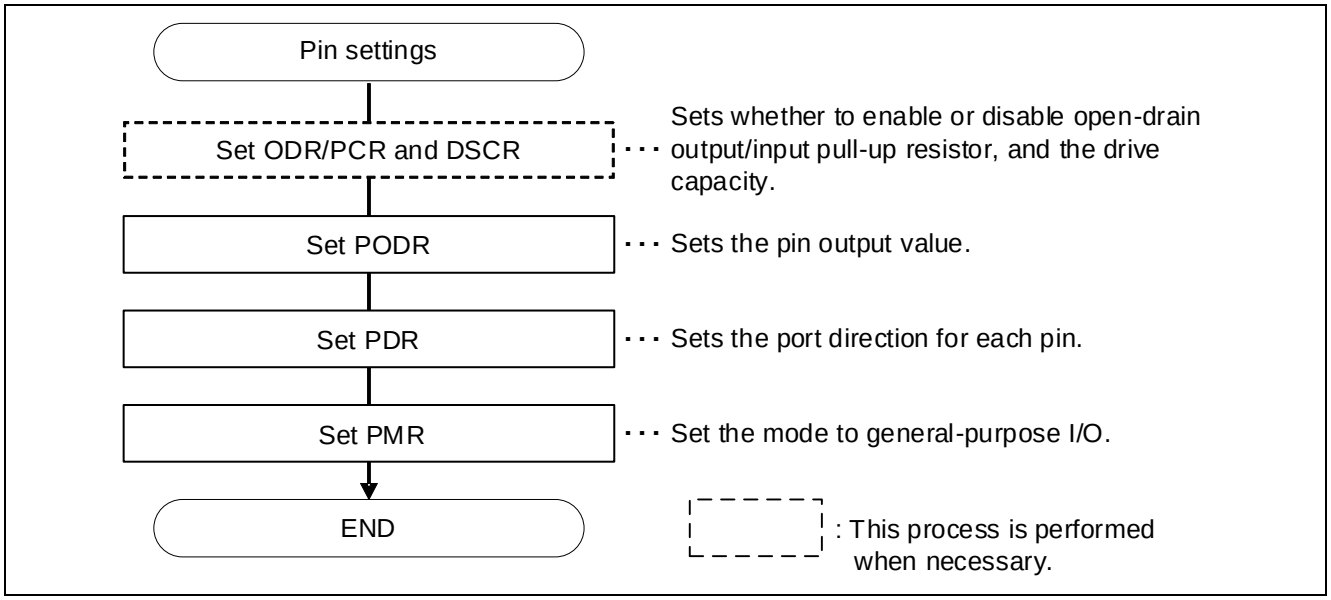
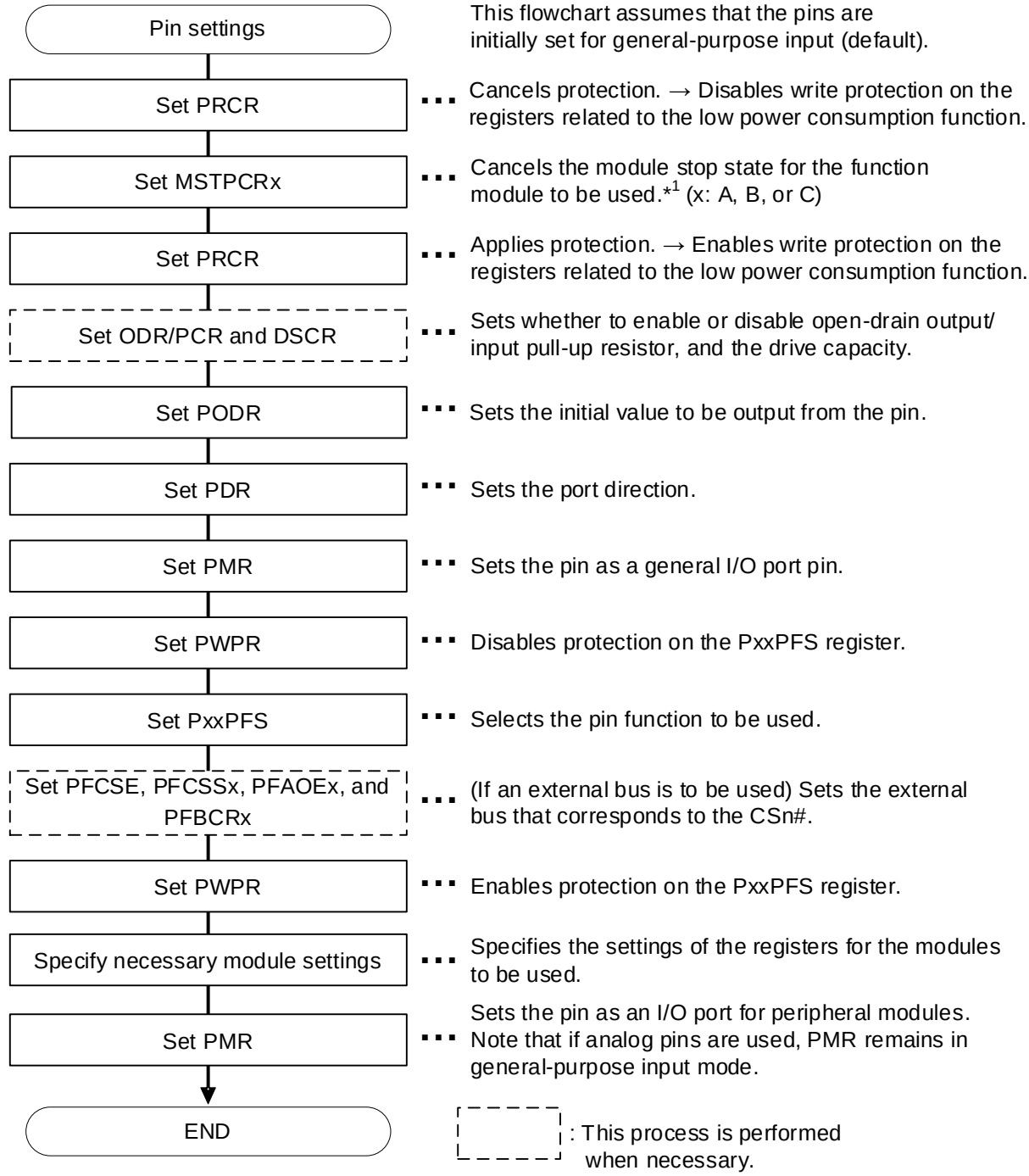


Figure 2.2 Sequence of Operations for Using a General-Purpose I/O Port on the RX231 Device

Figure 2.3 shows the sequence of initialization operations needed to use an I/O port on the RX231 device for pin functions. For example settings for using peripheral functions, including general-purpose input/output, refer to the relevant chapters.



Note: 1. When assigning pin functions to a port, keep in mind that the individual modules are in stopped state by default, except for the DMAC, DTC, and RAM. Therefore, it is necessary to cancel module stop by using the module stop control register (MSTPCRx) of the low power consumption function.

Figure 2.3 Sequence of Operations for Configuring Pin Functions on the RX231 Device

Note that the MSTPCRx 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 MSTPCRx register. DMAC, DTC, and RAM are operational by default.

2.3 Buses

2.3.1 Comparison of Specifications

Table 2.7 shows the differences in the bus specifications between the H8/3048 and RX231 devices.

Table 2.7 Differences in the Bus Specifications Between the H8/3048 and RX231 Devices

Item	H8/3048	RX231
External address space	The external address space can be divided into eight areas (areas 0 to 7) and different bus specifications can be set for each area.	The external address space can be divided into four areas (CS0 to CS3) and different bus specifications can be set for each area.
Area size	128 KB in 1 MB mode*1 2 MB in 16 MB mode*1	16 MB*2
Bus width	The 8-bit or 16-bit access space can be selected for each area.	The 8-bit or 16-bit bus space can be selected for each area.
Bus arbitration function	Each bus master has a fixed priority level: ▲ High External bus master Refresh controller DMAC CPU ▼ Low	Whether to use the fixed or toggled priority level can be selected. The CPU bus, which as a fixed priority level, is an exception.
Other access control	• Wait modes: — Programmable wait mode — Pin auto-wait mode — Pin wait mode 0 — Pin wait mode 1	• Wait control — Ability to set the assert/negate timing for chip select signals (CS0# to CS3#) — Ability to set assert timing for read signal (RD#) and write signals (WR0#, WR#, and WR1#) — Ability to set the data output start/stop timing • Endianness — Ability to set endianness separately for each area

Notes: 1. The size is common to areas 0 to 7.
2. The size is common to the CS0 to CS3 areas.

2.3.2 Bus Configuration

This section compares the bus configurations of the H8/3048 and RX231 devices.

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

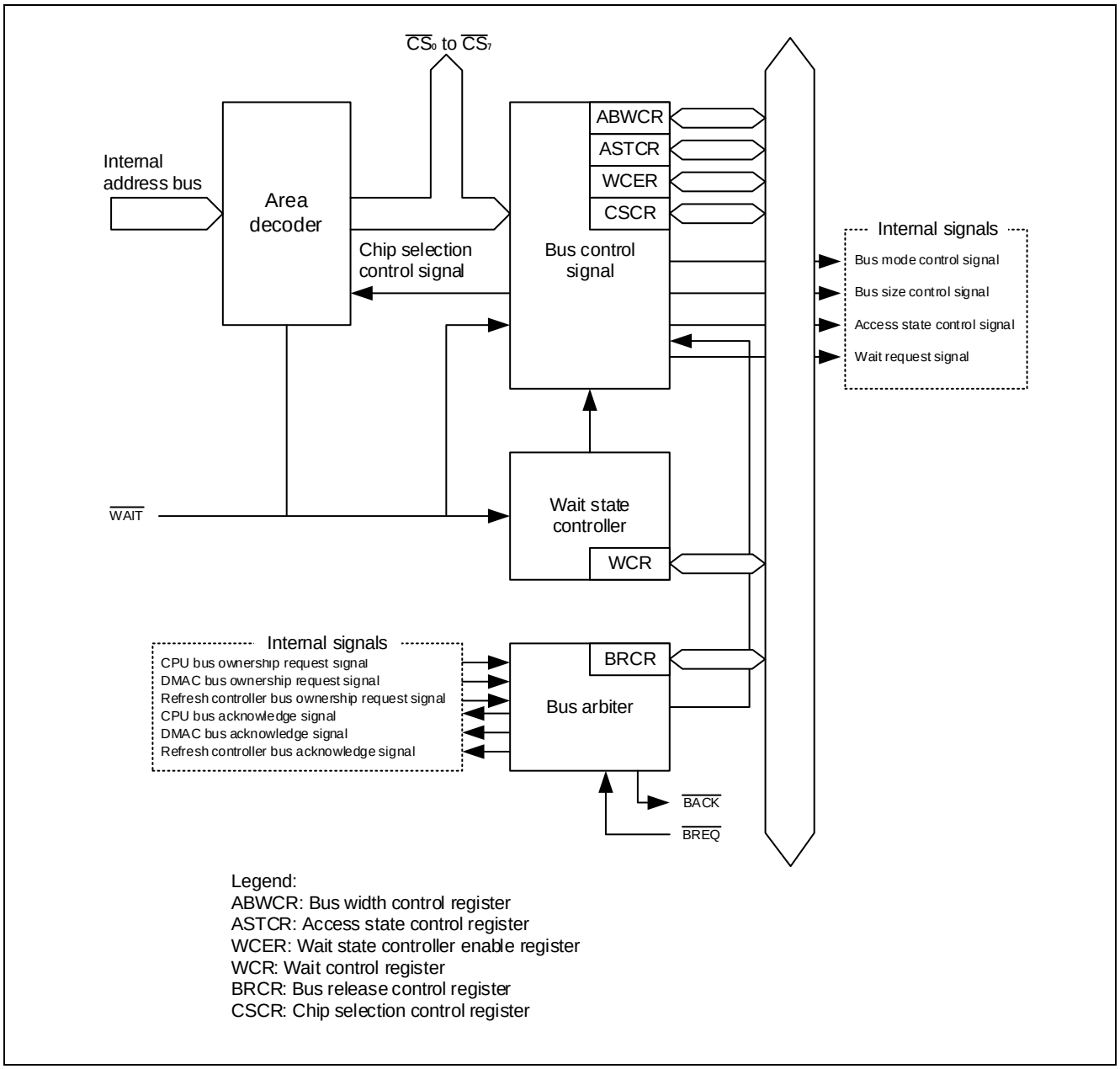
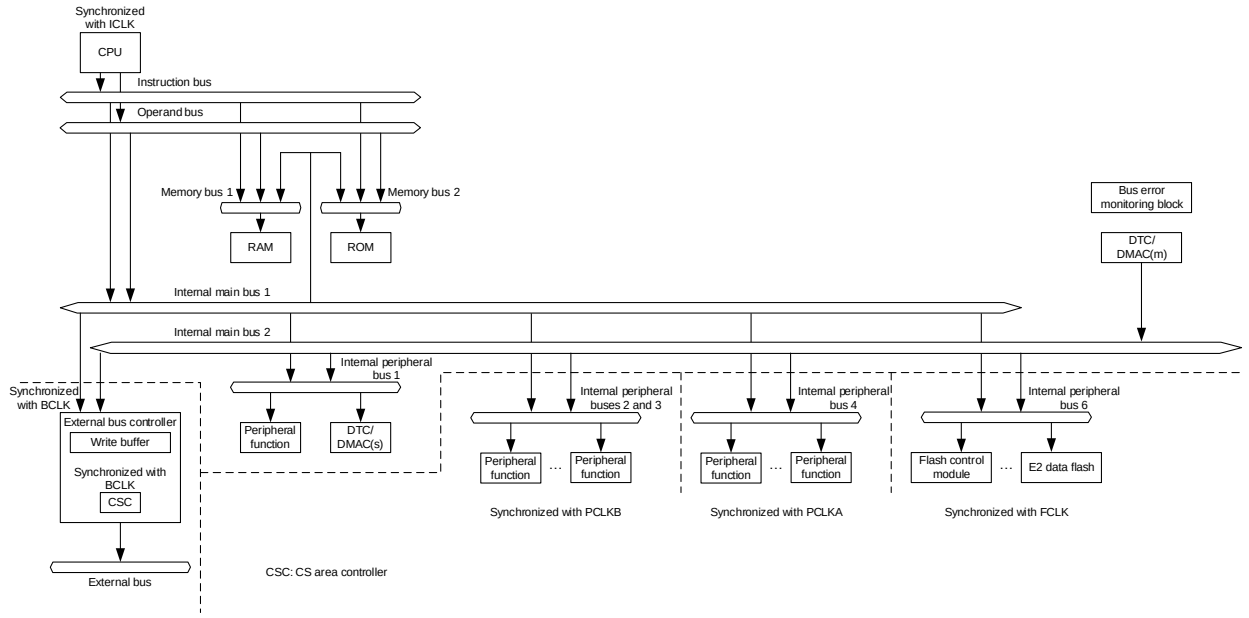


Figure 2.4 Configuration of the Bus Controller of the H8/3048 Device

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 RX231 device.



Notes: 1. Arrows with solid lines indicate the direction of access requests from the bus master.
2. DTC/DMAC(m) indicates master function access and DTC/DMAC(s) indicates register access.

Figure 2.5 Configuration of the Buses of the RX231 Device

The RX231 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 RX231 device. On the RX231 device, as shown in Table 2.8, the clock with which the bus is synchronized differs depending on the bus.

Table 2.8 Buses of the RX231 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, and 4	PCLKB
Internal peripheral bus 3	USB0, RSCAN, and CTSU	PCLKB
Internal peripheral bus 4	MTU2	PCLKA
Internal peripheral bus 6	Flash control module and E2 data flash	FCLK
External buses (CS areas)	External devices	BCLK

Note: 1. ICLK: System clock
PCLKA: Peripheral module clock A
PCLKB: Peripheral module clock B
FCLK: FlashIF clock
BCLK: External bus clock

2.3.3 Register Configuration

This section shows the registers that are used to configure the buses on the H8/3048 and RX231 devices. On the RX231 device, because it has registers related to bus errors, the CPU can be notified of a bus error to generate an interrupt. Bus errors include invalid address access errors and timeout errors.

Table 2.9 List of Bus Control Registers on the H8/3048 and RX231 Devices

H8/3048	RX231
Registers related to CS0 to CS7	**Registers related to CS0 to CS3**
Bus width control register (ABWCR)	CSn control register (CSnCR)*1
Access state control register (ASTCR)	CSn recovery cycle setting register (CSnREC)*1
Wait control register (WCR)	CS recovery cycle insertion enable register (CSRECEN)
Wait state controller enable register (WCER)	CSn mode register (CSnMOD)*1
Bus release control register (BRCR)	CSn wait control register 1 (CSnWCR1)*1
Chip select control register (CSCR)	CSn wait control register 2 (CSnWCR2)*1
—	**Registers related to bus errors**
	Bus error status clearing register (BERCLR)
	Bus error monitoring enable register (BEREN)
	Bus error status register 1 (BERSR1)
	Bus error status register 2 (BERSR2)
	Bus priority control register (BUSPRI)

Note: 1. n = 0 to 3

2.3.4 Wait Settings

When configuring bus settings, a wait must be set in accordance with the specifications of the device to be connected. To set a wait on the H8/3048 device, select an appropriate wait mode. Table 2.10 lists the wait modes available on the H8/3048 device. To set a wait on the RX231 device, configure the registers listed in Table 2.11.

Table 2.10 Wait Modes Available on the H8/3048 Device

Wait Mode	Description
Pin wait mode 0	Wait states can be set from only the #WAIT pin.
Programmable wait mode	Wait states are always inserted as configured with the WC1 and WC0 bits in the WCR register. Wait states cannot be inserted from the #WAIT pin.
Pin wait mode 1	Wait states are always inserted as configured with the WC1 and WC0 bits in the WCR register. In addition, wait states can also be inserted from the #WAIT pin.
Pin auto-wait mode	Wait states are inserted from the #WAIT pin as configured with the WC1 and WC0 bits in the WCR register.

Table 2.11 Setting Items for the External Bus Interface Registers on the RX231 Device (CS Area Settings)

Register	Symbol	Setting Item
CSnMOD*1	EWENB	Whether to enable an external wait*2
CSnWCR1*1	CSPWWAIT	Page write cycle wait
	CSPRWAIT	Page read cycle wait
	CSWWAIT	Normal write cycle wait
	CSRWAIT	Normal read cycle wait
CSnWCR2*1	CSROFF	Read-access CS extension cycle wait
	CSWOFF	Write-access CS extension cycle wait
	WDOFF	Write data output CS extension cycle wait
	AWAIT	Address cycle wait
	RDON	RD assert wait
	WRON	WR assert wait
	WDON	Write data output wait
	CSON	CS assert wait
CSnREC*1	RRCV	Read recovery cycle
	WRCV	Write recovery cycle
CSRECEN	RCVEN0 to RCVEN7	Separate bus recovery cycle insertion enable 0 to 7
	RCVENM0 to RCVENM7	Multiplexed bus recovery cycle insertion enable 0 to 7

Notes: 1. n = 0 to 3
2. Wait cycles can be inserted using the WAIT# signal by enabling external wait.

Make sure that the set value of the wait cycle does not violate the limitations described in the hardware manual for the RX231 Group. For the limitations to follow when setting the value, refer to:

- RX230 Group, RX231 Group User's Manual: Hardware (R01UH0496EJ)
  Section 16.5.7, "Limitations"

2.3.5 External Bus Pin Setting Example

For details on how to configure the settings for the external bus pins, refer to:

- RX230 Group, RX231 Group User's Manual: Hardware (R01UH0496EJ)
  Section 22.3, "Settings for the External Bus Interface"

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 RX231 device, P33 is used for the IRQ3 input pin.

Table 2.12 Interrupt Initial Setting Example (IRQ3 Settings)

Procedure		H8/3048	RX231
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 RX231 devices.

Table 2.13 Differences in the DMAC Specifications Between the H8/3048 and RX231 Devices (1/2)

Item		H8/3048 DMAC	RX231 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		• Internal interrupt • External request • Auto request	The activation source can be selected for each channel. • Software trigger • Interrupt request from a peripheral module • Trigger input to an external interrupt input pin
Channel priority		• Short address: Channel 0A > channel 0B > channel 1A > channel 1B (Channel 0A has the highest priority.) • Full address: Channel 0 > channel 1 (Channel 0 has the highest priority.)	• Channel 0 > channel 1 > channel 2 > channel 3 (Channel 0 has the highest priority.)
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		• Normal transfer mode • Repeat transfer mode*1 • Block transfer mode • I/O mode*1 • Idle mode*1	• Normal transfer mode • Repeat transfer mode • Block transfer mode

Table 2.13 Differences in the DMAC Specifications Between the H8/3048 and RX231 Devices (2/2)

Item	H8/3048 DMAC	RX231 DMACA
DMA interrupt request	• Issued when as many data units as specified by the transfer counter are transferred.	• Transfer end interrupt — Generated when as many data units as specified by the transfer counter are transferred. • Transfer escape end interrupt — Generated when as much data as the repeat size is transferred. — Generated when the extended repeat area overflows.
Low power consumption	The modules can be placed in standby status.	The modules can be placed in stopped state.
Other items	—	• Extended repeat area function • Event link function

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.

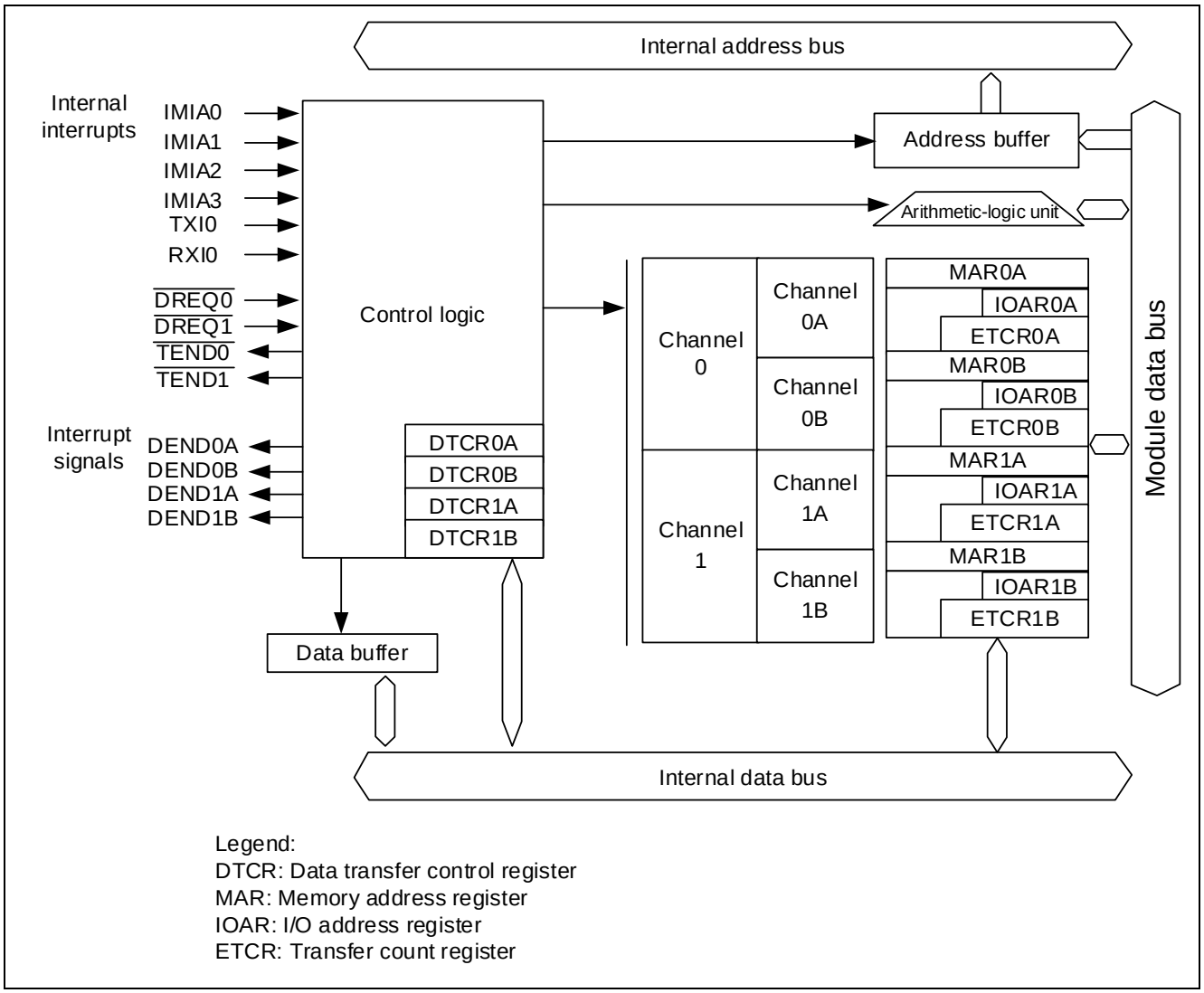


Figure 2.6 Block Diagram for the DMAC of the H8/3048 Device

The following figure shows the block diagram for the DMACA of the RX231 device.

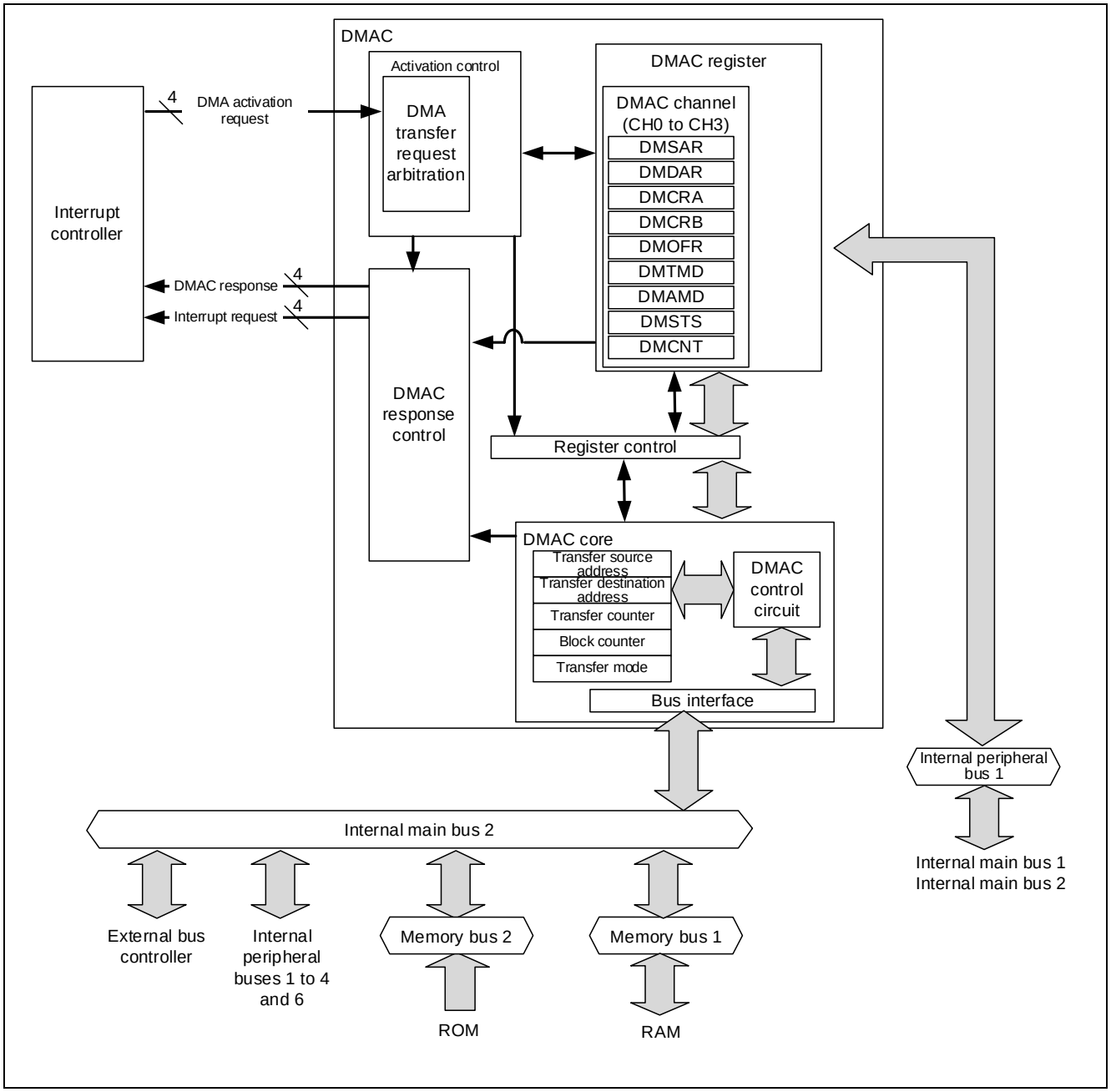


Figure 2.7 Block Diagram for the DMACA of the RX231 Device

2.5.3 List of Registers

The following table lists the registers for the DMA controllers of the H8/3048 and RX231 devices.

Table 2.14 List of Registers for the DMA Controller

H8/3048	RX231
DMAC	**DMACA**
Memory address register	DMA module start register (DMAST)
Short address mode: MARkm	DMA transfer source register n (DMACn.DMSAR)
Full address mode: MARkm and MARkm	DMA transfer destination register n (DMACn.DMDAR)
I/O address register	DMA transfer count register n (DMACn.DMCRA)
Short address mode: IOARkm	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: ETCRkm	DMA address mode register n (DMACn.DMAMD)
Full address mode: ETCRkm and ETCRkm	DMA offset register (DMACn.DMOFR)
Data transfer control register	DMA transfer enable register n (DMACn.DMCNT)
Short address mode: DTCRkm	DMA software start register n (DMACn.DMREQ)
Full address mode: DTCRkm and DTCRkm	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 2.15 compares the DMA controller activation sources of the H8/3048 and RX231 devices.

Table 2.15 Differences of DMA Controller Activation Sources Between the Devices

Activation Sources				H8/3048 DMAC				RX231 DMACA
				Short address mode		Full address mode		
				Channel 0A, 1A	Channel 0B, 1B	Normal	Block	All transfer modes
Internal interrupt	High-function timer	ITU compare match/ input capture interrupt (H8/3048)	IMIA0 to IMIA3	○		×	○	
		MTU compare match/ input capture interrupt (RX231)	TGIA0 to TGIA4					○
		TPU compare match/ input capture (RX231)	TGI0A to TGA5A					○
	SCI0 Transmit data empty interrupt Receive data full interrupt		TXI0	○		×		○
			RXI0	○		×		○
	Interrupts from peripheral modules other than the above	On the RX231 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 15.3.1 in the hardware manual for the RX231 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 RX231 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 RX231 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 2.16 shows the transfer mode specifications of the DMA controllers of the H8/3048 and RX231 devices.

Table 2.16 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	• IOAR registers: Non-change • MAR registers: Increment, decrement, or non-change can be selected.
		Idle mode		65,536	IOAR and MAR registers: Non-change
		Repeat mode		256	• IOAR registers: Non-change • MAR registers: Increment, decrement, or non-change can be selected.
	Full address mode	Normal mode		65,536	• MARA and MARB registers: Increment, decrement, or non-change can be selected.
		Block transfer mode (Maximum number of data units per block: 255)		65,536	
RX231 DMACA	Normal transfer mode		1 byte 1 word 1 longword	65,535	DMSAR and DMDAR registers: Increment, decrement, non-change, or increment-by-offset (DMAC0 only) can be selected for each register.
	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 RX231 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 RX231 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 RX231 device in each transfer mode.

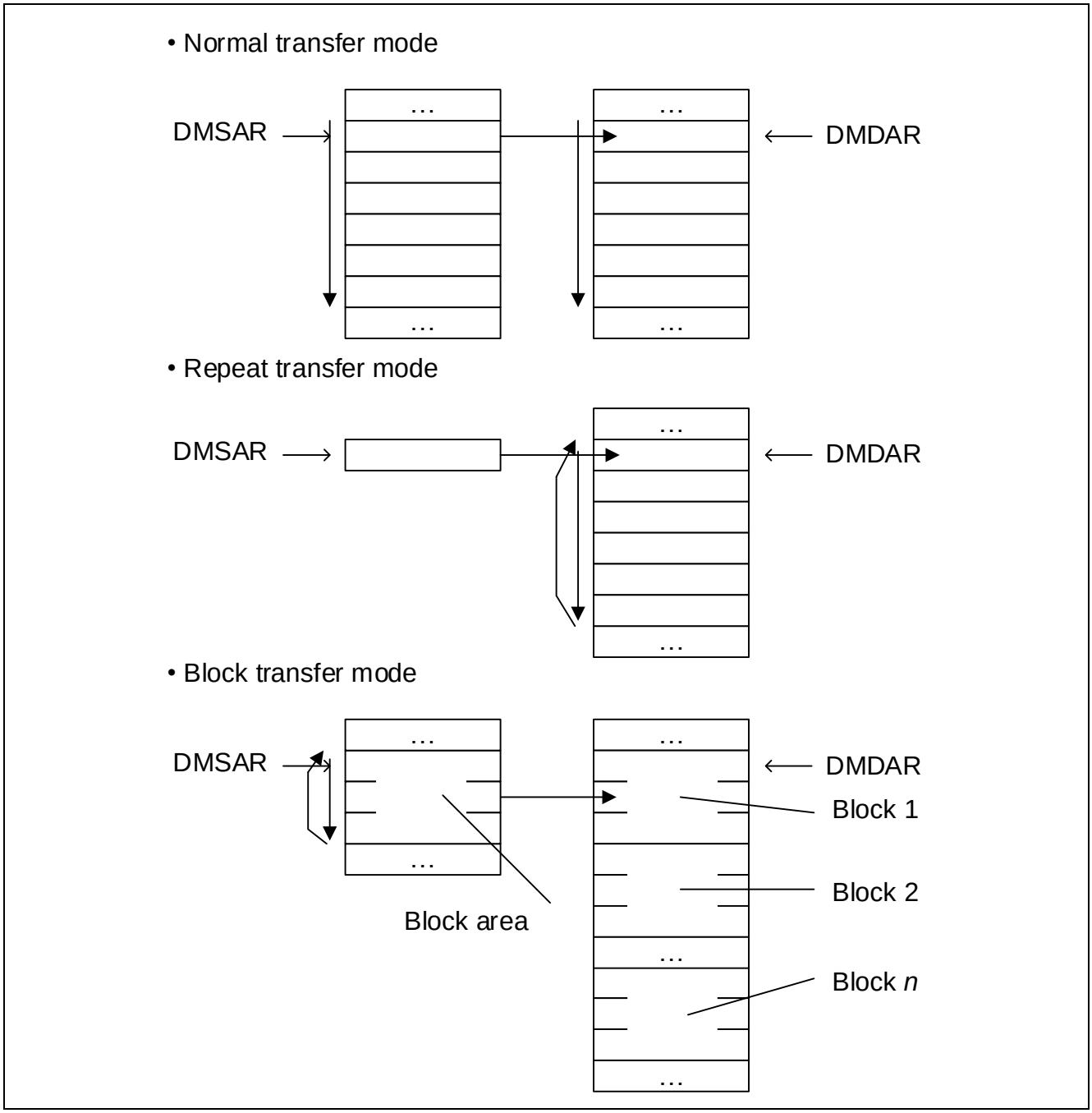


Figure 2.8 Example Operations of the RX231 Device in Transfer Modes

- 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 RX231 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 RX231 device provides two: SCIg and SCIh. SCIg 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. SCIg also supports the simplified I²C bus interface (single master operation) and simplified SPI bus interface. SCIh has all functions of SCIg and extended serial interface functions. For details on the transfer modes not available on the H8/3048 device, refer to the “RX230 Group, RX231 Group User's Manual: Hardware”.

Table 2.17 Differences in the Serial Communications Interface Between the Devices (1/2)

Item		H8/3048	RX231
Number of channels		2 (SCI0 and SCI1)	7 (SCIg: SCI0, SCI1, SCI5, SCI6, SCI8, and SCI9; SCIh: SCI12)
Serial communication modes		Asynchronous Clock-synchronous Smartcard interface (SCI0 only)	Asynchronous Clock-synchronous Smartcard interface Simplified I²C bus Simplified SPI bus
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 I²C bus supports only MSB-first.)
Interrupt source		Transmit data empty Transmit end Receive data full Receive error	Transmit data empty Transmit end Receive data full Receive error End of generating start, restart, or stop conditions (for simplified I²C mode only)
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.	

Table 2.17 Differences in the Serial Communications Interface Between the Devices (2/2)

Item		H8/3048	RX231
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 (channel SCI5 or SIC6 only).
	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 on SCIh (SCI12) only
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 2.18 List of SCI Registers (1/2)

H8/3048	RX231	Difference
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)	
Serial control register (SCR)	Serial control register (SCR)	
Serial status register (SSR)	Serial status register (SSR)	
Bit rate register (BRR)	Bit rate register (BRR)	
Smart card mode register (SCMR)	Smart card mode register (SCMR)	

Table 2.18 List of SCI Registers (2/2)

H8/3048	RX231	Difference
— (There is no corresponding register.)	Modulation duty register (MDDR)	New register added on the RX231
	Serial extended mode register (SEMR)*1	
	Noise filter setting register (SNFR)*1	
	I^2C mode registers 1 to 3 (SIMR1 to SIMR3)*1	
	I^2C status register (SISR)	
	SPI mode register (SPMR)*1	
	Extended serial mode enable register (ESMER)	
	Control registers 0 to 3 (CR0 to CR3)	
	Port control register (PCR)	
	Interrupt control register (ICR)	
	Status register (STR)	
	Status clearing register (STCR)	
	Control field 0 data register (CF0DR)	
	Control field 0 compare enable register (CF0CR)	
	Control field 0 receive data register (CF0RR)	
	Primary control field 1 data register (PCF1DR)	
	Secondary control field 1 data register (SCF1DR)	
	Control field 1 compare enable register (CF1CR)	
	Control field 1 receive data register (CF1RR)	
	Timer control register (TCR)	
	Timer mode register (TMR)	
	Timer prescaler register (TPRE)	
	Timer counter register (TCNT)	

Note: 1. Some bit fields in these registers are used for asynchronous or clock-synchronous serial communication settings.

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 RX231 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 RX231 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, MTU2a, and TPUa)

2.7.1 Comparison of Specifications

The MTU2a and TPUa on the RX231 device are based on the ITU of the H8/3048 device, and provide more advanced features. Some setting items supported by the ITU are not available on the TPUa but they are available with the MTU2a. For I/O pins, as shown in Table 2.20, the MTU2a and TPUa on the RX231 device have all I/O pins of the ITU on the H8/3048 device.

Table 2.19 Comparison of Specifications (Functions) of the ITU on the H8/3048 Device, and the MTU2a and TPUa on the RX231 Device (1/2)

Item		H8/3048	RX231	
		ITU	MTU2a	TPUa
Input clock	Internal clock	Internal clock: φ, φ/2, φ/4, and φ/8	MTU2a and TPUa support: PCLK/1, PCLK/4, PCLK/16, PCLK/64, PCLK/256, and PCLK/1024 TPUa also supports: PCLK/4096	
	External clock	TCLKA, TCLKB, TCLKC, and TCLKD	MTCLKA, MTCLKB, MTCLKC, and MTCLKD	TCLKA, TCLKB, TCLKC, and TCLKD
Number of channels		5 (channels 0 to 4)	6 (MTU0 to MTU5)	6 (TPU0 to TPU5)
Buffer register		Supported on channels 3 and 4 only	Supported on MTU0, MTU3, and MTU4 only	Supported on TPU0 and TPU3 only
Counter clearing source		Compare match and input capture	Compare match and input capture	Compare match and input capture
Compare match output		0, 1, or Toggle (Toggle is not supported on channel 2.)	0, 1, or Toggle	0, 1, or Toggle
Synchronization		Channels 0 to 4	MTU0 to MTU4	TPU0 to TPU5
PWM mode 1		Channels 0 to 4	MTU0 to MTU4	TPU0 to TPU5
Complementary PWM mode		Channels 3 and 4	MTU3 and MTU4	—
Reset-synchronized PWM mode		Channels 3 and 4	MTU3 and MTU4	—
Phase counting mode		Channel 2	MTU1 and MTU2	TPU1, TPU2, TPU4, and TPU5
DMAC activation		Channels 0 to 3	MTU0 to MTU4	TPU0 to TPU5
Number of interrupt sources		15	28	26
Low power consumption		The modules can be placed in standby state.	The modules can be placed in stopped state.	The modules can be placed in stopped state.
PWM mode 2		—	MTU0 to MTU2	—
AC synchronous motor drive mode		—	MTU0, MTU3, and MTU4	
DTC activation		—	MTU0 to MTU5	TPU0 to TPU4
A/D conversion start trigger		—	MTU0 to MTU4	TPU0 to TPU4
Event link function (output)		—	20 sources (MTU1 to MTU4)	

Table 2.19 Comparison of Specifications (Functions) of the ITU on the H8/3048 Device, and the MTU2a and TPUa on the RX231 Device (2/2)

Item	H8/3048	RX231	
	ITU	MTU2a	TPUa
Event link function (input)	—	MTU1 to MTU4 Count-start operation Input-capture operation Count-restart operation	—
A/D converter start request delaying function	—	MTU4	
Interrupt skipping	—	MTU3 and MTU4	

Table 2.20 Comparison of Specifications (Pins) of the ITU on the H8/3048 Device, and the MTU2a and TPUa on the RX231 Device

Item	H8/3048	RX231	
	ITU	MTU2a	TPUa
I/O pins	10 pins	16 pins (max.)	16 pins (max.)
Output pins	2 pins (channel 4)	—	—
Input pins	—	3 pins (MTU5)	—

2.7.2 Notes on Replacing the Timer

- If the ITU is to be replaced by the MTU2a, note that the interrupt flags for the timer status registers (TSR) are located in the interrupt controller.
- If the ITU is to be replaced by the TPUa, the interrupt flags for the timer status registers (TSR) are also located in the interrupt controller, although the timer status registers (TSR) themselves have their own interrupt flags.

2.7.3 List of Registers

Table 2.21 List of Registers for the ITU, MTU2a, and TPUa (1/2)

H8/3048	RX231		
ITU	**MTU2a**	**TPUa**	**Difference**
TSTR	MTU.TSTR	TPU.TSTR	Present*1
TSNC	MTU.TSYR	TPU.TSYR	Present*1
TMDR	MTU0.TMDR to MTU4.TMDR	TPU0.TMDR to TPU5.TMDR	Present*1
TFCR		—	Present*1
TOER	MTU.TOER	—	Present*1
TCNT0 to TCNT4	MTU0.TCNT to MTU4.TCNT, MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW	TPU0.TCNT to TPU5.TCNT	Not present
GRA0 to GRA4 and GRB0 to GRB4	MTU0.TGRA to MTU0.TGRF, MTU1.TGRA, MTU1.TGRB, MTU2.TGRA, MTU2.TGRB, MTU3.TGRA to MTU3.TGRD, MTU4.TGRA to MTU4.TGRD, MTU5.TGRU, MTU5.TGRV, and MTU5.TGRW	TPU0.TGRA to TPU0.TGRD, TPU1.TGRA, TPU1.TGRB, TPU2.TGRA, TPU2.TGRB, TPU3.TGRA to TPU3.TGRD, TPU4.TGRA, TPU4.TGRB, TPU5.TGRA, and TPU5.TGRB	Not present
BRA3, BRA4, BRB3, and BRB4	MTU0.TGRC, MTU3.TGRC, MTU4.TGRC, MTU0.TGRD, MTU3.TGRD, MTU4.TGRD, and MTU0.TGRF*2, *4	TPU0.TGRC, TPU3.TGRC, TPU0.TGRD, and TPU3.TGRD*3, *4	—
TOCR	MTU.TOCR1	—	Present*1
	MTU.TOCR2	—	Present*1
TCR0 to TCR4	MTU0.TCR to MTU4.TCR, MTU5.TCRU, MTU5.TCRV, and TCRW	TPU0.TCR to TPU5.TCR	Present*1
TIOR0 to TIOR4	MTU0.TIORH, MTU0.TIORL, MTU1.TIOR, MTU2.TIOR, MTU3.TIORH, MTU3.TIORL, MTU4.TIORH, and MTU4.TIORL	TPU0.TIORH, TPU0.TIORL, TPU1.TIOR, TPU2.TIOR, TPU3.TIORH, TPU3.TIORL, TPU4.TIOR, and TPU5.TIOR	Present*1
TSR0 to TSR4	—	TPU0.TSR to TPU5.TSR	Present*1
TIER0 to TIER4	MTU0.TIER to MTU4.TIER	TPU0.TIER to TPU5.TIER	Present*1

Table 2.21 List of Registers for the ITU, MTU2a, and TPUa (2/2)

H8/3048	RX231		
ITU	**MTU2a**	**TPUa**	**Difference**
—	MTU0.TSR to MTU4.TSR	—	—
	MTU0.NFCR to MTU4.NFCR	TPU0.NFCR to TPU5.NFCR	—
	MTU5.TCNTCMPCLR	TPU0.NFCR to TPU5.NFCR	—
	MTU1.TICCR	—	—
	MTU.TRWER		—
	MTU.TGCR		—
	MTU0.TBTM, MTU3.TBTM, and MTU4.TBTM		—
	MTU.TOLBR		—
	MTU.TCNTS		—
	MTU.TDDR		—
	MTU.TCDR		—
	MTU.TCBR		—
	MTU.TDER		—
	MTU.TWCR		—
	MTU4.TADCR		—
	MTU4.TADCORA and MTU4.TADCORB		—
	MTU4.TADCOBRA and MTU4.TADCOBRB		—
	MTU.TITCR		—
	MTU.TITCNT		—
	MTU.TBTER		—

Notes: 1. Bit assignments are different.
2. The TGRC and TGRD registers for the MTU0, MTU3, and MTU4 modules, and the TRGF register for the MTU0 module can be set to operate as buffer registers.
3. The TGRA and TGRB registers for the TPU0 and TPU3 modules can be set to operate as buffer registers.
4. Possible combinations of TGR registers and buffer registers are as follows: TGRA-TGRC, TGRB-TGRD, and TGRE-TGRF

2.7.4 Module Stop

On the RX231 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 RX231 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 RX231 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.

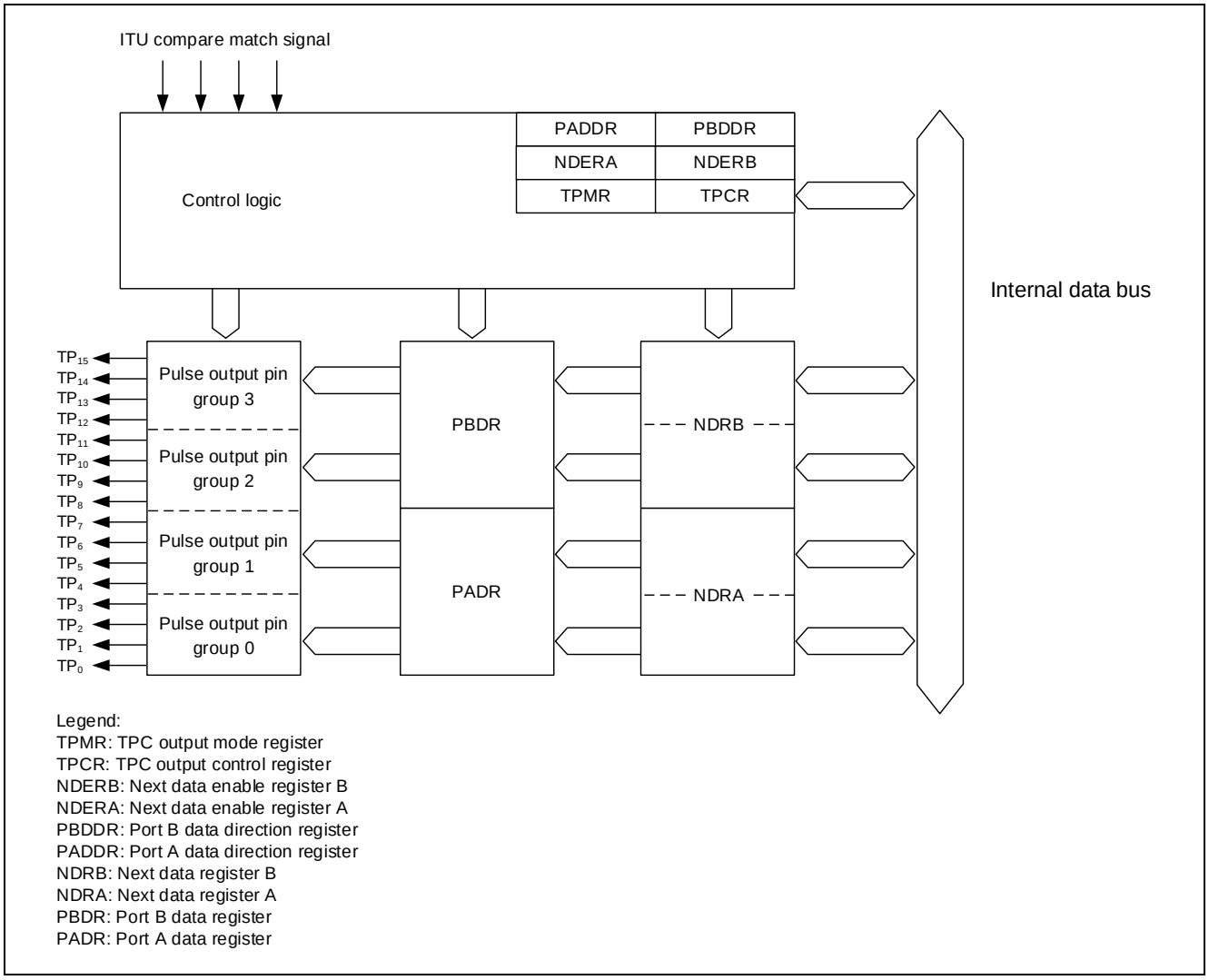


Figure 2.9 Block Diagram of the TPC on the H8/3048 Device

(2) Example settings for achieving a TPC-equivalent function on the RX231 device

Figure 2.10 shows an example of the settings for achieving a TPC-equivalent function on the RX231 device. In this example, general-purpose ports, the DMAC, and the CMT are used to achieve the function.

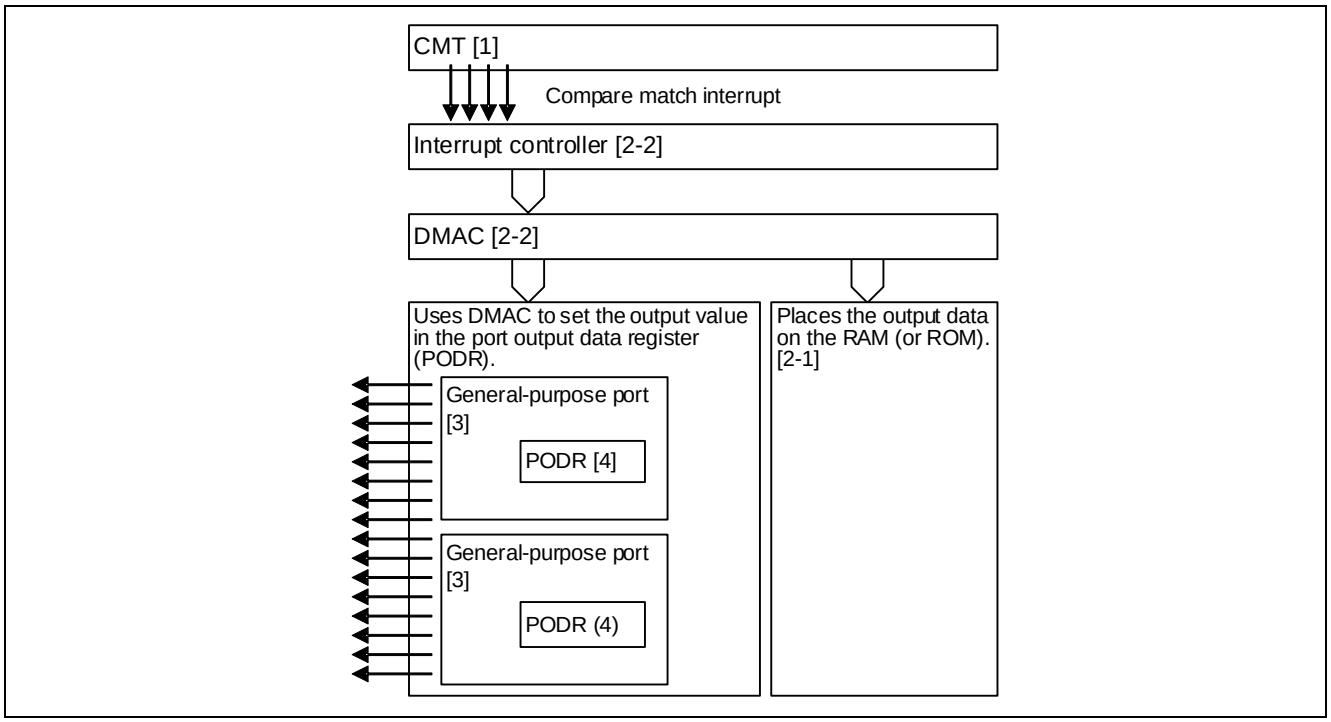


Figure 2.10 Block Diagram of a TPC-Equivalent Function on the RX231 Device

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.

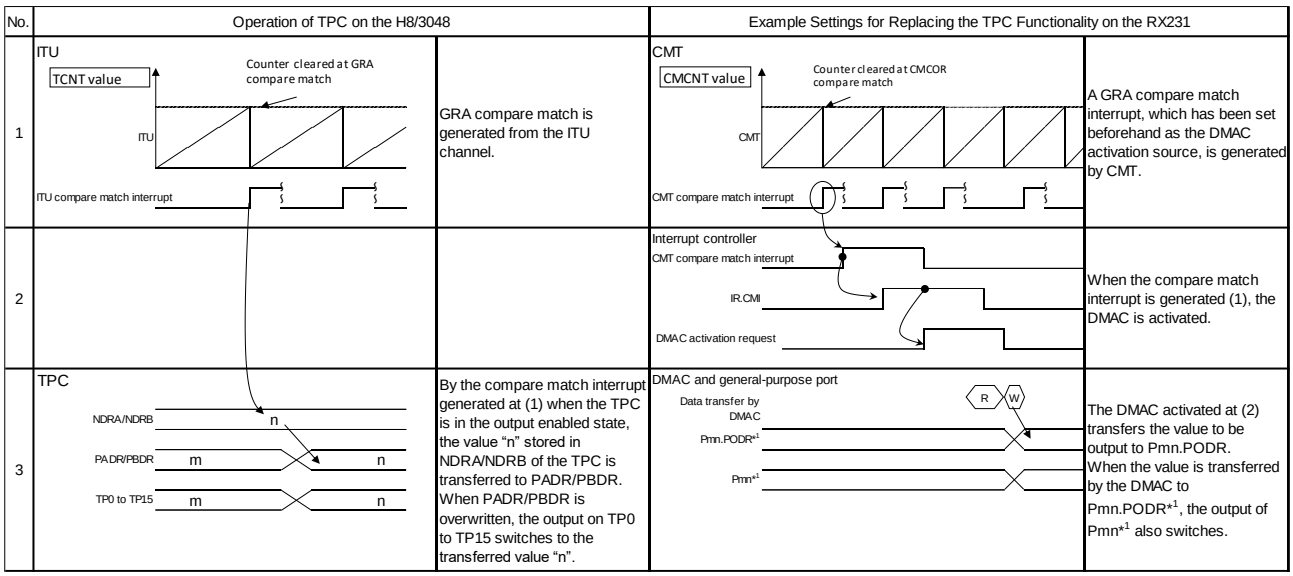
[2] Substitute for the TPC NDRA/NDRB on the H8/3048 device
[2-1]: A variable that stores the output pattern is prepared in the memory.
[2-2]: Upon receiving the compare match, the interrupt controller activates the DMAC.
The DMAC transfers the content of the variable (output pattern) to Pmn.PODR.

[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 RX231 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 RX231 device has two watchdog timer modules, WDTA and IWDTa. IWDTa operates based on an independent dedicated clock. Table 2.22 compares the specifications of these modules.

Table 2.22 Comparison of the Specifications of the Watchdog Timer Modules on the H8/3048 and RX231 Devices (WDT, WDTA, and IWDTa)

	H8/3048	**RX231**	
Item	**WDT**	**WDTA**	**IWDTa**
Count source	System clock	Peripheral module clock B (PCLKB)	IWDTCLK (IWDT-dedicated clock)
Clock division ratio	φ/2, φ/32, φ/64, φ/128, φ/256, φ/512, φ/2048, or φ/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	• Overflow of the up-counter	• Underflow of the down-counter • Refresh performed outside the period in which a refresh is permitted (refresh error)	
Behavior on underflow	—	Reset output or NMI interrupt	
Other items	• Use as an interval timer is possible.	• Use as an interval timer is not possible. • Window function • Auto-start mode • Register start mode	

2.10 A/D Converter

2.10.1 Comparison of Specifications

Table 2.23 compares the specifications of the A/D converters on the H8/3048 and RX231 device.

Table 2.23 Comparison of Specifications of the A/D Converters on the H8/3048 and RX231 Devices

Item	H8/3048 A/D converter	RX231 12-bit A/D converter (S12ADE)
Resolution	10 bits	12 bits
A/D conversion method	Successive approximation	Successive approximation
Conversion time	7.45 µs (min.) per channel (during operation at 18 MHz)	0.83 µs per channel (when the A/D conversion clock ADCLK is operating at 54 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	• Software • External trigger signal (_ADTRG)	• Software trigger • Synchronous trigger (trigger signal from MTU, ELC, or TPU) • Asynchronous trigger (ADTRG0# pin)
Other functions	Sample-and-hold function	• Sample-and-hold function • Variable number of sampling states • Self-diagnosis for 12-bit A/D converter • Addition mode or averaging mode can be selected for A/D conversion values • Analog input disconnection detection assist function • Double trigger mode • A/D data register automatic clearing • Compare (windows A and B) • 16 ring buffers available when the compare function is in use • Event link function
Interrupt source	(A/D conversion end interrupt request)	• The scan end interrupt request (S12ADI0) is generated when a scan ends in modes other than double trigger mode and group-scan mode. • In double trigger mode, the scan end interrupt request (S12ADI0) is generated when two scans end. • 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. • 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.
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 2.24 shows the correspondence between the conversion modes of the H8/3048 and RX231 devices.

Table 2.24 Correspondence of A/D Converter Operating Modes

No.	H8/3048	RX210
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 2.25 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.
RX231	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 2.26 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 RX231 device.

Table 2.26 Comparison of the Functions of the Registers for the AD Converters (1/2)

H8/3048	RX231	
A/D converter	**12-bit A/D converter**	**Difference**
A/D data registers A to D (ADDRA to ADDRD)	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)	Present
	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)	
— (There is no corresponding register.)	A/D data duplication register (ADDBLDR)	New register added on the RX231
	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 (ADCMPLRn) (n = 0 or 1)	

Table 2.26 Comparison of the Functions of the Registers for the AD Converters (2/2)

H8/3048	RX231	
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 RX231
	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 RX231 devices, these registers are 16-bit registers. Note that, on the H8/3048 device, the 10 active bits are left-aligned, whereas on the RX231 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 RX231 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 2.27 Comparison of Specifications of the D/A Converters on the H8/3048 and RX231 Devices

Item	H8/3048 D/A converter	RX231 12-bit D/A converter (R12DA)
Resolution	8 bits	12 bits
Number of input channels	2	2
Conversion speed	10 µs (at a load capacitance of 20 pF)	30 µs, max.
Output voltage	0 V to 255/256 × V_{REF}	0 V to 4095/4096 × V_{REF}*1
D/A output hold function in software standby mode	Implemented	Implemented
Low power consumption	—	The modules can be placed in stopped state.*2

Notes: 1. Voltage in the range from 0.35 V to (AVCC0 - 0.47) V can be output. One of the following can be selected: AVCC0/AVSS, Internal reference voltage/AVSS, or VREFH/VREFL
2. 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 RX231 devices.

Table 2.28 Comparison of Registers on the H8/3048 and RX231 Devices

H8/3048	RX231	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.
2. The RX231 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 RX231 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 2.29 Comparison of Specifications of Flash Memory on the H8/3048F and RX231 Devices

Item		H8/3048F	RX231
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		• 16 KB × 7 (112 KB) • 12 KB × 1 (12 KB) • 512 bytes × 8 (4 KB)	• User area: 2 KB × 256 (512 KB) • Data area: 1 KB × 8 (8 KB)
Operating mode		• Program mode • Program-and-verify mode • Erase mode • Erase-and-verify mode • Pre-write-and-verify mode	• E2 data flash access prohibition mode • ROM/E2 data flash read mode • ROM program/erase mode • E2 data flash program/erase mode
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 • Write: 50 µs or 1000 µs (per byte)*1 • Erase: 1 s or 30 s (per 128 KB (entire block))*1	FCLK: 32 MHz • Write: 52.3 µs or 491 µs (per 8 bytes)*1 • Erase: 66.7 ms or 1469 ms (per 512 KB (entire block))*1
	Maximum rewrite count	100	1000
E2 data flash	Unit of data to write/erase	—	Write: In bytes Erase: In blocks
	Write/erase time	—	FCLK: 32 MHz • Write: 40.8 µs or 376 µs (per byte)*1 • Erase: 12.9 ms or 368 ms (per 8 KB (entire block))*1
	Maximum rewrite count	—	1,000,000 / 100,000*2
Programming mode		• On-board programming Boot mode User program mode • PROM mode	• On-board programming Boot mode — FINE interface — USB interface — SCI interface • Self-programming (single-chip mode) • Off-board programming The user area and data area can be rewritten by using a flash programmer (serial or parallel programmer).
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 RX231 device also provides three low-power modes. On the H8/3048 and RX231 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 2.30 shows the operating states of the internal modules in each low-power mode of the H8/3048 device.

Table 2.30 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 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
φ clock output		φ 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 RX231 device

Table 2.31 shows the operating states of the internal modules in each low-power mode of the RX231 device.

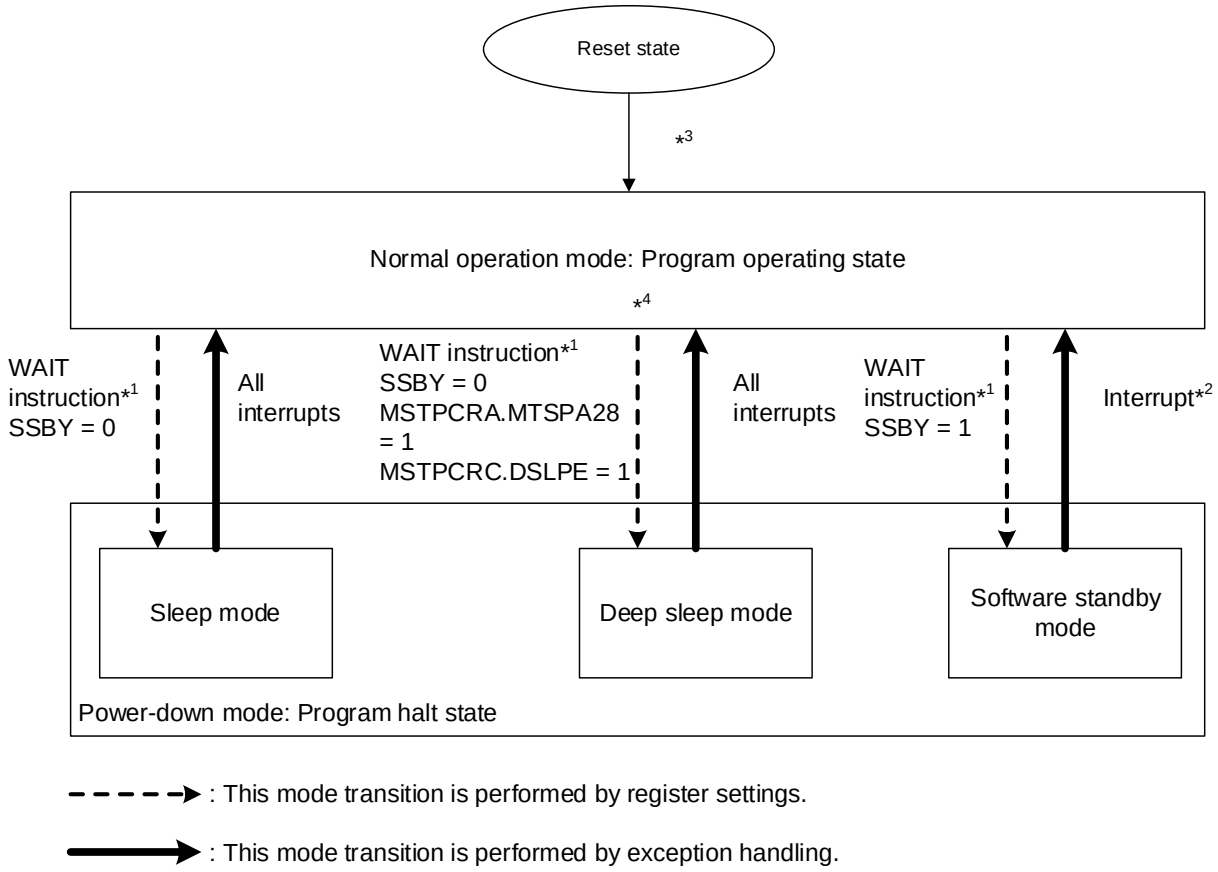
Table 2.31 Operating States of Internal Modules in Low-Power Modes of the RX231 Device

Module	Sleep mode	Deep sleep mode	Software standby mode
Main clock oscillator	Possible to operate	Possible to operate	Stopped
Subclock oscillator	Possible to operate	Possible to operate	Possible to operate
High-speed on-chip oscillator	Possible to operate	Possible to operate	Stopped
Low-speed on-chip oscillator	Possible to operate	Possible to operate	Stopped
IWDT-dedicated on-chip oscillator	Possible to operate	Possible to operate	Possible to operate
PLL	Possible to operate	Possible to operate	Stopped
USB-dedicated PLL	Possible to operate	Possible to operate	Stopped
CPU	Stopped (Settings retained)	Stopped (Settings retained)	Stopped (Settings retained)
RAM (0000 0000h to 0000 FFFFh)	Possible to operate (Settings retained)	Stopped (Settings retained)	Stopped (Settings retained)
Flash memory	Operating	Stopped (Settings retained)	Stopped (Settings retained)
Watchdog timer (WDT)	Stopped (Settings retained)	Stopped (Settings retained)	Stopped (Settings retained)
Independent watchdog timer (IWDT)	Possible to operate	Possible to operate	Possible to operate
Realtime clock (RTC)	Possible to operate	Possible to operate	Possible to operate
Low-power timer (LPT)	Possible to operate	Possible to operate	Possible to operate
Voltage detection circuit (LVD)	Possible to operate	Possible to operate	Possible to operate
Power-on reset circuit	Operating	Operating	Operating
Peripheral modules	Possible to operate	Possible to operate	Stopped (Settings retained)
I/O ports	Operating	Operating	Settings retained

Note: “Possible to operate” indicates that the operating state (operating or stopped) can be controlled by configuring control registers.
“Stopped (Settings retained)” indicates that the module is stopped and the settings of the internal registers are retained.

2.13.2 Mode Transitions

Figure 2.12 shows the transitions of low-power modes on the RX231 device.



Notes: 1.When the WAIT instruction is executed, if an interrupt that triggers cancellation is accepted during state transition to the program halt state, the transition is canceled and interrupt exception handling is performed.

2. External pin interrupt (NMI or IRQ0 to IRQ7), peripheral function interrupt (RTC alarm, RTC cycle, IWDT, voltage monitoring, detection of VBATT pin voltage reduction, USB, or ELC (LPT-dedicated interrupt))
3. Transition from the RESET state becomes the LOCO clock source in normal mode.
4. The mode changes from sleep mode, deep sleep mode, or software standby mode to the normal operating mode upon an interrupt.
   If the mode changes from sleep mode, the clock source to be used after sleep mode ends can be selected.
   For details, refer to the description of the RSTCLKCR register.
   If the mode changes from deep sleep mode or software standby mode, the clock source does not change when the mode changes.

Note: The state always changes to reset state if a RES# pin reset, power-on reset, voltage monitoring reset, WDT reset, IWDT reset, or software reset occurs.

Figure 2.12 Transitions of Modes on the RX231 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 RX231 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 RX231 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 2.32.

Table 2.32 List of Modules That Can Be Stopped

H8/3048	**RX231**
Module standby function	**Module stop function**
MSTCR: Module standby control register	SYSTEM.MSTPCRA: Module stop control register A
ITU	TMR
SCI	MTU
DMA	TPU
Refresh controller	CMT
A/D converter	S12AD
—	DA
	DMAC and DTC
	SYSTEM.MSTPCRB: Module stop control register B
	RSCAN0
	SCI0, SCI1, SCI5, SCI6, and SCI12
	DOC
	ELC
	Comparator B
	RSPI0
	USB0
	RIIC0
	CRC
	SYSTEM.MSTPCRC: Module stop control register C
	RAM
	CAC
	IRDA
	SCI8 and SCI9
	SYSTEM.MSTPCRD: Module stop control register D
	CTSU
	SSI
	SDHI
	Trusted Secure IP

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)
RX230 Group, RX231 Group User's Manual: Hardware (R01UH0496EJ)
RX231 Group Initial Setting (R01AN2185EJ)
RX65N Group, RX651 Group User's Manual: Hardware (R01UH0590EJ)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Mar. 27, 2023	—	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

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/.