RX71M Group

SH7214/SH7216 to RX71M Microcontroller Migration Guide

R01AN4044EJ0100
Rev.1.00
Feb 27, 2018

Summary

This application note describes points requiring special attention, points of difference, etc., that need to be borne in mind when replacing the SH7214 or SH7216 with the RX71M in a user system. For detailed information on each function, refer to the latest version of the User's Manual: Hardware.

In this application note the SH7214 and SH7216 are referred to collectively as the SH7216 Group, and the specifications of the SH7216 are treated as representative. Although there are minor differences in functions and pins among the products composing the SH7216 Group, functionally they are all basically equivalent to the SH7216. This application note therefore applies to the entire SH7216 Group.

Users are encouraged to make use of the available drivers for function modules such as the Ethernet controller, USB, and flash memory.

Target Device

RX71M

Contents

1. CPU Architecture

1.1 Data Format

The SH7216 Group supports double-precision floating-point data, but the RX71M does not.

1.2 System Registers

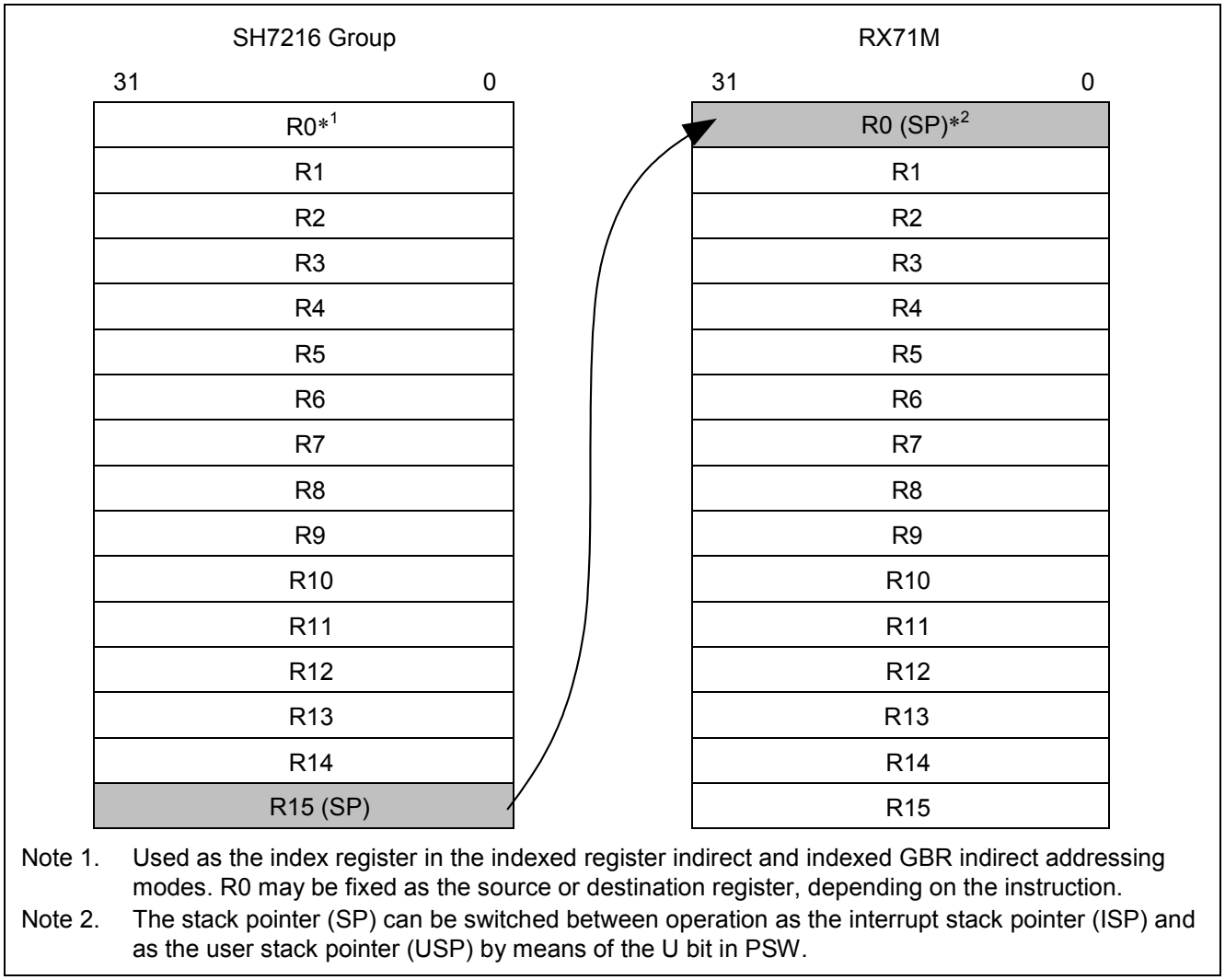
The points of difference between the registers of the SH7216 Group and the RX71M are described below.

1.2.1 General-Purpose Registers

The SH7216 Group and RX71M each have 16 32-bit general-purpose registers. They differ in that the register used as the stack pointer (SP) is different.

- SH7216 Group: R15
- RX71M: R0

Figure 1.1 shows the general-purpose registers of the SH7217 Group and RX71M. On the SH7216 Group R0 is also used as an index register.



Note 1. Used as the index register in the indexed register indirect and indexed GBR indirect addressing modes. R0 may be fixed as the source or destination register, depending on the instruction.

Note 2. The stack pointer (SP) can be switched between operation as the interrupt stack pointer (ISP) and as the user stack pointer (USP) by means of the U bit in PSW.

Figure 1.1 Differences Between General-Purpose Registers

1.2.2 Control Registers

Figure 1.2 shows the points of difference between the CPU registers (other than the general-purpose registers) of the SH7216 Group and the RX71M.

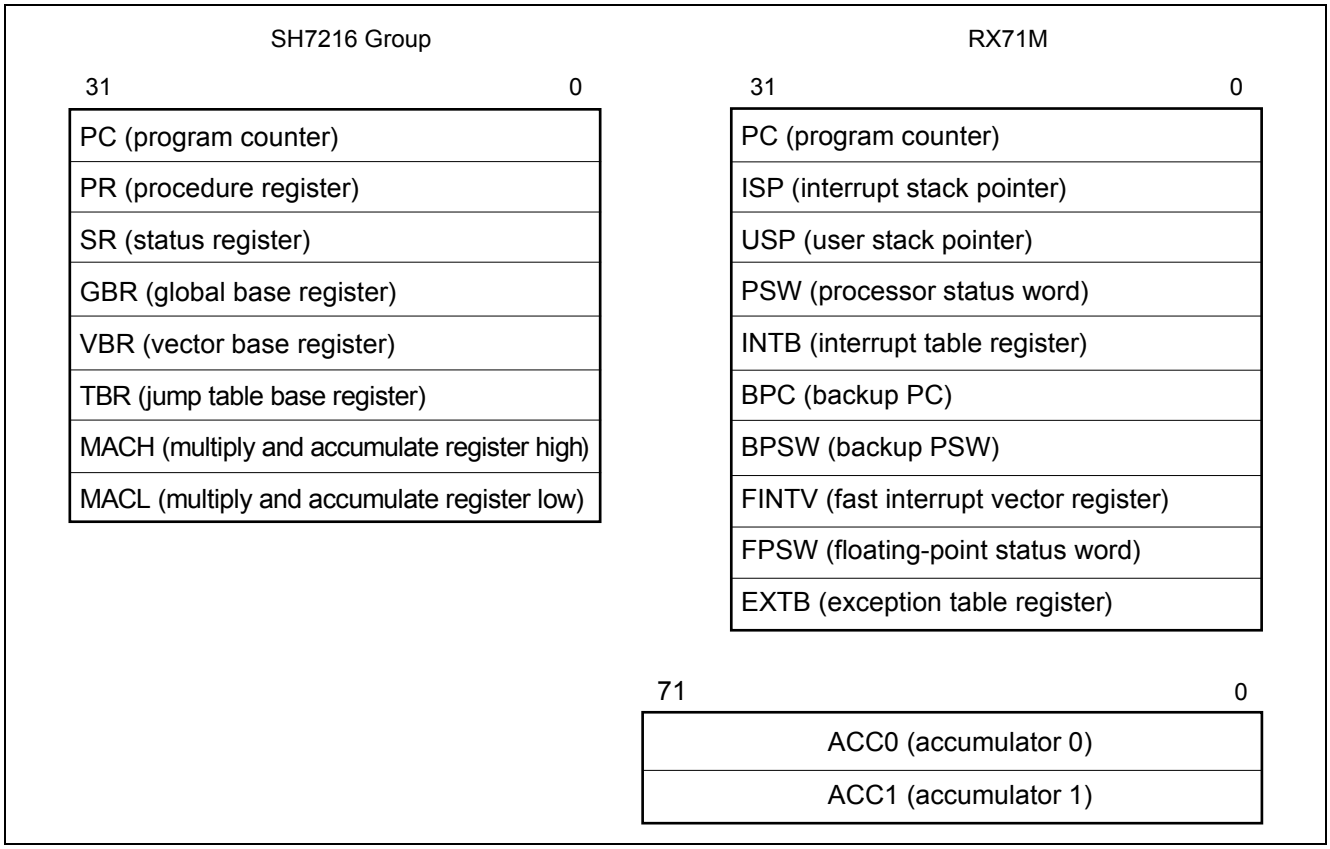


Figure 1.2 Points of Difference Between CPU Registers (Excluding General-Purpose Registers)

The RX71M has no registers corresponding to PR, GBR, and TBR on the SH7216 Group. The ACC0 and ACC1 registers on the RX71M corresponds to MACH and MACL on the SH7216 Group. An outline of the control registers that are implemented on the RX71M but not on the SH7216 Group is presented below.

Table 1.1 RX71M Control Registers Not Present on SH7216 Group

Register Name	Description
Interrupt stack pointer (ISP) User stack pointer (USP)	The RX71M has two types of stack pointers. The type of stack pointer used (ISP or USP) can be switched by means of the stack pointer select bit (U) in the processor status word (PSW) register.
Interrupt table register (INTB)*1	Specifies the start address of the interrupt vector table.
Exception table register (EXTB)*1	Specifies the start address of the exception vector table.
Backup PC (BPC) Backup PSW (BPSW)	The RX71M supports fast interrupts in addition to ordinary interrupts. For fast interrupts, the contents of PC and PSW are saved to dedicated registers (BPC and BPSW), thereby reducing the processing time needed to save the register data.
Fast interrupt vector register (FINTV)	This register specifies the jump destination when a fast interrupt occurs.
Floating-point status word (FPSW)	This register indicates the status of the calculation result (floating-point calculation result) generated by the RX71M's on-chip FPU.

Note 1. The functionality of this register is equivalent to that of VBR on the SH7216 Group.

Figure 1.3 and Table 1.2 show the points of difference between the status registers of the SH7216 Group and the RX71M.

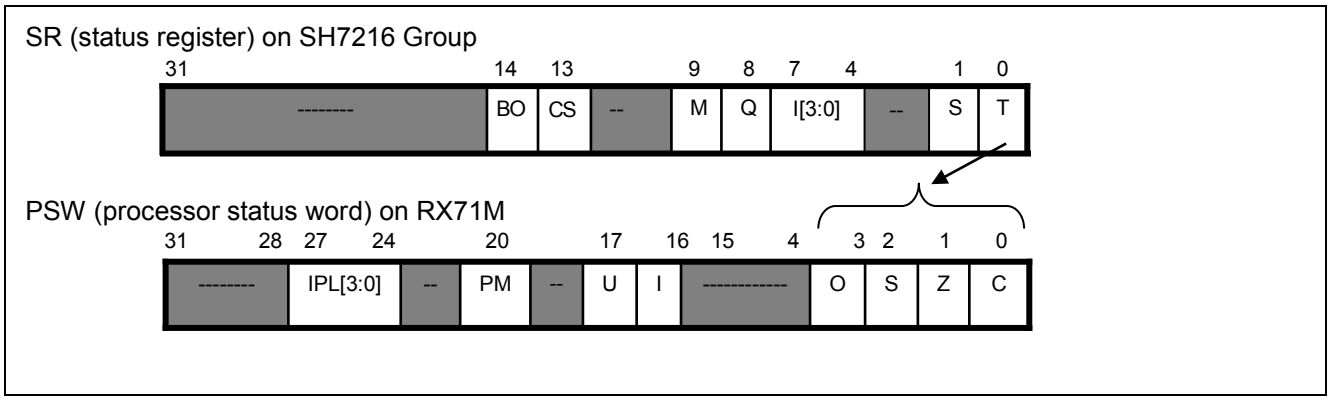


Figure 1.3 Differences Between SR (SH7216 Group) and PSW (RX71M)

Table 1.2 Differences Between SR (SH7216 Group) and PSW (RX71M)

SH Bit Name	RX Bit Name	Description
T	C Z S O	The calculation result (true/false, carry/borrow, etc.) indicated by the T bit on the SH7216 Group is shown by four flags (C, Z, S, and O) on the RX71M. C: Carry flag (0/1 = No carry has occurred./A carry has occurred.) Z: Zero flag S: Sign flag O: Overflow flag
S	—	Controls the functionality that prevents overflows during ALU arithmetic operations performed by the DSP unit of the SH7216 Group. On the RX71M there is no bit corresponding to the S bit, and the occurrence of an overflow during a floating-point operation is reported by the FPSW flag. It is also possible to perform exception handling when an overflow occurs.
I[3:0]	IPL[3:0]	These are the interrupt mask bits. Both the SH7216 Group and the RX71M support level settings from 0 (lowest) to 15 (highest). Only interrupts with a priority level higher than this setting are accepted.
Q	—	The Q bit is used by the DIV0U, DIV0S, and DIV1 instructions on the SH7216 Group. There is no corresponding bit on the RX71M.
M	—	The M bit is used by the DIV0U, DIV0S, and DIV1 instructions on the SH7216 Group. There is no corresponding bit on the RX71M.
CS	—	On the SH7216 Group the CS bit is used in CLIP instruction execution, but there is no equivalent bit on the RX71M.
BO	—	On the SH7216 Group the BO bit indicates that a register bank has overflowed, but there is no equivalent bit on the RX71M.
—	I	Interrupt enable bit 0: Interrupts are disabled. 1: Interrupts are enabled. This bit is used to enable interrupt requests on the RX71M. The initial state is 0, so it is necessary to set this bit to 1 in order to accept interrupts. It is set to 1 when a WAIT instruction is accepted and cleared to 0 when an exception is accepted. Note that the interrupt status flag of the interrupt controller is reset when an interrupt request occurs, regardless of the setting of this bit.
—	U	This bit specifies the stack pointer used by the RX71M. 0: Interrupt stack pointer (ISP) 1: User stack pointer (USP) This bit is cleared to 0 when an exception is accepted. It is set to 1 when a transition from supervisor mode to user mode occurs.
—	PM	This bit specifies the processor mode of the RX71M. 0: Supervisor mode 1: User mode This bit is cleared to 0 when an exception is accepted.

1.3 Option-Setting Memory

The RX71M is provided with an option-setting memory area containing registers for selecting the microcontroller state after a reset of the endian mode, watchdog timer operation, etc. Option-setting memory is allocated in the configuration setting area and user boot area of the flash memory, and the available setting methods are different for the two areas. For details, see the User's Manual: Hardware.

1.3.1 Outline of Option-Setting Memory

Figure 1.4 shows an outline of the option-setting memory area.

Address	Register name	Register overview
—	—	—
0012 0040h to 0012 0043h	Serial programmer command control register (SPCC)	Used to make settings for serial ID code protection, serial programmer connection, block erasure commands, programming commands, and read commands.
—	—	—
0012 0048h to 0012 004Bh	TM enable flag register (TMEF)	Used to enable or disable the TM function.
—	—	—
0012 0050h to 0012 005Fh	OCD/serial programmer ID setting register (OSIS)	Used to store the ID code for the ID code protection function of the OCD/serial programmer.
0012 0060h to 0012 0063h	TM identification data register (TMINF)	Area in which the user can store any 32 bits of data, such as a code used to identify the program stored in the area for which the TM function is enabled.
0012 0064h to 0021 0067h	Endian select register (MDE)	Used to specify the endian setting of the CPU.
0012 0068h to 0012 006Bh	Option function select register 0 (OFS0)	Used to make settings for the independent watchdog timer (IWDT) and watchdog timer (WDT).
0012 006Ch to 0012 006Fh	Option function select register 1 (OFS1)	Used to make settings for voltage monitoring after a reset and HOCO oscillation.
—	—	—
FF7F FFE8h to FF7F FFEFh	UB code A	Area for storing two 32-bit words required when using user boot mode (5573 6572h and 426F 6F74h).
FF7F FFF0h to FF7F FFF7h	UB code B	Area for storing two 32-bit words required when using user boot mode (FFFF FF07h and 0008 C04Ch).
—	—	—

Figure 1.4 RX71M Option-Setting Memory Area

Figure 1.5 to Figure 1.8 show sample settings for the option-setting memory.

```
/* Settings for big-endian */
#define __BIG
#pragma address MDE_REG = 0x00120064 // MDE register
#ifdef __BIG
      const unsigned long MDE_REG = 0xFFFFFFF8; // big
#else
      const unsigned long MDE_REG = 0xFFFFFFFF; // little
#endif
```

Figure 1.5 RX71M Endian Setting Example

```
/* Settings for enabling serial programmer connection and ID code protection after a reset. */
#pragma address SPCC_REG = 0x00120040 // SPCC register
const unsigned long SPCC_REG = 0x1EFFFFFF;

/* ID code settings for OCD/serial programmer */
/* ID1 =0x01, ID2 =0x02, ID3 =0x03, ID4 =0x04 */
/* ID5 =0x05, ID6 =0x06, ID7 =0x07, ID8 =0x08 */
/* ID9 =0x09, ID10=0x0A, ID11=0x0B, ID12=0x0C */
/* ID13=0x0D, ID14=0x0E, ID15=0x0F, ID16=0x10 */
#pragma address OSIS1_REG = 0x00120050 // OSIS register
const unsigned long OSIS1_REG = 0x04030201; // ID1, ID2, ID3, ID4

#pragma address OSIS5_REG = 0x00120054 // OSIS register
const unsigned long OSIS5_REG = 0x08070605; // ID5, ID6, ID7, ID8

#pragma address OSIS9_REG = 0x00120058 // OSIS register
const unsigned long OSIS9_REG = 0xC0B0A09; // ID9, ID10, ID11, ID12

#pragma address OSIS13_REG = 0x0012005C // OSIS register
const unsigned long OSIS13_REG = 0x100F0E0D; // ID13, ID14, ID15, ID16
```

Figure 1.6 RX71M OCD/Serial Programmer Setting Example

```
/* UB code A settings */
#pragma address UBA1_REG = 0xFF7FFFE8 // UB code A_1 register
const unsigned long UBA1_REG = 0x55736572;
#pragma address UBA2_REG = 0xFF7FFFEC // UB code A_2 register
const unsigned long FAW2_REG = 0x426F6F74;
```

Figure 1.7 RX71M UB Code A Setting Example

```
#pragma address OFS1_REG = 0x0012006C // OFS1 register
const unsigned long OFS1_REG = 0xFFFFFFFF;

#pragma address OFS0_REG = 0x00120068 // OFS0 register
const unsigned long OFS0_REG = 0xFFFFFFFF;
```

Figure 1.8 RX71M OFS0 and OFS1 Setting Example

1.3.2 Endian Setting

The SH7216 Group is fixed in big-endian mode. On the RX71M, instructions are fixed in little-endian, and the data order is selectable between little-endian and big-endian. The endian setting is specified by means of the endian select bits (MDE[2:0]) in the MDE register in the option-setting memory.

When switching from the SH7216 Group to the RX71M, it is possible to use big-endian order by specifying big-endian in the option settings of the genuine Renesas compiler. This allows migration without the need to be conscious of endianness in the user program.

The endian setting can be switched for each CS area in the external address space. However, instruction code cannot be allocated to an external space with an endian setting that differs from that of the MCU. When allocating instruction code to an external space, ensure that an area with the same endian setting as the MCU is used. (For details, see the User's Manual: Hardware.)

Endian settings based on the compiler option setting are illustrated in Figure 1.9. The files generated automatically based on the compiler option setting have been confirmed to run in the environment described in 3.1, Operating Environment.

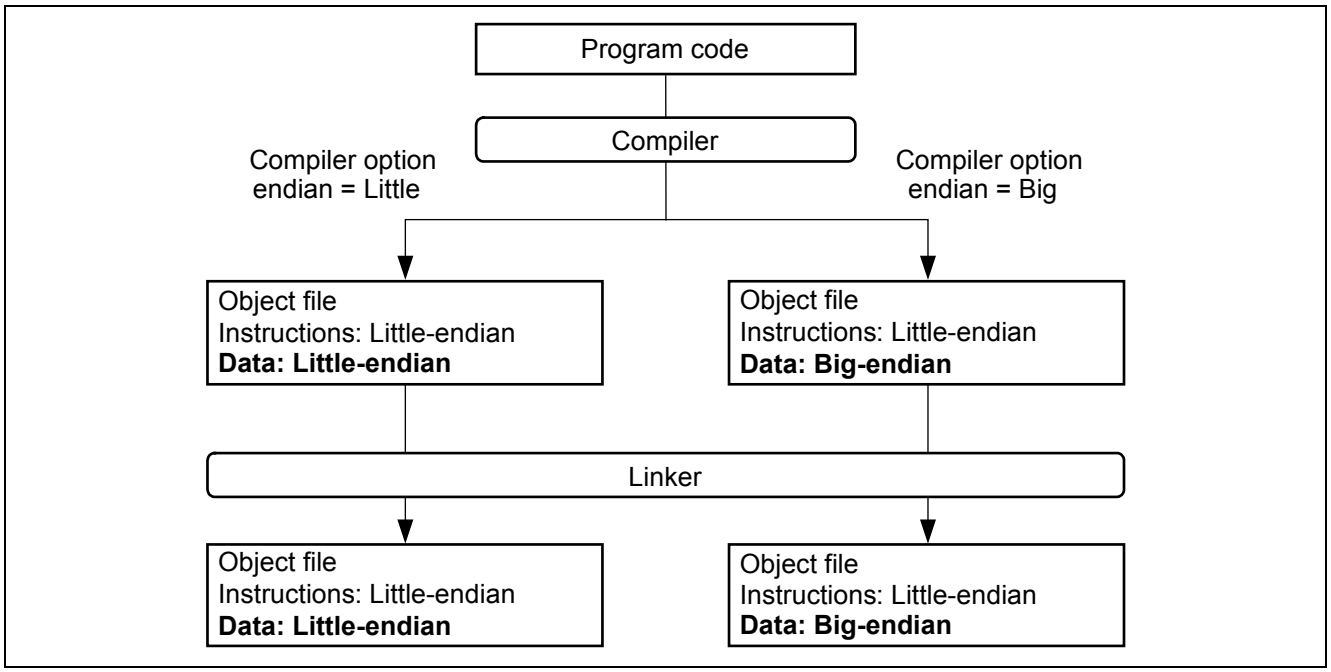


Figure 1.9 RX71M Specifying Endianness by Compiler Option

1.3.3 Specifying TM Identification Data and Setting TM Enable Flags

The RX71M is provided with a trusted memory (TM) function to prevent third parties from reading software stored in blocks 8 and 9 of the code flash memory. The TM function prevents reading of some designated areas even internally by the microcontroller, such as the on-chip flash memory, and allows instruction execution only.

This function is useful when storing software for processing encryption algorithms, device control processing software incorporating valuable intellectual property, commercial middleware, or the like in the code flash memory.

1.3.4 OCD/Serial Programmer Settings

The RX71M supports selection of serial programming functions by means of serial programmer commands. The SPCC register is used to enable serial programming.

When an OCD/serial programmer is connected, the data written in the option-setting memory is used to determine whether or not to accept the connection. A check is performed to determine if the code sent by the OCD/serial programmer matches the ID code in the option-setting memory. The connection to the OCD/serial programmer is enabled if the codes match, but no connection is possible if the codes do not match. The ID code of the OCD/serial programmer is stored in the OSIS register.

1.4 Reset Function

1.4.1 Reset Sources

Figure 1.3 lists the reset sources of the SH7216 Group and RX71M.

Table 1.3 Reset Sources

Item	SH7216 Group	RX71M
Reset type	• Power-on reset (RES# pin reset/H-UDI reset assert command/WDT overflow) • Manual reset (MRES# pin reset/WDT overflow)	• RES# pin reset • Power-on reset (internal reset) • Voltage monitor 0 reset • Voltage monitor 1 reset • Voltage monitor 2 reset • Deep software standby reset • Independent watchdog timer reset • Watchdog timer reset • Software reset

(1) **Reset Vector Configuration**

The SH7216 Group has separate vectors for power-on resets and for manual resets (PC and SP).*1

The RX71M has a single reset vector for multiple reset sources. The reset source is identified in reset status registers 0 to 2 during reset processing, and processing for the corresponding source is performed.

(2) **Stack Pointer**

On the SH7216 Group, it is necessary to specify the end address (+1) of the stack area in the reset vector. There is no stack pointer setting area in the vector table on the RX71M, so the stack pointer is set in ISP and USP.

Note 1. See 1.8.4, Vector Configuration, for details of the vector tables.

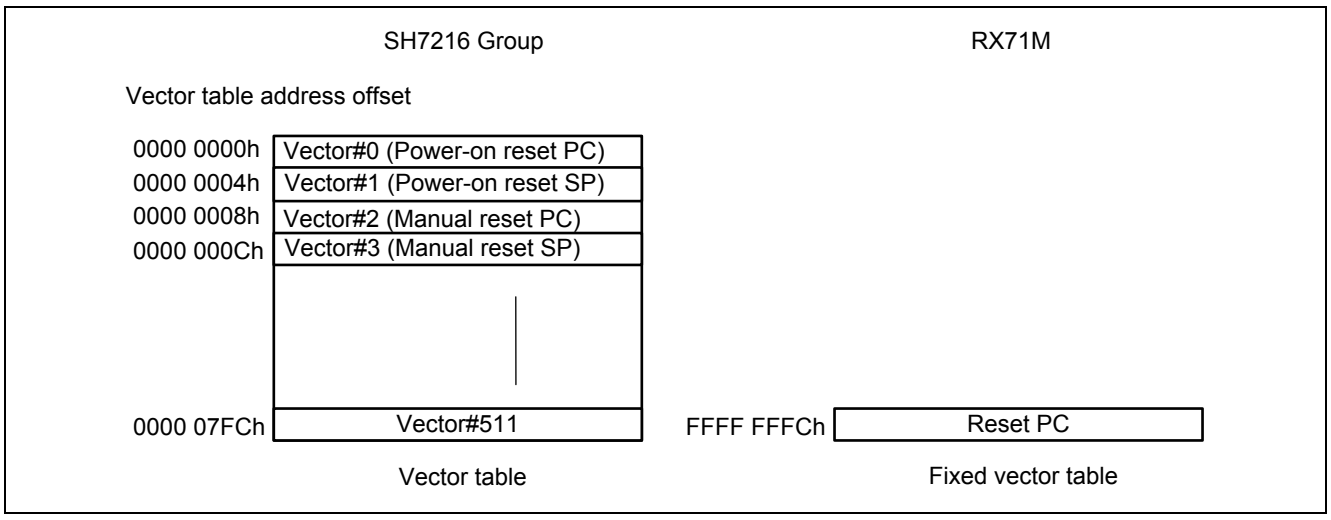


Figure 1.10 Reset Vector Comparison

1.4.2 Reset Sources and Initialization Scope

The initialization scope of the reset sources differs between the SH7216 Group and the RX71M. Table 1.4 lists the reset sources and their initialization scope on the SH7216 Group, and Table 1.5 lists the reset sources and their initialization scope on the RX71M. For details, see the User's Manual: Hardware.

Table 1.4 SH7216 Group Reset Sources and Initialization Scope

Item		CPU FPU	On-Chip Peripheral Module, I/O Port	WRCSR of WDT, FRQCR of CPG
Power-on reset	RES# pin reset	○	○	○
	H-UDI command	○	○	○
	WDT overflow	○	○	—
Manual reset	MRES# pin reset	○	—*1	—
	WDT overflow	○	—*1	—

○: Reset —: No reset

Note 1. The BN bit in IBNR of the INTC is initialized.

Table 1.5 RX71M Reset Sources and Initialization Scope

	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	Deep Software Standby Reset	Software Reset
Power-on reset detection flag	○	—	—	—	—	—	—	—	—
Cold start/warm start determination flag	—	○	—	—	—	—	—	—	—
Voltage monitor 0 reset detection flag	○	○	—	—	—	—	—	—	—
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	○	○	○	○	○	—	—	*1	—
Voltage monitor 2 reset detection flag	○	○	○	○	○	○	—	—	—
Voltage monitor function 2 registers	○	○	○	○	○	○	—	*2	—
Deep software standby reset detection flag	○	○	○	○	○	○	○	—	—
Software reset detection flag	○	○	○	○	○	○	○	○	—
Realtime clock registers*3	—	—	—	—	—	—	—	—	—
High-speed on-chip oscillator–related registers	○	○	○	○	○	○	○	—	○
Main clock oscillator–related registers	○	○	○	○	○	○	○	—	○
Pin states	○	○	○	○	○	○	○	—	○
Low power consumption–related registers*4	○	○	○	○	○	○	○	—	○
Registers other than the above, CPU, and internal state	○	○	○	○	○	○	○	○	○

○: Reset —: No reset

Note 1. Only LVD1CR1 and LVD1SR are initialized.

Note 2. Only LVD2CR1 and LVD2SR are initialized.

Note 3. Some control bits are initialized by all resets.
Note 4. The DPSBKRy registers are not initialized by any reset.

1.4.3 Cold/Warm Start Determination Function

On the RX71M it is possible to determine whether the most recent reset processing was caused by a power-on reset (cold start) or by a reset signal during operation (warm start).

When a power-on reset occurs because the external voltage (VCC) has exceeded the threshold, the cold/warm start determination flag (RSTSR1.CWSF) is cleared to 0, indicating a cold start. Since the flag is not cleared to zero by any other type of reset, 1 can be written to it by a program, indicating a warm start.

1.4.4 Write Protection

The RX71M has a register write protection function to protect important registers from being overwritten if program runaway occurs. The software reset register is protected by this function.

If necessary, set protect bit 1 (PRCR.PRC1) to 1 to enable writes before writing to the software reset register.

1.5 Clock Settings

1.5.1 Clock Sources

Table 1.6 lists the clock sources of the SH7216 Group and RX71M.

Table 1.6 Clock Sources

SH7216 Group	RX71M
Oscillator (EXTAL and XTAL) + PLL circuit USB oscillator (USBEXTAL and USBXTAL)	• Main clock oscillator (EXTAL and XTAL) + PLL circuit • Subclock oscillator (XCIN and XCOUT) • High-speed on-chip oscillator (HOCO) + PLL circuit • Low-speed on-chip oscillator (LOCO) • IWDT-dedicated on-chip oscillator

In the description below, the high-speed on-chip oscillator is referred to as the HOCO and the low-speed on-chip oscillator as the LOCO.

1.5.2 Clock Generation Circuit

On the SH7216 Group application of divider settings and oscillation stop detection control are performed in software. On the RX71M a variety of clock control operations are performed in software.

On the RX71M the LOCO operates as the clock source after a reset. The operation of necessary clock sources and PLL circuits other than the LOCO is started during system initialization, and various clocks are selected, such as the system clock and bus clocks. When making changes to clock-related settings, it is necessary to consider the register setting sequence and the oscillation and clock oscillation stabilization time.

See the following application note for details of the clock setting procedure.

RX71M Group Initial Settings (R01AN2459EJ)

1.5.3 Write Protection

The RX71M has a register write protection function to protect important registers from being overwritten if program runaway occurs, and the registers related to the clock generation circuit are protected by this function.

If necessary, set protect bit 0 (PRCR.PRC0) or protect bit 1 (PRCR.PRC1) to 1 to enable writes before writing to these registers.

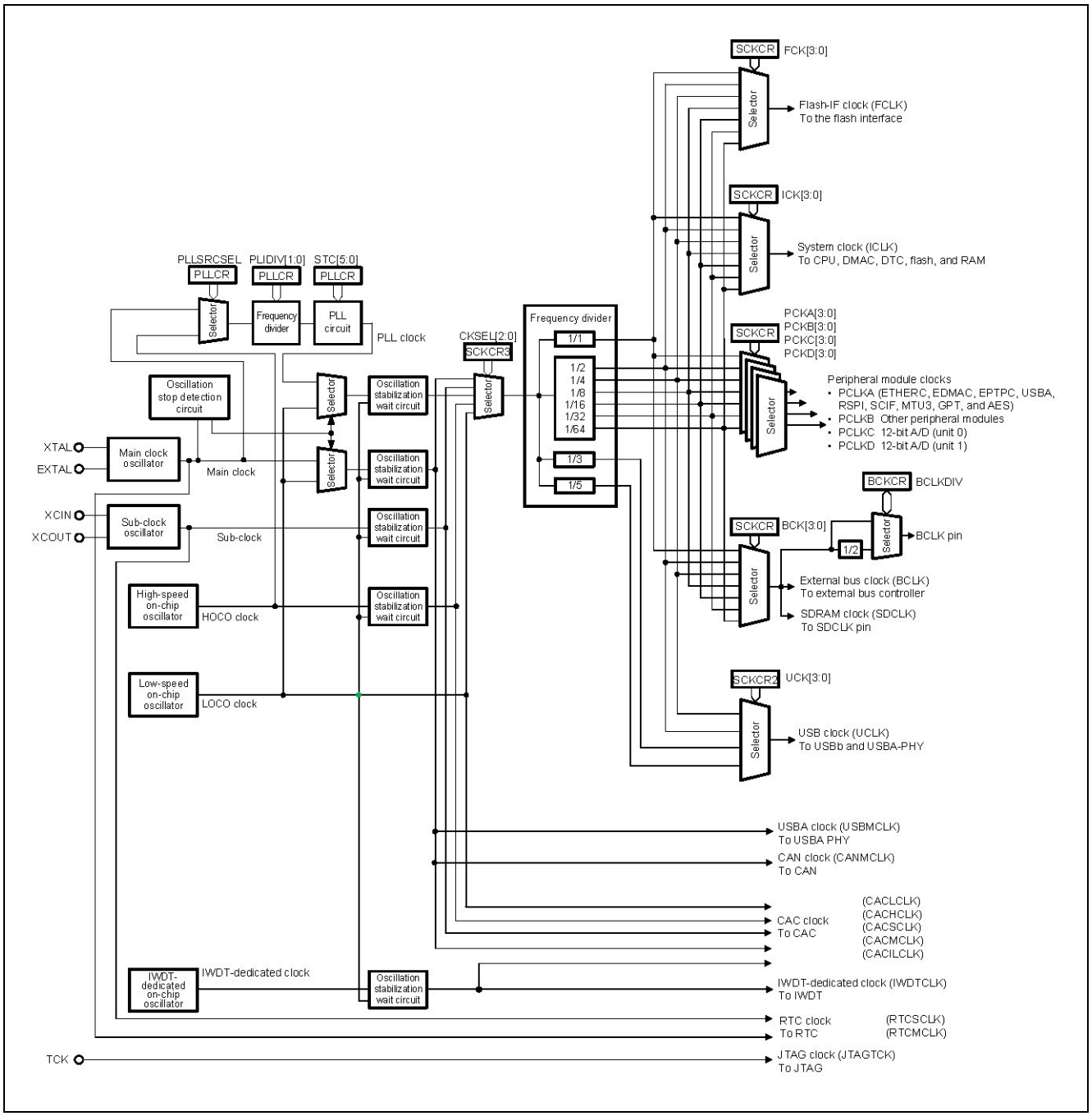


Figure 1.11 RX71M Block Diagram of Clock Generation Circuit

1.6 Operation Modes

1.6.1 Comparison of Operation Modes

Table 1.7 shows a comparison of the operation modes of the SH7216 Group and RX71M.

For details of each operation mode, see the User's Manual: Hardware.

Table 1.7 Comparison of Operation Modes

SH7216 Group	RX71M	Description
MCU extension mode 0	On-chip ROM disabled extended mode	An operation mode in which the on-chip ROM is disabled and the external address space is enabled. The external bus width differs from that of mode 0 and mode 1 on the SH7216 Group.
MCU extension mode 1		
MCU extension mode 2	On-chip ROM enabled extended mode	An operation mode in which the on-chip ROM is enabled and the external address space is enabled
Single-chip mode	Single-chip mode	An operation mode in which the external address space is disabled
Boot mode	Boot mode (SCI interface)	An operation mode in which the on-chip flash memory modifying program (boot program), which is stored in a dedicated area internal to the microcontroller, is run. The on-chip ROM can be programmed by a device external to the microcontroller by using the asynchronous serial interface.
USB boot mode	Boot mode (USB interface)	An operation mode in which the on-chip flash memory modifying program (boot program), which is stored in a dedicated area internal to the microcontroller, is run. The on-chip ROM (code flash memory) can be programmed by a device external to the microcontroller by using the USB interface.
User boot mode	User boot mode	An operation mode in which the on-chip flash memory modifying program (user boot program), which is stored in a dedicated area internal to the microcontroller, is run. The user MAT can be programmed by using a user-defined interface. Transition to this mode is possible after a reset is canceled.
User program mode	—	An operation mode in which the on-chip flash memory modifying program (user boot program), which is stored in a dedicated area internal to the microcontroller, is run. The user MAT can be programmed by using a user-defined interface. Transition to this mode is possible from MCU extension mode 2 or single-chip mode.

1.6.2 Comparison of Memory

Figure 1.12 shows a comparison of memory maps in on-chip ROM enabled mode.

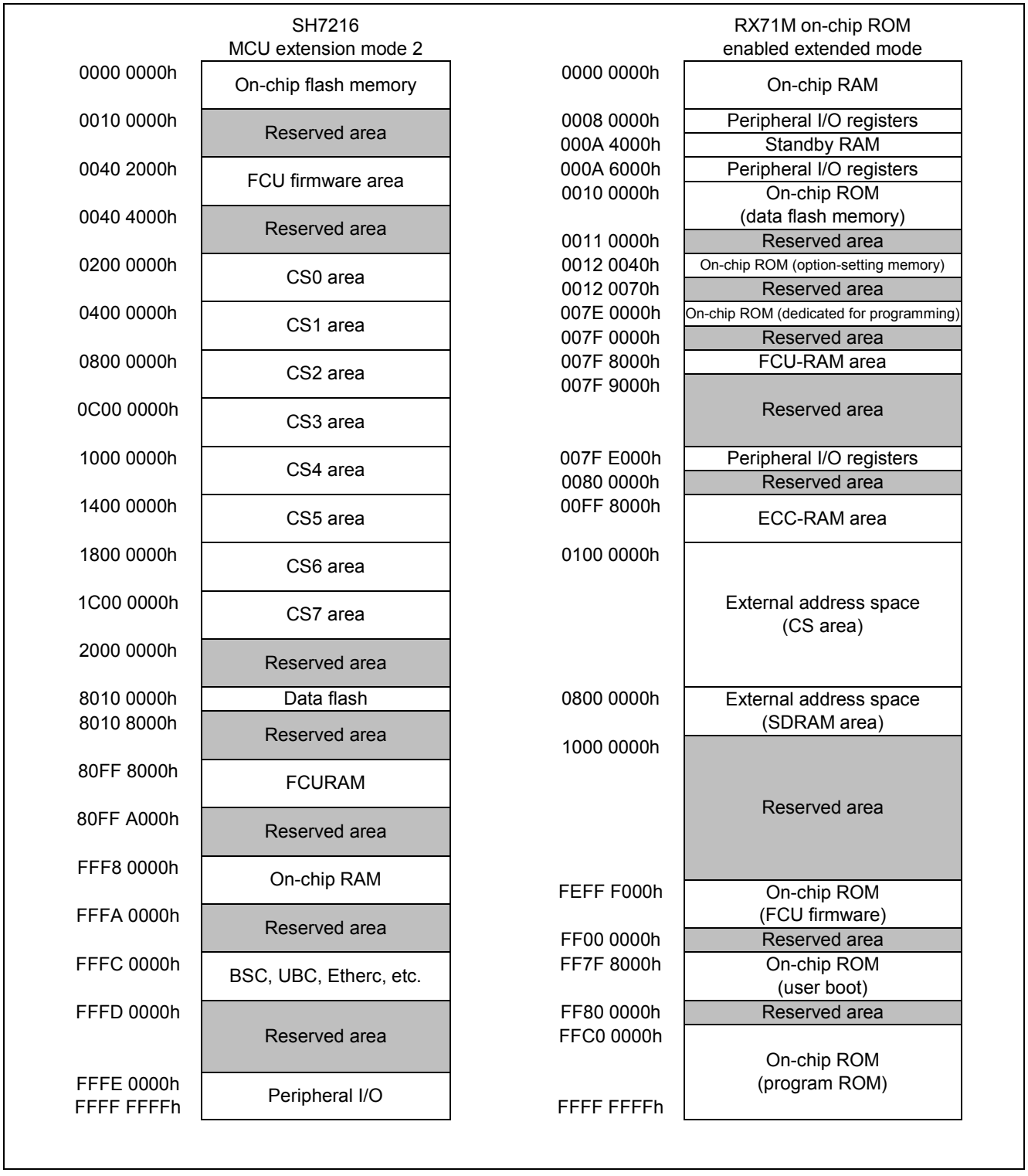


Figure 1.12 Memory Map Comparison (On-Chip ROM Enabled Mode)

Figure 1.13 shows a comparison of memory maps in single-chip mode.

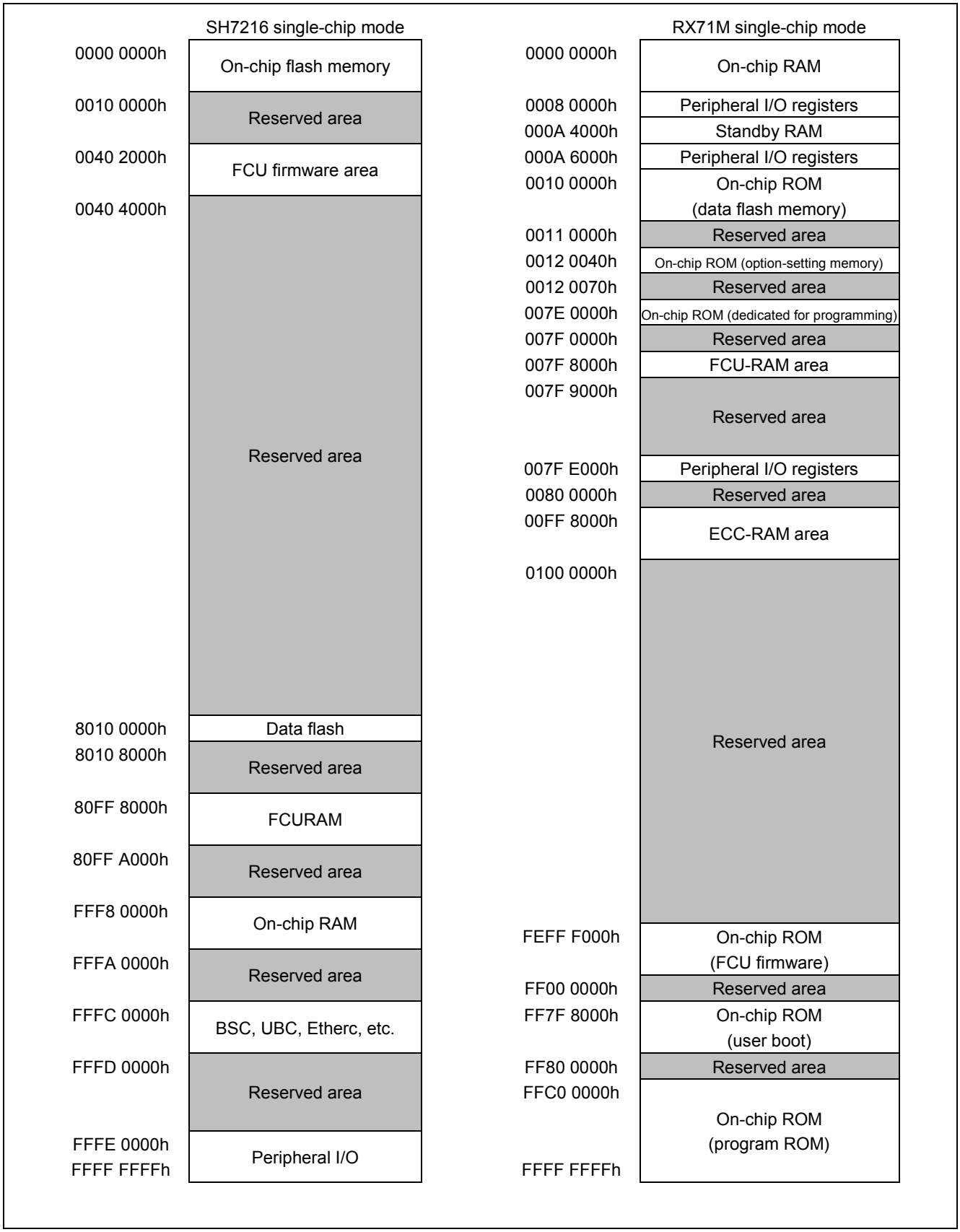


Figure 1.13 Memory Map Comparison (Single-Chip Mode)

Figure 1.14 shows a comparison of memory maps in on-chip ROM disabled mode.

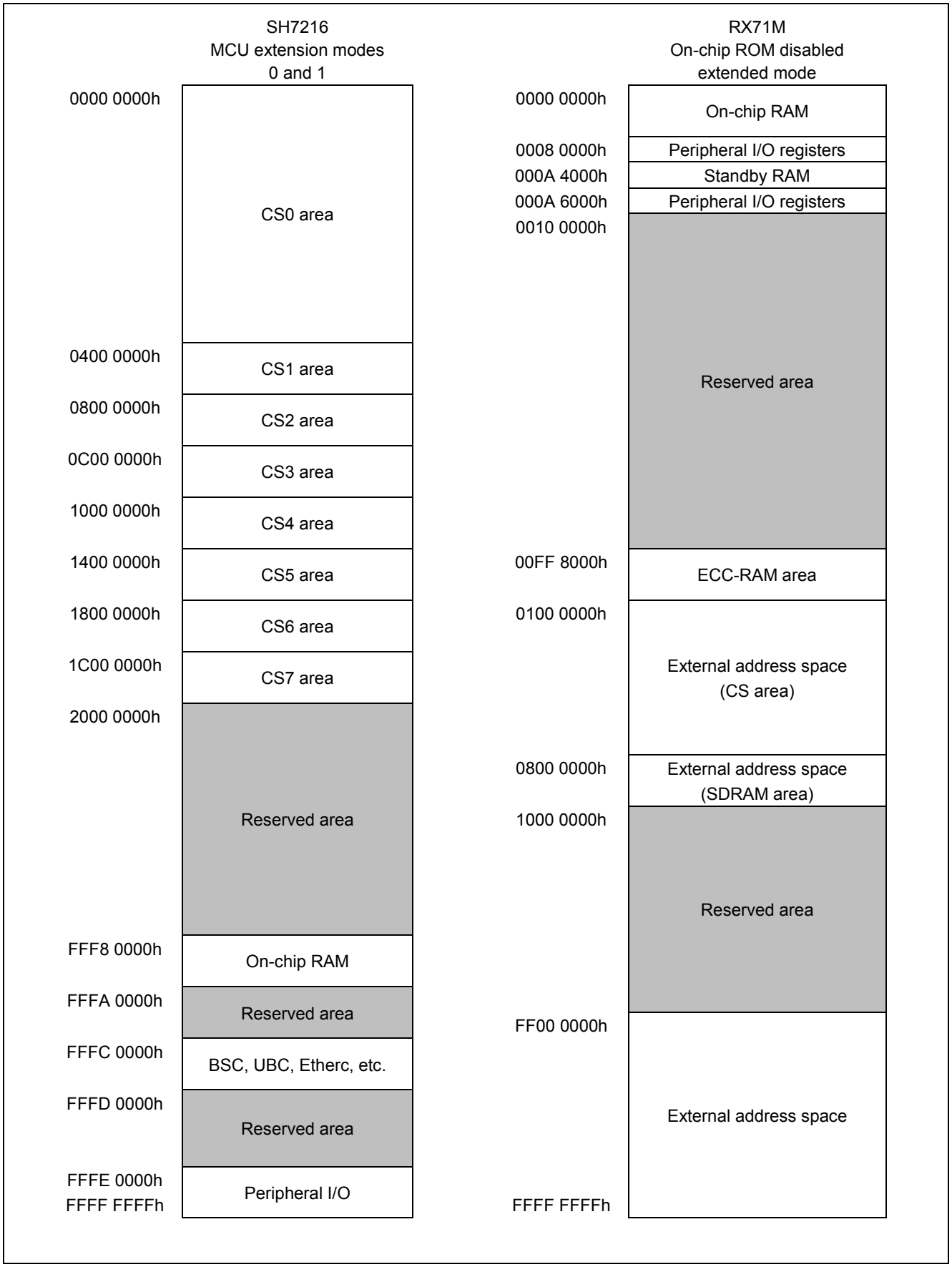


Figure 1.14 Memory Map Comparison (On-Chip ROM Disabled Mode)

- On the RX71M the RAM is allocated to addresses adjacent to 0000 0000h and ROM (for reading data) to addresses adjacent to FFFF FFFFh.
- On the RX71M the peripheral I/O registers are allocated within the address range from 0008 0000h to 000A 3FFFh and 000A 6000h to 000F FFFFh, and only the flash-related registers are allocated within the address range from 007F E000h to 007F FFFFh.
- On the RX71M the external address space is allocated within the address ranges from 0100 0000h to 0FFF FFFFh and FF00 0000h to FFFF FFFFh, and configured as up to eight CS spaces of 16 MB each and a 128 MB SDRAM space. In on-chip ROM enabled extended mode the CS0 area from FF00 0000h to FFFF FFFFh is disabled.

1.6.3 Operation Mode Settings

Whereas on the SH7216 Group operation mode settings are made only with the MD1, MD0, and FWE pins, on the RX71M operation mode settings can be made by means of the MD and UB pins when a reset is canceled, or by software after a reset is canceled.

Table 1.8 lists the operation modes that are determined by pin settings, and Table 1.9 lists the operation modes that are set in software after a reset is canceled.

Table 1.8 Pin Settings and Operation Modes on RX71M

Pin		
MD	**UB**	**Mode Name**
High	—	Single-chip mode
Low	Low	Boot mode (SCI interface)
	High	Boot mode (USB interface)
		User boot mode

Table 1.9 SYSCR0 Register Settings and Operation Modes on RX71M

SYSCR0 Register		
ROME Bit*1	**EXBE Bit**	**Mode Name**
0 (on-chip ROM disabled)	0 (external bus disabled)	Single-chip mode, user boot mode
1 (on-chip ROM enabled)*2	0 (external bus disabled)*2	
0 (on-chip ROM disabled)	1 (external bus enabled)	On-chip ROM disabled extended mode
1 (on-chip ROM enabled)	1 (external bus enabled)	On-chip ROM enabled extended mode

Note 1. Once the ROME bit is cleared to 0 it cannot be set to 1 again.
Note 2. After the STSCR0 register is reset, ROME = 1 and EXBE = 0.

1.6.4 Write Protection

The RX71M has a register write protection function to protect important registers from being overwritten if program runaway occurs, and the operation mode–related registers are protected by this function.

If necessary, set protect bit 1 (PRCR.PRC1) to 1 to enable writes before writing to these registers.

1.7 Processor Modes

The RX71M supports two processing modes: supervisor mode and user mode. These processor modes enable hierarchical CPU resource protection.

Table 1.10 RX71M Processor Modes

Processor Modes	Transition Conditions	Outline
Supervisor mode	• Reset cancellation • Exception occurrence (PSW.PM bit cleared to 0) When an exception occurs a transition to supervisor mode takes place, but the processor mode preceding the exception is restored following return from the exception handler.	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	• PSW.PM bit set to 1 In this case, first set to 1 the PSW.PM bit saved to the stack, then execute the RTE instruction. Alternately, first set to 1 the PSW.PM bit saved to BPSW, then execute the RTFI instruction.	Write access to some CPU resources, such as some bits in PSW and to BPC and BPSW, is restricted, and privileged instructions cannot be used. This is the mode in which user programs such as application programs ordinarily operate.

Transitioning from supervisor mode to user mode

The C/C++ Compiler Package for RX Family provides the intrinsic function chg_pmusr()*1 for switching to user mode.

The intrinsic function can be declared in the C source code. The output code does not perform a normal function call, but outputs the corresponding assembler code.

Note 1. The __chg_pmusr() function is available for use in C/C++ Compiler Package for RX Family (V.2.05.00) and later.

Figure 1.15 Processor Mode Setting Example (User Mode)

Transitioning from user mode to supervisor mode

An exception is generated by using the INT instruction or BRK instruction to generate an unconditional trap. A transition to supervisor mode occurs during exception handling.

Figure 1.16 Processor Mode Setting Example (Supervisor Mode)

1.8 Exception Handling

The points of difference regarding exception handling in general on the SH7216 Group and RX71M, including interrupts, are described below.

1.8.1 Types of Exception Handling

Table 1.11 shows a comparative listing of exception sources on the SH7216 Group and RX71M.

Table 1.11 Exception Source Comparison

SH7216 Group	RX71M	Main Points of Difference
Power-on reset	Reset	On the RX71M there is a single reset vector. Reset status registers 0 to 2 are checked during reset interrupt handling to determine the reset source, and appropriate processing is performed.
Manual reset		
Address error	—	The exception in question does not apply to the RX71M.
—	Access exception	This exception occurs when a memory protection error takes place. The exception in question does not apply to the SH7216 Group.
Interrupt	Non-maskable interrupt	The RX71M has separate vector tables for maskable and non-maskable interrupts.
	Interrupt	
Register bank error	—	The exception in question does not apply to the RX71M.
TRAP instruction (TRAPA instruction)	Unconditional trap (INT, BRK instruction)	The SH7216 Group has 32 sources, but the RX71M has 16 sources with dedicated vectors (up to 256 sources when sources also used for interrupts are included).
General illegal instruction	Undefined instruction	On the RX71M there is a single vector regardless of whether or not the exception occurs immediately after a delayed branch instruction.
Illegal slot instruction		
Integer division instruction	Floating-point exception	There is no exception corresponding to an integer division instruction on the RX71M.
Floating-point operation instruction		
—	Privileged instruction	The SH7216 Group has no exception that occurs when a privileged instruction is detected in user mode.

1.8.2 Exception Handling Priority

Table 1.12 shows the comparative priority of exception sources on the SH7216 Group and the RX71M.

Table 1.12 Exception Event Priority

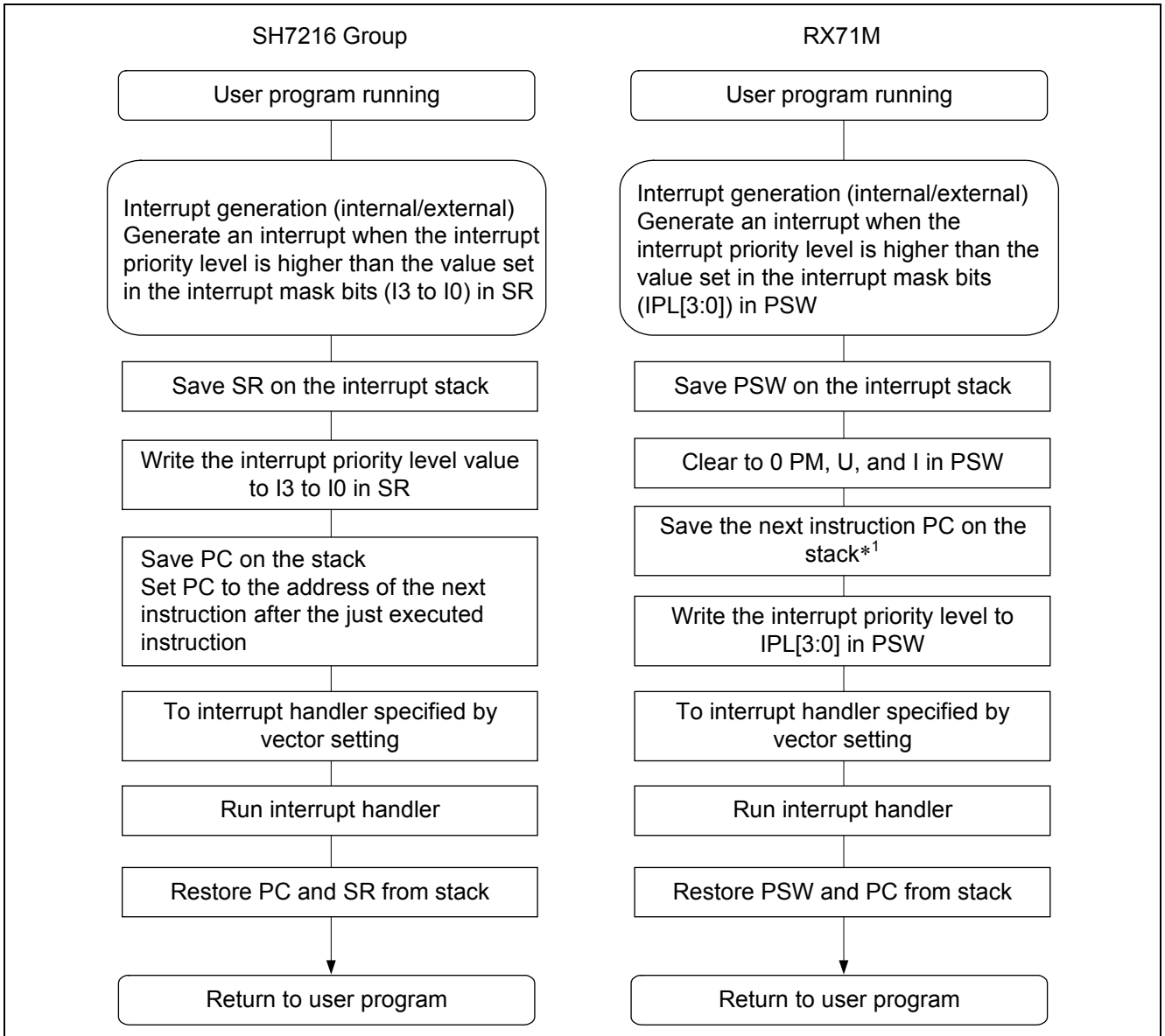
Priority*1	SH7216 Group	RX71M
High	Power-on reset	Reset
▲	Manual reset	Non-maskable interrupt
	Address error	Interrupt
	Floating-point operation instruction, integer division instruction	Access exception (instruction access exception)
	Register bank error	Undefined instruction exception, privileged instruction exception
	Interrupt	Unconditional trap
	TRAP instruction	Access exception (operand access exception)
Low	General illegal instruction, illegal slot instruction	Floating-point exception

Note 1. Among interrupts, the priority is determined by the interrupt controller.

Whereas on the SH7216 Group interrupts have low priority, on the RX71M they have high priority.

1.8.3 Basic Processing Sequence of Exception Handling

Figure 1.17 is a flowchart of interrupt exception handling on the SH7216 Group and the RX71M.



Note 1. When the instruction being executed is RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE, the PC of the instruction being executed is saved on the stack.

Figure 1.17 Interrupt (Internal/External) Processing Sequence

1.8.4 Vector Configuration

Both the SH7216 Group and the RX71M have a relocatable vector configuration, which allows vector tables to be reallocated.

On the SH7216 Group VBR (the vector base register) specifies the start of the vector table. (Note that VBR is initialized to 0 after a reset, so it is not possible to change the reset vector.)

On the RX71M INTB (the interrupt table register) specifies the start of the interrupt vector table, and EXTB (the exception table register) specifies the start of the exception vector table. Relocatable interrupt and unconditional trap vectors are assigned in the interrupt vector table. System exceptions are assigned in the exception vector table. The RX71M has a fixed reset vector. Also, the fast interrupt vector address is set in the FINTV register.

Figure 1.18 shows the differences between the vector tables.

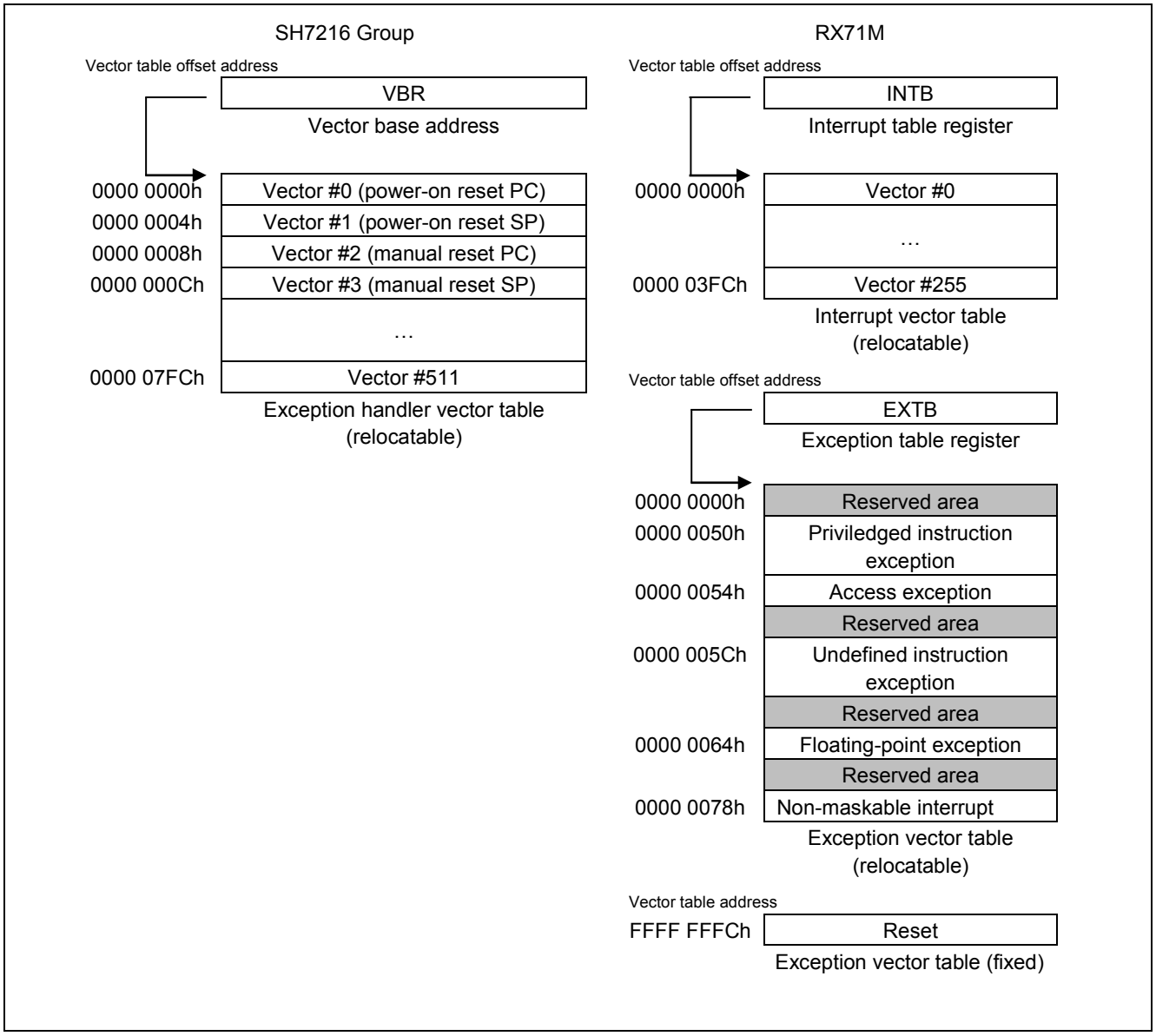


Figure 1.18 Vector Table Settings

1.8.5 Interrupt Masking by SR (SH7216 Group) and PSW (RX71M)

On the RX71M the I bits in control register PSW are used to set the interrupt mask level. The I bits indicate which interrupts are enabled and which are disabled.

Table 1.13 Interrupt-Related Bits in SR and PSW

SH7216 Group	RX71M	
SR Register	**PSW Register**	**Description**
I[3:0]	IPL[3:0]	CPU interrupt mask level (priority level) Setting value: 0 to Fh (levels 0 to 15) When an interrupt request occurs, this level setting is compared with the priority level set for the individual interrupt source, and the interrupt is enabled if its level setting is higher than the mask level.
—	I	Interrupt enable bit 0: Interrupts are disabled. 1: Interrupts are enabled. When an interrupt occurs, the interrupt status flag in the interrupt controller is set to 1. After a system reset, this bit is set to 1, enabling acceptance of interrupts. When an exception is accepted, this bit is cleared to 0 and no interrupts are accepted while its value remains 0.

1.9 Interrupt Handling

This section describes the differences in interrupt handling between the SH7216 Group and RX71M, with the focus on the interrupt controller.

1.9.1 Interrupt Controller

Table 1.14 lists the differences in the interrupt controller specifications.

Table 1.14 Comparison of SH7216 Group and RX71M Specifications (Interrupt Controller)

Item		SH7216 Group	RX71M
Interrupts	Peripheral function interrupts	• Interrupts from peripheral modules • Interrupt detection: Edge	• Interrupts from peripheral modules • Interrupt detection: Edge/level*1 • Group interrupt function support
	External pin interrupts	• IRQ0 to IRQ7 pins • Sources: 8 • Interrupt detection: Low level, falling edge, rising edge, or both edges can be specified for each source.	• IRQ0 to IRQ15 pins • Sources: 16 • Interrupt detection: Low level, falling edge, rising edge, or both edges can be specified for each source. • Noise canceler function
	Other sources	• User break interrupt • H-UDI interrupt • Memory error interrupt	None
	Noise cancellation	None	Digital filter settings are supported for the IRQi pins.
	Software interrupts	None	Supported
	Interrupt priority	A level from 0 to Fh can be specified for each source by a register setting.	A level from 0 to Fh can be specified for each source by a register setting.
	Fast interrupt function	None	Supported
	DTC/DMAC activation	DTC/DMAC activation supported*2	DTC/DMAC activation supported
	EXCMAC control	None	A software configurable interrupt can be used to start the EXDMAC.
Non-maskable interrupts	NMI pin interrupts	• Interrupt detection method (selection of falling or rising edge) • NMI input level read bit provided	• Interrupt detection method (selection of falling or rising edge)
	Other sources	None	• Interrupt at oscillation stop detection • WDT underflow or refresh error • IWDT underflow or refresh error • Voltage monitor 1 interrupt • Voltage monitor 2 interrupt • RAM error interrupt
	Noise cancellation	Checking of the NMI input level can be used as a noise canceler function.	Noise filter settings are supported for the NMI pin.
Register banks		15 register banks	None

Note 1. The detection method is fixed for fixed-connection peripheral modules.

Note 2. On the SH7216 Group activation source setting is performed on the DTC or DMAC.

Figure 1.19 shows the points of difference between the interrupt controller of the SH7216 Group and the RX71M.

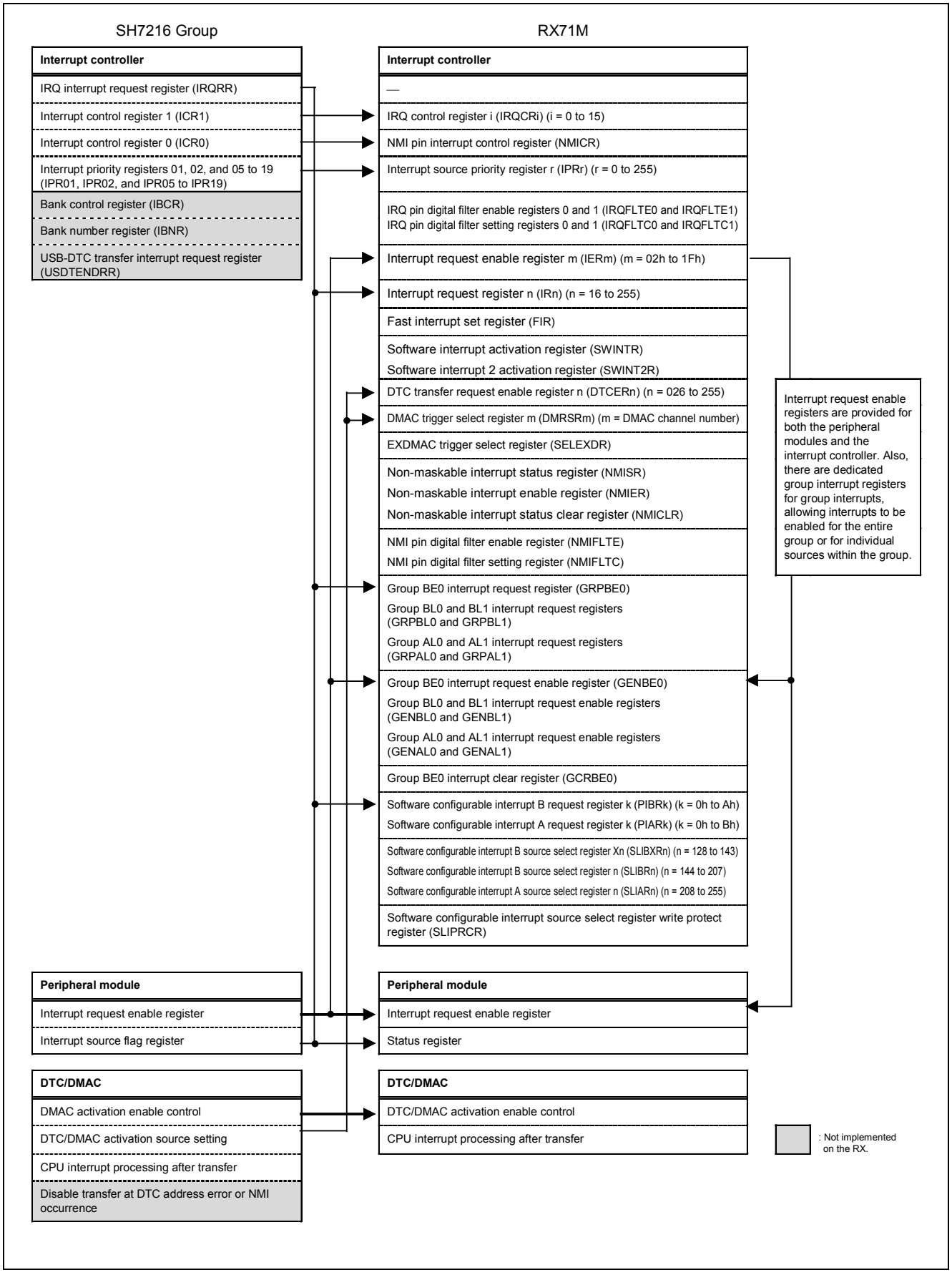


Figure 1.19 Differences Between Interrupt Controller Registers

The interrupt controller of the SH7216 Group controls IRQ interrupt flags, while peripheral module interrupt flags are controlled by the peripheral modules.

On the RX71M the interrupt controller controls all interrupt status flags, for both IRQs and peripheral modules.*1 In addition, the interrupt controller controls the activation source settings for the DTC and DMAC. The disable transfer at NMI occurrence function of the DTC and DMAC on the SH7216 Group is not implemented on the RX71M.

Note 1. The interrupt controller contains an interrupt request register for each interrupt source, but there are also interrupt enable bits implemented in the peripheral modules. (For details, see the User's Manual: Hardware.)

1.9.2 Interrupt Flag Management

When a peripheral module of the SH7216 Group generates an interrupt by edge detection, the corresponding interrupt source flag is cleared (the flag is cleared and a dummy read is performed) by the interrupt handler. This is done because the interrupt will be generated once again if the flag is not cleared by the handler.

On the RX71M the interrupt status flags are managed internally by the interrupt controller, and interrupt requests are sent to the CPU or DTC/DMAC. The interrupt controller has a function whereby, when edge detection is used, the corresponding interrupt status flag is cleared automatically when a response is received indicating acceptance of an interrupt. When level detection is used, both the request flag within the peripheral module and the corresponding interrupt status flag are cleared automatically. For details, see the User's Manual: Hardware.

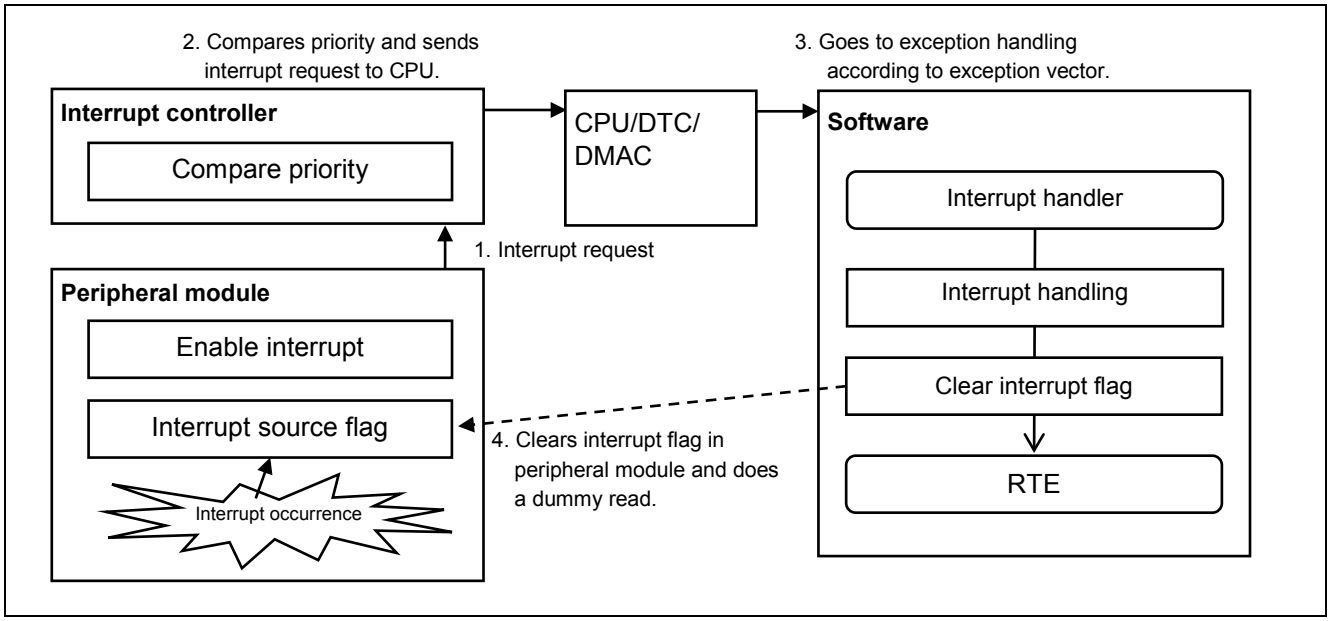


Figure 1.20 SH7216 Group Peripheral Module Interrupt (Edge Detection)

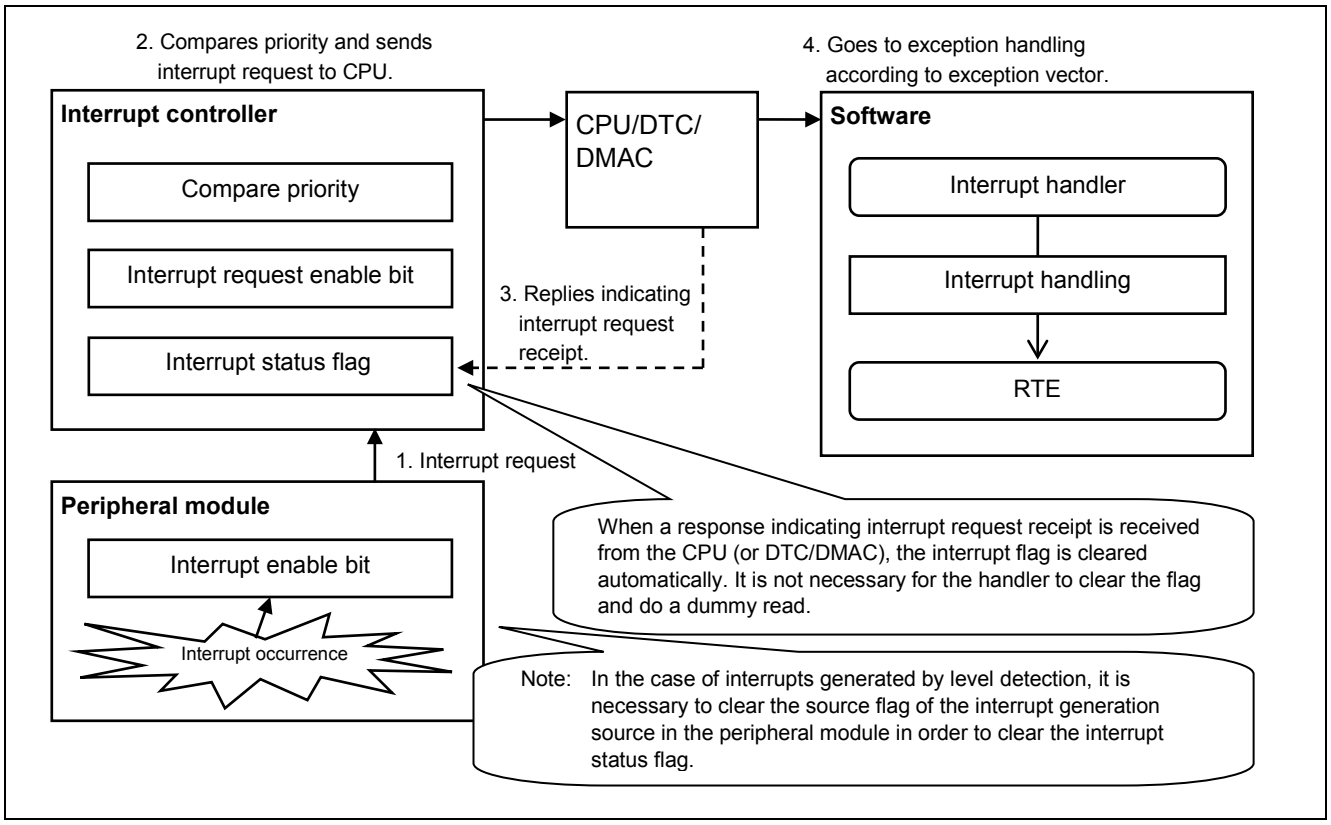


Figure 1.21 RX71M Peripheral Module Interrupt (Edge Detection)

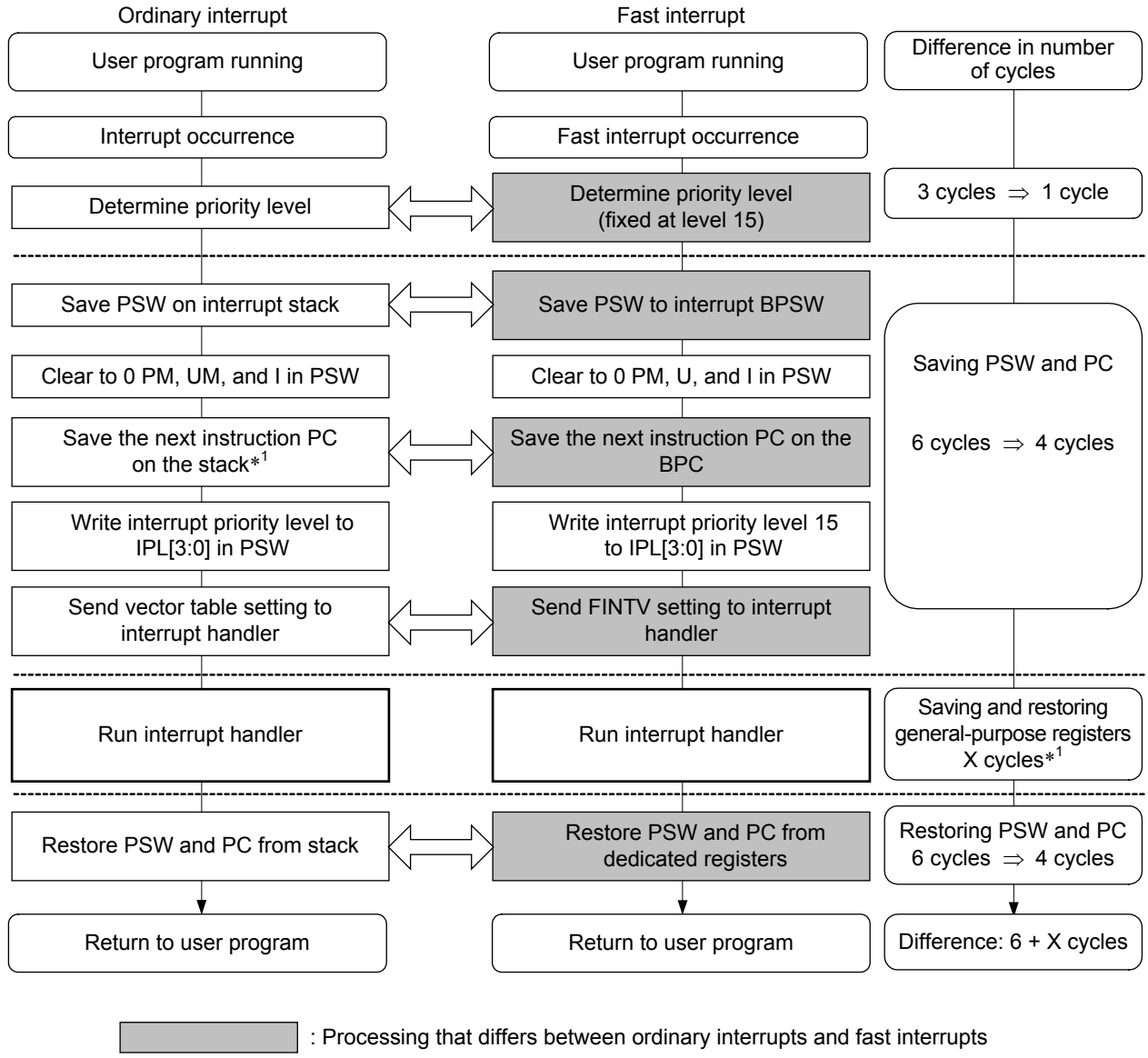
1.9.3 Fast Interrupt Control

In addition to ordinary interrupts, the RX71M supports fast 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 RAM or the external RAM by software.

Fast interrupt: Operation gives the interrupt the highest priority. When the interrupt occurs, the contents of the control registers are saved to dedicated registers, allowing interrupt activation to be realized faster than an ordinary interrupt.

It is possible to assign a portion of the general-purpose registers to exclusive use for interrupts by setting a compiler option. This eliminates the need to save and restore the contents of the general-purpose registers, further speeding up the interrupt.



Note 1. Fast interrupts have a single save register stage. Therefore, multistage fast interrupts are not possible. It is necessary to decide on a single function that will use fast interrupts.

Figure 1.22 Differences Between Ordinary Interrupts and Fast Interrupts on the RX71M

1.9.4 Noise Cancellation

The SH7216 Group has an NMI level bit (ICR0.NMIL) that indicates the state of the NMI pin. An interrupt handler service routine can check the pin state by reading this bit, and this capability can be used as a noise canceler function.

The RX71M is provided with a digital filter function for signals input on the IRQi pins and NMI pin. The sampling clock for the digital filter can be specified, and interrupt signals that do not last for at least three cycles of the sampling clock base are not accepted as interrupts. This improves the system's noise tolerance.

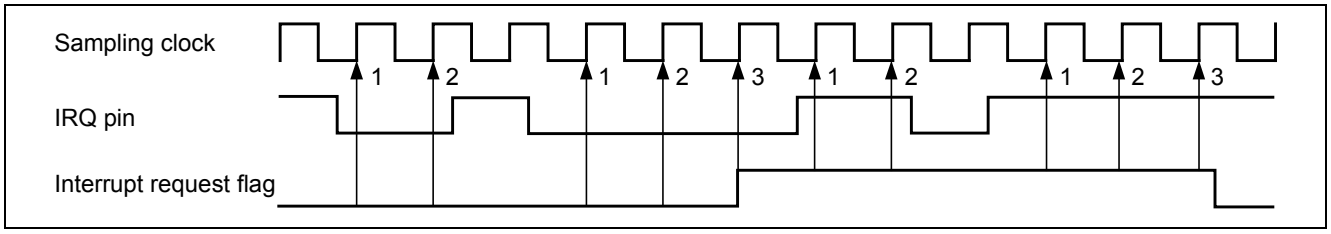


Figure 1.23 RX71M Digital Filter Operation Example

1.9.5 Multiple Interrupts

On the SH7216 Group if a high-priority interrupt occurs while a low-priority interrupt handler is running, the low-priority interrupt handler is suspended and the high-priority interrupt handler is executed. Once the high-priority interrupt handler finishes, the suspended low-priority interrupt handler is restarted.

On the RX71M if a high-priority interrupt occurs while a low-priority interrupt handler is running, the high-priority interrupt is not accepted until the low-priority interrupt handler finishes. This is because the PSW.I bit is cleared to 0 (interrupts are disabled) in a normal interrupt handler. In order to realize handling of multiple interrupts equivalent to that of the SH7216 Group, it is necessary to set the PSW.I bit to 1 (interrupts are enabled) in the interrupt handler.

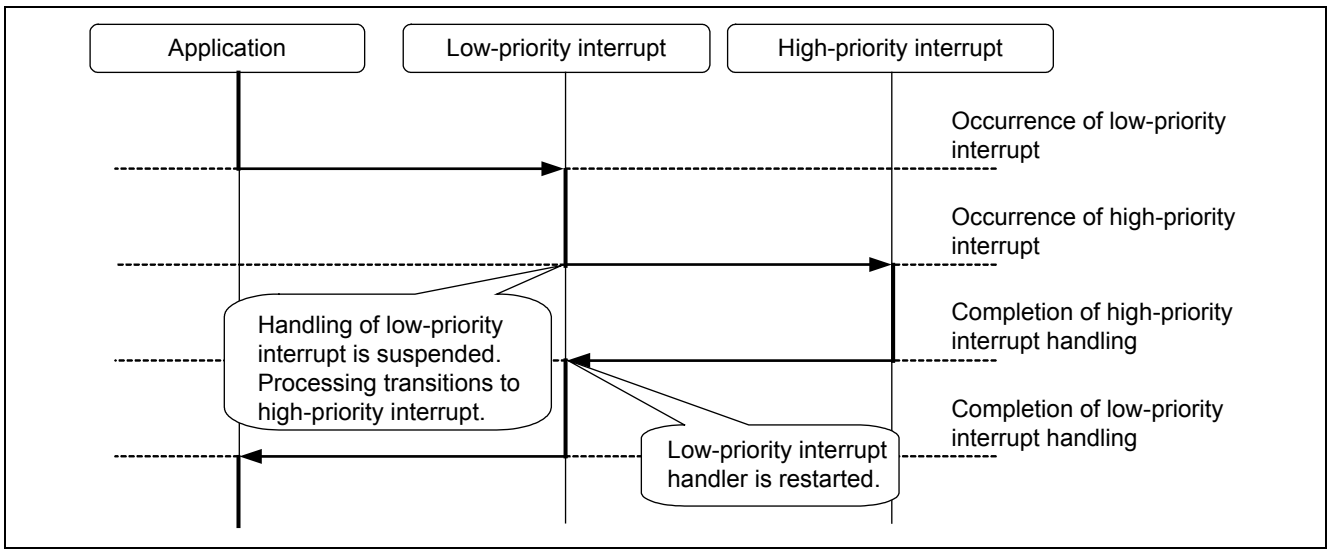


Figure 1.24 SH7216 Group Multiple Interrupt Sequence

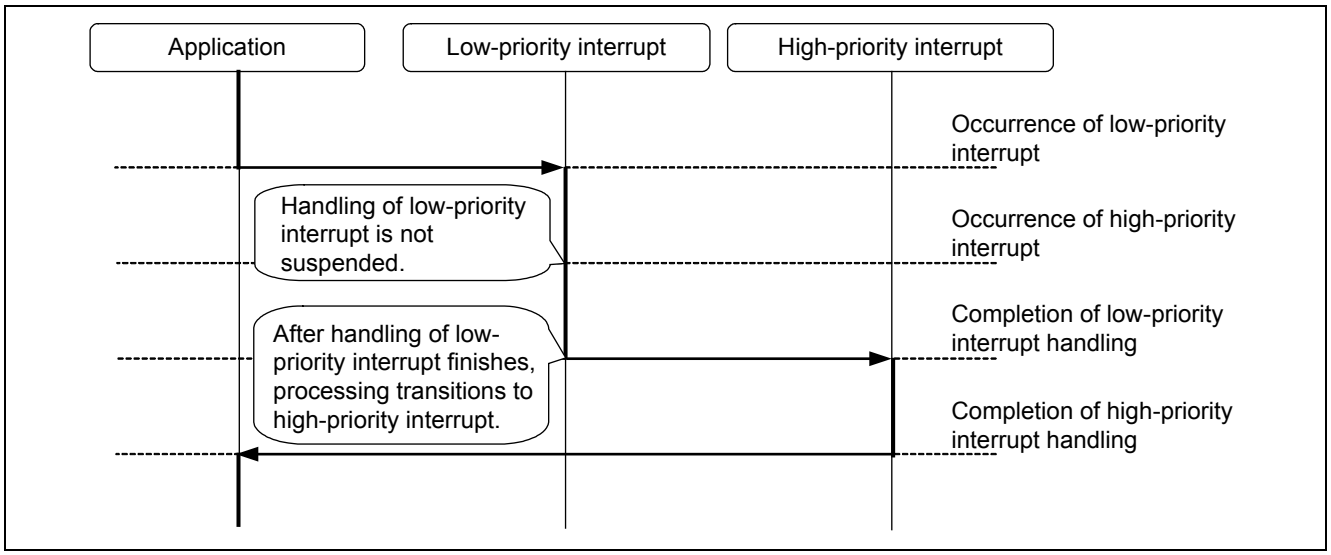


Figure 1.25 RX71M Interrupt Sequence (Not Controlled by PSW.I Bit)

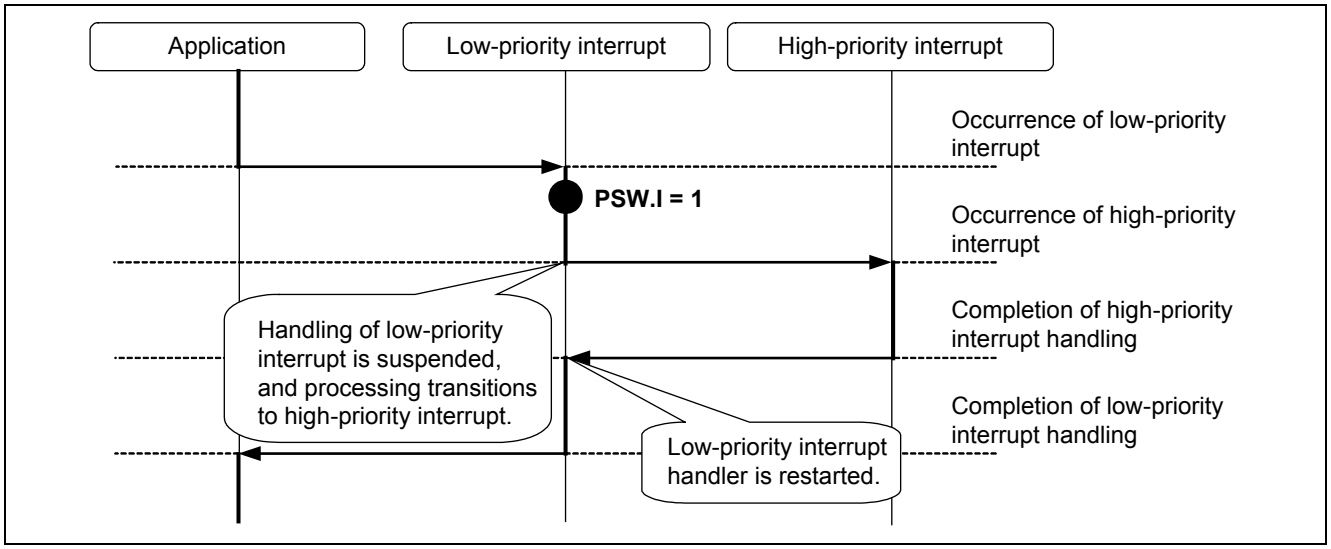


Figure 1.26 RX71M Interrupt Sequence (Controlled by PSW.I Bit)

1.9.6 Group Interrupts

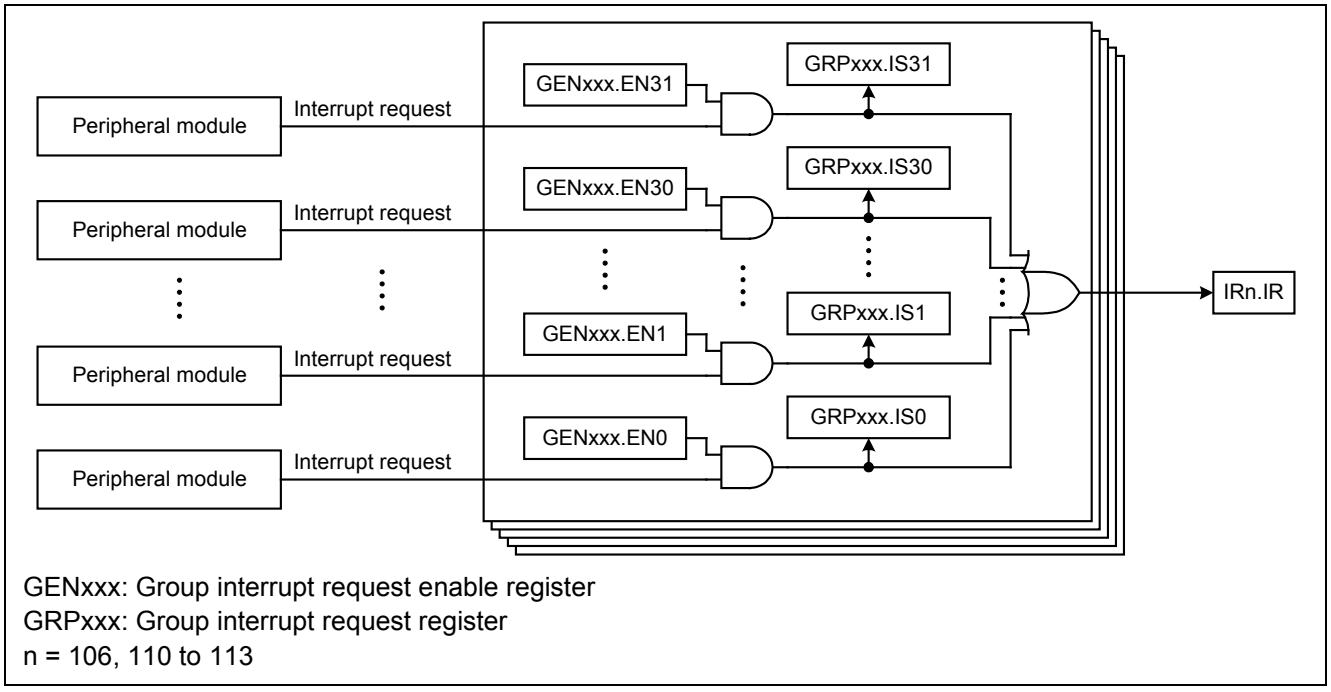
Group interrupts allow multiple interrupt sources to be assigned to a single vector. Group interrupt detection is by means of a logical OR operation on all the interrupt requests assigned to the group. This means that when an interrupt request is detected, it is necessary to identify the interrupt request from among those in the group by means of software.

Interrupt sources are assigned to different groups according to the operating clock of the peripheral module and the interrupt detection method.

The clearing condition for each group interrupt status flag differs according to the interrupt detection method. Table 1.15 lists the types of group interrupts and the clearing conditions of their status flags.

Table 1.15 RX71M Group Interrupt Types

Group	Peripheral Module Operating Clock	Interrupt Detection Method	Group Interrupt Status Flag
Group BE0	PCLKB	Edge detection	Cleared automatically when 1 is written to the corresponding interrupt source clear bit (GCRBE0.CLRn) of the interrupt controller.
Group BL0		Level detection	Cleared automatically when the peripheral module's interrupt status flag is cleared. Also cleared automatically when the interrupt controller's interrupt request enable bit (ENj in GENBL0, GENBL1, GENAL0, or GENAL1) is cleared to 0.
Group BL1			
Group AL0	PCLKA		
Group AL1			



GENxxx: Group interrupt request enable register
GRPxxx: Group interrupt request register
n = 106, 110 to 113

Figure 1.27 Group Interrupt Configuration on the RX71M

1.9.7 Software Configurable Interrupts

A single interrupt source among multiple peripheral modules can be selected for each software configurable interrupt, which is then assigned an interrupt vector number from 128 to 255.

Software configurable interrupts are classified into two types, A and B, according to the peripheral module operating clock. Table 1.16 lists the types of software configurable interrupts.

The software configurable interrupt status flags are not cleared automatically, but there is no effect on the generation of interrupt requests even if the corresponding flags are not cleared.

Table 1.16 Types of Software Configurable Interrupts on the RX71M

Software Configurable Interrupt Name	Peripheral Module Operating Clock	Interrupt Detection Method	Software Configurable Interrupt Status Flag
Software Configurable Interrupt A	PCLKA	Edge detection	Not cleared automatically, but there is no effect on interrupt request generation even if the flag is not cleared.
Software Configurable Interrupt B	PCLKB		

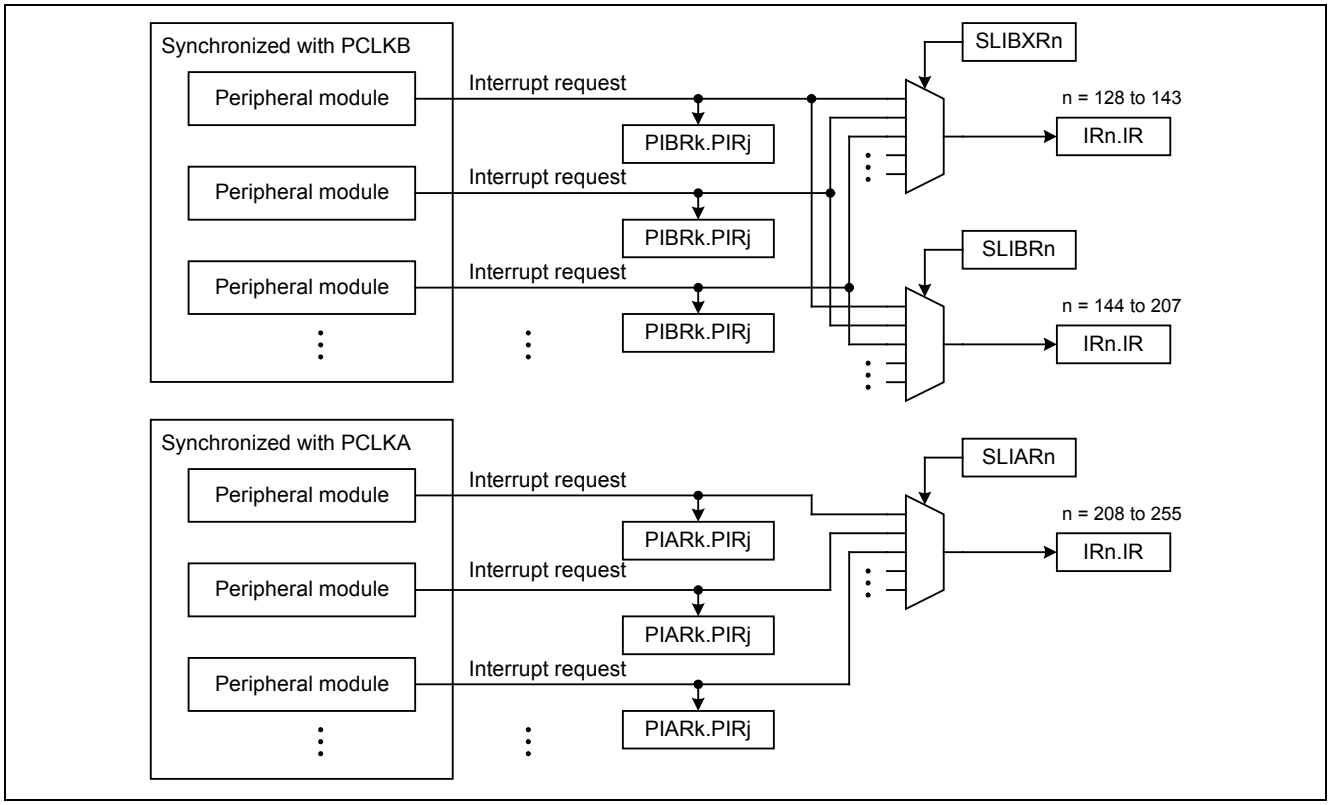


Figure 1.28 Software Configurable Interrupt Configuration on the RX71M

2. On-Chip Functions

2.1 List of On-Chip Functions

Table 2.1 lists the on-chip functions of the SH7216 Group and RX71M.

Table 2.1 On-Chip Functions

SH7216 Group	RX71M
Clock pulse generator (CPG)	Clock generation circuit
Interrupt controller (INTC)	Interrupt controller (ICUA)
User break controller (UBC)	—
Data transfer controller (DTC)	Data transfer controller (DTCa)
Bus state controller (BSC)	Bus
Direct memory access controller (DMAC)	DMA controller (DMACAa) EXDMA controller (EXDMACa)
Multi-function timer pulse unit 2 (MTU2) Multi-function timer pulse unit 2S (MTU2S)	Multi-function timer pulse unit 3 (MTU3a)
Port output enable (POE2)	Port output enable 3 (POE3)
Watchdog timer (WDT)	Watchdog timer (WDTA) Independent watchdog timer (IWDTa)
Serial communication interface (SCI) Serial communication interface with FIFO (SCIF)	Serial communication interface (SCIg, SCIh) FIFO embedded serial communication interface (SCIFA)
Renesas serial peripheral interface (RSPI)	Serial peripheral interface (RSPIa)
I^2C bus interface 3 (IIC3)	I^2C bus interface (RIICa)
A/D converter (ADC)	12-bit A/D converter (S12ADC)
Controller area network (RCAN-ET)	CAN module (CAN)
USB function module (USB)	USB 2.0 FS host/function module (USBb) USB2.0 high-speed host/function module (USBAa)
Ethernet controller (EtherC)	Ethernet controller (ETHERC)
Ethernet controller direct memory access controller (E-DMAC)	DMA controller for the Ethernet controller (EDMACa)
Compare match timer (CMT)	Compare match timer (CMT) Compare match timer W (CMTW)
Pin function controller (PFC)	Multi-function pin function controller (MPC)
I/O port	I/O port
Flash memory Data flash	Flash memory*1
On-chip RAM (max. 128 KB)	RAM (max. 512 KB, 32 KB) Standby RAM (max. 8 KB)
Power-down mode	Low power consumption function

SH7216 Group	RX71M
User debugging interface (H-UDI)	Voltage detection circuit (LVDA) Clock frequency accuracy measurement circuit (CAC) Battery backup function Register write protection function Memory-protection unit (MPU) Event link controller (ELC) General PWM timer (GPTa) 16-bit timer pulse unit (TPUa) Programmable pulse generator (PPG) 8-bit timer (TMR) Realtime clock (RTCd) PTP module for the Ethernet controller (EPTPCa) Quad serial peripheral interface (QSPI) CRC calculator (CRC) Serial sound interface (SSI) Sample rate converter (SRC) SD host interface (SDHI) MultiMediaCard interface (MMCIF) Parallel data capture unit (PDC) Boundary scan AESa DES SHAa RNG 12-bit D/A converter (R12DA) Temperature sensor (TEMPS) Data operation circuit (DOC)

Note 1. The flash memory of the RX71M includes code flash memory and data flash memory.

2.2 I/O Ports/Pin Function Controller (PFC)

2.2.1 Number of I/O Ports

Table 2.2 lists the number of I/O ports on the SH7216 Group and RX71M.

Table 2.2 Number of I/O Ports

Item	Package	Port Function
Number of I/O ports on SH7216 Group	PLQP0176KB-A PLQP0176LB-A PLBG0176GA-A	I/O: 100 Input: 10 Total: 110 Pull-up resistor: 100
Number of I/O ports on RX71M	PTLG0177KA-A (in planning) PLQP0176KB-A PLBG0176GA-A (in planning)	I/O: 127 Input: 1 Pull-up resistor: 127 Open-drain output: 127 5 V tolerant: 19
	PTLG0145KA-A (in planning) PLQP0144KA-A	I/O: 111 Input: 1 Pull-up resistor: 111 Open-drain output: 111 5 V tolerant: 18
	PTLG0100JA-A (in planning) PLQP0100KB-A	I/O: 78 Input: 1 Pull-up resistor: 78 Open-drain output: 78 5 V tolerant: 17

2.2.2 I/O Settings

Both the SH7216 Group and RX71M have multiplexed pins. Therefore, it is necessary to make pin settings to assign each pin to either general I/O or an on-chip module function.

On the SH7216 Group port functions are determined by settings made to the pin function controller (PFC). The I/O ports are configured as ports A to F.

The SH7216 Group's I/O port register settings are shown in Figure 2.1, the I/O port register configuration in Table 2.3, and the pin function controller (PFC) register configuration in Table 2.4.

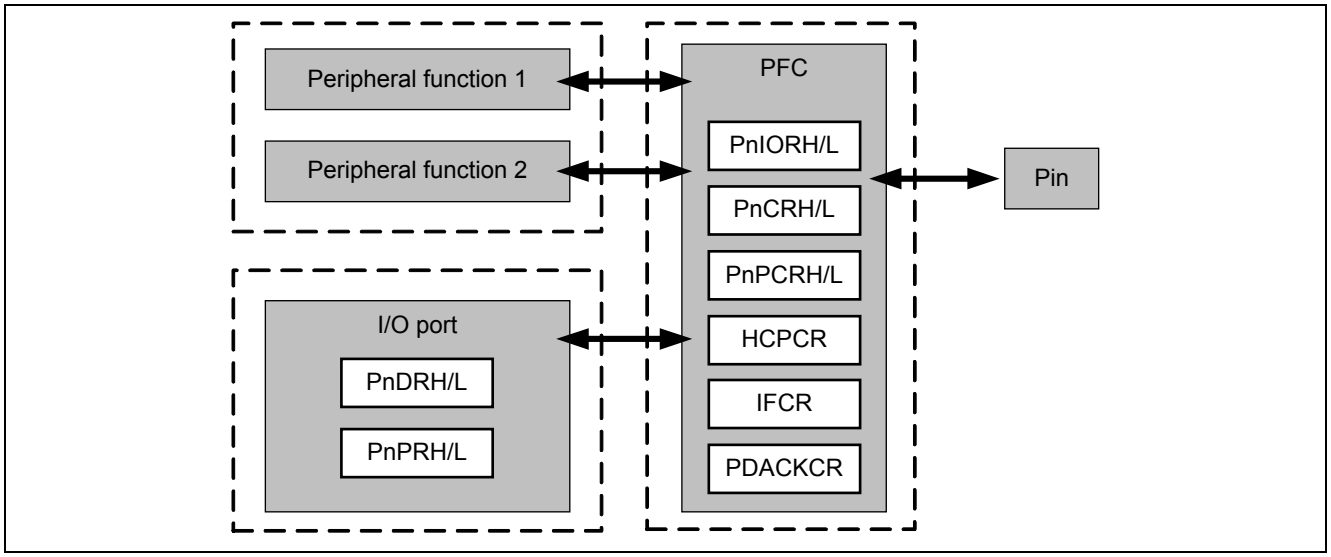


Figure 2.1 SH7216 Group I/O Settings

Table 2.3 SH7216 Group Register Configuration (I/O Ports)

Register	Function Name	Function
PnDRH	Port n data register H	Port n data registers
PnDRL	Port n data register L	Pin function is general output: Stores pin output data. Pin function is general input: Reflects pin states.
PnPRH	Port n port register H	Port n data read-only registers
PnPRL	Port n port register L	They reflect pin states.

n: Port name (n = A to F)

Table 2.4 SH7216 Group Register Configuration (PFC)

Register	Function Name	Function
PnIORH	Port n IO register H	Pin input/output direction selection
PnIORL	Port n IO register L	
PnCRHm	Port n control register Hm	Multiplexed pin function selection
PnCRLm	Port n control register Lm	
PnPCRH	Port n pull-up MOS control register H	Selects the input pull-up MOS setting.
PnPCRL	Port n pull-up MOS control register L	
HCPCR	High-current port control register	Sets the state of high-current ports.
IFCR	IRQOUT function control register	Sets the state of IRQ output pin.
PDACKCR	DACK output timing control register	Sets the DACK pin output timing.

n: Port name (n = A to E)
m: Setting number (m = 1 to 4)

Note that the functions that can be assigned to pins and the functions that can be specified by the PFC differ according to the SH7216 Group's operation mode (microcontroller mode 0, 1, or 2, or single-chip mode).

On the RX71M port functions are specified by making settings to the multi-function pin controller (MPC). The I/O ports are configured as ports 0 to 9, A to G, and J.

Unlike the SH7216 Group, where registers are provided for each port, on the RX71M registers are provided for each pin for selection of pin functions.

The following types of I/O port settings are supported on the RX71M.

- Open drain control register: Port output format selection
  CMOS output, N-channel open-drain output, or P-channel open-drain output
- Pull-up control register: Input pull-up resistor on/off selection
- Drive capacity control register: Selection between normal drive output and high drive output
- 5 V tolerant input ports are provided.

The RX71M's I/O port register settings are shown in Figure 2.2, the I/O port register configuration in Table 2.5, and the multi-function pin controller (MPC) register configuration in Table 2.6.

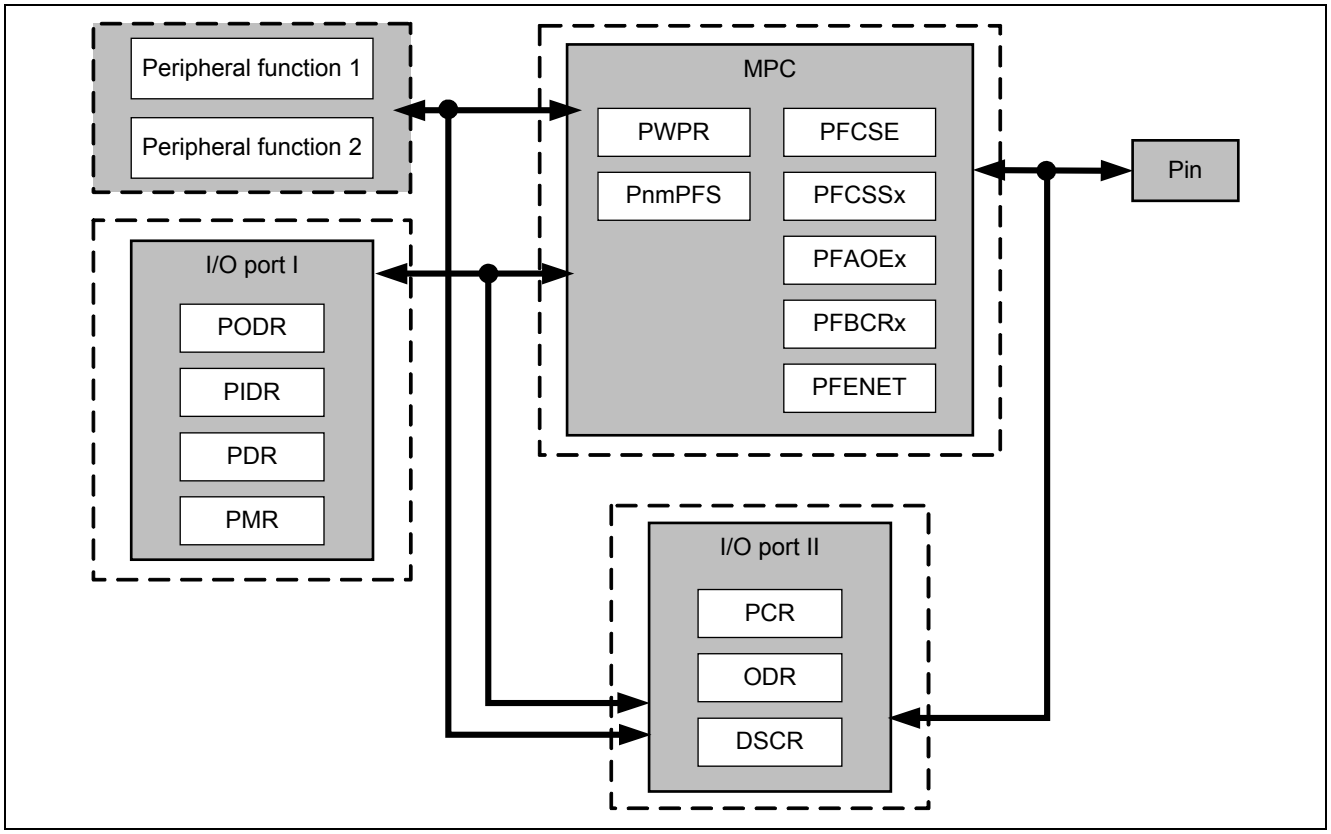


Figure 2.2 I/O Settings on the RX71M

To use a pin as a general I/O pin it is sufficient to make a setting in the appropriate I/O port register. Figure 2.3 shows the initialization sequence for using pins as general I/O pins on the RX71M.

The pin function control registers (PnmPFS) of the MPC are used to assign peripheral functions to pins. For setting examples when using peripheral functions that include general I/O, refer to the individual chapters for each of the peripheral functions. Figure 2.4 shows the initialization sequence for assigning pins to peripheral functions on the RX71M.

Table 2.5 RX71M Register Configuration (I/O Ports)

Register	Function Name	Function
PDR	Port direction register	Specifies input or output for pins selected as general I/O ports.
PODR	Port output register	Stores pin output data for general output ports.
PIDR	Port input register	Reflects port pin states.
PMR	Port mode register	Used for port pin function settings. Specifies whether each pin is used as a general I/O port or for a peripheral function.
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
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	Turns the port input pull-up resistor on or off.
DSCR	Drive capacity control register	Specifies the drive capacity. • Normal drive output • High drive output

Table 2.6 RX71M Register Configuration (MPC)

Register	Function Name	Function
PWPR	Write-protect register	Write-protect function for PnmPFS register
PnmPFS	Pnm pin function control register	Selects functions of multiplexed pins.
PFCSE	CS output enable register	Disables or enables output on CSn# (n: 0 to 7).
PFCSS0	CS output pin select register 0	Selects output pins for CS0 to CS3.
PFCSS1	CS output pin select register 1	Selects output pins for CS4 to CS7.
PFAOE0	Address output enable register 0	Settings when using pins for address bus
PFAOE1	Address output enable register 1	Settings when using pins for address bus
PFBCR0	External bus control register 0	Settings when using pins for external bus
PFBCR1	External bus control register 1	Settings when using pins for external bus
PFENET	Ethernet control register	Settings when using Ethernet PHY mode

n: Port name (n = 0 to 9, A to G, J)
m: Pin number (m = 0 to 7)

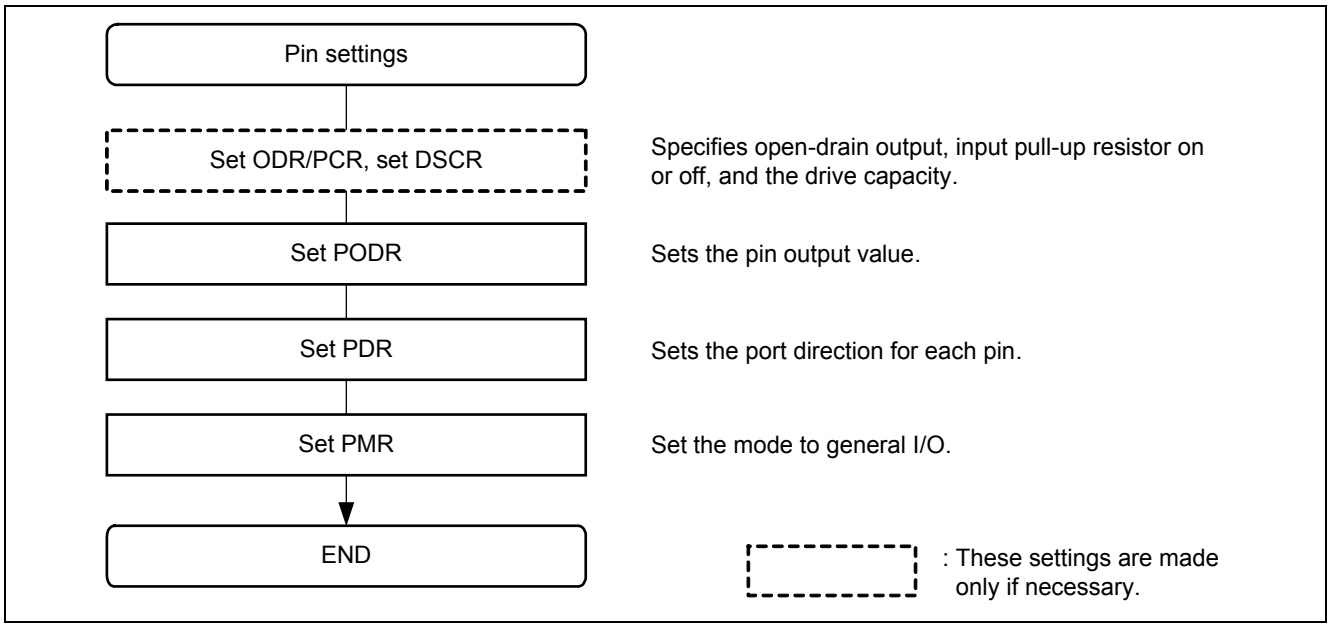
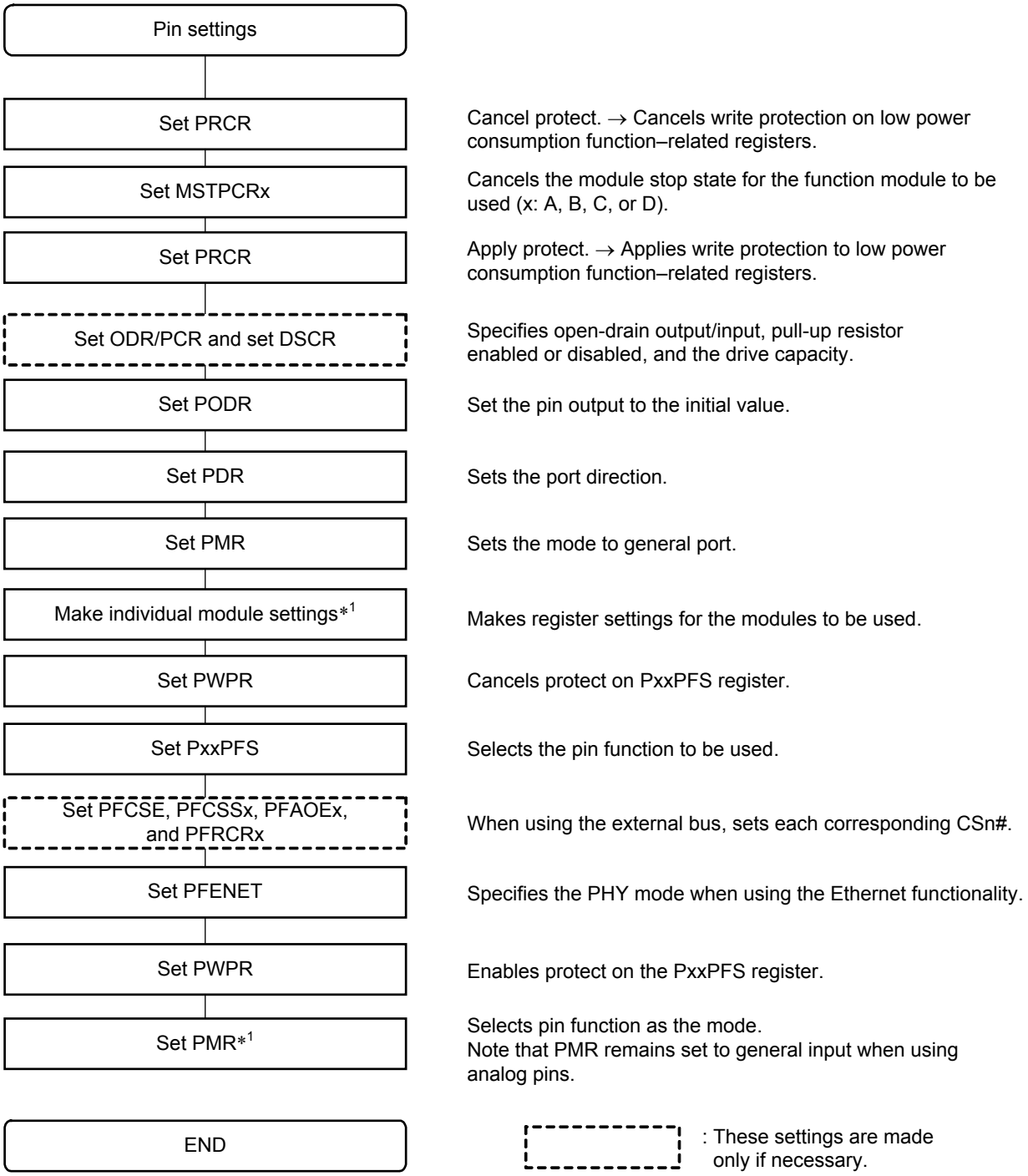


Figure 2.3 RX71M Pin General I/O Setting Flowchart



Note: For details on the MPC settings used to assign functions to pins, see the section describing the specific peripheral function.

Note 1. The order of PMR settings and module settings differs according to the module.

Figure 2.4 RX71M Pin Peripheral Function Setting Flowchart

2.2.3 General I/O Setting Examples

General I/O port setting examples for the SH7216 Group and RX71M are shown below. For the RX71M, information is also shown for pins connected to LEDs on the Renesas Starter Kit+ for RX71M.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.7 General I/O Port Setting Example Specifications

Procedure	SH7216 Group	RX71M
Pins used	PA5/port	P03/general (LED0)
	—	P05/general (LED1)
	—	P26/general (LED2)
	—	P27/general (LED3)

Figure 2.5 presents flowcharts showing examples of processing for setting I/O ports. The names of the processing steps shown in the flowcharts correspond to the names in the setting examples.

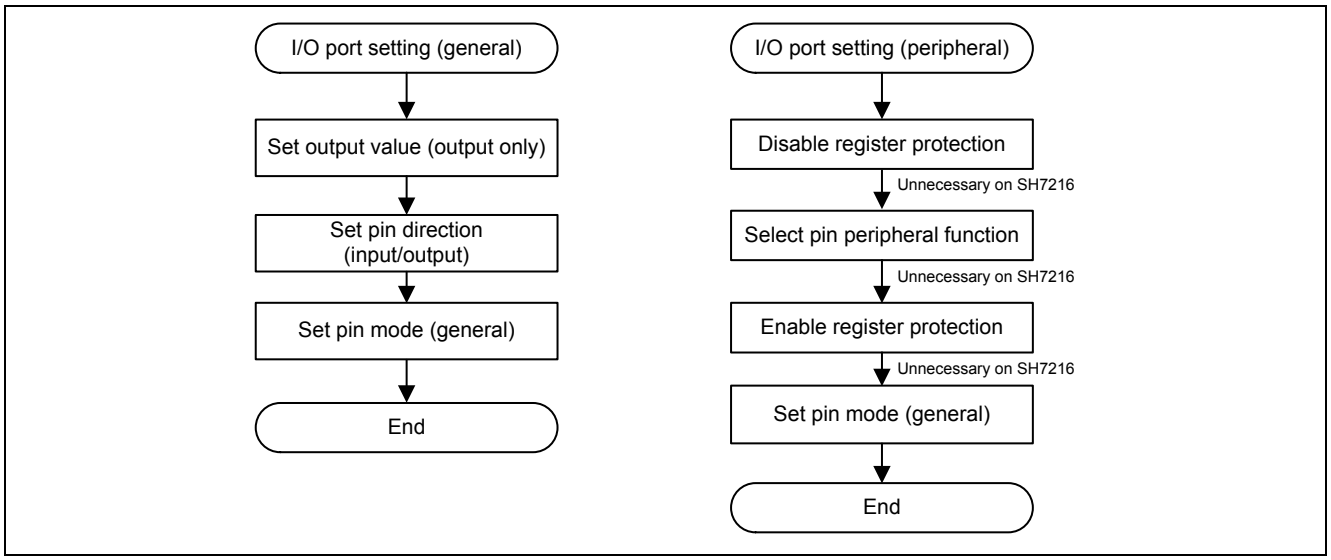


Figure 2.5 Example Flowcharts of I/O Port Setting

Table 2.8 shows examples of general input settings, and Table 2.9 of general output settings, for PA5 on the SH7216 Group and P03 on the RX71M.

Table 2.8 General Input Setting Examples

Processing	SH7216 Group	RX71M
Pin direction setting (input)	PFC.PAIORL.B5 = 0b	PORT0.PDR.B3 = 0b
Pin mode setting (general)	PFC.PACRL2.PA5MD = 000b	PORT0.PMR.B3 = 0b

Table 2.9 General Output Setting Examples

Processing	SH7216 Group	RX71M
Output value setting (output 1: LED = off)	PA.DR.B5 = 1b	PORT0.PODR.B3 = 1b
Pin direction setting (output)	PFC.PAIORL.B5 = 1b	PORT0.PDR.B3 = 1b
Pin mode setting (general)	PFC.PACRL2.PA5MD = 000b	PORT0.PMR.B3 = 0b

2.3 Buses

2.3.1 Comparison of Specifications

The SH7216 Group incorporates a BSC that provides bus state controller functionality.

Table 2.10 is a comparative listing of the specifications of the SH7216 Group and RX71M.

Table 2.10 Comparison of SH7216 Group and RX71M Specifications (Bus)

Item	SH7216 Group (BSC)	RX71M
External bus address space	• External address space designated as areas CS0 to CS7 (max. 64 MB each) Ability to select SDRAM for up to two CS areas (max. 64 MB)	• External address space designated as areas CS0 to CS7 (16 MB each) • Independent SDRAM space (max. 128 MB)
Bus width	Ability to select the data bus width (8, 16, or 32 bits) for each area	Ability to select the data bus width (8, 16, or 32 bits) for each area Note that selection of multiplex bus for address or data is not possible when 32-bit bus width is specified.
Endianness	• Data Area 0: Fixed big-endian Areas 1 to 7: Endian setting selectable independently by area • Instructions Fixed big-endian to match the CPU	• Data Endian setting by area*1 • Instructions The endian setting must match that of the CPU.
Bus arbitration	• CPU bus and external bus have fixed priority. • Ability to output bus enable (BACK) after receiving a bus request (BREQ) from an external device.	• Fixed or toggled priority — Memory bus — Internal peripheral bus — External bus • Fixed priority — CPU bus — Internal main bus
Other	• CS area — Access wait control — CSn assert duration extension — MPX I/O interface (address data multiplexed) — Support for SRAM with byte selection — Burst ROM (synchronous/asynchronous) support • SDRAM area — Auto refresh and self-refresh — CAS latency setting	• CS area — Ability to insert recovery cycles — Cycle wait function — CSn# signal timing control — RD# and WR# signal control timing — Write access mode — Ability to access address and data multiplexed I/O devices • SDRAM area — Multiplexed output of row and column addresses — Auto refresh and self-refresh — CAS latency setting • Write buffer — Write buffer function

Note 1. Refer to 1.3.2, Endian Setting for information on endian settings.

2.3.2 Bus Block Diagrams

Comparative bus block diagrams of the SH7216 Group and RX71M are presented below.

Figure 2.6 is a block diagram of the BSC of the SH7216 Group, and Figure 2.7 is a bus block diagram of the RX71M.

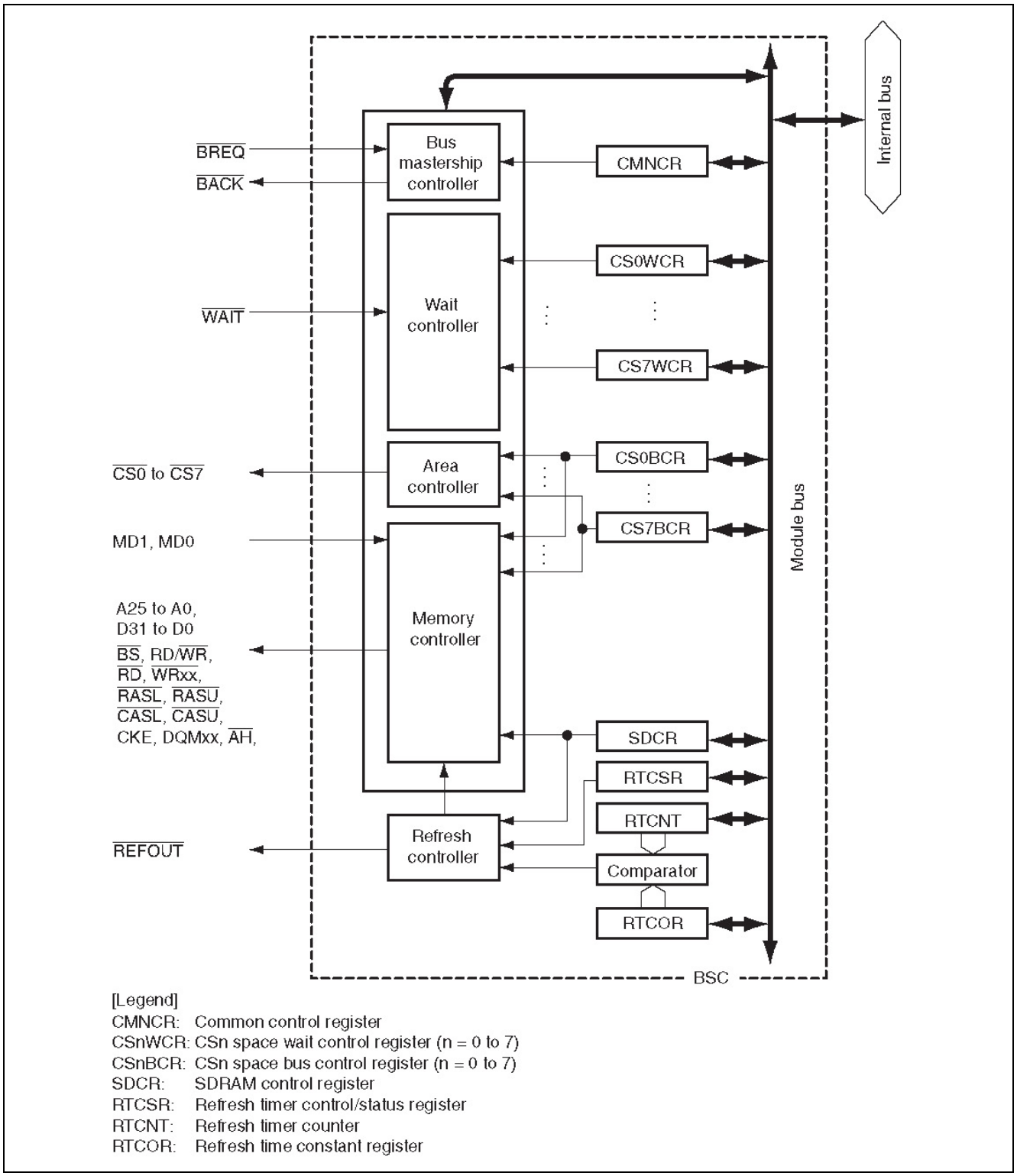


Figure 2.6 SH7216 Group Bus Block Diagram

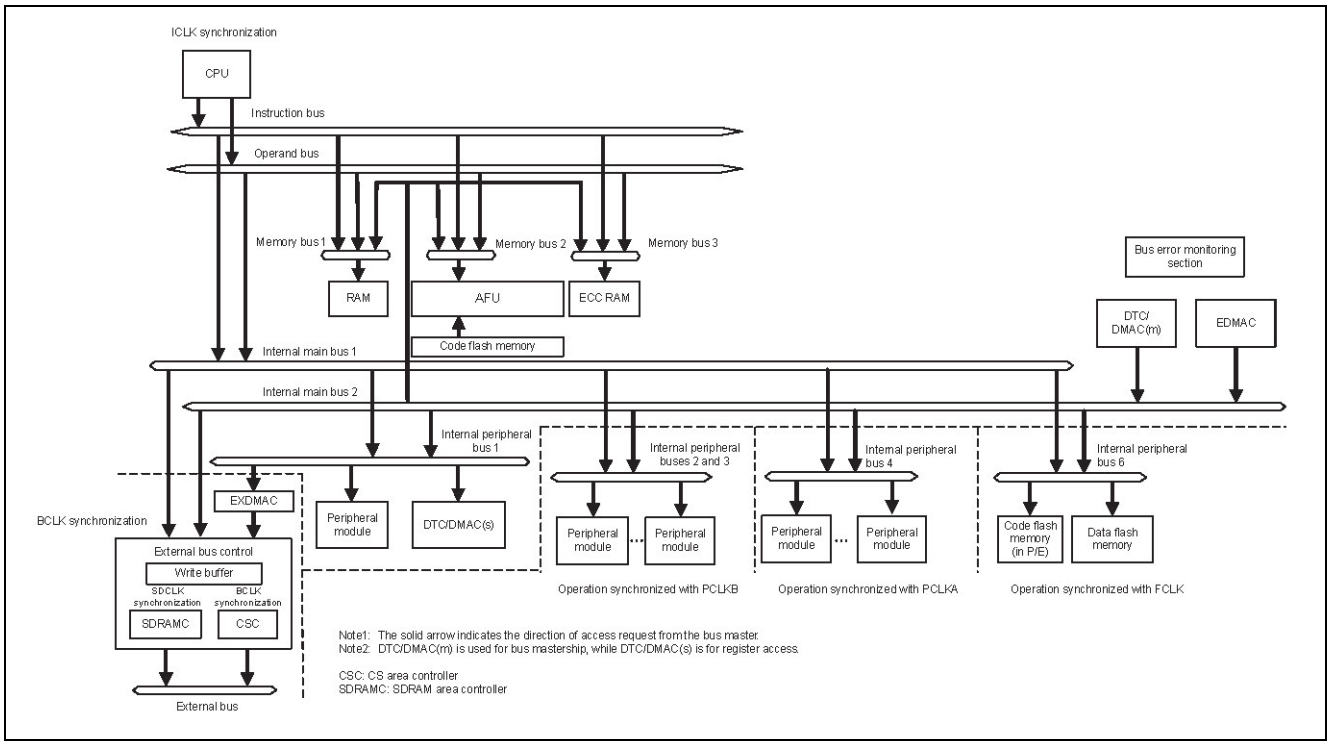


Figure 2.7 RX71M Bus Block Diagram

Table 2.11 shows the bus types on the RX71M. The RX71M has a different bus architecture than the SH7216 Group, and the memory buses, internal buses, and peripheral buses each have multiple stages. This enables parallel operation by the CPU and DMAC, DTC, or EDMAC and between the modules on the peripheral buses, thereby speeding up operation overall.

Table 2.11 RX71M Buses

Bus	Connected Modules, etc.	Clock
CPU buses (instruction bus and operand bus)	Instruction bus: CPU, on-chip memory Operand bus: CPU, on-chip memory	ICLK
Memory bus 1	On-chip RAM	ICLK
Memory bus 2	Code flash memory	ICLK
Memory bus 3	ECCRAM	
Internal main bus 1	CPU	ICLK
Internal main bus 2	DTC, DMAC, EDMAC, on-chip memory	ICLK
Internal peripheral bus 1	Peripheral functions (DTC, DMAC, EXDMAC, interrupt controller, bus error monitoring block)	ICLK (EXDMAC: BCLK)
Internal peripheral bus 2	Peripheral functions (peripheral functions other than those connected to peripheral buses 1, 3, 4, and 5)	PCLKB
Internal peripheral bus 3	Peripheral functions (USBb, PDC, standby RAM)	PCLKB
Internal peripheral bus 4	Peripheral functions (EDMAC, ETHERC, EPTPC, MTU3, GPT, SCIF, RSPI, USBA, AES)	PCLKA
Internal peripheral bus 5	Reserved area	—
Internal peripheral bus 6	Code flash memory (P/E), data flash memory	FCLK
External buses (CS areas)	External devices	BCLK
External buses (SDRAM)	SDRAM	SDCLK

ICLK: System clock PCLKA, PCLKB: Peripheral module clock
FCLK: FlashIF clock BCLK: External bus clock SDCLK: SDRAM clock

2.3.3 SDRAM Read/Write Setting Example

As an example of bus settings on the SH7216 Group and RX71M, the SDRAMC is used to make read/write settings for a 128 Mbit SDRAM area (MT48LC8M16A2P-6A from Micron Technology: 2 megawords × 16 bits × 4 banks).

Operational Overview

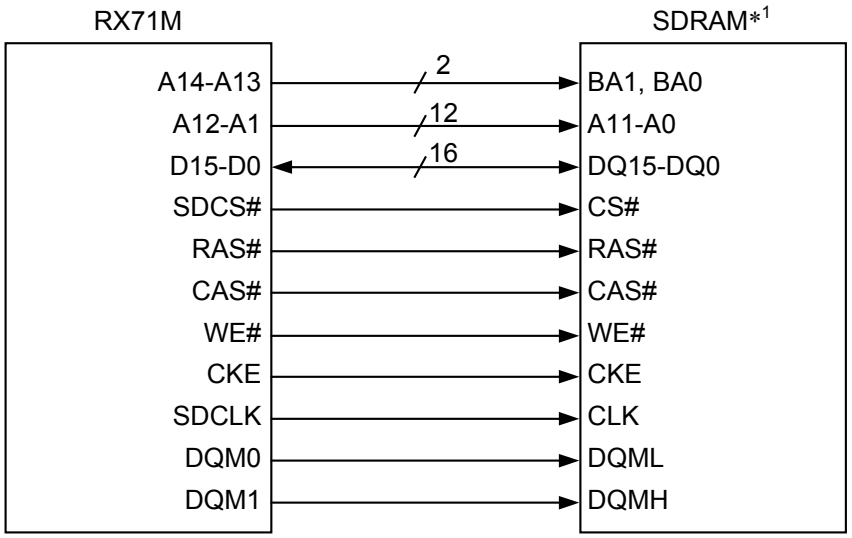
- Data is written to, then read from, the SDRAM area.
- All the data is written to the SDRAM area.
- After writing of all the data to the SDRAM area finishes, the previously written data is read from the SDRAM area.
- The write data and read data are compared.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by "SH7216:" in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.12 SDRAM Read/Write Operation Specifications

Item		Description	Remarks
Clock		SDCLK = 60 MHz	SH7216: Bϕ = 50 MHz
External bus area		SDCS	SH7216: CS3
Address interface		Address multiplexing	
Access mode		Burst read/single write	
Endian setting		Same endian setting as operating mode	SH7216: big-endian
Bus width		Word	
Write/read data		16 MB Repetition of values from 0h to ffffh	
Pins used	A0 to A11	A1 to A12 A1 to A7/PA1 to PA7, A8 to A12/PB0 to /PB4	SH7216: A1 to A12/PC1 to PC12
	BA0, BA1	A13, A14/PB5, PB6	SH7216: A13, A14/PC13, PC14
	DQ0 to DQ15	D0 to D15 D0 to D7/PD0 to PD7, D8 to D15/PE0 to PE7	SH7216: D0 to D15/PD0 to PD15
	CLK	P70/SDCLK	SH7216: PA18/CK
	CS#	P61/SDCS#	SH7216: PA9/CS3#
	RAS#	P62/RAS#	SH7216: PB2/RASL#
	CAS#	P63/CAS#	SH7216: PB3/CASL#
	WE#	P64/WE#	SH7216: PB0/RD/WR#
	CKE	P65/CKE	SH7216: PA21/CKE
	DQML	P66/DQM0	SH7216: PA15/DQMLU
	DQMH	P67/DQM1	SH7216: PA16/DQMLL
	LED0	P03/general	Lights when comparison result matches.
	LED1	P05/general	Lights when comparison result does not match.



Note 1. Using MT48LC8M16A2P-6A (2 megawords × 16 bits × 4 banks).

Figure 2.8 SDRAM Connection

Table 2.13 SDRAM (MT48LC8M16A2P-6A) Specifications

Item	Description
Configuration	2 megawords × 16 bits × 4 banks (Micron Technology)
Capacity	128 MB
Row addresses	A11 to A0
Column addresses	A8 to A0
Auto-refresh interval	4,096 refresh cycles (max.) every 64 msec.
CAS latency	3 cycles
Initialization auto-refresh count	2 or more
Auto-refresh time (tRFC)	60 ns (min.)
Write recovery time (tWR)	Auto-precharge mode: 1 CLK + 6 ns (min.) Precharge mode: 12 ns (min.)
Precharge command time (tRP)	18 ns (min.)
Time from active command to precharge command (tRAS)	42 ns (min.) to 120,000 ns (max.)
Delay time from active command to read/write command (tRCD)	18 ns (min.)

Setting Values

The SDRAM setting values used in the setting examples for the initialization sequence, auto refresh, and read/write timing are listed below.

Table 2.14 Initialization Sequence Setting Values with SDRAM (MT48LC8M16A2P-6A) Connected

Item	Symbol	Description	SH7216 Group Setting	RX71M Setting
Initialization auto refresh interval	tRP	18 ns (min.)	CS3WCR.WTRP = 00b (0 cycles: 20 ns)	SDIR.PRC = 000b (3 cycles: approx. 50 ns)
Initialization refresh count	tRFC	60 ns (min.)	CS3WCR.WTRC = 00b (2 cycles: 60 ns)	SDIR.ARFI = 001b (4 cycles: 66 ns)
Initialization auto refresh interval	—	2 times or more	Fixed at 8 times	SDIR.ARFC = 0010b (2 times)

Table 2.15 Auto Refresh Setting Values with SDRAM (MT48LC8M16A2P-6A) Connected

Item	Symbol	Description	SH7216 Group Setting	RX71M Setting
Auto refresh interval	—	15.625 µs (max.) (tREF/low address count)*1	RTCOR = A55A00C3h (195 cycles: 15.6 µs) RTCSR.CKS = 001b*2 (count-up clock = Bϕ/4)	SDRFCR.RFC = 3A9h (937 cycles: 15.617 µs)
Auto refresh cancel cycle	tRFC	60 ns (min.)	CS3WCR.WTRC = 00b (2 cycles: 60 ns)	SDRFCR.REFW = 0011b (4 cycles: approx. 66 ns)

Note 1. Refresh cycle (tREF) = 64 ms (max.), low address count = 4,096

Note 2. RTCSR requires 32-bit access and cancellation of write protection. Selecting the input clock by means of the CKS bits in the RTCSR register causes the refresh time counter to start.

Table 2.16 Read/Write Access Timing Setting Values with SDRAM (MT48LC8M16A2P-6A) Connected

Item	Symbol	Description	SH7216 Group Setting	RX71M Setting
Column latency*2	—	3 cycles*1	CS3WCR.A3CL = 10b	SDTR.CL = 011b
Write recovery period	tWR	1CLK + 6 ns (min.) SH7216: 26 ns (min.) RX71M: 22.66 ns (min.)	CS3WCR.TRWL = 01b (1 cycle: 40 ns)	SDTR.WR = 1b (2 cycles: approx. 33 ns)
Low precharge period	tRP	18 ns (min.)	CS3WCR.WTRP = 00b (0 cycles: 20 ns)	SDTR.RP = 001b (2 cycles: approx. 33 ns)
Low active period*2	tRAS	42 ns (min.)	—	SDTR.RAS = 010b (3 cycles: approx. 50 ns)
Low column latency*2	tRCD	18 ns (min.)	CS3WCR.WTRCD = 00b (0 cycles: 20 ns)	SDTR.RCD = 01b (2 cycles: approx. 33 ns)

Note 1. A setting of 3 is selected in the SDRAM mode register.

Note 2. The low active period should be set to a value less than or equal to (low column latency + column latency).

Table 2.17 lists the SDRAM mode register setting values for the SDRAM used.

Whereas on the SH7216 Group writing data to the SDRAM mode register is accomplished by writing in word units to a specified address, on the RX71M writing data to the SDRAM mode register is accomplished by setting register values.

Table 2.17 SDRAM Mode Register Setting Values with SDRAM (MT48LC8M16A2P-6A) Connected

Bit	Symbol	Setting Value
b2 to b0	BurstLength	000: 1
b3	BurstType	0: Sequential
b6 to b4	CASLatency	011: 3
b8, b7	OperatingMode	00: StandardOperation
b9	WriteBurstMode	1: SingleLocationAccess
b11, b10	Reserved	00: Reserved

Processing Flowchart

Figure 2.9 shows an example processing flowchart when SDRAM is connected. The names of the processing steps in this flowchart correspond to the names in the setting examples.

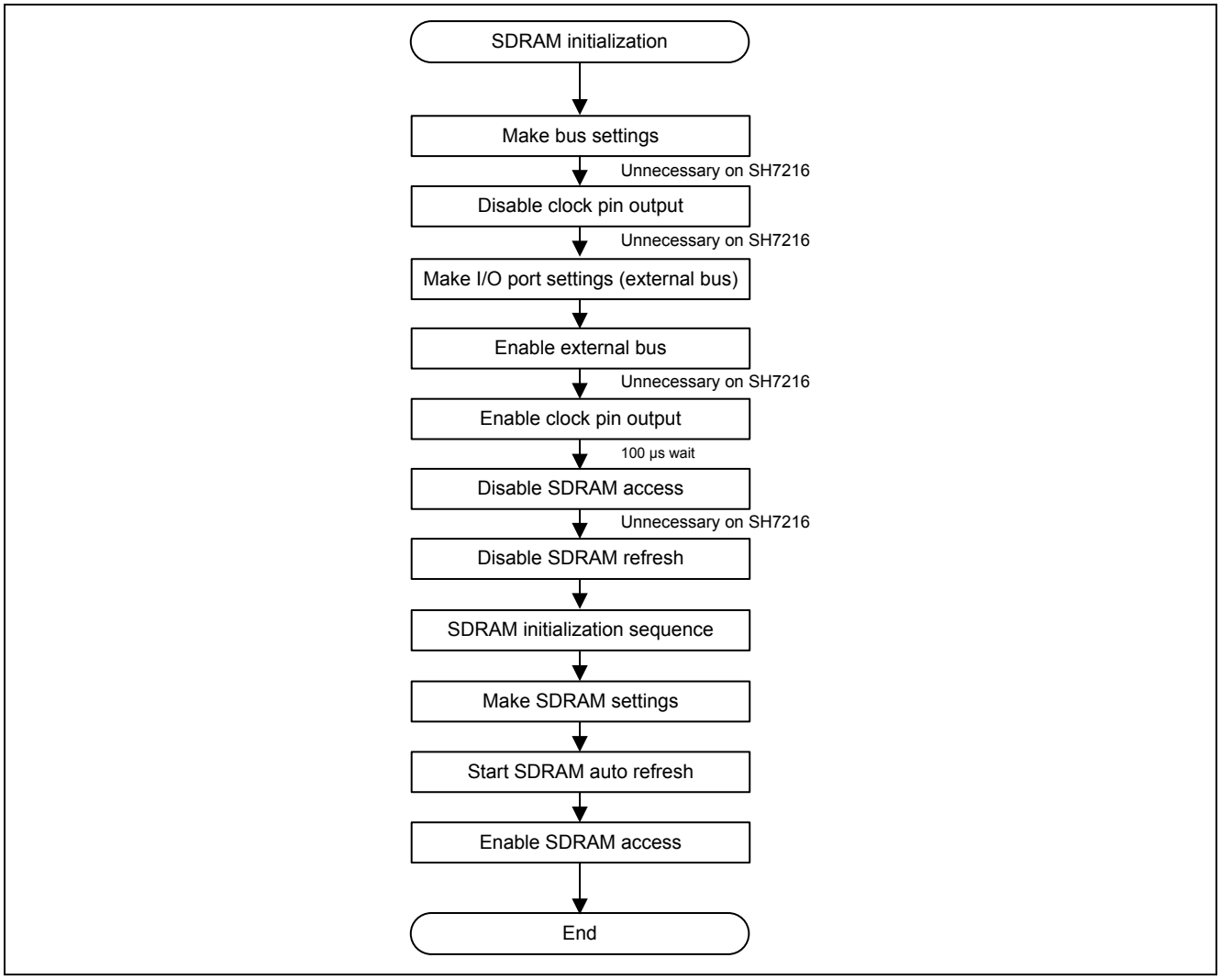


Figure 2.9 External Bus (SDRAM) Processing Flowchart

Setting Examples

Setting examples with SDRA connected are shown below. The names of the processing steps in these setting examples correspond to the names in the flowchart.

Table 2.18 Bus Initialization Setting Example (SDRAM Connected)

Procedure	SH7216 Group Setting Example	RX71M Setting Example
Make bus settings.	—	Bus error settings (detection disabled) BSC.BEREN.IGAEN = 0b (illegal address) BSC.BEREN.TOEN = 0b (timeout) Bus priority setting (fixed) BSC.BUSPRI.BPEB = 00b
Disable clock pin output.	—	External bus clock pin output disabled SYSTEM.PRCR = A503h SYSTEM.SCKCR.PSTOP0 = 1b (SDCLK) SYSTEM.SCKCR.PSTOP1 = 1b (BCLK) Register write protection is set to writing disabled after clock pin output is enabled.
Make I/O port settings (external bus).		External bus control pins enabled MPC.PFBCR0 = 11h (A0 to A7, D8 to D15) MPC.PFBCR1 = D0h (SDCLK, DQM1, CKE, SDCS#, RAS#, CAS#, WE#, DQM0) MPC.PFAOE0 = 7Fh (A14 to A8) Setting of D0 to D7 not required.
	Pin mode settings PFC.PCCRL4 = 1111h (A12 to A15) PFC.PCCRL3 = 1111h (A8 to A11) PFC.PCCRL2 = 1111h (A4 to A7) PFC.PCCRL1 = 1111h (A0 to A3) PFC.PDCRL4 = 1111h (D12 to D15) PFC.PDCRL3 = 1111h (D8 to D11) PFC.PDCRL2 = 1111h (D4 to D7) PFC.PDCRL1 = 1111h (D0 to D3) PFC.PACRL4 = 1011h (CK, WRH#/DQMLU, WRL#/DQMLL) PFC.PACRL3 = 0055h (CKE, RDWR) PFC.PACRL2 = 2000h (CS3#) PFC.PBCRL2 = 0044h (CASL#, RASL#)	Pin mode settings (general) PORTA.PMR = 00h (A0 to A7) PORTB.PMR &= 80h (A8 to A14) PORTD.PMR = 00h (D0 to D7) PORTE.PMR =00h (D8 to D15) PORT6.PMR &= 01h (SDCS#, RAS#, CAS#, WE#, CKE, DQM0, DQM1) PORT7.PMR &= FEh (SDCLK)
Enable external bus.	—	SYSTEM.SYSCR0 = 5A03h Confirm update of SYSTEM.SYSCR0.EXBE.
Enable clock pin output.	—	External bus clock pin output enabled SYSTEM.SCKCR.PSTOP0 = 0b (SDCLK) SYSTEM.PRCR = A500h
Disable SDRAM access.	—	BSC.SDCCR.EXENB = 0b

Procedure	SH7216 Group Setting Example	RX71M Setting Example
Disable SDRAM refresh.	SDRAM refresh control BSC.SDCR.RFSH = 0b	SDRAM auto refresh disabled BSC.SDRFEN.RFEN = 0b SDRAM self-refresh disabled BSC.SDSELF.SFEN = 0b
SDRAM initialization sequence	Initialization sequence settings*1 BSC.CS3WCR.WTRP = 00b BSC.CS3WCR.WTRC = 00b	Initialization sequence settings*1 BSC.SDIR.PRC = 000b BSC.SDIR.ARFC = 0010b BSC.SDIR.ARFI = 0001b Initialization sequence start BSC.SDICR.INIRQ = 1b Wait until BSC.SDSR.INIST = 0b.
Make SDRAM settings	Bus width setting (16 bits) BSC.CS3BCR.BSZ = 10b Memory type setting BSC.CS3BCR.TYPE = 100b (SDRAM) SDRAM mode register setting*2 SDMR3 address = FFFC5460h Auto refresh settings*3 BSC.RTCOR = A55A00C3h BSC.CS3WCR.WTRC = 00b Endian setting BSC.CS3BCR.ENDIAN = 0b (big-endian) Read/write access timing settings*4 BSC.CS3WCR.WTRCD = 00b BSC.CS3WCR.WTRP = 00b BSC.CS3WCR.A3CL = 10b BSC.CS3WCR.TRWL = 01b Address multiplexing settings BSC.SDCR.A3ROW = 01b (low address: 12 bits) BSC.SDCR.A3COL = 01b (column address: 9 bits)	Bus width setting (16 bits) BSC.SDCCR.BSIZE = 00b SDRAM mode register setting*2 BSC.SDMOD = 0230h Auto refresh settings*3 BSC.SDRFCR.RFC = 3A9h BSC.SDRFCR.REFW = 0011b Endian setting BSC.SDCMOD.EMODE = 0b (same as endian setting of operating mode) Access mode setting BSC.SDAMOD.BE = 0 (continuous access disabled) Read/write access timing settings*4 BSC.SDTR.RCD = 01b BSC.SDTR.RP = 001b BSC.SDTR.CL = 011b BSC.SDTR.WR = 1b BSC.SDTR.RAS = 010b Address multiplexing setting BSC.SDADR.MXC = 01b (low address: shifted 9 bits)
Enable SDRAM refresh	Auto refresh selection BSC.SDCR.RMODE = 0b SDRAM refresh control BSC.SDCR.RFSH = 1b	Auto refresh selection BSC.SDRFEN.RFEN = 1b
Make auto refresh settings	BSC.RTCSR = A55A0008h Refresh count: 1 Refresh timer counter: Bϕ/4	—
Enable SDRAM access	Write 0 to SDMR3 address. (Write to SDRAM mode register.)	BSC.SDCCR.EXENB = 1b

Note 1. See Table 2.14 for details of setting values.
Note 2. See Table 2.17 for details of setting values. The count-up clock setting is made after auto refresh is enabled on the SH7216.
Note 3. See Table 2.15 for details of setting values.
Note 4. See Table 2.16 for details of setting values.

2.4 Interrupt Controller

2.4.1 IRQ Setting Example

A setting example using IRQ on the SH7216 Group and RX71M is shown below.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.19 IRQ Setting Example Specifications

Item	Description	Remarks
Detection condition	Falling-edge detection	
Interrupt priority	Level 15	
Noise cancellation	Sampling at PCLKB/64 (PCLKB = 60 MHz)	SH7216: No such function
Pins used	P41/general (IRQ9-DS)	SH7216: PC13/IRQ0

List of Related Registers

Table 2.20 and Table 2.21 show the IRQ interrupt-related registers used in the SH7216 Group and RX71M setting examples broken down by source.

Table 2.20 SH7216 Interrupt-Related Registers (INTC)

Item	Vector No.	Name	Interrupt Enable	Status	Priority Level
Setting register	—	—	—	IRQRR	IPR01
Setting position of each source					
IRQ0	64	IRQ0	—	IRQ0F	bit 12 to 15

Table 2.21 RX71M Interrupt-Related Registers (ICUA)

Item	Vector No.	Name	Interrupt Enable	Status	Priority Level
Setting register	—	—	IERm	IRr	IPRr
Setting position of each source					
IRQ9-DS	73	IRQ9	IER09.IEN1	IR073	IPR073

When making settings to the ICUA interrupt-related registers on the RX71M, iodefine.h can be used to make settings as follows:

- IERm: IEN (ICU or ICUA interrupt name)
- IPRr: IPR (ICU or ICUA interrupt name)
- IRr: IR (ICU or ICUA interrupt name)

Flowcharts of processing using IRQ interrupts and setting examples are shown below. The names of the processing steps shown in the flowcharts correspond to the names in the setting examples.

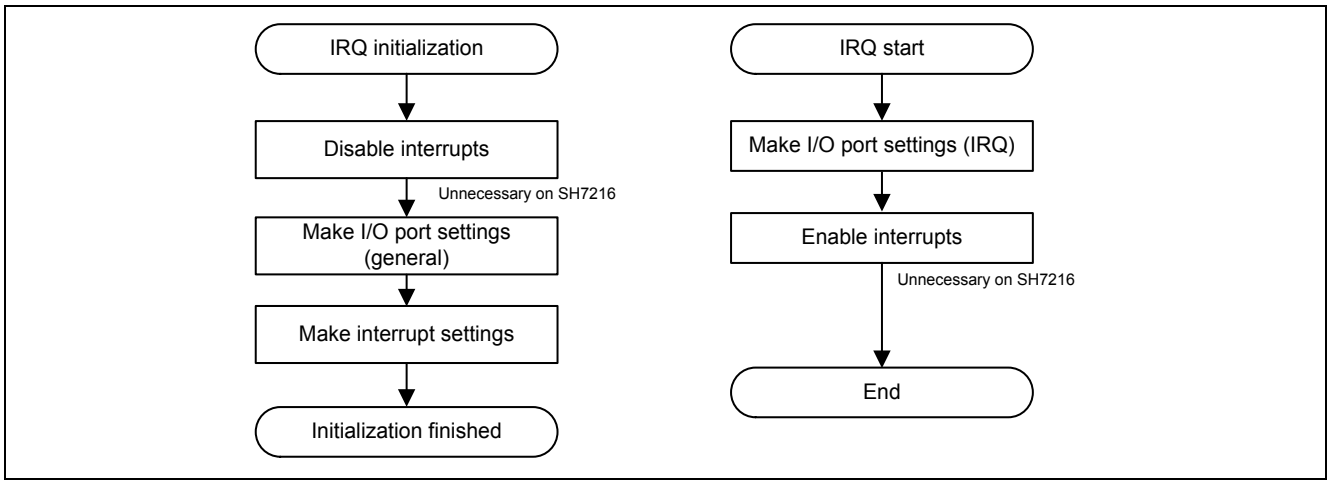


Figure 2.10 Example Flowcharts of IRQ Processing

Table 2.22 Example of Initial IRQ Settings

Processing	SH7216 Group	RX71M
Disable interrupts	—	IEN(ICU, IRQ9) = 0b
Make I/O port settings (general).	Pin direction setting PFC.PCIORL.B13 = 0b (input/PC13) Pin mode setting (port) PFC.PCCRL4.PC13MD = 000b (PC13)	Pin direction setting PORT4.PDR.B1 = 0b (input/P41) Pin mode setting (general) PORT4.PMR.B1 = 0b (P41)
Make interrupt settings.	Detection edge setting INTC.ICR1.IRQ0S = 01b Priority setting (level 15) INTC.IPR01._IRQ0 = 15 Clearing of ICU status Reading value of IRQRR.IRQ0F IRQRR.IRQ0F = 0b	Detection edge setting ICU.IRQCR[9].IRQMD = 01b Digital filter settings ICU.IRQFLTC1.FCLKSEL9 = 11b ICU.IRQFLTE1.FLTEN9 = 1b Priority setting (level 15) IPR(ICU, IRQ9) = 15 Clearing of ICU status IR (ICU, IRQ9) = 0b

Table 2.23 Example of IRQ Start Settings

Processing	SH7216 Group	RX71M
Make I/O port settings (IRQ).	Pin mode setting PFC.PCCRL4.PC13MD = 011b (IRQ0)	Cancellation of register protection MPC.PWPR.B0WI = 0b MPC.PWPR.PFSWE = 1b Pin peripheral function selection MPC.P41PFS.ISEL = 1b (IRQ9-DS) Register protection settings MPC.PWPR.PFSWE = 0b MPC.PWPR.B0WI = 1b
Enable interrupts.	—	IEN(ICU, IRQ9) = 1b

2.5 Data Transfer Controller (DTC)

2.5.1 Comparison of Specifications

Data transfer controller functionality is provided on the SH7216 Group by the DTC and on the RX71M by the DTCa.

On both the SH7216 Group and RX71M transfer information is located in the RAM and specified by means of DTC vectors. The basic operation of the three transfer modes (normal transfer mode, repeat transfer mode, and block transfer mode) is identical on the two platforms. Table 2.24 is a comparative listing of the specifications of the SH7216 Group and RX71M.

Table 2.24 Comparison of SH7216 Group and RX71M Specifications (DTC)

Item	SH7216 Group (DTC)	RX71M (DTCa)
Transfer modes	• Normal transfer mode • Repeat transfer mode • Block transfer mode	
Activation sources	• External interrupt • Peripheral function interrupt	• External interrupt • Peripheral function interrupt • Software interrupt
Activation enable/ disable control	Activated by DTC enable register of DTC module.	Activated by DTC activation enable register of interrupt controller.
Transfer spaces	Transfer between the following spaces is possible: • On-chip memory space • On-chip peripheral module space (excluding DMAC, DTC, BSC, UBC, and FLASH) • External memory space • Memory-mapped external device space At a minimum, the on-chip peripheral module space must be specified as either the transfer source or transfer destination.	Transfer between the following spaces is possible: • On-chip memory space • On-chip peripheral module space • External memory space
Transfer units	• Normal transfer mode: Selectable among 8, 16, and 32 bits • Repeat transfer mode: Selectable among 8, 16, and 32 bits • Block transfer mode: Selectable within range from 8 bits to 256 longwords	
Transfer counts	• Normal transfer mode: 1 to 65,536 • Repeat transfer mode: 1 to 256 times (repeat after completion of specified transfer count) • Block transfer mode: 1 to 65,536	
CPU interrupt requests	• An interrupt generated by a CPU interrupt request may be used as the DTC activation source. • A CPU interrupt at single data unit transfer-end may be used. • A CPU interrupt after transfer of a specified number of data units may be used.	
Method	Control information is allocated for each interrupt source by using DTC vectors.	
Other	• Chain transfer • Transition to module-stop state • The following functions can be used to shorten the transfer duration and reduce memory usage: — Transfer information read skipping — Write-back skipping — Short-address mode — Bus mastership release timing setting	• Chain transfer • Event link • Transition to module-stop state • The following functions can be used to shorten the transfer duration and reduce memory usage: — Transfer information read skipping — Write-back skipping — Short-address mode

2.5.2 Register Comparison

On the SH7216 Group operation of the DTC is enabled by canceling the module-stop state for the DTC. On the RX71M, in addition to canceling the module-stop state for the DTC, it is necessary to make a setting in the DTC module start register (DTCST) to enable DTC operation.

Table 2.25 provides a comparative listing of the registers of the SH7216 Group and the RX71M.

Guide to Symbols in "Changes" Column of Table

◎: Register with same bit assignments on SH7216 Group and RX71M

△: Register with different bit assignments on SH7216 Group and RX71M

—: Register not present on SH7216 Group or RX71M

Table 2.25 SH7216 Group and RX71M Register Comparison (DTC)

SH7216 Group (DTC)	RX71M (DTCa)	Changes
DTC mode register A (MRA)	DTC mode register A (MRA)	◎
DTC mode register B (MRB)	DTC mode register B (MRB)	◎
DTC source address register (SAR)	DTC transfer source register (SAR)	◎
DTC destination address register (DAR)	DTC transfer destination register (DAR)	◎
DTC transfer count register A (CRA)	DTC transfer count register A (CRA)	◎
DTC transfer count register B (CRB)	DTC transfer count register B (CRB)	◎
DTC control register (DTCCR)	DTC control register (DTCCR)	△
DTC vector base register (DTCVBR)	DTC vector base register (DTCVBR)	◎
Bus function extending register (BSCEHR) DTC short address mode (DTSA bit)	DTC address mode register (DTCADMOD)	△
DTC enable registers A to E (DTCERA to DTCERE)*1	—	—
—	DTC module start register (DTCST)	—
	DTC status register (DTCSTS)	

Note 1. On the RX71M transfer request settings from peripheral modules are made by means of the interrupt controller.

2.5.3 Activation Source Settings

On the SH7216 Group peripheral modules can activate the DTC by making settings in activation source DTC enable registers A to E (DTCERA to DTCERE) of the DTC module. On the RX71M DTC activation sources are specified by means of settings to DTC transfer request enable register n (DTCERn) of the interrupt controller. This allows specific interrupts to be enabled as activation sources for enabling the DTC.

2.5.4 DTC Vector Configuration

The DTC vector configuration of the SH7216 Group and RX71M is shown below.

On the SH7216 Group the upper 20 bits of the start address of the DTC vector table are the DTC vector base address (DTCVBR) and the lower 12 bits are calculated as “400h + vector number × 4”. The base address of the DTC vector table is aligned with a 4 KB boundary such that the lower 12 bits are 0.

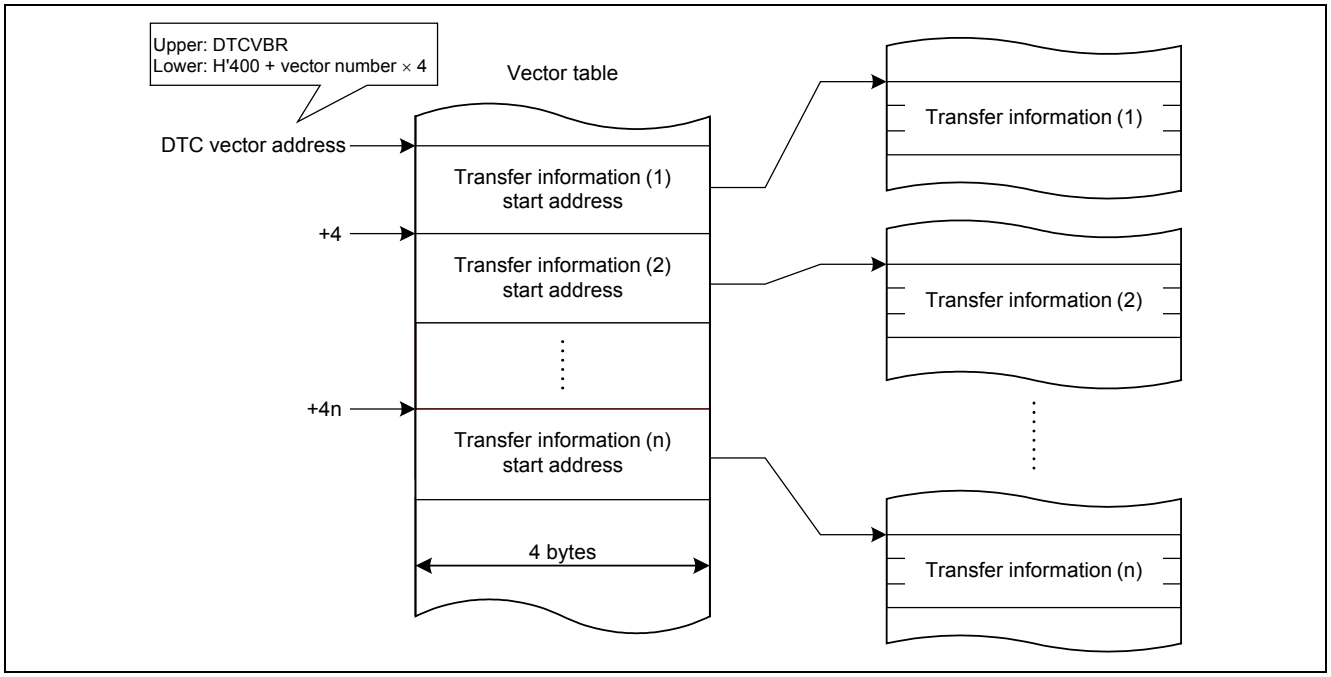


Figure 2.11 DTC Vector Configuration on SH7216 Group

On the RX71M the start address of the DTC vector table is calculated as “DTC vector base address (DTCVBR) + (vector number × 4)”. The base address of the DTC vector table is aligned with a 1 KB boundary such that the lower 10 bits are 0.

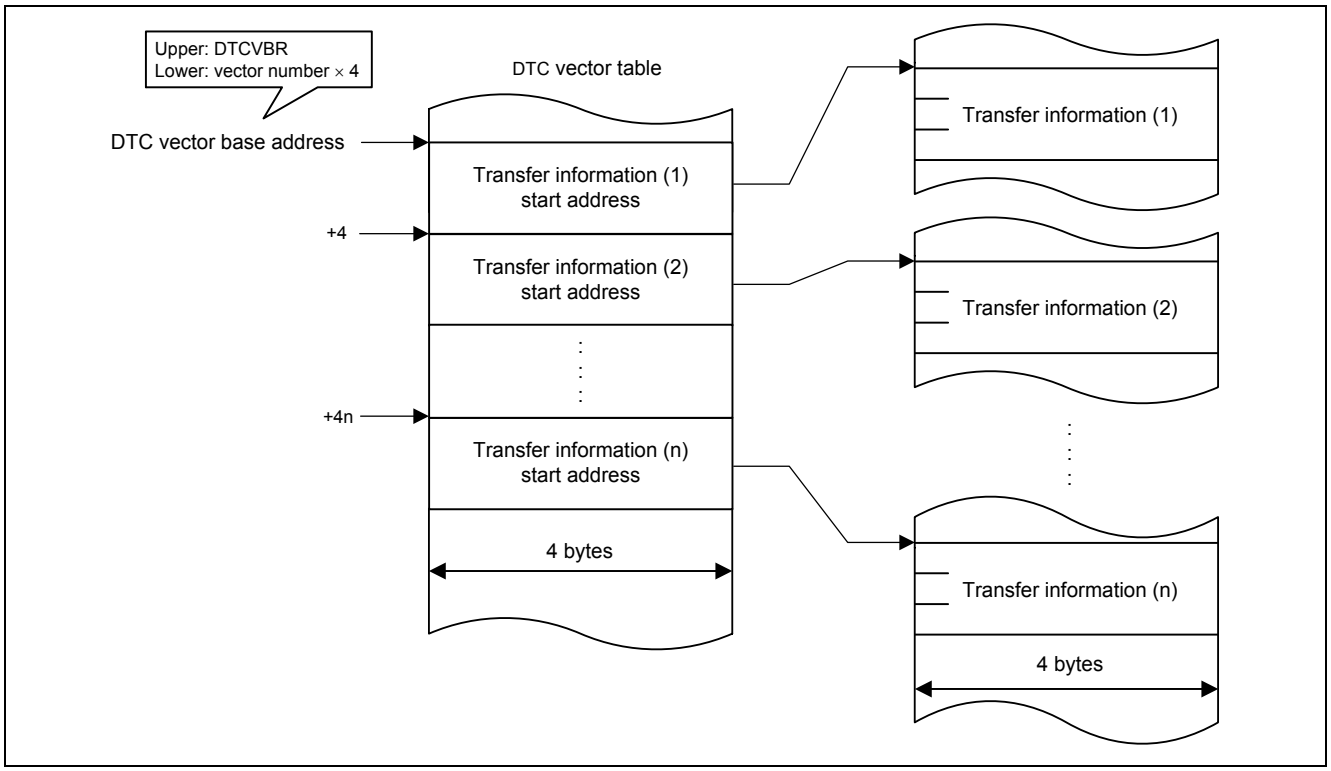


Figure 2.12 DTC Vector Configuration on RX71M

2.5.5 Allocation of Transfer Information

On the SH7216 Group the format of DTC transfer information is fixed at big-endian. On the RX71M the endian setting for DTC transfer information depends on the allocation area. The allocation of transfer information other than the endian setting is identical.

Short address mode is selected on the SH7216 Group by making a setting in the bus function extending register (BSCEHR) of the BSC and on the RX71M by making a setting in the DTC address mode register (DTCADMOD). Figure 2.13 illustrates the DTC transfer source and transfer destination addresses in short address mode.

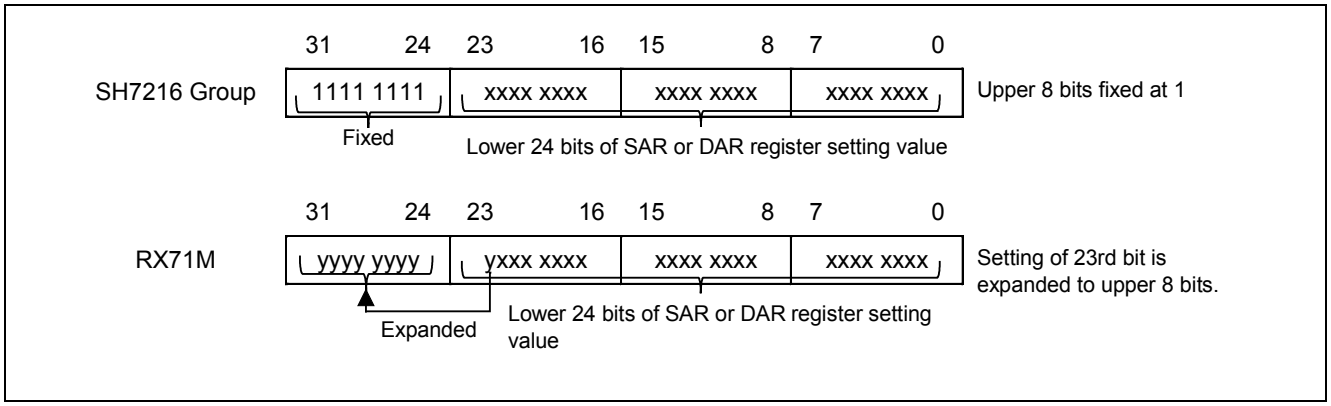


Figure 2.13 Transfer Source and Transfer Destination Addresses in Short Address Mode

Figure 2.14 shows the allocation of DTC transfer information on the SH7216 Group and RX71M.

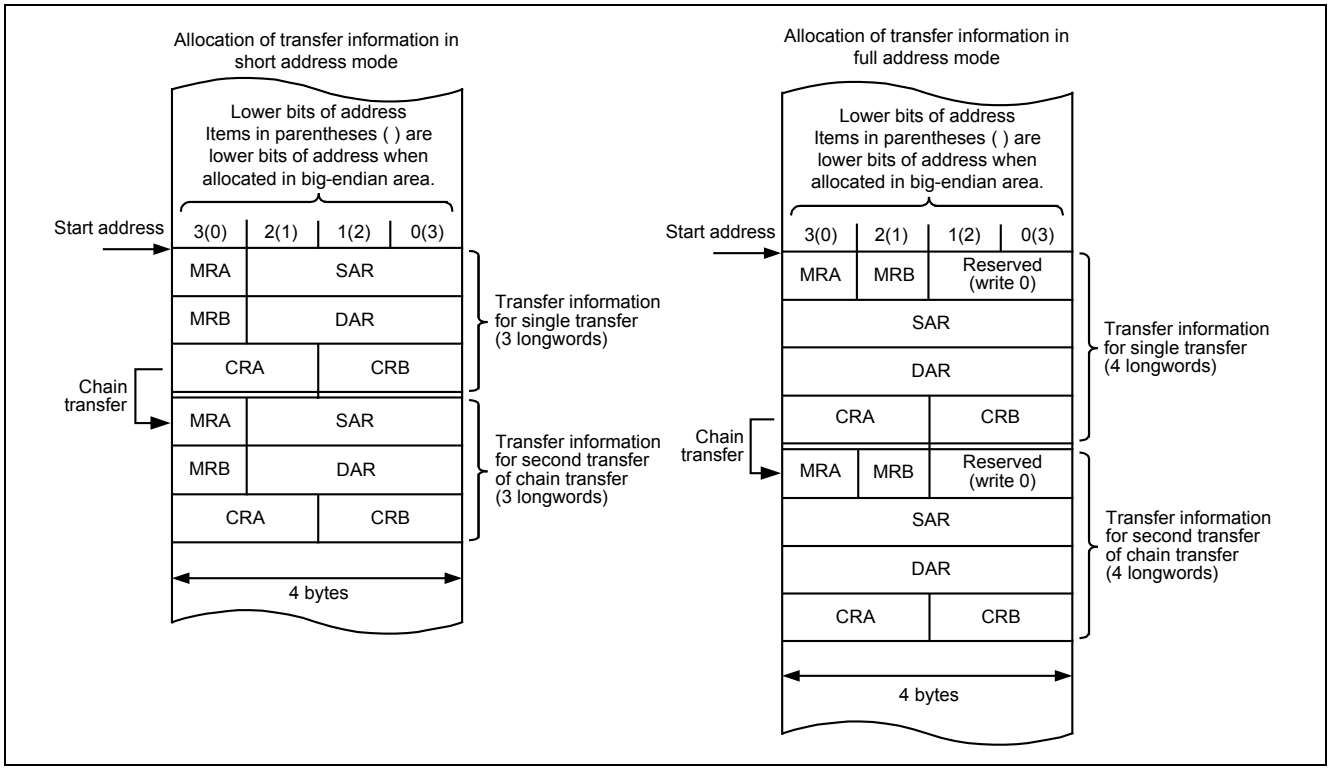


Figure 2.14 Allocation of Transfer Information

2.5.6 Module Stop

On the RX71M the DTCa module-stop state is canceled after a reset.

The module-stop setting bit (MSTPCRA.MSTPA28) is common to both the DTCa and DMACAa on the RX71M, so module-stop control for these two modules is simultaneous.

Refer to 2.21, Low Power Consumption Function for information on the module-stop state.

2.5.7 Setting Examples for Data Transfer between SCI and On-Chip RAM

Setting examples for performing data transfer between the SCI and on-chip RAM using the data transfer controller on the SH7216 Group and RX71M are presented below. Additionally, refer to the SCI initial setting examples in 2.10.6, Asynchronous Transmit/Receive Setting Examples (Interrupt/Polling). The examples presented below only show settings for activating the DTC by means of SCI interrupts.

Operational Overview

- The SCI is used to perform asynchronous transmission and reception.
- The DTC is activated by a transmit data-empty interrupt request from the SCI and transfers data from the on-chip RAM to a register of the SCI.
- The DTC is activated by a receive data-full interrupt request from the SCI and transfers data from the SCI register to the on-chip RAM.
- Transmission and reception of all the data takes place without intervention by software.
- When transmission of all the data finishes (at DTC transfer end), the SCI generates a transmit data-empty interrupt.
- When reception of all the data finishes (at DTC transfer end), the SCI generates a receive data-full interrupt.
- After transmission and reception of all the data finishes, SCI and DTC operation ends.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by “SH7216:” in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.26 Operation Specifications of Data Transfer between SCI and On-Chip RAM

Item		Description	Remarks
SCI specifications		See Table 2.70.	SCI2 asynchronous transmission/reception
Transfer mode		Normal transfer mode	
Transfer count		32	
Transfer size		Byte	
Transmit	Transfer source	On-chip RAM (incremented following transfer)	
	Transfer destination	SCI transmit data register (fixed address)	
	Activation sources	SCI transmit data-empty interrupt	
Receive	Transfer source	SCI receive data register (fixed address)	
	Transfer destination	On-chip RAM (incremented following transfer)	
	Activation sources	SCI receive data-full interrupt	
Address mode		Full address mode	
Interrupts		Interrupt to CPU when transfer of specified data finishes	
Pins used	TXD	P50/TXD2	SH7216: PD3/TXD2
	RXD	P52/RXD2	SH7216: PD2/RXD2
	LED1	P05/general	Lights at transfer end.
	LED2	P26/general	Lights when error detected.

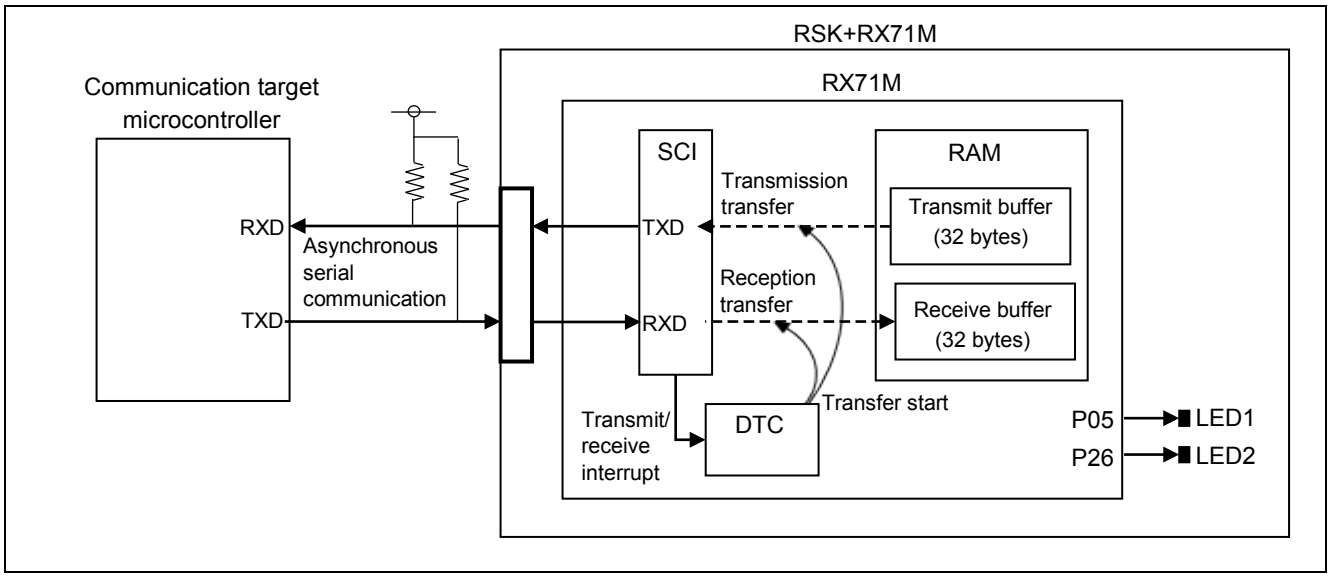


Figure 2.15 Connection Example for Data Transfer between SCI and On-Chip RAM

Note

On the Renesas Starter Kit+ for RX71M in the initial state the pins used for SCI2 in the connection example are connected for use with the external bus, so make appropriate modifications to the board as necessary.

Processing Flowcharts

Table 2.16 shows example flowcharts of processing using the DTC. The names of the processing steps shown in the flowcharts correspond to the names in the setting examples. For SCI-related processing, refer to 2.10.6, Asynchronous Transmit/Receive Setting Examples (Interrupt/Polling).

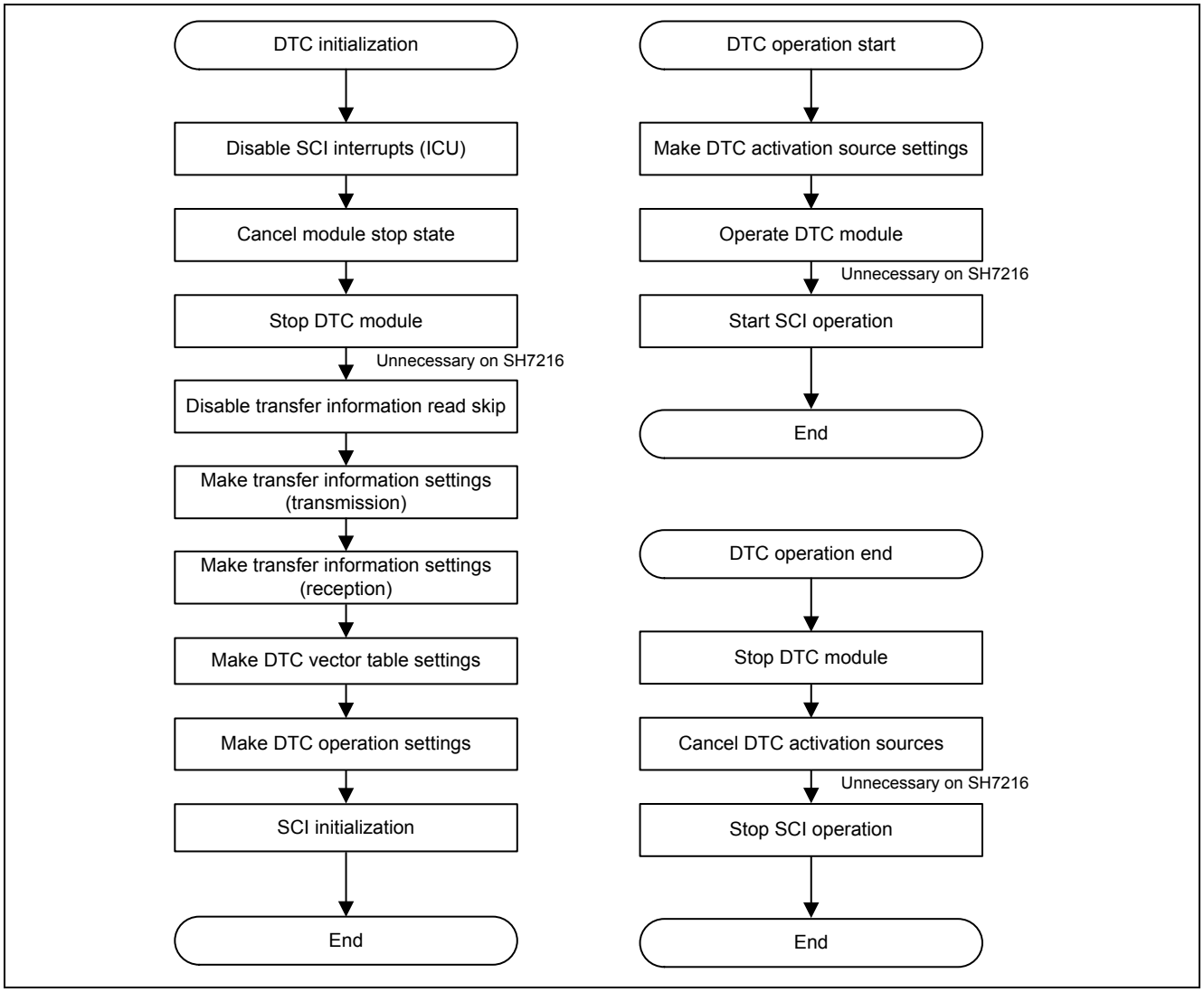


Figure 2.16 Example Flowcharts of DTC Processing

Setting Examples

Setting examples for data transfer between the SCI and the on-chip RAM are presented below. The names of the processing steps in these setting examples correspond to the names in the flowcharts. Refer to the flowcharts for the processing procedure.

The setting examples presented below do not cover the SCI, so also refer to 2.10.6, Asynchronous Transmit/Receive Setting Examples (Interrupt/Polling).

The setting examples use the following structures for DTC transfer information:

- DTC_TX: Transmit transfer information
- DTC_RX: Receive transfer information

The setting examples allocate addresses to arrays for the DTC vector table as follows:

- SH7216:

```
#pragma address DTC_VECT_TABLE=0xFFF80400 (user-defined address)
volatile unsigned long DTC_VECT_TABLE[256];
```

- RX71M:

```
#pragma address DTC_VECT_TABLE=0x0 (user-defined address)
volatile unsigned long DTC_VECT_TABLE[256];
```

When setting SCI2 interrupts as DTC activation sources, iodefine.h can be used to make settings as follows:

- DTCE (SCI2 or ICUA interrupt name)

Table 2.27 DTC Initialization Setting Examples (Data Transfer between SCI and On-Chip RAM)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Cancel module stop state.	STB.CR2._DTC = 0b	SYSTEM.PRCR = A502h SYSTEM.MSTPCRA.MSTPA28 = 0b SYSTEM.PRCR = A500h
Stop DTC module.	—	DTC.DTCST.DTCST = 0b
Disable transfer information read skip.	DTC.DTCCR.RRS = 0b	DTC.DTCCR.RRS = 0b
Make transfer information settings (transmission).	DTC_TX.MRA.MD = 00b DTC_TX.MRA.Sz = 00b DTC_TX.MRA.SM = 10b DTC_TX.MRB.CHNE = 0b DTC_TX.MRB.DISEL = 0b DTC_TX.MRB.DM = 00b DTC_TX.SAR = RAM area start address DTC_TX.DAR = SCI2.SCTDR register address DTC_TX.CRA = 32	DTC_TX.MRA.MD = 00b DTC_TX.MRA.SZ = 00b DTC_TX.MRA.SM = 10b DTC_TX.MRB.CHNE = 0b DTC_TX.MRB.DISEL = 0b DTC_TX.MRB.DM = 00b DTC_TX.SAR = RAM area start address DTC_TX.DAR = SCI2.TDR register address DTC_TX.CRA = 32
Make transfer information settings (reception).	DTC_RX.MRA.MD = 00b DTC_RX.MRA.Sz = 00b DTC_RX.MRA.SM = 00b DTC_RX.MRB.CHNE = 0b DTC_RX.MRB.DISEL = 0b DTC_RX.MRB.DM = 10b DTC_RX.SAR = SCI2.SCRDR register address DTC_RX.DAR = RAM area start address DTC_RX.CRA = 32	DTC_RX.MRA.MD = 00b DTC_RX.MRA.SZ = 00b DTC_RX.MRA.SM = 00b DTC_RX.MRB.CHNE = 0b DTC_RX.MRB.DISEL = 0b DTC_RX.MRB.DM = 10b DTC_RX.SAR = SCI2.RDR register address DTC_RX.DAR = RAM area start address DTC_RX.CRA = 32
Make DTC vector table settings.	DTC_VECT_TABLE[249] = DTC_RX address DTC_VECT_TABLE[250] = DTC_TX address DTC.DTCVBR = DTC_VECT_TABLE-400h (DTC_VECT_TABLE: array address)	DTC_VECT_TABLE[62] = DTC_RX address DTC_VECT_TABLE[63] = DTC_TX address DTC.DTCVBR = DTC_VECT_TABLE (DTC_VECT_TABLE: array address)
Make DTC operation settings.	Transfer information read skip setting DTC.DTCCR.RRS = 1b Address mode setting BSC.BSCEHR.DTSA = 0b	Transfer information read skip setting DTC.DTCCR.RRS = 1b Address mode setting DTC.DTCADMOD.SHORT = 0b

Table 2.28 DTC Operation Start Setting Examples

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make DTC activation source settings.	DTC.DTCERE.RXI2 = 1b DTC.DTCERE.TXI2 = 1b	DTCE(SCI2, RXI2) = 1b DTCE(SCI2, TXI2) = 1b
Operate DTC module.	—	DTC.DTCST.DTCST = 1b

Table 2.29 DTC Operation End Setting Examples

Processing	SH7216 Group Setting Example	RX71M Setting Example
Stop DTC module.	—	DTC.DTCST.DTCST = 0b
Clear DTC activation sources.	DTC.DTCERE.RXI2 = 0b DTC.DTCERE.TXI2 = 0b	DTCE(SCI2, RXI2) = 0b DTCE(SCI2, TXI2) = 0b

2.6 Direct Memory Access Controller (DMAC)

2.6.1 Comparison of Specifications

Direct memory access control functionality is implemented on the SH7216 Group by an on-chip DMAC and on the RX71M by an on-chip DMACAa and by a dedicated on-chip EXDMACa for transfers between external areas.

The internal bus configuration of the RX71M differs from that of the SH7216 Group. It supports independent data transfers by CPU instruction execution and by the DMAC or DTC for improved transfer performance. Table 2.30 is a comparative listing of the specifications of the SH7216 Group and RX71M.

Table 2.30 Comparison of SH7216 Group and RX71M Specifications (DMAC)

Item		SH7216 Group	RX71M	
		DMAC	DMACAa	EXDMACa
Number of channels		8 channels	8 channels	2 channels
Maximum transfer count (maximum transfer data unit count on RX)		16 M (16,777,216)	64 M data units (block transfer mode max. total transfer count: 1,024 data units × 65,536 blocks) Free running is also supported.	1 M data units (block transfer mode max. total transfer count: 1,024 data units × 1,024 blocks)
Activation sources		• External request • On-chip module request • Auto request (software trigger equivalent)	(External requests not supported.) • On-chip module request • Software trigger • External interrupt	• External request • On-chip module request • Software trigger
Channel priority		Selectable between the following: • Channel 0 > channel 1 > ... > channel 7 • Channel 0 > channel 4 > ... > channel 3 > channel 7 • Round robin	Fixed (channel 0 > channel 1 > ... > channel 3)	Fixed (channel 0 > channel 1)
Transfer data	1 data unit	8 bits, 16 bits, 32 bits, 128 bits	8 bits, 16 bits, 32 bits	8 bits, 16 bits, 32 bits
	Repeat size	—	Data units: 1 to 1,024	Data units: 1 to 1,024
	Block size	—	Data units: 1 to 1,024	Data units: 1 to 1,024
	Cluster size	—	—	Data units: 1 to 8
Transfer modes		None (The transfer mode on the SH is equivalent to normal transfer mode on the RX.)	• Normal transfer mode • Repeat transfer mode • Block transfer mode	• Normal transfer mode • Repeat transfer mode • Block transfer mode • Cluster transfer mode
Bus modes		• Cycle-steal mode • Burst mode	—	—
Address modes		• Single address mode • Dual address mode	—	• Single address mode • Dual address mode
Interrupt request	Transfer-end interrupt	• When data transfer finishes • When 1/2 of data transfer finishes	• When data transfer finishes	• When data transfer finishes
	Transfer escape-end interrupt	—	Generated after completion of data transfer equivalent to the repeat size or when the extended repeat area overflows.	
Other		• Reload function • Output of transfer-end signal	• Extended repeat area • Event link • Offset address updating	• Extended repeat area • Offset address updating

2.6.2 DMAC Block Diagram

Figure 2.17 is a block diagram of the SH7216 Group's DMAC.

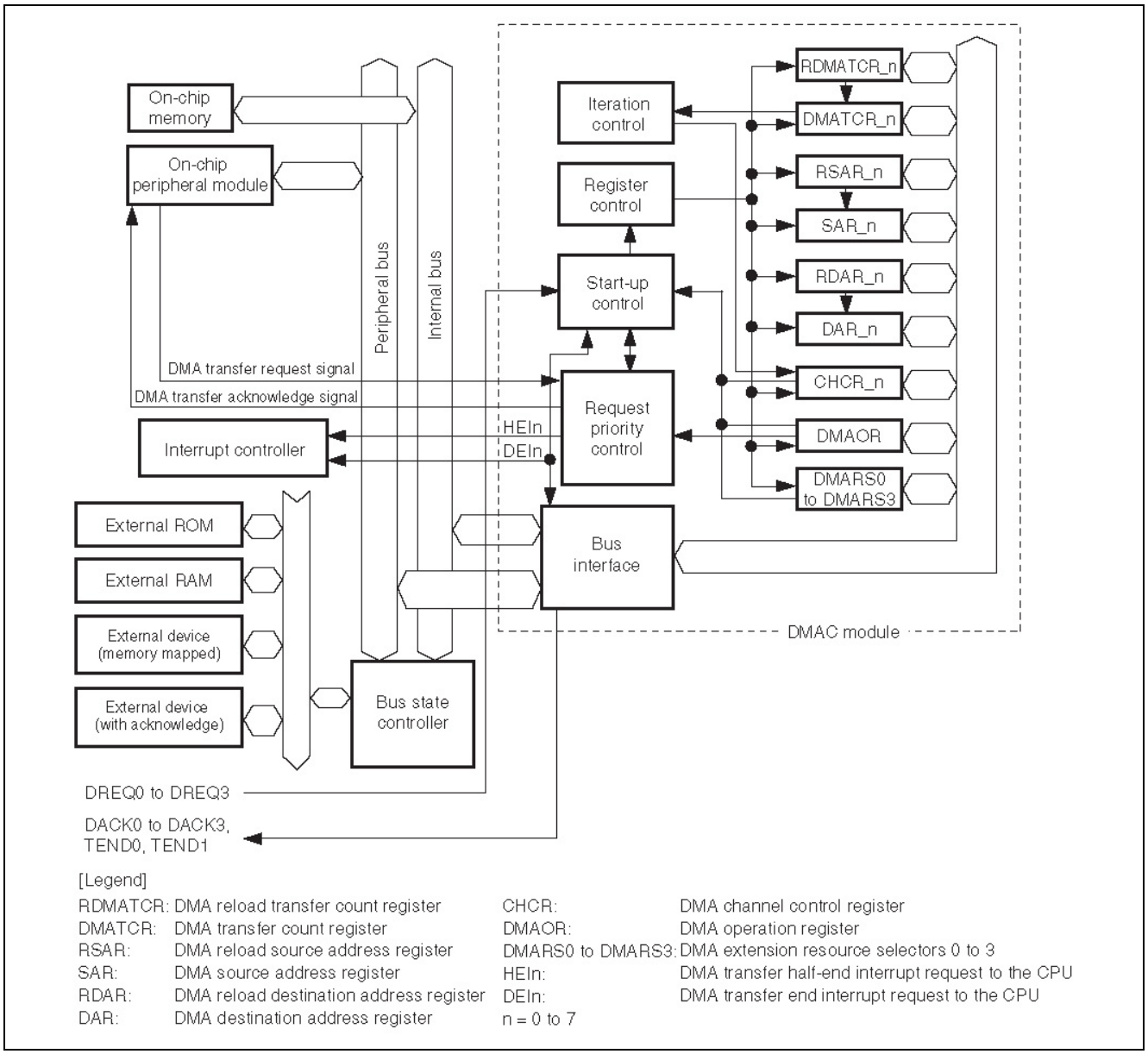


Figure 2.17 SH7216 Group DMAC Block Diagram

Figure 2.18 is a block diagram of the RX71M's DMACAa.

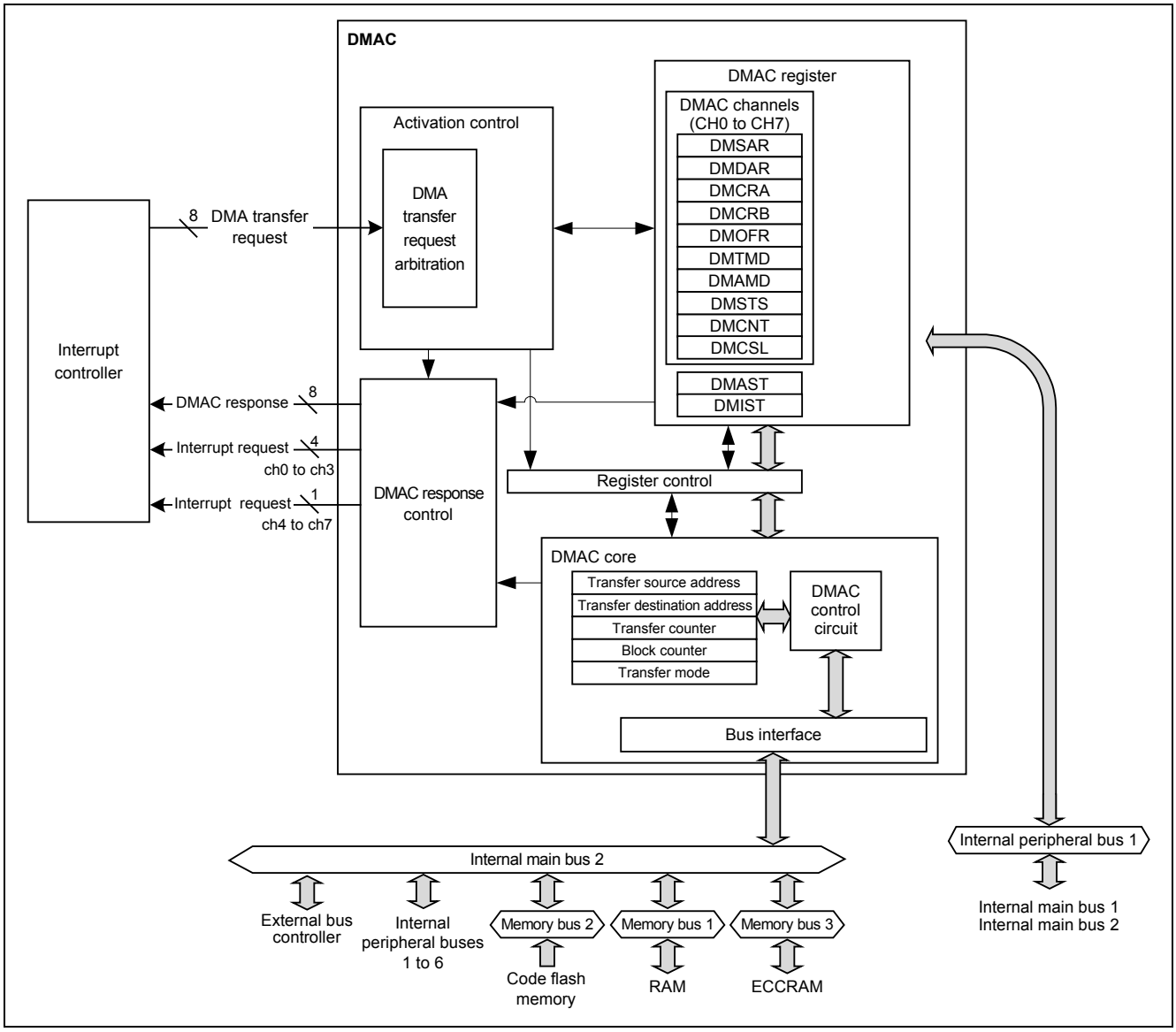
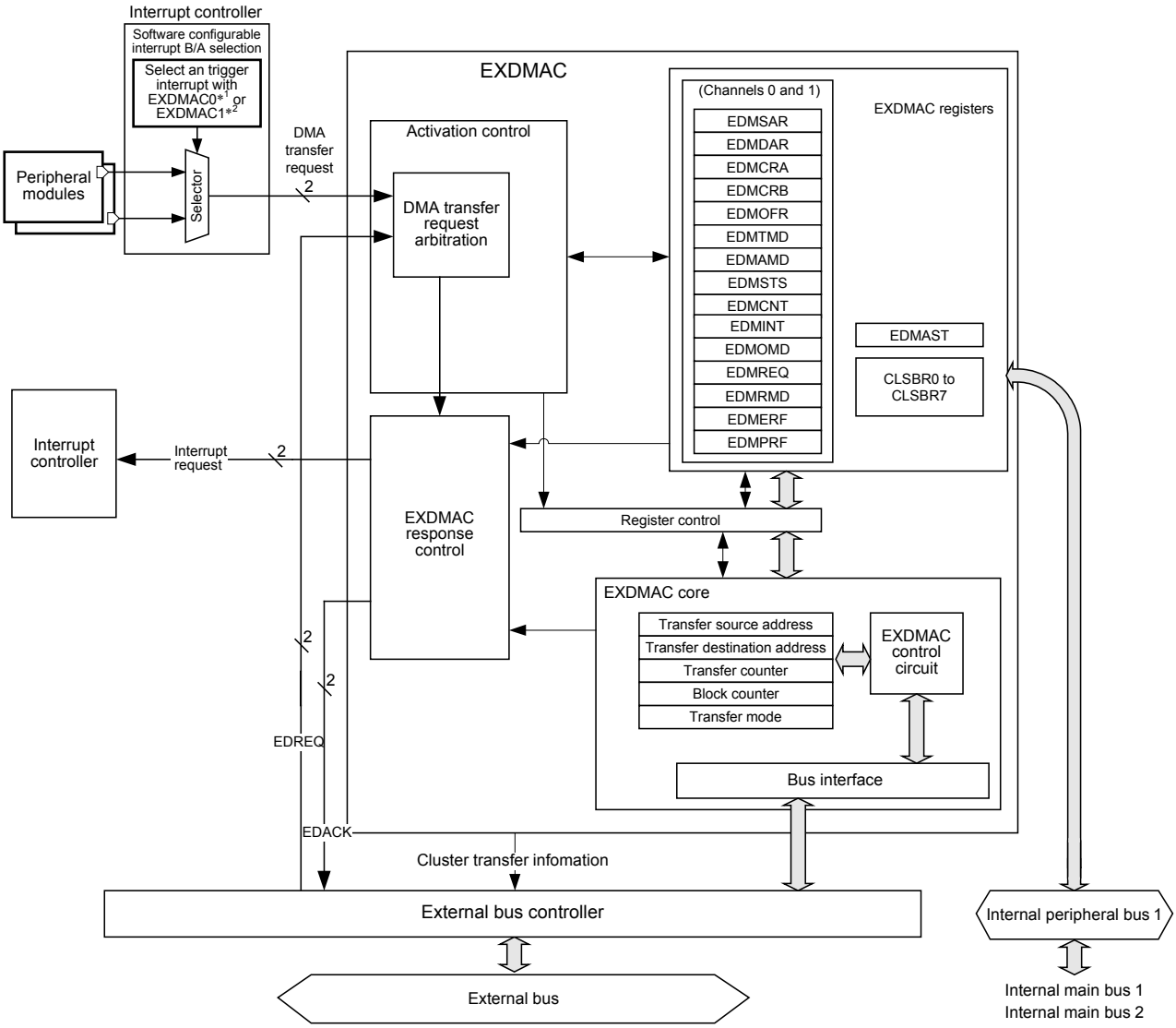


Figure 2.18 RX71M DMACAa Block Diagram

Figure 2.19 is a block diagram of the RX71M's EXDMACa.



Note 1. A software configurable interrupt B request from TPU1.TRGA selected in ICU.SLIBR144 or a software configurable interrupt A request from MTU1.TRGA selected in ICU.SLIAR208

Note 2. A software configurable interrupt B request from TPU1.TRGA selected in ICU.SLIBR145 or a software configurable interrupt A request from MTU1.TRGA selected in ICU.SLIAR209

Figure 2.19 RX71M EXDMACa Block Diagram

2.6.3 Register Comparison

Table 2.31 and Table 2.32 provide a comparative listing of the registers of the SH7216 Group and the RX71M.

Guide to Symbols in "Changes" Column of Table

◎: Register with same bit assignments on SH7216 Group and RX71M

△: Register with different bit assignments on SH7216 Group and RX71M

—: Register not present on SH7216 Group or RX71M

Table 2.31 SH7216 Group and RX71M Register Comparison (DMAC/DMACAa)

SH7216 Group (DMAC)*1	RX71M (DMACAa)*2	Changes
DMA operation register (DMAOR)	DMAC module start register (DMAST)	△
DMA source address register n (SAR_n)	DMA transfer source address register (DMACm.DMSAR)	◎
DMA destination address register n (DAR_n)	DMA transfer destination address register (DMACm.DMDAR)	◎
DMA transfer count register n (DMATCR_n)	DMA transfer count register (DMACm.DMCRA)	◎
DMA channel control register n (CHCR_n)*3	DMA transfer mode register (DMACm.DMTMD) DMA address mode register (DMACm.DMAMD) DMA interrupt setting register (DMACm.DMINT) DMA transfer enable register (DMACm.DMCNT) DMA status register (DMACm.DMSTS) DMA software start register (DMACm.DMREQ)	△
—	DMA block transfer count register (DMACm.DMCRB)	—
	DMA activation source flag control register (DMACm.DMCSL)	
	DMA offset register (DMAC0.DMOFR)	
	DMAC74 interrupt status monitor register (DMIST)	
DMA extension resource selectors 0 to 3 (DMARS0 to DMARS3) DMA reload source address register n (RSAR_n) DMA reload destination address register n (RDAR_n) DMA reload transfer count register n (RDMATCR_n)	—	—

Note 1. DMAC n: 0 to 7

Note 2. DMACAa m: 0 to 7

Note 3. On the RX71M transfer request settings from peripheral modules are made by means of the interrupt controller.

Table 2.32 SH7216 Group and RX71M Register Comparison (DMAC/EXDMACa)

SH7216 Group (DMAC)*1	RX71M (EXDMACa)*2	Changes
DMA operation register (DMAOR)	EXDMA module start register (EDMAST)	△
DMA source address register n (SAR_n)	EXDMA transfer source address register (EXDMACm.EDMSAR)	◎
DMA destination address register n (DAR_n)	EXDMA transfer destination address register (EXDMACm.EDMDAR)	◎
DMA transfer count register n (DMATCR_n)	EXDMA transfer count register (EXDMACm.EDMCRA)	◎
DMA channel control register n (CHCR_n)*3	EXDMA transfer mode register (EXDMACm.EDMTMD) EXDMA address mode register (EXDMACm.EDMAMD) EXDMA interrupt setting register (EXDMACm.EDMINT) EXDMA transfer enable register (EXDMACm.EDMCNT) EXDMA external request sense mode register (EXDMACm.EDMRMD) EXDMA output setting register (EXDMACm.EDMOMD) EXDMA status register (EXDMACm.EDMSTS) EXDMA software start register (EXDMACm.EDMREQ)	△
—	EXDMA block transfer count register (EXDMACm.EDMCRB)	—
	EXDMA offset register (EXDMAC0.EDMOFR)	
	EXDMA external request flag register (EXDMACm.EDMERF)	
	EXDMA peripheral request flag register (EXDMACm.EDMPRF)	
	Cluster buffer register y (CLSBRy) (y = 0 to 7)	
DMA extension resource selectors 0 to 3 (DMARS0 to DMARS3) DMA reload source address register n (RSAR_n) DMA reload destination address register n (RDAR_n) DMA reload transfer count register n (RDMATCR_n)	—	—

Note 1. DMAC n: 0 to 7

Note 2. DMACAa m: 0 or 1

Note 3. On the RX71M transfer request settings from peripheral modules are made by means of the interrupt controller.

2.6.4 Activation Source Settings

On the SH7216 Group activation sources that enable peripheral modules to activate the DMA are specified by setting the resource select bits in the DMA channel control registers (RS[3:0] in CHCR_0 to CHCR_7) and making appropriate settings in the DMA extension resource selectors (DMARSm). On the RX71M DMA activation sources are specified by setting activation source vector numbers in the DMAC trigger select registers (DMRSRm) of the interrupt controller, thereby enabling DMA activation by the corresponding interrupts.

Table 2.33 lists the types of DMA activation sources.

Table 2.33 DMA Activation Source Comparison

DMA Activation Sources	SH7216 Group	RX71M	
	DMAC	DMACAa	EXDMACa
Activation by software	Supported	Supported	Supported
Activation by external device via request pin	Supported (DREQn pin) Rising edge Falling edge Low level High level	Not supported	Supported (DREQm pin) Rising edge Falling edge Low level
Activation by interrupt from external interrupt input pin	Not supported	supported (IRQ pin)	Not supported
Activation by peripheral module	Supported (MTU, ADC, SCIF, IIC, CMT, USB, RSPI, CAN)	Supported (CMT, CMTW, USB, RSPI, QSPI, SDHI, MMCIF, SSI, SRC, RIIC, SCI, PDC, SCIF, MTU, GPT, EPTPC, AES, TPU, ADC, SHA, DES, RNG, ELC)	Supported (TPU, MTU)

n, m: Number of DMA channels (n = 0 to 3, m = 0 or 1)

2.6.5 Transfer Count

The RX71M supports free running operation, in which transfer count is not specified. Table 2.34 lists transfer count settings in normal transfer mode on the SH7216 Group and RX71M.

Table 2.34 Transfer Count Setting Values

Transfer count	SH7216 Group	RX71M (DMACAa, EXDMACa)
1	00000001h	0001h
65,535	FFFFh	FFFFh (max. transfer count)
16,777,215	00FFFFFFh	—
16,777,216	00000000h (max. transfer count)	—
Free running (no transfer count specified)	—	0000h

2.6.6 Transfer Sources and Destinations

Table 2.35 to Table 2.37 list the transfer sources and destinations supported by each DMA controller.

Table 2.35 SH7216 Group DMAC Transfer Sources and Destinations

	Transfer Destination				
Transfer Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Memory**	**On-Chip Peripheral Module**
External Device with DACK	—	●○	●○	—	—
External Memory	●○	○	○	○	○
Memory-Mapped External Device	●○	○	○	○	○
On-Chip Peripheral Module	—	○	○	○	○
On-Chip Memory	—	○	○	○	○

●: Single address mode transfers supported. ○: Dual address mode transfers supported.
—: Transfer not supported

Table 2.36 RX71M DMACAa Transfer Sources and Destinations

	Transfer Destination				
Transfer Source	**External Device with DACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Memory**	**On-Chip Peripheral Module**
External Device with DACK	—	—	—	—	—
External Memory	—	○	○	○	○
Memory-Mapped External Device	—	○	○	○	○
On-Chip Peripheral Module	—	○	○	○	○
On-Chip Memory	—	○	○	○	○

○: Transfers supported. —: Transfer not supported

Table 2.37 RX71M EXDMACa Transfer Sources and Destinations

	Transfer Destination				
Transfer Source	**External Device with EDACK**	**External Memory**	**Memory-Mapped External Device**	**On-Chip Memory**	**On-Chip Peripheral Module**
External Device with EDACK	—	●	●	—	—
External Memory	●	○	○	—	—
Memory-Mapped External Device	●	○	○	—	—
On-Chip Peripheral Module	—	—	—	—	—
On-Chip Memory	—	—	—	—	—

●: Single address mode transfers supported. ○: Dual address mode transfers supported.
—: Transfer not supported

2.6.7 Address Modes

The SH7216 Group has two address modes: single address mode and dual address mode.

The EXDMACa of the RX71M has a single address mode and a dual address mode like the SH7216 Group. In single address mode a DMA transfer can be completed in a single bus cycle. Two bus cycles are required to complete a DMA transfer in dual address mode. On the DMACAa the address mode concept does not apply, but the method of specifying addresses and the operation are equivalent to dual address mode on the SH7216 Group.

2.6.8 Bus Modes

On the SH7216 Group the bus mode can be specified as either cycle-steal mode or burst mode. In cycle-steal mode the bus is released to another bus master when a single transfer finishes. In burst mode the bus is not released after the start of a DMA transfer until the transfer finishes.

On the RX71M it is not possible to specify the bus mode of the DMACAa or EXDMACa. This is because the bus architecture differs from that of the SH7216 Group. The RX71M supports parallel operation when the bus master accesses a different slave. On the RX71M it is possible for the DMAC to perform transfers between the peripheral bus and the external bus while the CPU is accessing the ROM to fetch CPU instructions or the RAM to manipulate operands.

Figure 2.20 shows an example in which the DMAC accesses the peripheral bus and the external bus using internal main bus 2 while the CPU is accessing the code flash memory and RAM.

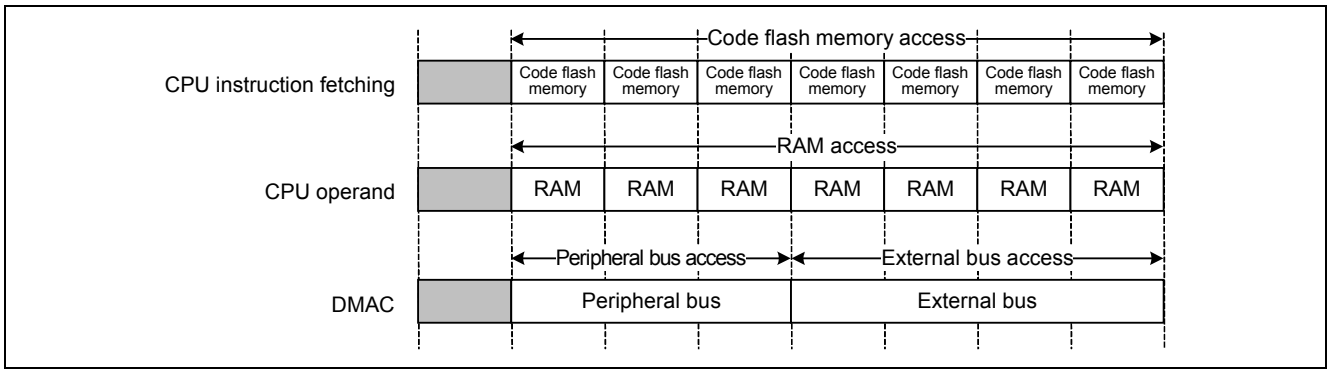


Figure 2.20 RX71M Parallel Bus Operation

2.6.9 Module Stop

On the RX71M the DMACAa and EXDMACa module-stop state is canceled after a reset.

The module-stop setting bit (MSTPCRA.MSTPA28) is common to both the DTCa and DMACAa on the RX71M, so module-stop control for these two modules is simultaneous. The EXDMACa has an independent module-stop setting bit (MSTPCRA.MSTPA29), allowing it to be controlled individually.

Refer to 2.21, Low Power Consumption Function for information on the module-stop state.

2.6.10 Setting Example for Data Transfer between SCIF and On-Chip RAM

Setting examples for data transfer between the SCIF and the on-chip RAM using the direct memory access controller on the SH7216 Group and RX71M are presented below. Additionally, refer to the SCIF initial setting examples in 2.11.7, Clock-Synchronous Slave Reception Setting Examples. The examples presented below only show settings for activating the DMAC by means of SCIF interrupts.

Operational Overview

- The SCIF is used to perform clock-synchronous slave reception.
- The DMAC is activated by a receive data-full interrupt request from the SCIF and transfers data from a SCIF register to the on-chip RAM.
- Reception of all the data takes place without intervention by software.
- When reception of all the data finishes (at DMAC transfer end), the DMAC generates a transfer-end interrupt.
- After reception of all the data finishes, SCIF and DMAC operation end.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by "SH7216:" in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.38 Operation Specifications of Data Transfer between SCIF and On-Chip RAM

Item		Description	Remarks
SCIF specifications		See Table 2.98.	SCIF clock-synchronous slave reception (RX71M: SCIFA8) (SH7216: SCIF3)
DMAC channel		DMAC0	
Transfer mode		Normal transfer mode	
Transfer count		32	
Transfer size		Byte	
Transfer source		SCIF receive FIFO data register (fixed address)	
Transfer destination		On-chip RAM (incremented following transfer)	
Activation sources		SCIF receive data-full interrupt	
Interrupts		DMAC transfer-end interrupt	Generated after specified number of data transfers.
Pins used	SCK	PC5/SCK8	SH7216: PE4/SCK3
	RXD	PC6/RXD8	SH7216: PE6/RXD3
	LED1	P05/general	Lights at transfer end.
	LED2	P26/general	Lights when error detected.

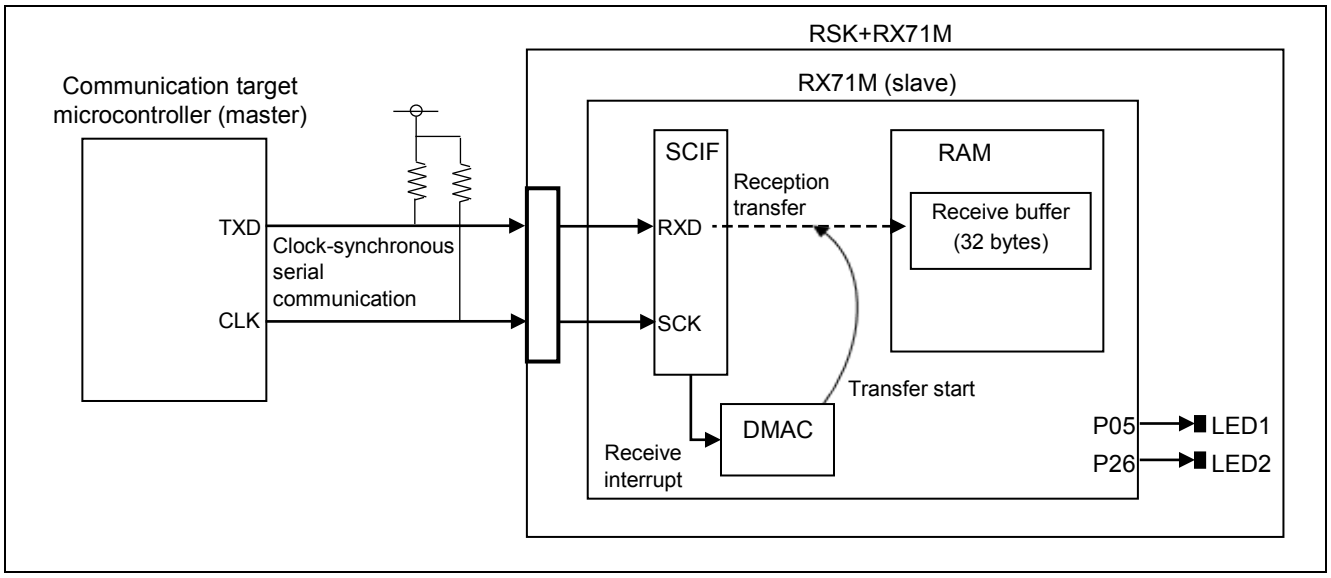


Figure 2.21 Connection Example for Data Transfer between SCIF and On-Chip RAM

Note

On the Renesas Starter Kit+ for RX71M in the initial state the pins used for SCIFA8 in the connection example are connected to the Ethernet-PHY, so make appropriate modifications to the board as necessary.

List of Related Registers

Table 2.39 shows the interrupt-related registers used in the SH7216 Group setting example broken down by source.

Table 2.39 SH7216 Group Interrupt-Related Registers (DMAC and INTC)

			DMAC		INTC
Item	**Vector No.**	**Name**	**Interrupt Enable**	**Status**	**Priority Level**
Setting register	—	—	DMAC0.CHCR	DMAC0.CHCR	IPR06
Setting position of each source (DMAC0)					
Data transfer-end interrupt	108	DEI0	IE	TE	Bits 12 to 15

Table 2.40 and Table 2.41 show the interrupt-related registers used in the RX71M setting example broken down by source.

Table 2.40 RX71M Interrupt-Related Registers (DMACAa)

Item	**Name**	**Interrupt Enable**	**Status**
Setting register	—	DMAC0.DMINT	DMAC0.DMSTS
Setting position of each source (DMAC0)			
Transfer end	—	DTIE	DTIF

Table 2.41 RX71M Interrupt-Related Registers (ICUA)

Item	**Vector No.**	**Name**	**Interrupt Enable**	**Status**	**Priority Level**
Setting register	—	—	IERm	IRr	IPRr
Setting position of each source (DMAC0)					
Transfer end	120	DMAC0I	IER0F.IEN0	IR120	IPR120

When making settings to the ICUA interrupt-related registers on the RX71M, iodefine.h can be used to make settings as follows:

- IERm: IEN (DMAC or ICUA interrupt name)
- IPRr: IPR (DMAC or ICUA interrupt name)
- IRr: IR (DMAC or ICUA interrupt name)

Processing Flowcharts

Table 2.22 shows example flowcharts of processing using the DMAC. The names of the processing steps shown in the flowcharts correspond to the names in the setting examples. For SCIF-related processing, refer to 2.11.7, Clock-Synchronous Slave Reception Setting Examples.

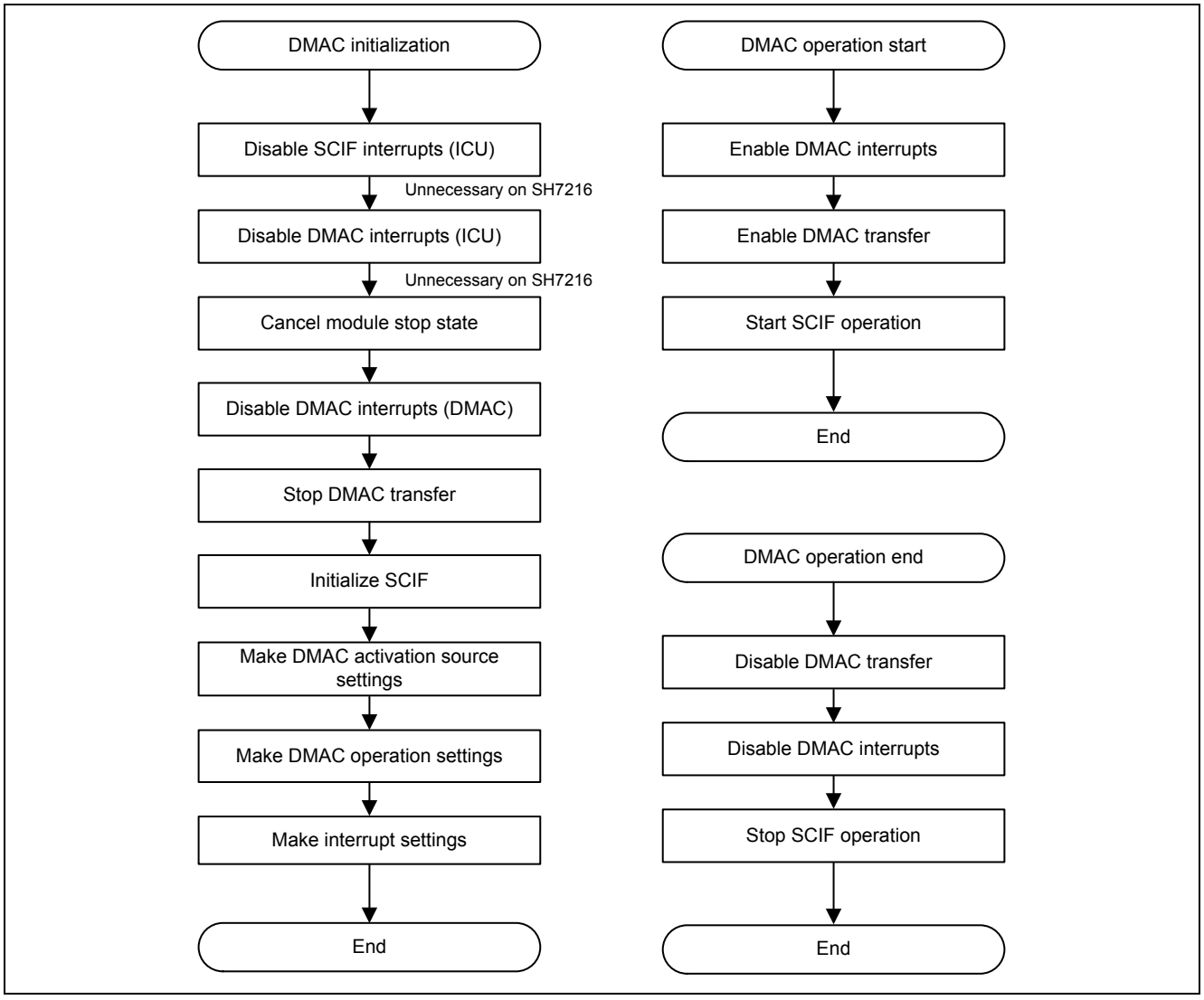


Figure 2.22 Example Flowcharts of DMAC Processing

Setting Examples

Setting examples for data transfer between the SCIF and the on-chip RAM are presented below. The names of the processing steps in these setting examples correspond to the names in the flowcharts. Refer to the flowcharts for the processing procedure.

The setting examples presented below do not cover the SCIF, so also refer to 2.11.7, Clock-Synchronous Slave Reception Setting Examples.

Table 2.42 DMAC Initialization Setting Examples (Data Transfer between SCIF and On-Chip RAM)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Disable interrupts (ICU).	—	Disabling of ICU interrupts IEN(DMAC, DMAC0I) = 0b
Cancel module stop state.	STB.CR2._DMAC = 0b	SYSTEM.PRCR = A502h SYSTEM.MSTPCRA.MSTPA28 = 0b SYSTEM.PRCR = A500h
Disable interrupts (DMAC).	DMAC0.CHCR.IE = 0b (DEI0)	DMAC0.DMINT.DTIE = 0b
Stop DMAC transfer.	DMAC.DMAOR.DME = 0b DMAC0.CHCR.DE = 0b	DMAC.DMAST.DMST = 0b DMAC0.DMCNT.DTE = 0b
Make DMAC activation source settings.	DMAC.DMARS0.C0MID = 100011b (SCIF3) DMAC.DMARS0.C0RID = 10b (RXI3)	ICU.DMRSR0 = 100 (RXIF8)
Make DMAC operation settings.	Transfer source information settings DMAC0.CHCR.SM = 00b (fixed address) DMAC0.SAR = SCIF3.SCFRDR register address Transfer destination information settings DMAC0.CHCR.DM = 01b (increment) DMAC0.DAR = RAM area start address Transfer request source setting DMAC0.CHCR.RS = 1000b (extension resource) Transfer information DMAC0.CHCR.TS = 00b (8 bits) DMAC0.DMATCR = 32 (transfer data size)	Transfer source information settings DMAC0.DMAMD.SM = 00b (fixed address) DMAC0.DMSAR = SCIFA8.FRDR register address Transfer destination information settings DMAC0.DMAMD.DM = 10b (increment) DMAC0.DMDAR = RAM area start address Transfer request source setting DMAC0.DMTMD.DCTG = 01b (peripheral) Transfer mode setting DMAC0.DMTMD.MD = 00b (normal transfer) Transfer source interrupt flag clear setting DMAC0.DMCSL.DISEL = 0b (flag cleared at transmit start) Transfer information DMAC0.DMTMD.SZ = 00b (8 bits) DMAC0.DMCRA = 32 (transfer data size)
Make interrupt settings.	Priority setting INTC.IPR06._DMAC0 = 5 Clearing of DMAC status* DMAC0.CHCR.TE = 0b * Clear to 0 after reading value as 1.	Priority setting IPR(DMAC, DMAC0I) = 5 (RXIF8) Clearing of ICU status IR(DMAC, DMAC0I) = 0b (RXIF8) No need to clear the DMAC status flag because it is cleared when transfer is enabled.

Table 2.43 DMAC Operation Start Setting Examples

Processing	SH7216 Group Setting Example	RX71M Setting Example
Enable DMAC interrupts.	Enabling of DMAC interrupts DMAC0.CHCR.IE = 1b (DEI0)	Enabling of DMAC interrupts DMAC0.DMINT.DTIE = 1b Enabling of ICU interrupts IEN(DMAC, DMAC0I) = 1b
Enable DMAC transfer.	DMAC.DMAOR.DME = 1b DMAC0.CHCR.DE = 1b	DMAC.DMAST.DMST = 1b DMAC0.DMCNT.DTE = 1b

Table 2.44 DMAC Operation End Setting Examples

Processing	SH7216 Group Setting Example	RX71M Setting Example
Disable DMAC transfer.	DMAC0.CHCR.DE = 0b DMAC.DMAOR.DME = 0b	DMAC0.DMCNT.DTE = 0b DMAC.DMAST.DMST = 0b
Disable DMAC interrupts.	Disabling of DMAC interrupts DMAC0.CHCR.IE = 0b (DEI0)	Disabling of ICU interrupts IEN(DMAC, DMAC0I) = 0b Disabling of DMAC interrupts DMAC0.DMINT.DTIE = 0b

2.7 Multi-function Timer Pulse Unit (MTU)

2.7.1 Comparison of Specifications

Multi-function timer pulse unit functionality is provided on the SH7216 Group by the MTU2 and MTU2S and on the RX71M by the MTU3a.

The RX71M includes the MTU functionality of the SH7216 Group (backward compatibility). Table 2.45 lists comparative specifications of the SH7216 Group and RX71M.

Table 2.45 Comparison of SH7216 Group and RX71M Specifications (MTU)

		SH7216 Group		RX71M
Item		**MTU2**	**MTU2S**	**MTU3a**
Functional compatibility by channel	16-bit timer	MTU0	—	MTU0
		MTU1	—	MTU1
		MTU2	—	MTU2
		MTU3	MTU3S	MTU3, MTU6
		MTU4	MTU4S	MTU4, MTU7
		MTU5	MTU5S	MTU5
	32-bit timer	—	—	MTU8
Pulse I/O		Max. 16	Max. 8	Max. 28
Pulse input		3	3	3
Count clock		Selectable for each channel among eight clocks employing the peripheral clock (Pϕ) and external clocks (TCLKA, TCLKB, TCLKC, and TCLKD)	Selectable for each channel among six clocks employing the MTU2S clock (Mϕ).	Selectable for each channel among 14 clocks employing the peripheral module clock (PCLKA) and external clocks (MTCLKA, MTCLKB, MTCLKC, MTCLKD, and MTIOC1A).
DTC/DMAC activation		DTC/DMAC activation supported	DTC activation supported	DTC/DMAC activation supported
A/D conversion start triggers		Trigger generation supported	Trigger generation supported	Trigger generation supported
Interrupt sources		28	13	43
Noise cancellation		None	None	Ability to enable noise filtering for external clock pins
Other		• Cascade connection	—	• Event link • Cascade connection

2.7.2 Interrupts

The RX71M does not have timer status register (TSR) interrupt flags, but equivalent processing can be accomplished by using the corresponding MTU interrupt request registers of the interrupt controller.

The MTU2S of the SH7216 Group can activate the DTC only, but the RX71M can activate the DTC and DMAC on all channels.

The RX71M is provided with software configurable interrupt A. The interrupt controller's software configurable interrupt A status flags (PIARk.PIRn) are not cleared automatically, but even if left uncleared they do not affect the generation of interrupt requests.

Refer to 1.9, Interrupt Handling for information about interrupts.

Table 2.46 List of MTU Interrupt Sources on SH7216 Group and RX71M

	SH7216 Group							
	MTU0	**MTU1**	**MTU2**	**—**	**MTU3 MTU3S**	**MTU4 MTU4S**	**MTU5 MTU5S**	**—**
	RX71M							
Item	**MTU0**	**MTU1**	**MTU2**	**MTU1 MTU2*2**	**MTU3 MTU6**	**MTU4 MTU7**	**MTU5**	**MTU8**
Compare match nA*3	○	○	○	—	○	○	—	○
Input capture nA*3	○	○	○	○	○	○	—	○
Compare match nB*3	○	○	○	—	○	○	—	○
Input capture nB*3	○	○	○	○	○	○	—	○
Compare match nC*3	○	—	—	—	○	○	—	○
Input capture nC*3	○	—	—	—	○	○	—	○
Compare match nD*3	○	—	—	—	○	○	—	○
Input capture nD*3	○	—	—	—	○	○	—	○
Overflow	○	○	○	○	○	○	—	○
Underflow	—	○	○	○	—	○*1	—	—
Compare match nE	○	—	—	—	—	—	—	—
Compare match nF	○	—	—	—	—	—	—	—
Compare match nU*3	—	—	—	—	—	—	○	—
Input capture nU*3	—	—	—	—	—	—	○	—
Compare match nV*3	—	—	—	—	—	—	○	—
Input capture nV*3	—	—	—	—	—	—	○	—
Compare match nW*3	—	—	—	—	—	—	○	—
Input capture nW*3	—	—	—	—	—	—	○	—

n: Channel number ○: Supported —: Not supported

Note 1. Complementary PWM mode only

Note 2. 32-bit access

Note 3. "S" is appended at the end to indicate MTU2S.

2.7.3 Register Comparison

Table 2.47 is a comparative listing of the registers on the SH7216 Group and RX71M.

Guide to Symbols in "Changes" Column of Table

◎: Register with same bit assignments on SH7216 Group and RX71M

△: Register with different bit assignments on SH7216 Group and RX71M

—: Register not present on SH7216 Group or RX71M

Table 2.47 SH7216 Group and RX71M Register Comparison (MTU)

Register Name*1	SH7216 Group (MTU2, MTU2S)	RX71M (MTU3a)	Changes
Timer control register	TCR_0 to TCR_4 TCRU/V/W_5 TCR_3/4S	MTU0.TCR to MTU4.TCR MTU5.TCRU/V/W MTU6/7.TCR	◎
	TCRU/V/W_5S	—	—
	—	MTU8.TCR	—
Timer control register 2	—	MTU0.TCR2 to MTU4.TCR2 MTU6.TCR2 to MTU8.TCR2 MTU5.TCR2U/V/W	—
Timer mode register (SH7216 Group) Timer mode register 1 (RX71M)	TMDR_0/3/4 TMDR_3/4S	MTU0/3/4.TMDR1 MTU6/7.TMDR1	◎
	TMDR_1/2	MTU1/2.TMDR1	△
	—	MTU8.TMDR1	—
Timer mode register 2	—	MTU.TMDR2A/B	—
Timer mode register 3	—	MTU1.TMDR3	—
Timer I/O control register	TIORH_0 TIOR_1	MTU0.TIORH MTU1.TIOR	△
	TIORL_0 TIOR_2 TIORU/V/W_5 TIORH/L_3/4 TIORH/L_3/4S	MTU0.TIORL MTU2.TIOR MTU5.TIORU/V/W MTU3/4.TIORH/L MTU6/7.TIORH/L	◎
	TIORU/V/W_5S	—	—
	—	MTU8.TIORH/L	—
Timer compare match clear register	TCNTCMPCLR	MTU5.TCNTCMPCLR	◎
	TCNTCMPCLRS	—	—
Timer interrupt enable register	TIER_0 to TIER_5 TIER_3/4S TIER2_0	MTU0.TIER to MTU5.TIER MTU6/7.TIER MTU0.TIER2	◎
	TIER_5S	—	—
	—	MTU8.TIER	—

Register Name*1	SH7216 Group (MTU2, MTU2S)	RX71M (MTU3a)	Changes
Timer status register	TSR_1 to TSR_4 TSR_3/4S	MTU1.TSR to MTU4.TSR MTU6/7.TSR	△
	TSR_0 TSR2_0 TSR_5/5S	—	—
Timer buffer operation transfer mode register	TBTM_0/3/4 TBTM_3/4S	MTU0/3/4.TBTM MTU6/7.TBTM	◎
Timer input capture control register	TICCR	MTU1.TICCR	◎
Timer A/D conversion start request control register	TADCR, TADCRS	MTU4.TADCR, MTU7.TADCR	◎
Timer A/D conversion start request cycle set register	TADCORA/B_4 TADCORA/B_4S	MTU4.TADCORA/B MTU7.TADCORA/B	◎
Timer A/D conversion start request cycle set buffer register	TADCOBRA/B_4 TADCOBRA/B_4S	MTU4.TADCOBRA/B MTU7.TADCOBRA/B	◎
Timer counter	TCNT_0 to TCNT_4 TCNTU/V/W_5 TCNT_3/4S	MTU0.TCNT to MTU4.TCNT MTU5.TCNTU/V/W MTU6/7.TCNT	◎
	TCNTU/V/W_5S	—	—
	—	MTU8.TCNT	—
Timer longword counter	—	MTU1.TCNTLW	—
Timer general register	TGR_0 (A to F) TGR_1/2 (A, B) TGR_3/4 (A to D) TGR_5 (U, V, W) TGR_3/4S (A to D)	MTU0.TGR (A to F) MTU1/2.TGR (A, B) MTU3/4.TGR (A to D) MTU5.TGR (U, V, W) MTU6/7.TGR (A to D)	◎
	TGR_5S (U, V, W)	—	—
	—	MTU3/6.TGR (E) MTU4/7.TGR (E, F) MTU8.TGR (A to D)	—
Timer longword general register	—	MTU1.TGRA/BLW	—
Timer start register	TSTR	MTU.TSTRA	◎
	TSTRS, TSTR_5	MTU.TSTRB, MTU5.TSTR	◎
	TSTR_5S	—	—
Timer synchronous register	TSYR, TSYRS	MTU.TSYRA, MTU.TSYRB	◎
Timer synchronous clear register	TSYCRS	MTU6.TSYCR	◎
Timer counter synchronous start register	TCSYSTR	MTU.TCSYSTR	◎
Timer read/write enable register	TRWER, TRWERS	MTU.TRWERA, MTU.TRWERB	◎
Timer output master enable register	TOER, TOERS	MTU.TOERA, MTU.TOERB	◎
Timer output control register 1	TOCR1, TOCR1S	MTU.TOCR1A, MTU.TOCR1B	◎
Timer output control register 2	TOCR2, TOCR2S	MTU.TOCR2A, MTU.TOCR2B	◎
Timer output level buffer register	TOLBR, TOLBRS	MTU.TOLBRA, MTU.TOLBRB	◎

Register Name*1	SH7216 Group (MTU2, MTU2S)	RX71M (MTU3a)	Changes
Timer gate control register (SH7216 Group)	TGCR	MTU.TGCRA	◎
Timer gate control register A (RX71M)	TGCRS	—	—
Timer sub counter	TCNTS, TCNTSS	MTU.TCNTSA, MTU.TCNTSB	◎
Timer period data register	TCDR, TCDRS	MTU.TCDRA, MTU.TCDRB	◎
Timer period buffer register	TCBR, TCBRS	MTU.TCBRA, MTU.TCBRB	◎
Timer dead time data register	TDDR, TDDRS	MTU.TDDRA, MTU.TDDRB	◎
Timer dead time enable register	TDER, TDERS	MTU.TDERA, MTU.TDERB	◎
Timer buffer transfer set register	TBTER, TBTERS	MTU.TBTERA, MTU.TBTERB	◎
Timer waveform control register	TWCR, TWCRS	MTU.TWCRA, MTU.TWCRB	◎
Timer interrupt skipping set register (SH7216 Group) Timer interrupt skipping set register 1 (RX71M)	TITCR, TITCRS	MTU.TITCR1A, MTU.TITCR1B	◎
Timer interrupt skipping set register 2	—	MTU.TITCR2A, MTU.TITCR2B	—
Timer interrupt skipping counter (SH7216 Group) Timer interrupt skipping counter 1 (RX71M)	TITCNT, TITCNTS	MTU.TITCNT1A, MTU.TITCNT1B	◎
Timer interrupt skipping counter 2	—	MTU.TITCNT2A, MTU.TITCNT2B	—
Timer interrupt skipping mode register	—	MTU.TITMRA, MTU.TITMRB	—
Noise filter control register n	—	NFCR0 to NFCR4 in MTU0 to MTU4 NFCR6 to NFCR8 in MTU6 to MTU8 MTU0.NFCRC	—
Noise filter control register 5	—	MTU5.NFCR5	—

Note 1. On the SH7216 Group MTU2S register names have S appended at the end.

2.7.4 Module Stop

As on the SH7216 Group, the MTU3a of the RX71M is set to the module-stop state after a reset, and no clock is supplied.

Refer to 2.21, Low Power Consumption Function for information on the module-stop state.

2.7.5 Compare Match Pulse Output Setting Examples

Setting examples for the use the multifunction timer pulse unit to generate pulse output with the compare match function on the SH7216 Group and RX71M are presented below.

Operational Overview

- Toggle output on a pin is produced using the compare match function.
- Pulse output continues with a duty ratio of 50% without intervention by software.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by "SH7216:" in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.48 Compare Match Pulse Output Operation Specifications

Item		Description	Remarks
MTU channel		MTU4	
Count clock		Count at rising edge of PCLKA/1 (PCLKA = 120 MHz)	Count at rising edge of SH7216: Pϕ/1 (Pϕ = 50 MHz)
Operating mode		Normal mode	
Counter clear source		TGRB compare match	
Timer general register		Used as output compare register.	
Compare match value		0xEA5F (0.5 ms @120 MHz)	SH7216: 0x61A8 (0.5 ms @50 MHz)
MTU pin operation		Initial output is low output Output toggled at TGRB compare match	
Noise cancellation		Not used.	
Interrupts		Not used.	
Pins used	MTIOC	PC2/MTIOC4B	SH7216: PE13/TIOC4B
	LED0	P03/general	Lights when pulse output starts.

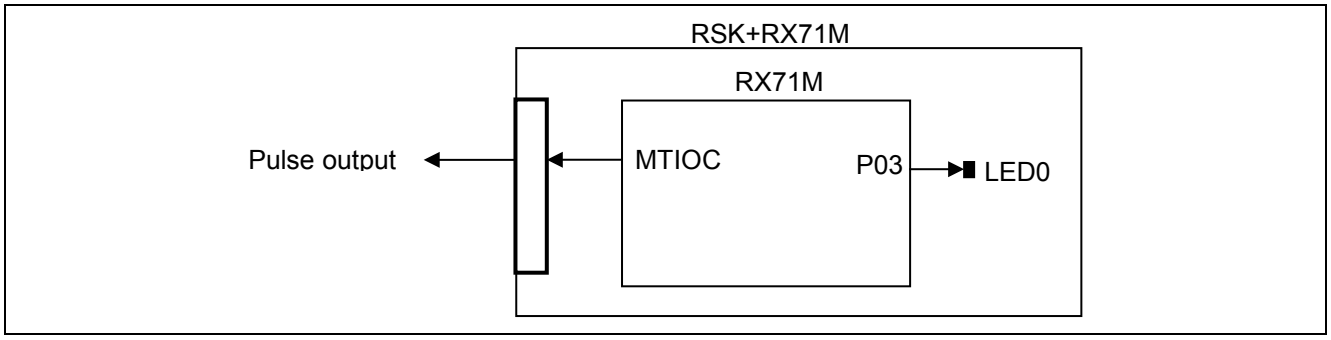


Figure 2.23 Compare Match Pulse Output Connection Example

Note

On the Renesas Starter Kit+ for RX71M in the initial state the pins used for MTU4 in the connection example are connected to the Ethernet-PHY, so make appropriate modifications to the board as necessary.

Processing Flowcharts

Figure 2.24 shows a flowchart of processing using the MTU. The interrupt-related processing is not necessary when interrupts are not used. The names of the processing steps shown in the flowcharts correspond to the names in the setting examples.

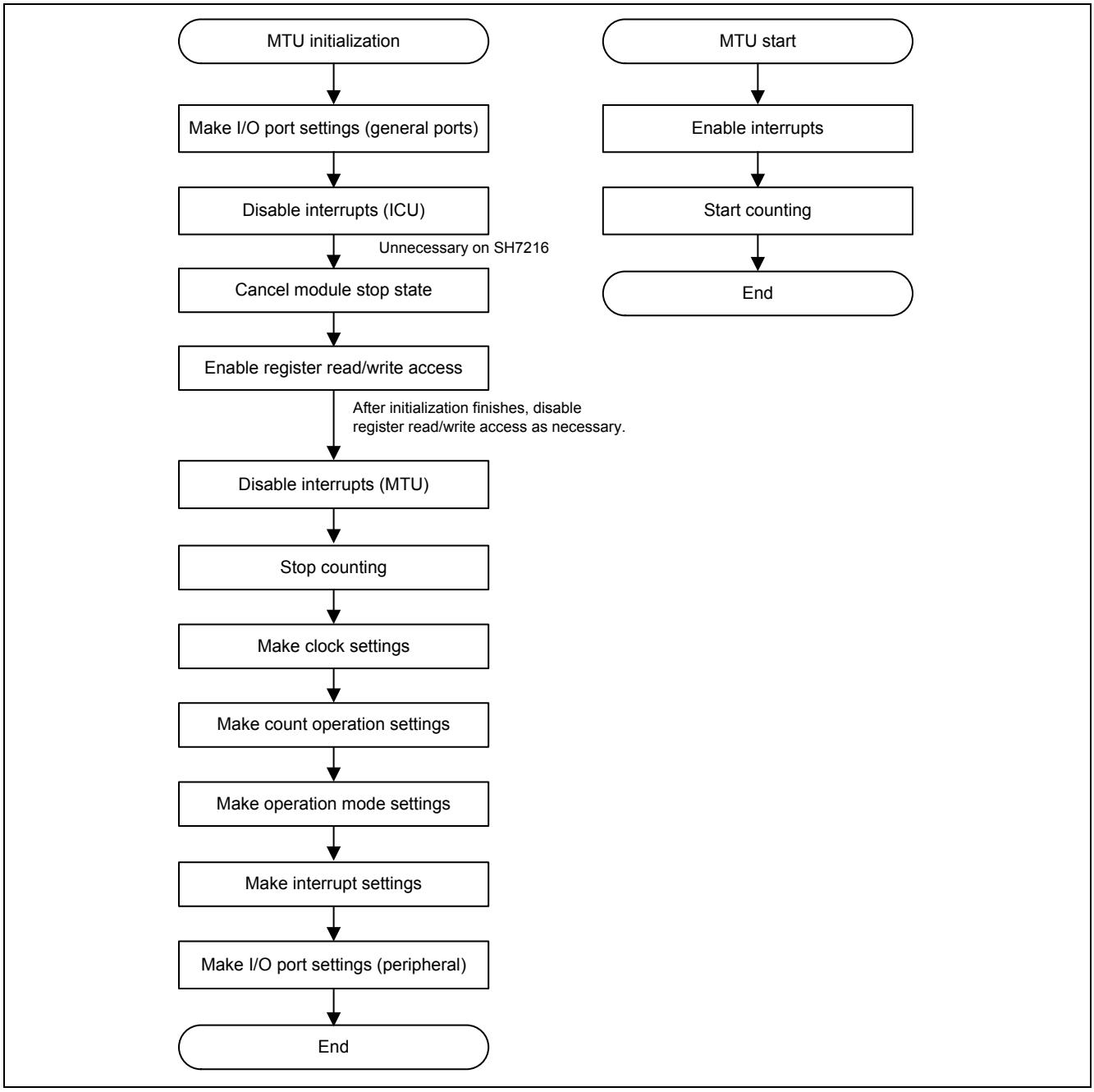


Figure 2.24 Example Flowcharts of MTU Processing

Setting Examples

Setting examples for pulse output using the compare match function are presented below. The names of the processing steps shown in the flowcharts correspond to the names in the setting examples.

Table 2.49 MTU Initialization Setting Examples (Compare Match Pulse Output)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make I/O port settings (general ports).	Output value setting (output 0) PE.DR.B13 = 0b (PE13) Pin direction setting PFC.PEIORL.B13 = 1b (output/PE13) Pin mode setting (port) PFC.PECRL4.PE13MD = 000b (PE13)	Output value setting (output 0) PORTC.PODR.B2 = 0b (PC2) Pin direction setting PORTC.PDR.B2 = 1b (output/PC2) Pin mode settings (general) PORTC.PMR.B2 = 0b (PC2)
Cancel module stop state.	STB.CR3._MTU2 = 0b	SYSTEM.PRCR = A502h SYSTEM.MSTPCRA.MSTPA9 = 0b SYSTEM.PRCR = A500h
Enable register read/write access.	MTU2.TRWER.RWE = 1b	MTU.TRWERA.RWE = 1b
Stop counting.	MTU2.TSTR.CST4 = 0b	MTU.TSTRA.CST4 = 0b
Make clock settings.	Internal clock frequency division ratio setting MTU24.TCR.TPSC = 000b Count edge setting MTU24.TCR.CKEG = 00b	Internal clock frequency division ratio settings MTU4.TCR.TPSC = 000b MTU4.TCR2.TPSC2 = 000b Count edge setting MTU4.TCR.CKEG = 00b
Make count operation settings.	Clearing the counter MTU24.TCNT = 0 Independent operation setting MTU2.TSYR.SYNC4 = 0b Counter clear source setting MTU24.TCR.CCLR = 010b Pin operation setting MTU24.TIOR.IOB = 0011b Enabling pin output MTU2.TOER.OE4B = 1b Cycle setting MTU24.TGRB = 61A8h	Clearing the counter MTU4.TCNT = 0 Independent operation setting MTU.TSYRA.SYNC4 = 0b Counter clear source setting MTU4.TCR.CCLR = 010b Pin operation setting MTU4.TIORH.IOB = 0011b Enabling pin output MTU.TOERA.OE4B = 1b Cycle setting MTU4.TGRB = EA5Fh
Make operation mode settings.	MTU24.TMDR.BFB = 0b MTU24.TMDR.MD = 0000b	MTU4.TMDR1.BFB = 0b MTU4.TMDR1.MD = 0000b
Disable register read/write access.	Reading value of MTU2.TRWER.RWE MTU2.TRWER.RWE = 0b	Reading value of MTU.TRWERA.RWE MTU.TRWERA.RWE = 0b
Make I/O port settings (peripheral).	 Pin mode setting PFC.PECRL4.PE13MD = 100b (TIOC4B)	Cancellation of register protection MPC.PWPR.B0WI = 0b MPC.PWPR.PFSWE = 1b Pin peripheral function selection MPC.PC2PFS.PSEL = 000001b (MTIOC4B) Register protection settings MPC.PWPR.PFSWE = 0b MPC.PWPR.B0WI = 1b Pin mode setting (peripheral) PORTC.PMR.B2 = 1b (MTIOC4B)

Table 2.50 MTU Compare Match Operation Start Setting Examples

Processing	SH7216 Group Setting Example	RX71M Setting Example
Start counting.	MTU2.TSTR.CST4 = 1b	MTU.TSTRA.CST4 = 1b

2.7.6 Input Capture Pulse Width Measurement Setting Examples

Setting examples for the use the multifunction timer pulse unit to measure input pulse width with the input capture function on the SH7216 Group and RX71M are presented below.

Operational Overview

- Input capture is used to measure the pulse width of input on a pin.
- An input capture interrupt is used to obtain the high-width duration and the value is stored in the RAM.
- An overflow interrupt is used to count the number of overflows, and an error is generated when an upper limit value is exceeded.
- Input capture operation ends when an error is detected.
- Operation continues until an error is detected.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by "SH7216:" in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.51 Input Capture Pulse Width Measurement Operation Specifications

Item		Description	Remarks
MTU channel		MTU4	
Count clock		Count at rising edge of PCLKA/1 (PCLKA = 120 MHz)	Count at rising edge of SH7216: Pϕ/1 (Pϕ = 50 MHz)
Operating mode		Normal mode	
Counter clear source		TGRB input capture	
Timer general register		Used as input capture register.	
Overflow upper limit value		0xFFFF times	
MTU pin operation		Input capture at both edges	
Noise cancellation		Not used.	
Interrupts		TGRB input capture TCNT overflow	Priority level 5 Priority level 6
Pins used	MTIOC	PC2/MTIOC4B	SH7216: PE13/TIOC4B
	LED0	P03/general	Lights when pulse width measurement starts.
	LED1	P05/general	Lights when error detected.

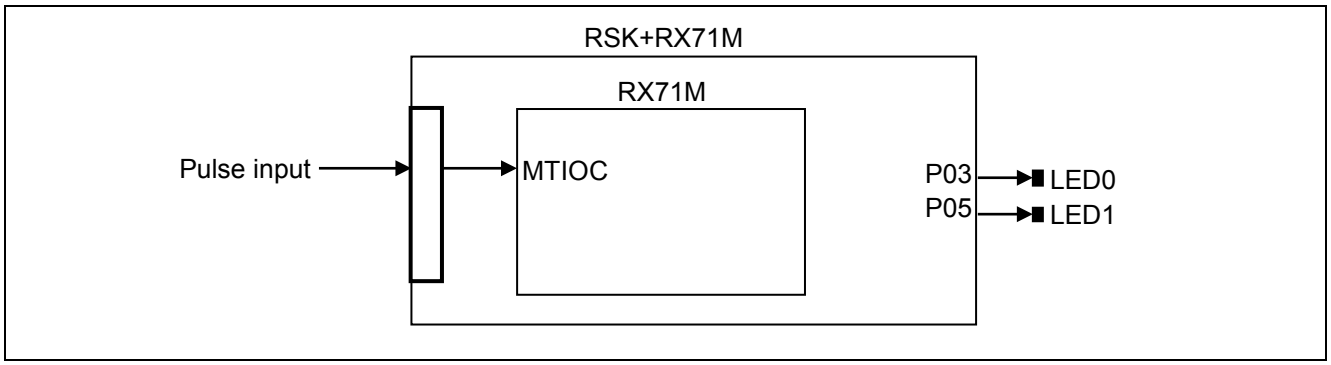


Figure 2.25 Input Capture Pulse Width Measurement Connection Example

Note

On the Renesas Starter Kit+ for RX71M in the initial state the pins used for MTU4 in the connection example are connected to the Ethernet-PHY, so make appropriate modifications to the board as necessary.

List of Related Registers

Table 2.52 shows the interrupt-related registers used in the SH7216 Group setting example broken down by source.

Table 2.52 SH7216 Group Interrupt-Related Registers (MTU2 and INTC)

			MTU2		INTC
Item	**Vector No.**	**Name**	**Interrupt Enable**	**Status**	**Priority Level**
Setting register	—	—	MTU2.TIER_4	MTU2.TSR_4	IPR11
Setting position of each source (MTU4-TGRB)					
Input capture/compare match	189	TGIB_4	TGIEB	TGFB	Bits 12 to 15
Underflow/overflow	192	TCIV_4	TCIEV	TCFV	Bits 8 to 11

Table 2.53 and Table 2.54 show the interrupt-related registers used in the RX71M setting example broken down by source. On the RX71M MTU-related interrupts are assigned to software configurable interrupt A. Software configurable interrupt A vectors 210 and 211 are used in the setting example, as follows:

- SLIAR210 = 22: Assigns TGIB4 to vector 210 of software configurable interrupt A.
- SLIAR211 = 25: Assigns TCIV4 to vector 211 of software configurable interrupt A.

Table 2.53 RX71M Interrupt-Related Registers (MTU3a)

Item	**Name**	**Interrupt Enable**	**Status**
Setting register	—	MTU4.TIER	—
Setting position of each source (MTU4-TGRB)			
Input capture/compare-match	TGIB4	TGIEB	—
Overflow/underflow	TCIV4	TCIEV	—

Table 2.54 RX71M Interrupt-Related Registers (ICUA)

Item	**Vector No.**	**Name**	**Interrupt Enable**	**Status**	**Priority Level**	**Software Configurable Interrupt Source Selection**
Setting register	—	—	IERm	IRr	IPRr	SLIARn
Setting position of each source (MTU4-TGRB)						
Input capture/compare match	210	TGIB4	IER1A.IEN2	IR210	IPR210	SLIAR210 = 22
Overflow/underflow	211	TCIV4	IER1A.IEN3	IR211	IPR211	SLIAR211 = 25

↑

Software configurable interrupt settings

When making settings to the ICUA interrupt-related registers on the RX71M, iodefine.h can be used to make settings as follows. The notation “xx” represents a vector number assigned to a software configurable interrupt.

- IERm : IEN (PERIA, INTAxx)
- IPRr : IPR (PERIA, INTAxx)
- IRr : IR (PERIA, INTAxx)

Setting Examples

Setting examples for pulse width measurement using the input capture function are presented below. The names of the processing steps in these setting examples correspond to the names in the flowcharts.

Table 2.55 MTU Initialization Setting Examples (Input Capture Pulse Width Measurement)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make I/O port settings (general ports).	Pin direction setting PFC.PEIORL.B13 = 0b (input/PE13) Pin mode setting (port) PFC.PECRL4.PE13MD = 000b (PE13)	Pin direction setting PORTC.PDR.B2 = 0b (input/PC2) Pin mode setting (general) PORTC.PMR.B2 = 0b (PC2)
Disable interrupts (ICU).	—	Disabling of ICU interrupts IEN(PERIA, INTA210) = 0b IEN(PERIA, INTA211) = 0b
Cancel module stop state.	STB.CR3._MTU2 = 0b	SYSTEM.PRCR = A502h SYSTEM.MSTPCRA.MSTPA9 = 0b SYSTEM.PRCR = A500h
Enable register read/write access.	MTU2.TRWER.RWE = 1b	MTU.TRWERA.RWE = 1b
Disable interrupts (MTU).	MTU24.TIER.TGIEB = 0b (TGIB_4) MTU24.TIER.TCIEV = 0b (TCIV_4)	MTU4.TIER.TGIEB = 0b (TGIB4) MTU4.TIER.TCIEV = 0b (TCIV4)
Stop counting.	MTU2.TSTR.CST4 = 0b	MTU.TSTRA.CST4 = 0b
Make clock settings.	Internal clock frequency division ratio setting MTU24.TCR.TPSC = 000b Count edge setting MTU24.TCR.CKEG = 00b	Internal clock frequency division ratio settings MTU4.TCR.TPSC = 000b MTU4.TCR2.TPSC2 = 000b Count edge setting MTU4.TCR.CKEG = 00b
Make count operation settings.	Clearing the counter MTU24.TCNT = 0 MTU24.TGRB = 0 Independent operation setting MTU2.TSYR.SYNC4 = 0b Counter clear source setting MTU24.TCR.CCLR = 010b Pin operation setting MTU24.TIOR.IOB = 1010b	Clearing the counter MTU4.TCNT = 0 MTU4.TGRB = 0 Independent operation setting MTU.TSYRA.SYNC4 = 0b Counter clear source setting MTU4.TCR.CCLR = 010b Pin operation setting MTU4.TIORH.IOB = 1010b
Make operation mode settings.	MTU24.TMDR.BFB = 0b MTU24.TMDR.MD = 0000b	MTU4.TMDR1.BFB = 0b MTU4.TMDR1.MD = 0000b
Make interrupt settings.	Priority setting INTC.IPR11._MTU24G = 5 (TGIB_4) INTC.IPR11._MTU24C = 6 (TCIV_4)	Software configurable interrupt vector assignments ICU.SLIAR210 = 22 (TGIB4) ICU.SLIAR211 = 25 (TCIV4) Protecting the software configurable interrupt source select register ICU.SLIPRCR.WPRC = 1b*1 Wait until ICU.SLIPRCR.WPRC is set to 1. Priority setting IPR(PERIA, INTA210) = 5 (TGIB4) IPR(PERIA, INTA211) = 6 (TCIV4) Clearing of ICU status IR(PERIA, INTA210) = 0b (TGIB4) IR(PERIA, INTA211) = 0b (TCIV4)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make interrupt settings.	Clearing of MTU status* MTU24.TSR.TGFB = 0b (TGIB_4) MTU24.TSR.TCFV = 0b (TCIV_4) * Clear to 0 after reading value as 1.	
Make I/O port settings (peripheral).		Cancellation of register protection MPC.PWPR.B0WI = 0b MPC.PWPR.PFSWE = 1b Pin peripheral function selection MPC.PC2PFS.PSEL = 000001b (MTIOC4B) Register protection settings MPC.PWPR.PFSWE = 0b MPC.PWPR.B0WI = 1b
	Pin mode setting PFC.PECRL4.PE13MD = 100b (TIOC4B)	Pin mode setting (peripheral) PORTC.PMR.B2 = 1b (MTIOC4B)

Note 1. Once ICU.SLIPRCR.WPRC is set to 1, it cannot be cleared to 0 by software.

Table 2.56 MTU Input Capture Operation Start Setting Examples

Processing	SH7216 Group Setting Example	RX71M Setting Example
Enable interrupts.	Enabling of MTU interrupts MTU24.TIER.TGIEB = 1b (TGIB_4) MTU24.TIER.TCIEV = 1b (TCIV_4)	Enabling of MTU interrupts MTU4.TIER.TGIEB = 1b (TGIB4) MTU4.TIER.TCIEV = 1b (TCIV4) Enabling of ICU interrupts IEN(PERIA, INTA210) = 1b (TGIB4) IEN(PERIA, INTA211) = 1b (TCIV4)
Start counting.	MTU2.TSTR.CST4 = 1b	MTU.TSTRA.CST4 = 1b

2.8 Port Output Enable (POE)

2.8.1 Comparison of Specifications

Port output enable functionality is provided on the SH7216 Group by the POE2 and on the RX71M by the POE3.

The RX71M includes the POE functionality of the SH7216 Group (backward compatibility). Table 2.57 lists comparative specifications of the SH7216 Group and RX71M.

Table 2.57 Comparison of SH7216 Group and RX71M Specifications (POE)

Item	SH7216 Group (POE2)	RX71M (POE3)
Clock source	Peripheral clock (Pϕ)	Peripheral module clock (PCLKB)
Pins subject to high-impedance control	• MTU0 pins • MTU high-current pins — MTU3 pins — MTU4 pins — MTU3S pins — MTU4S pins	• MTU0 pins • MTU complementary PWM output pins — MTU3 pins — MTU4 pins — MTU6 pins — MTU7 pins • GPT output pins — GPT0 pins — GPT1 pins — GPT2 pins — GPT3 pins
High-impedance request generation conditions	• Change in input pin state — Falling edge — Low level for Pϕ/8 × 16 cycles — Low level for Pϕ/16 × 16 cycles — Low level for Pϕ/128 × 16 cycles • Combined output signal level match for 1 cycle or more (short) • Register setting	• Change in input pin state — Falling edge — Low level for PCLKB/8 × 16 cycles — Low level for PCLKB/16 × 16 cycles — Low level for PCLKB/128 × 16 cycles • Combined output signal level match for 1 cycle or more (short) • Register setting • Detection of clock generation circuit oscillation stop
Interrupt sources	• High-impedance request by change in input pin state • High-impedance request by output signal level comparison	• High-impedance request by change in input pin state • High-impedance request by output signal level comparison
Other	—	Ability to add high-impedance control conditions for MTU complementary PWM output pins, MTU0 pins, and GPT pins

2.8.2 Input/Output Pins

On the SH7216 Group the only supported input pins are POE0# to POE8# for the MTU, but on the RX71M, in addition to the MTU input pins, POE10# and POE11# input signals are supported.

On the SH7216 Group the MTU0 pins are high-impedance only when assigned as general I/O pins when the MTU2 or MTU2S function is selected. On the RX71M multiplexed MTU complementary PWM output pins, MTU0 pins, and GPT pins are high-impedance even when the MTU or GPT is not selected.

Table 2.58 lists the input pins on the SH7216 Group and RX71M, and Table 2.59 provides a comparative listing of output pin combinations.

Table 2.58 POE Input Pins

SH7216 Group	RX71M	Subject to High-Impedance Control*1
POE0# to POE3#	POE0#	SH7216 Group: MTU3 and MTU4 pins RX71M: MTU3, MTU4, and GPT output pins
POE4#	POE4#	SH7216 Group: MTU3S and MTU4S pins RX71M: MTU6 and MTU7 pins
POE8#	POE8#	MTU0 pins
—	POE10#	GPT0 and GPT1 pins
—	POE11#	GPT2 and GPT3 pins

Note 1. On the RX71M the addition of high-impedance control conditions enables control of other pins as well.

Table 2.59 POE Output Pin Combinations

SH7216 Group	RX71M	Subject to High-Impedance Control
TIOC3B and TIOC3D	MTIOC3B and MTIOC3D	MTU3 and MTU4 pins
TIOC4A and TIOC4C	MTIOC4A and MTIOC4C	
TIOC4B and TIOC4D	MTIOC4B and MTIOC4D	
TIOC3BS and TIOC3DS	MTIOC6B and MTIOC6D	SH7216 Group: MTU3S and MTU4S pins RX71M: MTU6 and MTU7 pins
TIOC4AS and TIOC4CS	MTIOC7A and MTIOC7C	
TIOC4BS and TIOC4DS	MTIOC7B and MTIOC7D	
—	GTIOC0A and GTIOC0B	GPT0 to GPT2 pins
—	GTIOC1A and GTIOC1B	
—	GTIOC2A and GTIOC2B	

2.8.3 Register Comparison

On the SH7216 Group the port impedance state is specified by making settings to the port output enable control registers (POECR1 and POECR2). On the RX71M the port impedance state is specified by making settings to the port output enable control registers (POECR1 and POECR2), and the ports assigned to the various pins are specified by making settings to the pin select registers (M0SELR1 and M0SELR2, M3SELR, and M4SELR1 and M4SELR2) of the MTU channels.

Table 2.60 is a comparative listing of the registers on the SH7216 Group and RX71M.

Guide to Symbols in "Changes" Column of Table

◎: Register with same bit assignments on SH7216 Group and RX71M

△: Register with different bit assignments on SH7216 Group and RX71M

—: Register not present on SH7216 Group or RX71M

Table 2.60 SH7216 Group and RX71M Register Comparison (POE)

SH7216 Group (POE2)	RX71M (POE3)	Changes
Input level control/status register 1 (ICSR1)	Input level control/ status register 1 (ICSR1)	◎
Input level control/status register 2 (ICSR2)	Input level control/ status register 2 (ICSR2)	◎
Input level control/status register 3 (ICSR3)	Input level control/ status register 3 (ICSR3)	◎
—	Input level control/ status register 4 (ICSR4) Input level control/ status register 5 (ICSR5) Input level control/ status register 6 (ICSR6)	—
Output level control/status register 1 (OCSR1)	Output level control/ status register 1 (OCSR1)	◎
Output level control/status register 2 (OCSR2)	Output level control/ status register 2 (OCSR2)	◎
Software port output enable register (SPOER)	Software port output enable register (SPOER)	△
Port output enable control register 1 (POECR1)	Port output enable control register 1 (POECR1) MTU0 pin select register 1 (M0SELR1) MTU0 pin select register 2 (M0SELR2)	△
Port output enable control register 2 (POECR2)	Port output enable control register 2 (POECR2) MTU3 pin select register (M3SELR) MTU4 pin select register 1 (M4SELR1) MTU4 pin select register 2 (M4SELR2)	△
—	Active level setting register 1 (ALR1)	—
	Port output enable control register 3 (POECR3) Port output enable control register 4 (POECR4) Port output enable control register 5 (POECR5) Port output enable control register 6 (POECR6)	
	GPT0 pin select register (G0SELR) GPT1 pin select register (G1SELR) GPT2 pin select register (G2SELR) GPT3 pin select register (G3SELR)	
	MTU/GPT pin function select register (MGSELR)	

2.8.4 High-Impedance Control by Oscillation Stop Detection

The RX71M provides the ability to transition user-specified MTU complementary PWM output pins, MTU0 pins , GPT output pins, or GPT3 pins to the high-impedance state when oscillation stop is detected by the oscillation stop detection function of the clock generation circuit.

Pins that transition to the high-impedance state when oscillation stop is detected revert to the default state after a reset, but the high-impedance state is canceled by means of register settings.

2.8.5 Addition of High-Impedance Control Conditions

The RX71M supports the addition of high-impedance control conditions for the MTU complementary PWM output pins, MTU0 pins, and GPT output pins. Table 2.61 lists the high-impedance control conditions that can be added.

Table 2.61 Additional High-Impedance Control Conditions on RX71M

Subject to High-Impedance Control	Additional High-Impedance Control Conditions
MTU3 and MTU4 pins GPT0 to GPT2 pins	Input level detection on POE4#, POE8#, POE10#, and POE11#
MTU6 and MTU7 pins	Input level detection on POE0#, POE8#, POE10#, and POE11#.
MTU0 pins	Input level detection on POE0#, POE4#, POE10#, and POE11#.
GPT0 and GPT1 pins	Input level detection on POE0#, POE4#, POE8#, and POE11#.
GPT2 and GPT3 pins	Input level detection on POE0#, POE4#, POE8#, and POE10#.

2.8.6 Interrupts

On the RX71M POE3 is assigned to group interrupt BL1. The interrupt controller’s group BL1 interrupt status flag (GRPBL1.ISn) is automatically cleared when the corresponding bit in the module’s status register is cleared.

Refer to 1.9, Interrupt Handling for information about interrupts.

2.9 Watchdog Timers (WDT)

2.9.1 Comparison of Specifications

The SH7216 Group incorporates the WDT as its watchdog timer module. The RX71M incorporates, in addition to the WDTA, the IWDTa, which operates on a dedicated independent clock and is able to operate when the microcontroller is in the low-power-consumption state.

Table 2.62 lists comparative specifications of the SH7216 Group and RX71M.

Table 2.62 Comparison of SH7216 Group and RX71M Specifications (WDT)

Item	SH7216 Group (WDT)	RX71M (WDTA, IWDTa)
Clock sources	Peripheral clock (Pϕ)	WDTA: Peripheral module clock (PCLKB) IWDTa: IWDT dedicated clock (IWDTCLK) PCLKB ≥ 4 × IWDTCLK frequency after IWDTCLK frequency division
Clock frequency division ratio	Pϕ /1, 64, 128, 256, 512, 1,024, 4,096, 16,384	WDTA: PCLKB/4, 64, 128, 512, 2048, 8192 IWDTa: IWDTCLK/1, 16, 32, 64, 128, 256
Count operation	8-bit up-counter	14-bit down-counter
Operating modes	• Watchdog timer mode • Interval timer mode	None • Reset output enabled (equivalent to watchdog timer mode) • Interrupt requests enabled (equivalent to interval timer mode)
Count start condition	• Timer enable bit setting	In auto-start mode • After a reset • After an underflow • After a refresh error In register start mode • Refresh operation
Count stop condition	• Timer enable bit setting • After internal reset caused by overflow • Power-on reset caused by RES pin (counter and setting initialization)	• Reset (counter and setting initialization) • Underflow • Refresh error
Operation at overflow/ underflow	Watchdog timer mode • Internal reset (power-on reset and manual reset) • WDTOVF output Interval timer mode • Interrupt	When reset output enabled • Internal reset When interrupt request output enabled • Interrupt
Interrupt sources	• Overflow of up-counter	• Underflow of down-counter • Refresh error
Other	—	• Event link (IWDTa only) • Window function • Also operates in low-power-consumption state (IWDTa only) • Settings made in option function select register 0 in auto-start mode — Clock division ratio — Refresh window start/end — Timeout period — Enabling of interrupt requests and resets

2.9.2 Count Start Conditions

On the SH7216 Group count operation starts when 1 is written to the timer enable bit. The RX71M supports a register start mode, in which count operation is started with a write to a register (writing a setting to the option function select register), as on the SH7216 Group, and an auto-start mode, in which count operation starts automatically after a reset.

When auto-start mode is selected on the RX71M, count operation starts automatically after a reset, in accordance with the setting of option function select register (OFS0). When register start mode is selected, count operation is started by a refresh, after the appropriate register settings are made following reset cancelation.

2.9.3 Refresh Operation

On the RX71M the count is refreshed after 00h and then FFh is written to the WDT refresh register (WDTRR). Writes to the WDT refresh register must take place within the refresh-enabled interval. To refresh the count of IWDTa, perform the same write operation to the IWDT refresh register (IWDTRR) within the refresh-enabled interval.

Table 2.63 Comparison of Refresh Operation

Item	SH7216 Group	RX71M (WDTA)
Refresh condition	Write to watchdog timer counter (WTCNT)	00h and then FFh written to refresh register (WDTRR) within refresh-enabled interval
Counter initial value after refresh	Value written to watchdog timer counter (WTCNT)	Register start mode • Value selected by timeout period selection bits in WDT control register (WDTCR.TOPS) Auto start mode • Value selected by WDT timeout period select bits in option function select register (OFS0.WDTTOPS)

2.9.4 Register Write Limitations

Limitations apply when writing to the WDT registers of the SH7216 Group and RX71M. These register write limitations are summarized below.

Table 2.64 SH7216 Group Register Write Limitations

Item	Write Limitations
Watchdog timer counter (WTCNT) Watchdog reset control/status register (WRCSR) • Reset enable (WRCSR.RSTE) • Reset select (WRCSR.RSTS)	Writing in word-size units in the following configuration: • Upper byte: 5Ah • Lower byte: Write data
Watchdog timer control/status register (WTCSR) Watchdog reset control/status register (WRCSR) • Watchdog timer overflow (WRCSR.WOVF)	Writing in word-size units in the following configuration: • Upper byte: A5h • Lower byte: Write data

Table 2.65 RX71M Register Write Limitations

Item	Write Limitations
WDT control register (WDTCR) WDT reset control register (WDTRCR) IWDT control register (IWDTCR) IWDT reset control register (IWDTRCR) IWDT count stop control register (IWDTCSTPR)	Can be written to once in the interval between reset cancellation and the first refresh operation.

2.9.5 Interrupts

On the RX71M WDTA and IWDTa interrupts may be non-maskable or maskable. The interrupt controller interrupt status flag (IRn.IR) is cleared automatically when the corresponding interrupt is accepted.

Refer to 1.9, Interrupt Handling for information about interrupts.

2.9.6 All-Module Stop

The WDTA and IWDTa do not support a module-stop function.

The WDTA and IWDTa behave differently when the RX71M is in the all-module stop state. Table 2.66 lists the states of these modules when the microcontroller is in the all-module stop state.

Table 2.66 Module States in All-Module Stop State on RX71M

Module Name	Module State
Watchdog timer (WDTA)	Count stopped (state retained)
Independent watchdog Timer (IWDTa)	Selectable in option setting memory

2.9.7 Option Settings

On the RX71M it is possible to specify the microprocessor's state after a reset by setting the start mode select bits (OFS0.IWDTSTRT and OFS0.WDTSTRT).

2.10 Serial Communication Interface (SCI)

2.10.1 Comparison of Specifications

The SH7216 Group incorporates the SCI, which provide serial communication interface functionality, and the RX71M incorporates the SCIg and SCIh.

The SCIg provides, in addition to conventional asynchronous and clock-synchronous transfer capabilities, extended asynchronous functionality that supports a smartcard (IC card) interface. In addition, it supports simple I^2C bus interface single-master operation and simple SPI bus interface operation. The SCIh adds to the functions of the SCIg support for an extended serial interface. For details of the transfer methods not supported on the SH7216 Group, refer to RX71M Group User's Manual: Hardware.

Table 2.67 provides a comparative listing of the specifications of the SH7216 Group and RX71M.

Table 2.67 Comparison of SH7216 Group and RX71M Specifications (SCI)

Item		SH7216 Group (SCI)	RX71M (SCIg, SCIh)
Number of channels		4 channels (SCI0 to SCI2, SCI4)	SCIg: 8 channels (SCI0 to SCI7) SCIh: 1 channel (SCI12)
Clock source		Peripheral clock (Pϕ)	Peripheral module clock (PCLKB)
Serial communication modes		• Asynchronous • Clock-synchronous	• Asynchronous • Clock-synchronous • Smartcard interface • Simple I^2C bus • Simple SPI bus
Transfer speed		Any bit rate may be selected using the on-chip baud rate generator.	
Full-duplex communication		Double-buffer configurations for transmission and reception to enable continuous transmission and continuous reception	
Data transfer		Selectable between LSB-first and MSB-first (except for asynchronous 7-bit data)	Selectable between LSB-first and MSB-first (MSB-first only on simple I^2C bus)
DTC/DMAC activation		DTC activation supported	DTC/DMAC activation supported
Interrupt sources		• Transmit data-empty • Transmit end • Receive data-full • Receive error	• Transmit data-empty • Transmit end • Receive data-full • Receive error Used in simple I^2C mode. • Start condition • Restart condition • Stop condition generation-end
Asynchronous mode	Data length	7 bits, 8 bits	7 bits, 8 bits, 9 bits
	Stop bits	1 bit, 2 bits	
	Parity function	Even parity, odd parity, or no parity	
	Receive error detection	Parity error, overrun error, or framing error	
	Hardware flow control	No	Supported (controllable with CTSn# and RTSn# pins)
	Break detection	Detection of when a framing error occurs is possible by directly reading the level of the RXDn pin.	
	Clock source	Selectable between internal and external clock	Selectable between internal and external clock Ability to input transfer rate clock from TMR (SCI5, SCI6, and SCI12)

Item		SH7216 Group (SCI)	RX71M (SCIg, SCIh)
Asynchronous mode	Multi-processor communication	Yes	
	Noise cancellation	No	On-chip digital noise filter for input on RXDn pins
	Other	—	• Double-speed mode • Selectable start bit detection condition
Clock-synchronous mode	Data length	8 bits	
	Receive error detection	Overrun error	
	Hardware flow control	No	Supported (controllable with CTSn# and RTSn# pins)
Other		—	• Event link (SCI5 only) • Expanded serial mode (SCI12 only) • Bit rate modulation

2.10.2 Register Comparison

Table 2.68 is a comparative listing of the registers on the SH7216 Group and RX71M.

Guide to Symbols in "Changes" Column of Table

◎: Register with same bit assignments on SH7216 Group and RX71M

△: Register with different bit assignments on SH7216 Group and RX71M

—: Register not present on SH7216 Group or RX71M

Table 2.68 SH7216 Group and RX71M Register Comparison (SCI)

SH7216 Group (SCI)*1	RX71M (SCIg, SCIh)*2	Changes
Transmit data register n (SCTDR_n)	Transmit data register (SCIm.TDR)	◎
Transmit shift register (SCTSR)	Transmit shift register (TSR)	◎
Receive data register n (SCRDR_n)	Receive data register (SCIm.RDR)	◎
Receive shift register (SCRSR)	Receive shift register (RSR)	◎
Serial mode register n (SCSMR_n)	Serial mode register (SCIm.SMR)	◎
Serial control register n (SCSCR_n)	Serial control register (SCIm.SCR)	◎
Serial status register n (SCSSR_n)	Serial status register (SCIm.SSR)	◎
Bit rate register n (SCBRR_n)	Bit rate register (SCIm.BRR)	◎
Serial direction control register n (SCSDCR_n)	Smartcard mode register (SCIm.SCMR)	△
Serial port register n (SCSPTR_n)	—	—
—	Receive data register HL (SCIm.RDRHL)	—
	Transmit data register HL (SCIm.TDRHL)	
	Modulation duty register (SCIm.MDDR)	
	Serial extended mode register (SCIm.SEMR)	
	Noise filter setting register (SCIm.SNFR)	
	I^2C mode registers 1 to 3 (SCIm.SIMR1 to SCIm.SIMR3)	
	I^2C status register (SCIm.SISR)	
	SPI mode register (SCIm.SPMR)	
	Extended serial mode enable register (SCI12.ESMER)	
	Control registers 0 to 3 (SCI12.CR0 to SCI12.CR3)	
	Port control register (SCI12.PCR)	
	Interrupt control register (SCI12.ICR)	
	Status register (SCI12.STR)	
	Status clear register (SCI12.STCR)	
	Control field 0 data register (SCI12.CF0DR)	
	Control field 0 compare enable register (SCI12.CF0CR)	
	Control field 0 receive data register (SCI12.CF0RR)	
	Primary control field 1 data register (SCI12.PCF1DR)	
	Secondary control field 1 data register (SCI12.SCF1DR)	
	Control field 1 compare enable register (SCI12.CF1CR)	
	Control field 1 receive data register (SCI12.CF1RR)	
	Timer control register (SCI12.TCR)	
	Timer mode register (SCI12.TMR)	
	Timer prescaler register (SCI12.TPRE)	
	Timer count register (SCI12.TCNT)	

Note 1. SCI n: 0 to 2, 4
Note 2. SCI m: 0 to 7, 12

2.10.3 Clock Source Selection

TMR clock input (SCI5, SCI6, or SIC12 only) may be selected as the clock source for asynchronous mode communication on the RX71M. Also, whereas on the SH7216 Group a 16-bit clock is fixed as the base clock for one bit period, on the RX71M an 8-bit or 16-bit clock can be selected.

2.10.4 Interrupts

Whereas on the SH7216 Group a receive data-full or transmit data-empty interrupt can be used to activate the DTC only, on the RX71M these interrupts can be used to activate both the DTC and the DMAC.

On the RX71M when a receive data-full or transmit data-empty interrupt occurs while the corresponding interrupt status flag (IRn.IR) is set to 1, the interrupt request is also stored internally by the module, and after the interrupt status flag (IRn.IR) is cleared to 0 it is reset to 1 by the stored request.

On the RX71M some interrupts are assigned to group interrupt BL0. The interrupt controller's interrupt status flag (IRn.IR) is cleared automatically when the corresponding interrupt is accepted. Group BL0 interrupt status flag (GRPBL0.ISn) is cleared automatically when the corresponding bit in the module's status register is cleared.

Table 2.69 lists interrupt sources for the SH7216 Group and RX71M.

Refer to 1.9, Interrupt Handling for information about interrupts.

Table 2.69 SCI Interrupt Sources

		Activation by Interrupt	
Priority	**Interrupt Source**	**SH7216 Group**	**RX71M**
High	Receive error	Not possible	Not possible
↑	Receive data-full	DTC activation possible	DMAC and DTC activation possible
	Transmit data-empty		
Low	Transmit end	Not possible	Not possible

2.10.5 Module Stop

As on the SH7216 Group, the SCIg and SCIh of the RX71M is set to the module-stop state after a reset and no clock is supplied.

Refer to 2.21, Low Power Consumption Function for information on the module-stop state.

2.10.6 Asynchronous Transmit/Receive Setting Examples (Interrupt/Polling)

Setting examples for asynchronous transmit and receive operation using the serial communications interface of the SH7216 Group and RX71M are presented below.

Operational Overview

- Asynchronous transmit and receive are performed.
- Transmission is activated by a transmit data-empty interrupt.*1
- Reception is activated by a receive data-full interrupt, and the receive data is stored in the RAM.*1
- After transmission and reception of all of the data finishes, SCI operation ends.
- SCI operation ends if an error is detected.

Note 1. When polling is used, the state of the transmit data-empty flag in the status register is checked and transmission is activated, and the state of the receive data-full flag is checked and reception is activated, without making use of interrupts.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by "SH7216:" in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.70 Asynchronous Transmit/Receive Operation Specifications

Item		Description	Remarks
SCI channel		SCI2	
Clock		PCLKB = 60 MHz	SH7216: Pϕ = 50 MHz
Communication mode		Asynchronous serial communication	
Operating mode		Full-duplex synchronous transmission/ reception	
Transfer speed		38,400 bps	
Data length		8 bits	
Stop bits		1 bit	
Parity		Not added	
Hardware flow control		Not used.	
Bit order		LSB-first	
Clock source		Internal clock	
Transmit data		32 bytes (value from 1 to 32)	
Receive data		32 bytes	
Noise cancellation		Not used.	
Interrupts		All interrupts used.	Priority: level 5
Pins used	TXD	P50/TXD2	SH7216: PD3/TXD2
	RXD	P52/RXD2	SH7216: PD2/RXD2
	LED0	P03/general	Lights when transmission/ reception is possible.
	LED1	P05/general	Lights when transmission/ reception ends.
	LED2	P26/general	Lights when error detected.

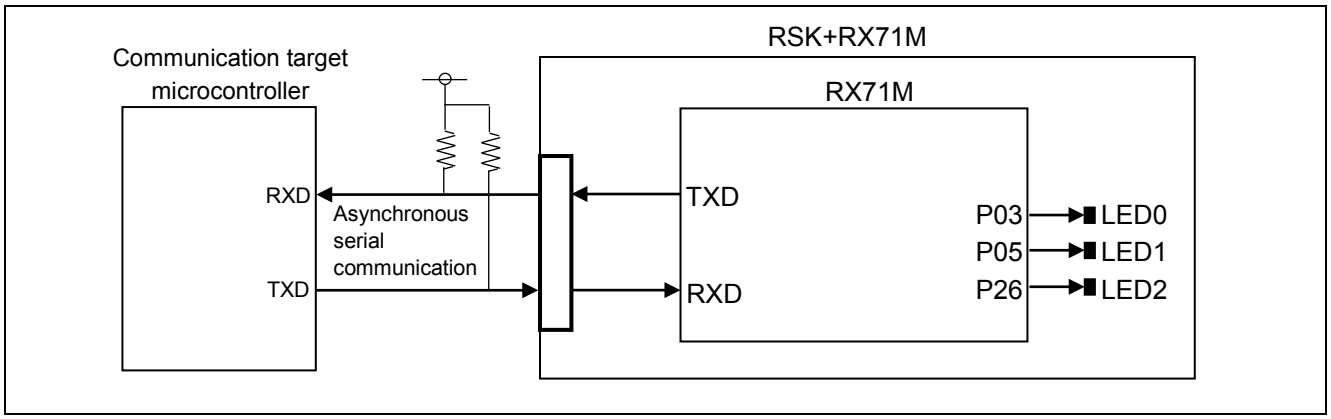


Figure 2.26 Asynchronous Transmit/Receive Connection Example

Note

On the Renesas Starter Kit+ for RX71M in the initial state the pins used for SCI2 in the connection example are connected for use with the external bus, so make appropriate modifications to the board as necessary.

List of Related Registers

Table 2.71 shows the interrupt-related registers used in the SH7216 Group setting example broken down by source.

Table 2.71 SH7216 Group Interrupt-Related Registers (SCI and INTC)

			SCI		INTC
Item	**Vector No.**	**Name**	**Interrupt Enable**	**Status**	**Priority Level**
Setting register	—	—	SCI2.SCSCR	SCI2.SCSSR	IPR16
Setting position of each source (SCI2)					
Transmit end	251	TEI2	TEIE	TEND	Bits 4 to 7
Receive error (overrun error)	248	ERI2	RIE	ORER	
Receive error (framing error)				FER	
Receive error (parity error)				PER	
Receive data-full	249	RXI2	RIE	RDRF	
Transmit data-empty	250	TXI2	TIE	TDRE	

Table 2.72 and Table 2.73 show the interrupt-related registers used in the RX71M setting example broken down by source. On the RX71M some SCI-related interrupts are assigned as group BL0 interrupts.

Table 2.72 RX71M Interrupt-Related Registers (SCIg)

Item	**Name**	**Interrupt Enable**	**Status**
Setting register	—	SCI2.SCR	SCI2.SSR
Setting position of each source (SCI2)			
Transmit end	TEI2	TEIE	TEND
Receive error (overrun error)	ERI2	RIE	ORER
Receive error (framing error)			FER
Receive error (parity error)			PER
Receive data-full	RXI2	RIE	RDRF
Transmit data-empty	TXI2	TIE	TDRE

Table 2.73 RX71M Interrupt-Related Registers (ICUA)

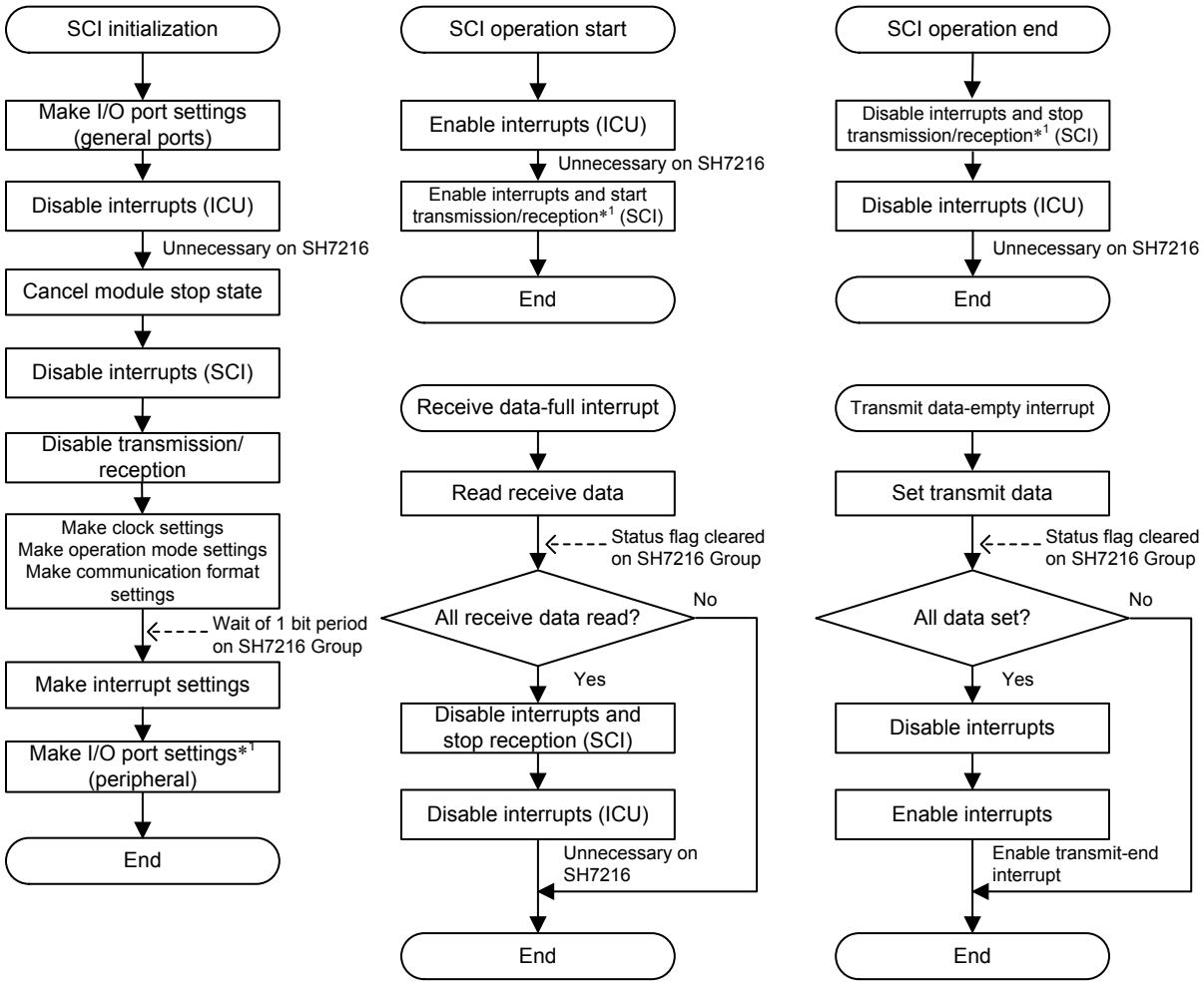
Item	**Vector No.**	**Name**	**Interrupt Enable**		**Status**		**Priority Level**
Setting register	—	—	IERm	GENBL0	IRr	GRPBL0	IPRr
Setting position of each source (SCI2)							
Transmit end	110	TEI2	IER0D.IEN6	EN4	IR110	IS4	IPR110
Receive error (overrun error)		ERI2	(group BL0)	EN5	(group	IS5	(group
Receive error (framing error)					BL0)		BL0)
Receive error (parity error)							
Receive data-full	62	RXI2	IER07.IEN6	—	IR062	—	IPR062
Transmit data-empty	63	TXI2	IER07.IEN7	—	IR063	—	IPR063
				↑		↑	
				Group BL0 interrupt settings			

When making settings to the ICUA interrupt-related registers on the RX71M, iodefine.h can be used to make settings as follows. GROUPBL0 defines settings for group BL0 interrupts.

- IERm : IEN (SCI2 or ICUA interrupt name), IEN (ICU or GROUPBL0)
- IPRr : IPR (SCI2 or ICUA interrupt name), IPR (ICU or GROUPBL0)
- IRr : IR (SCI2 or ICUA interrupt name), IR (ICU or GROUPBL0)
- GENBL0 : EN (SCI2 or ICUA interrupt name)
- GRPBL0 : IS (SCI2 or ICUA interrupt name)

Processing Flowcharts

Figure 2.27 shows example flowcharts of processing using the SCI. The names of the processing steps shown in the flowcharts correspond to the names in the setting examples.

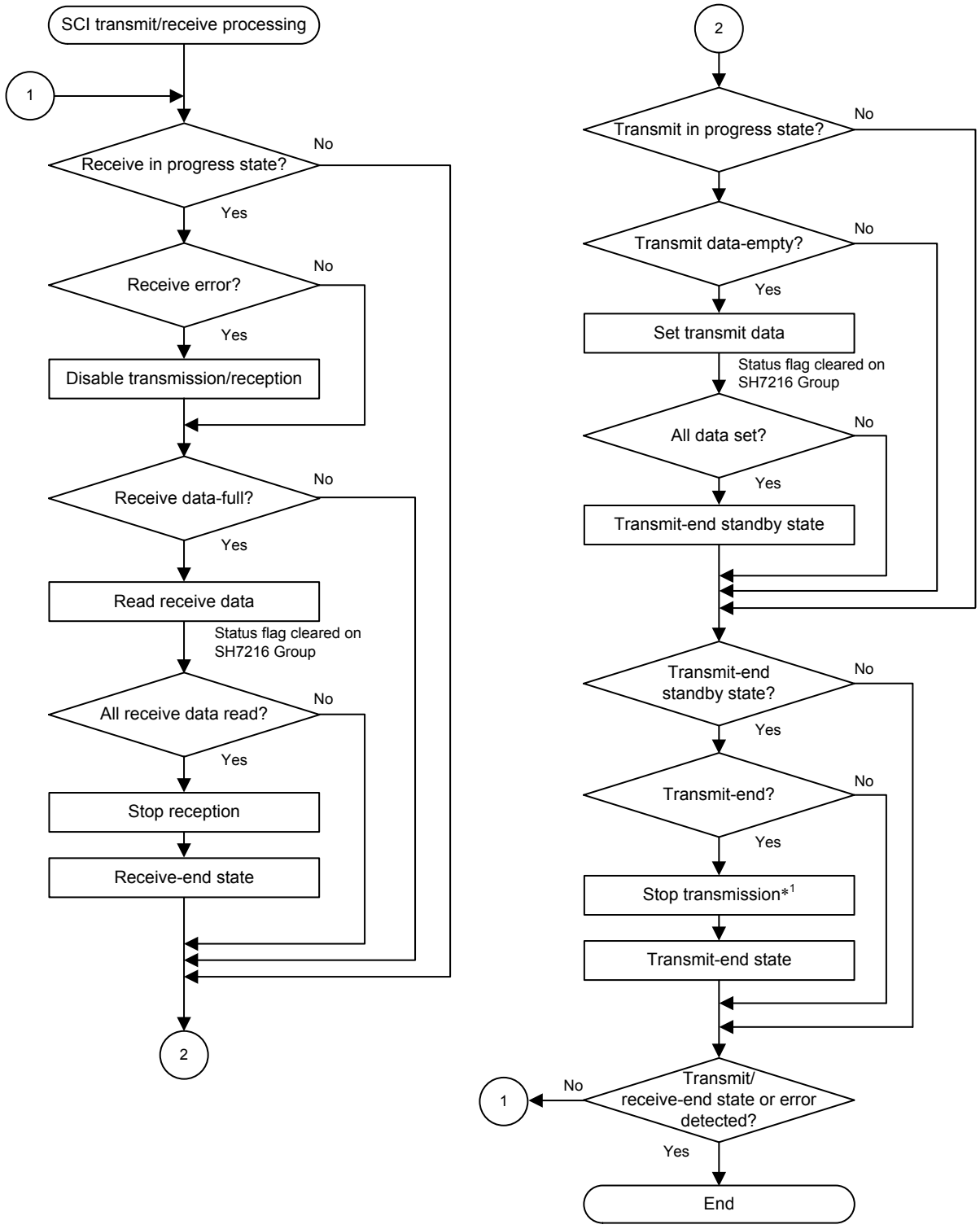


Note 1. On the RX71M if transmit operation ends with the pin function set to “TXDn,” the pin enters the high-impedance state. Use either of the following methods to prevent the TXDn line from entering the high-impedance state:

- Connect a pull-up resistor to the TXDn line.
- Change the pin function to “general” before transmit operation ends. Then change the pin function to “TXDn” after transmit operation starts.

Figure 2.27 Example Flowcharts of SCI Processing

Figure 2.28 is a flowchart of transmit and receive processing using polling with the SCI. In the example shown in this flowchart processing takes place in three states: receive in progress, transmit in progress, and transmit-end standby. Refer to Figure 2.27 for a flowchart of initialization processing. It is not necessary to enable ICU or SCI interrupts when polling is used.



Note 1. If transmit operation ends with the pin function set to “TXDn,” the pin enters the high-impedance state. Use either of the following methods to prevent the TXDn line from entering the high-impedance state:

- Connect a pull-up resistor to the TXDn line.
- Change the pin function to “general” before transmit operation ends. Then change the pin function to “TXDn” after transmit operation starts.

Figure 2.28 Example Flowcharts of SCI Processing (Transmit/Receive Using Polling)

Setting Examples

Setting examples for asynchronous transmit/receive are presented below. The names of the processing steps in these setting examples correspond to the names in the flowcharts. Refer to the flowcharts for the processing procedure.

Note that the setting examples below apply to both interrupt and polling operation.

Table 2.74 SCI Initialization Setting Example (Asynchronous Transmit/Receive)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make I/O port settings (general ports).	Output value setting (output 1) PD.DR.B3 = 1b (PD3) Pin direction settings PFC.PDIORL.B3 = 1b (output/PD3) PFC.PDIORL.B2 = 0b (input/PD2) Pin mode settings (port) PFC.PDCRL1.PD3MD = 000b (PD3) PFC.PDCRL1.PD2MD = 000b (PD2)	Output value setting (output 1) PORT5.PODR.B0 = 1b (P50) Pin direction settings PORT5.PDR.B0 = 1b (output/P50) PORT5.PDR.B2 = 0b (input/P52) Pin mode settings (general) PORT5.PMR.B0 = 0b (P50) PORT5.PMR.B2 = 0b (P52)
Disable interrupts (ICU).	—	Disabling of ICU interrupts IEN(SCI2, RXI2) = 0b IEN(SCI2, TXI2) = 0b IEN(ICU, GROUPBL0) = 0b (group BL0) Disabling interrupts in group BL0 EN(SCI2, TEI2) = 0b EN(SCI2, ERI2) = 0b
Cancel module stop state.	STB.CR5._SCI2 = 0b	SYSTEM.PRCR = A502h SYSTEM.MSTPCRB.MSTPB29 = 0b SYSTEM.PRCR = A500h
Disable interrupts (SCI).	Disabling of SCI interrupts SCI2.SCSCR.TEIE = 0b (TEI2) SCI2.SCSCR.RIE = 0b (RXI2, ERI2) SCI2.SCSCR.TIE = 0b (TXI2)	Disabling of SCI interrupts SCI2.SCR.TEIE = 0b (TEI2) SCI2.SCR.RIE = 0b (RXI2, ERI2) SCI2.SCR.TIE = 0b (TXI2)
Disable transmission/reception.	SCI2.SCSCR.RE, TE = 0b	SCI2.SCR.RE, TE = 0b
Make clock settings. Make operation mode settings. Make communication format settings.	Clock settings SCI2.SCSCR.CKE = 00b SCI2.SCSMR.CKS = 00b Operation mode setting SCI2.SCSMR.CA = 0b (asynchronous) Communication format settings SCI2.SCSMR._PE = 0b SCI2.SCSMR.STOP = 0b SCI2.SCSMR.MP = 0b SCI2.SCSDCR.DIR = 0b Data length setting (8 bits) SCI2.SCSMR.CHR = 0b Bit rate setting SCI2.SCBRR = 40 (38400 bps @50 MHz)	Clock settings SCI2.SCR.CKE = 00b SCI2.SMR.CKS = 00b SCI2.SEMR.ABCS = 0b (1 bit = 16 cycles of basic clock) SCI2.SEMR.BGDM = 0b (no doubling) Operation mode setting SCI2.SMR.CM = 0b (asynchronous) Communication format settings SCI2.SMR.PE = 0b SCI2.SMR.STOP = 0b SCI2.SMR.MP = 0b SCI2.SCMR.SDIR = 0b Data length settings (8 bits) SCI2.SCMR.CHR1 = 1b SCI2.SMR.CHR = 0b Bit rate setting SCI2.BRR = 48 (38400 bps @60 MHz)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make interrupt settings.*1	Priority setting (level 5) INTC.IPR16._SCI2 = 5	Priority settings (level 5) IPR(SCI2, RXI2) = 5 IPR(SCI2, TXI2) = 5 IPR(ICU, GROUPBL0) = 5 (group BL0) Clearing of ICU status IR(SCI2, RXI2) = 0b IR(SCI2, TXI2) = 0b IR(ICU, GROUPBL0) = 0b (group BL0)
	Clearing of SCI status flag* SCI2.SCSSR.ORER = 0b (ERI2) SCI2.SCSSR.FER = 0b (ERI2) SCI2.SCSSR.PER = 0b (ERI2) SCI2.SCSSR.RDRF = 0b (RXI2) * Clear to 0 after reading value as 1.	Clearing of SCI status flag* SCI2.SSR.ORER = 0b (ERI2) SCI2.SSR.FER = 0b (ERI2) SCI2.SSR.PER = 0b (ERI2) * Clear to 0 after reading value as 1.
Make I/O port settings (peripheral).		Cancellation of register protection MPC.PWPR.B0WI = 0b MPC.PWPR.PFSWE = 1b Pin peripheral function selection MPC.P50PFS.PSEL = 001010b (TXD2) MPC.P52PFS.PSEL = 001010b (RXD2) Register protection settings MPC.PWPR.PFSWE = 0b MPC.PWPR.B0WI = 1b
	Pin mode settings PFC.PDCRL1.PD3MD = 110b (TXD2) PFC.PDCRL1.PD2MD = 110b (RXD2)	Pin mode settings (peripheral) PORT5.PMR.B0 = 1b (TXD2) PORT5.PMR.B2 = 1b (RXD2)

Note 1. It is not necessary to clear the priority setting and the ICU status when polling is used.

Table 2.75 SCI Operation Start Setting Example

Processing	SH7216 Group Setting Example	RX71M Setting Example
Enable interrupts*1 (ICU). • Except for transmit-end interrupt	—	Enabling interrupts in group BL0 EN(SCI2, ERI2) = 1b Enabling of ICU interrupts IEN(SCI2, RXI2) = 1b IEN(SCI2, TXI2) = 1b IEN(ICU, GROUPBL0) = 1b (group BL0)
Enable interrupts*1 and start transmit/receive operation (SCI). • Except for transmit-end interrupt	SCI2.SCSCR.RE, TE, RIE, TIE = 1b	SCI2.SCR.RE, TE, RIE, TIE = 1b

Note 1. It is not necessary to enable ICU or SCI interrupts when polling is used.

Table 2.76 SCI Operation End Setting Examples

Processing	SH7216 Group Setting Example	RX71M Setting Example
Disable interrupts*1 and stop transmit/receive operation (SCI). • All interrupts	SCI2.SCSCR.RE, TE, RIE, TIE, TEIE = 0b	SCI2.SCR.RE, TE, RIE, TIE, TEIE = 0b
Disable interrupts*1 (ICU). • All interrupts	—	Disabling of ICU interrupts IEN(SCI2, RXI2) = 0b IEN(SCI2, TXI2) = 0b IEN(ICU, GROUPBL0) = 0b (group BL0) Disabling interrupts in group BL0 EN(SCI2, TEI2) = 0b EN(SCI2, ERI2) = 0b

Note 1. It is not necessary to disable interrupts when polling is used.

The sample code does not specify any particular interrupt handling. The setting examples below apply to a portion of the processing involved in interrupt handling.

Table 2.77 Setting Examples in SCI Receive Data-Full Interrupt Handling

Processing	SH7216 Group Setting Example	RX71M Setting Example
Read receive data.	Reading value of SCI2.SCRDR	Reading value of SCI2.RDR
Clear status flag.	SCI2.SCSSR.RDRF = 0b Clear to 0 after reading value as 1.	—
Disable interrupts and stop receive operation (SCI). • Receive error interrupt • Receive data-full interrupt	SCI2.SCSCR.RE, RIE = 0b	SCI2.SCR.RE, RIE = 0b
Disable interrupts (ICU). • Receive error interrupt • Receive data-full interrupt	—	Disabling of ICU interrupts IEN(SCI2, RXI2) = 0b (RXI2) Disabling interrupts in group BL0 EN(SCI2, ERI2) = 0b Group BL0 is used by TEI2, so not necessary to disable.

Table 2.78 Setting Examples in SCI Transmit Data-Empty Interrupt Handling

Processing	SH7216 Group Setting Example	RX71M Setting Example
Set transmit data.	Setting the value of SCI2.SCTDR	Setting the value of SCI2.TDR
Clear status flag.	SCI2.SCSSR.TDRE = 0b Clear to 0 after reading value as 1.	—
Disable interrupts. • Transmit data-empty interrupt	Disabling of SCI interrupts SCI2.SCSCR.TIE = 0b (TXI2)	Disabling of ICU interrupts IEN(SCI2, TXI2) = 0b (TXI2) Disabling of SCI interrupts SCI2.SCR.TIE = 0b (TXI2)
Enable interrupts. • Transmit end interrupt	Enabling of SCI interrupts SCI2.SCSCR.TEIE = 1b (TEI2)	Enabling of SCI interrupts SCI2.SCR.TEIE = 1b (TEI2) Enabling interrupts in group BL0 EN(SCI2, TEI2) = 1b Group BL0 is used by ERI2, so not necessary to enable.

2.10.7 Clock-Synchronous Master Transmit Setting Examples (Interrupt/Polling)

Setting examples for clock-synchronous master transmit processing using the serial communication interface of the SH7216 Group and RX71M are presented below.

Operational Overview

- Clock-synchronous master transmit operation is performed.
- Transmission is activated by a transmit data-empty interrupt.*1
- After transmission of all the data finishes, SCI operation ends.

Note 1. When polling is used, the state of the transmit data-empty flag in the status register is checked and transmission is activated, without making use of interrupts.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by "SH7216:" in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.79 Clock-Synchronous Master Transmit Operation Specifications

Item		Description	Remarks
SCI channel		SCI2	
Clock		PCLKB = 60 MHz	SH7216: Pϕ = 50 MHz
Communication mode		Clock-synchronous serial communication	
Operating mode		Master transmit	
Transfer speed		100 kbps	
Data length		8 bits	
Bit order		LSB-first	
Clock source		Internal clock	
Transmit data		32 bytes (value from 1 to 32)	
Noise cancellation		Not used.	
Interrupts		Transmit data-full Transmit end	Priority: level 5
Pins used	SCK	P51/SCK2	SH7216: PD4/SCK2
	TXD	P50/TXD2	SH7216: PD3/TXD2
	LED0	P03/general	Lights when transmission starts.
	LED1	P05/general	Lights when transmission ends.

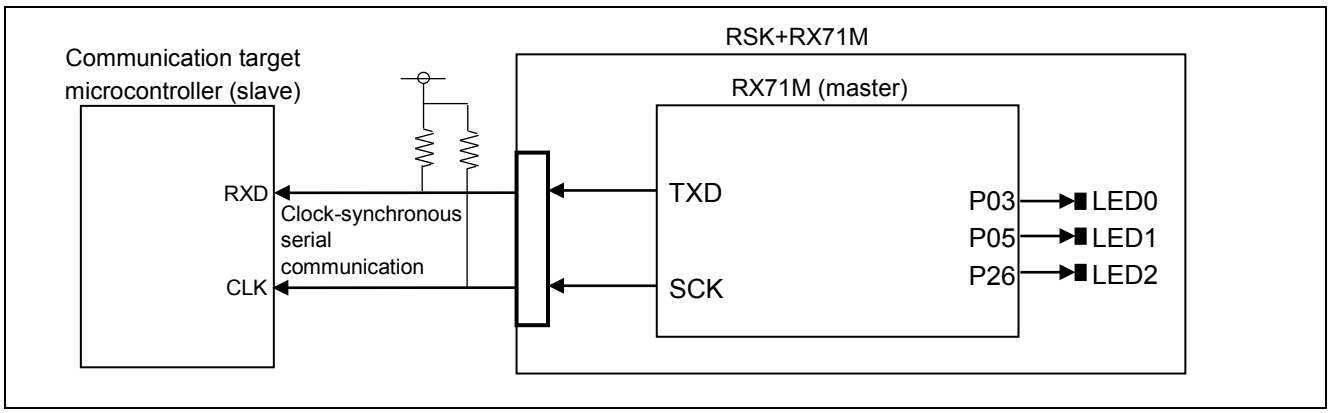


Figure 2.29 Clock-Synchronous Master Transmit Connection Example

Note

On the Renesas Starter Kit+ for RX71M in the initial state the pins used for SCI2 in the connection example are connected for use with the external bus, so make appropriate modifications to the board as necessary.

List of Related Registers

For the interrupt-related registers used in the setting examples, refer to the description of registers related to asynchronous operation in 2.10.6, Asynchronous Transmit/Receive Setting Examples (Interrupt/Polling).

Processing Flowcharts

For example flowcharts of processing using the SCI, refer to the example flowcharts of asynchronous processing in 2.10.6, Asynchronous Transmit/Receive Setting Examples (Interrupt/Polling). In the clock-synchronous master transmit setting examples only transmit operation takes place, so reception-related processing is not necessary.

Setting Examples

Setting examples for clock-synchronous master transmit are presented below. The names of the processing steps in these setting examples correspond to the names in the flowcharts. Refer to the flowcharts for the processing procedure.

Note that the setting examples below apply to both interrupt and polling operation.

Table 2.80 SCI Initialization Setting Example (Clock-Synchronous Master Transmit)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make I/O port settings (general ports).	Output value settings (output 1) PD.DR.B4 = 1b (PD4) PD.DR.B3 = 1b (PD3) Pin direction settings PFC.PDIORL.B4 = 1b (output/PD4) PFC.PDIORL.B3 = 1b (output/PD3) Pin mode settings (port) PFC.PDCRL2.PD4MD = 000b (PD4) PFC.PDCRL1.PD3MD = 000b (PD3)	Output value settings (output 1) PORT5.PODR.B1 = 1b (P51) PORT5.PODR.B0 = 1b (P50) Pin direction settings PORT5.PDR.B1 = 1b (output/P51) PORT5.PDR.B0 = 1b (output/P50) Pin mode settings (general) PORT5.PMR.B1 = 0b (P51) PORT5.PMR.B0 = 0b (P50)
Disable interrupts (ICU).	—	Disabling of ICU interrupts IEN(SCI2, RXI2) = 0b IEN(SCI2, TXI2) = 0b IEN(ICU, GROUPBL0) = 0b (group BL0) Disabling interrupts in group BL0 EN(SCI2, TEI2) = 0b EN(SCI2, ERI2) = 0b
Cancel module stop state.	STB.CR5._SCI2 = 0b	SYSTEM.PRCR = A502h SYSTEM.MSTPCRB.MSTPB29 = 0b SYSTEM.PRCR = A500h
Disable interrupts (SCI).	Disabling of SCI interrupts SCI2.SCSCR.TEIE = 0b (TEI2) SCI2.SCSCR.RIE = 0b (RXI2, ERI2) SCI2.SCSCR.TIE = 0b (TXI2)	Disabling of SCI interrupts SCI2.SCR.TEIE = 0b (TEI2) SCI2.SCR.RIE = 0b (RXI2, ERI2) SCI2.SCR.TIE = 0b (TXI2)
Disable transmission/reception.	SCI2.SCSCR.RE, TE = 0b	SCI2.SCR.RE, TE = 0b

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make clock settings. Make operation mode settings. Make communication format settings.	Clock settings SCI2.SCSCR.CKE = 00b SCI2.SCSMR.CKS = 00b Operation mode setting SCI2.SCSMR.CA = 1b (clock-synchronous) Communication format setting SCI2.SCSDCR.DIR = 0b Bit rate setting SCI2.SCBRR = 124 (100 kbps@50 MHz)	Clock settings SCI2.SCR.CKE = 00b SCI2.SMR.CKS = 00b Operation mode setting SCI2.SMR.CM = 1b (clock-synchronous) Communication format setting SCI2.SCMR.SDIR = 0b Bit rate setting SCI2.BRR = 149 (100 kbps@60 MHz)
Make interrupt settings.*1	Priority setting (level 5) INTC.IPR16._SCI2 = 5	Priority settings (level 5) IPR(SCI2, TXI2) = 5 IPR(ICU, GROUPBL0) = 5 (group BL0) Clearing of ICU status IR(SCI2, TXI2) = 0b IR(ICU, GROUPBL0) = 0b (group BL0)
Make I/O port settings (peripheral).		Cancellation of register protection MPC.PWPR.B0WI = 0b MPC.PWPR.PFSWE = 1b Pin peripheral function selection MPC.P51PFS.PSEL = 001010b (SCK2) MPC.P50PFS.PSEL = 001010b (TXD2) Register protection settings MPC.PWPR.PFSWE = 0b MPC.PWPR.B0WI = 1b
	Pin mode settings PFC.PDCRL2.PD4MD = 110b (SCK2) PFC.PDCRL1.PD3MD = 110b (TXD2)	Pin mode settings (peripheral) PORT5.PMR.B1 = 1b (SCK2) PORT5.PMR.B0 = 1b (TXD2)

Note 1. It is not necessary to clear the priority setting and the ICU status when polling is used.

Refer to the setting examples for asynchronous processing in 2.10.6, Asynchronous Transmit/Receive Setting Examples (Interrupt/Polling), except regarding initialization of clock-synchronous master transmit operation. In the clock-synchronous master transmit setting examples only transmit operation takes place, so reception-related processing is not necessary.

2.10.8 Clock-Synchronous Slave Reception Setting Examples (Interrupt/Polling)

Setting examples for clock-synchronous slave reception using the serial communication interface of the SH7216 Group and RX71M are presented below.

Operational Overview

- Clock-synchronous slave reception is performed.
- Reception is activated by a receive data-full interrupt, and the receive data is stored in the RAM.*1
- After reception of all the data finishes, SCI operation ends.
- SCI operation ends when an error is detected.

Note 1. When polling is used, the state of the receive data-full flag in the status register is checked and reception is activated, without making use of interrupts.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by "SH7216:" in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.81 Clock-Synchronous Slave Receive Operation Specifications

Item		Description	Remarks
SCI channel		SCI2	
Communication mode		Clock-synchronous serial communication	
Operating mode		Slave receive	
Data length		8 bits	
Bit order		LSB-first	
Clock source		External clock	
Noise cancellation		Not used.	
Interrupts		Receive data-empty Receive error	Priority: level 5
Pins used	SCK	P51/SCK2	SH7216: PD4/SCK2
	RXD	P52/RXD2	SH7216: PD2/RXD2
	LED0	P03/general	Lights when reception starts.
	LED1	P05/general	Lights when reception ends.
	LED2	P26/general	Lights when error detected.

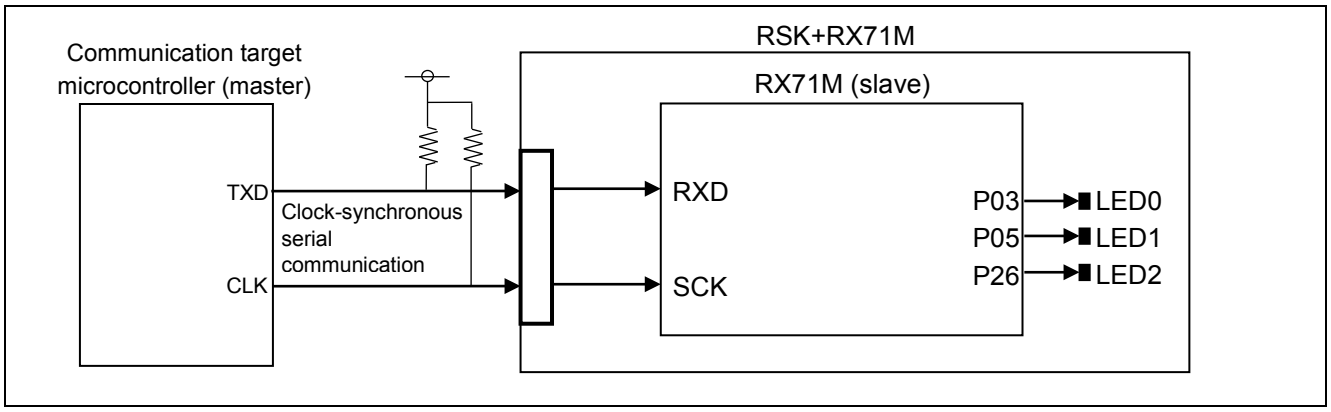


Figure 2.30 Clock-Synchronous Slave Receive Connection Example

Note

On the Renesas Starter Kit+ for RX71M in the initial state the pins used for SCI2 in the connection example are connected for use with the external bus, so make appropriate modifications to the board as necessary.

List of Related Registers

For the interrupt-related registers used in the setting examples, refer to the description of registers related to asynchronous operation in 2.10.6, Asynchronous Transmit/Receive Setting Examples (Interrupt/Polling).

Processing Flowcharts

For example flowcharts of processing using the SCI, refer to the example flowcharts of asynchronous processing in 2.10.6, Asynchronous Transmit/Receive Setting Examples (Interrupt/Polling). In the clock-synchronous slave receive setting examples only receive operation takes place, so transmission-related processing is not necessary.

Setting Examples

Setting examples for clock-synchronous slave receive are presented below. The names of the processing steps in these setting examples correspond to the names in the flowcharts. Refer to the flowcharts for the processing procedure.

Note that the setting examples below apply to both interrupt and polling operation.

Table 2.82 SCI Initialization Setting Example (Clock-Synchronous Slave Receive)

Procedure	SH7216 Group Setting Example	RX71M Setting Example
Make I/O port settings (general ports).	Pin direction settings PFC.PDIORL.B4 = 0b (input/PD4) PFC.PDIORL.B2 = 0b (input/PD2) Pin mode settings (port) PFC.PDCRL2.PD4MD = 000b (PD4) PFC.PDCRL1.PD2MD = 000b (PD2)	Pin direction settings PORT5.PDR.B1 = 0b (input/P51) PORT5.PDR.B2 = 0b (input/P52) Pin mode settings (general) PORT5.PMR.B1 = 0b (P51) PORT5.PMR.B2 = 0b (P52)
Disable interrupts (ICU).	—	Disabling of ICU interrupts IEN (SCI2, RXI2) = 0b IEN(SCI2, TXI2) = 0b IEN(ICU, GROUPBL0) = 0b (group BL0) Disabling interrupts in group BL0 EN(SCI2, TEI2) = 0b EN(SCI2, ERI2) = 0b
Cancel module stop state.	STB.CR5._SCI2 = 0b	SYSTEM.PRCR = A502h SYSTEM.MSTPCRB.MSTPB29 = 0b SYSTEM.PRCR = A500h
Disable interrupts (SCI).	Disabling of SCI interrupts SCI2.SCSCR.TEIE = 0b (TEI2) SCI2.SCSCR.RIE = 0b (RXI2, ERI2) SCI2.SCSCR.TIE = 0b (TXI2)	Disabling of SCI interrupts SCI2.SCR.TEIE = 0b (TEI2) SCI2.SCR.RIE = 0b (RXI2, ERI2) SCI2.SCR.TIE = 0b (TXI2)
Disable transmission/ reception.	SCI2.SCSCR.RE, TE = 0b	SCI2.SCR.RE, TE = 0b
Make clock settings. Make operation mode settings. Make communication format settings.	Clock settings SCI2.SCSCR.CKE = 10b SCI2.SCSMR.CKS = 00b Operation mode setting SCI2.SCSMR.CA = 1b (clock-synchronous) Communication format setting SCI2.SCSDCR.DIR = 0b	Clock settings SCI2.SCR.CKE = 10b SCI2.SMR.CKS = 00b Operation mode setting SCI2.SMR.CM = 1b (clock-synchronous) Communication format setting SCI2.SCMR.SDIR = 0b

Procedure	SH7216 Group Setting Example	RX71M Setting Example
Make interrupt settings.*1	Priority setting (level 5) INTC.IPR16._SCI2 = 5	Priority settings (level 5) IPR(SCI2, RXI2) = 5 IPR(ICU, GROUPBL0) = 5 (group BL0) Clearing of ICU status IR(SCI2, RXI2) = 0b IR(ICU, GROUPBL0) = 0b (group BL0)
	Clearing of SCI status flag* SCI2.SCSSR.ORER = 0b (ERI2) SCI2.SCSSR.RDRF = 0b (RXI2) * Clear to 0 after reading value as 1.	Clearing of SCI status flag* SCI2.SSR.ORER = 0b (ERI2) * Clear to 0 after reading value as 1.
Make I/O port settings (peripheral).		Cancellation of register protection MPC.PWPR.B0WI = 0b MPC.PWPR.PFSWE = 1b Pin peripheral function selection MPC.P51PFS.PSEL = 001010b (SCK2) MPC.P52PFS.PSEL = 001010b (RXD2) Register protection settings MPC.PWPR.PFSWE = 0b MPC.PWPR.B0WI = 1b
	Pin mode settings PFC.PDCRL2.PD4MD = 110b (SCK2) PFC.PDCRL1.PD2MD = 110b (RXD2)	Pin mode settings (peripheral) PORT5.PMR.B1 = 1b (SCK2) PORT5.PMR.B2 = 1b (RXD2)

Note 1. It is not necessary to clear the priority setting and the ICU status when polling is used.

Refer to the setting examples for asynchronous processing in 2.10.6, Asynchronous Transmit/Receive Setting Examples (Interrupt/Polling), except regarding initialization of clock-synchronous master transmit operation. In the clock-synchronous master transmit setting examples only transmit operation takes place, so reception-related processing is not necessary.

2.11 Serial Communications Interface with FIFO (SCIF)

2.11.1 Comparison of Specifications

The serial communications interface with FIFO functionality is provided on the SH7216 Group by the SCIF and on the RX71M by the SCIFA.

Table 2.83 provides a comparative listing of the specifications of the SH7216 Group and RX71M.

Table 2.83 Comparison of SH7216 Group and RX71M Specifications (SCIF)

Item		SH7216 Group (SCIF)	RX71M (SCIFA)
Number of channels		1 channel (SCIF3)	4 channels (SCIFA8 to SCIFA11)
Clock source		Peripheral clock (Pϕ)	Peripheral module clock (PCLKA)
Serial communication mode		• Asynchronous • Clock-synchronous	
Transfer speed		Any bit rate may be selected using the on-chip baud rate generator.	
Full-duplex communication		Continuous transmit and receive operation possible using 16-stage FIFO buffering	
Data transfer		LSB-first	Selectable between LSB-first and MSB-first
DTC/DMAC control		DTC/DMAC control supported	
Interrupt Source		• Transmit FIFO data-empty • Break • Receive FIFO data-full • Receive error	• Transmit FIFO data-empty • Break • Receive FIFO data-full • Receive error • Transmit-end • Receive data-ready
Asynchronous mode	Data length	7 bits, 8 bits	
	Stop bits	1 bit, 2 bits	
	Parity function	Even parity, odd parity, or no parity	
	Receive error detection	Parity error, overrun error, flaming error	
	Hardware flow control	No	Supported (controllable with CTSn# and RTSn# pins)
	Break detection	Break detection is possible. Also, framing errors can be detected by reading the level of the RXDn pin directly.	Break detection is possible.
	Clock source	Selectable between internal and external clock	
	Noise cancellation	No	On-chip digital noise filter for input on RXDn pins
Clock-synchronous mode	Data length	8 bits	
	Receive error detection	Overrun error	
	Hardware flow control	No	Supported (controllable with CTSn# and RTSn# pins)
Other		—	Bit rate modulation

2.11.2 Register Comparison

Table 2.84 is a comparative listing of the registers on the SH7216 Group and RX71M.

Guide to Symbols in "Changes" Column of Table

◎: Register with same bit assignments on SH7216 Group and RX71M

△: Register with different bit assignments on SH7216 Group and RX71M

—: Register not present on SH7216 Group or RX71M

Table 2.84 SH7216 Group and RX71M Register Comparison (SCIF)

SH7216 Group (SCIF)*1	RX71M (SCIFA)*2	Changes
Transmit FIFO data register n (SCFTDR_n)	Transmit FIFO data register (SCIFAm.FTDR)	◎
Transmit shift register (SCTSR)	Transmit shift register (TSR)	◎
Receive FIFO data register n (SCFRDR_n)	Receive FIFO data register (SCIFAm.FRDR)	◎
Receive shift register (SCRSR)	Receive shift register (RSR)	◎
Serial mode register n (SCSMR_n)	Serial mode register (SCIFAm.SMR)	◎
Serial control register n (SCSCR_n)	Serial control register (SCIFAm.SCR)	◎
Serial status register n (SCFSR_n)	Serial status register (SCIFAm.FSR) Line status register (SCIFAm.LSR)	△
Bit rate register n (SCBRR_n)	Bit rate register (SCIFAm.BRR)	◎
Serial port register n (SCSPTR_n)	Serial port register (SCIFAm.SPTR)	◎
FIFO control register n (SCFCR_n)	FIFO control register (SCIFAm.FCR)	◎
FIFO data count register n (SCFDR_n)	FIFO data count register (SCIFAm.FDR)	◎
Line status register n (SCLSR_n)	Line status register (SCIFAm.LSR)	◎
Serial extended mode register n (SCSEMR_n)	Serial extended mode register (SCIFAm.SEMR)	△
—	Modulation duty register (SCIFAm.MDDR)	—
	FIFO trigger control register (SCIFAm.FTCR)	

Note 1. SCI n: 3
Note 2. SCI m: 8 to 11

2.11.3 Interrupts

On both the SH7216 Group and the RX71M the receive FIFO data-full and transmit FIFO data-empty interrupts can be used to activate the DTC and DMAC.

On the RX71M some interrupts are assigned to group interrupt AL0. The interrupt controller's interrupt status flag (IRn.IR) is cleared automatically when the corresponding interrupt is accepted. Group AL0 interrupt status flag (GRPAL0.ISn) is cleared automatically when the corresponding bit in the module's status register is cleared.

Table 2.85 lists interrupt sources on the SH7216 Group, and Table 2.86 lists interrupt sources on the RX71M.

Refer to 1.9, Interrupt Handling for information about interrupts.

Table 2.85 SCIF Interrupt Sources on SH7216 Group

Interrupt Source	Activation by Interrupt	Priority
Break or overrun	Not possible	High
Receive error		↑
Receive FIFO data-full or receive data-ready	DMAC and DTC activation possible	
Transmit FIFO data-empty		Low

Table 2.86 SCIFA Interrupt Sources on RX71M

Interrupt Source	Activation by Interrupt	Priority
Break or overrun	Not possible	High
Receive error (framing error or parity error)		↑
Receive FIFO data-full	DMAC and DTC activation possible	
Transmit FIFO data-empty		
Transmit-end	Not possible	
Receive data-ready		Low

2.11.4 Module Stop

As on the SH7216 Group, the SCIFA of the RX71M is set to the module-stop state after a reset and no clock is supplied.

Refer to 2.21, Low Power Consumption Function for information on the module-stop state.

2.11.5 Asynchronous Transmit/Receive Setting Examples

Setting examples for asynchronous transmission and reception using the serial communication interface with FIFO on the SH7216 Group and RX71M are presented below.

Operational Overview

- Asynchronous transmit and receive are performed.
- Transmission is activated by a transmit FIFO data-empty interrupt.
- Reception is activated by a receive FIFO data-full interrupt, and the receive data is stored in the RAM.
- After transmission and reception of all of the data finishes, SCIF operation ends.
- SCIF operation ends if an error is detected.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by "SH7216:" in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.87 Asynchronous Transmit/Receive Operation Specifications

Item		Description	Remarks
SCIF channel		SCIFA8	SH7216: SCIF3
Clock		PCLKA = 120 MHz	SH7216: Pϕ = 50 MHz
Communication mode		Asynchronous serial communication	
Operating mode		Full-duplex synchronous transmission/reception	
Transfer speed		38,400 bps	
Data length		8 bits	
Stop bits		1 bit	
Parity		No	
Hardware flow control		No	
Bit order		LSB-first	
Transmit FIFO data count trigger		8	
Receive FIFO data count trigger		8	
Clock source		Internal clock	
Transmit data		128 bytes (value from 1 to 128)	
Receive data		128 bytes	
Noise cancellation		Not used.	
Interrupts		Transmit-end (RX71M only) Transmit FIFO data-empty Receive FIFO data-full Receive error Break detection	Priority: level 5
Pins used	TXD	PC7/TXD8	SH7216: PE5/TXD3
	RXD	PC6/RXD8	SH7216: PE6/RXD3
	LED0	P03/general	Lights when transmission/reception is possible.
	LED1	P05/general	Lights when transmission/reception ends.
	LED2	P26/general	Lights when error detected.

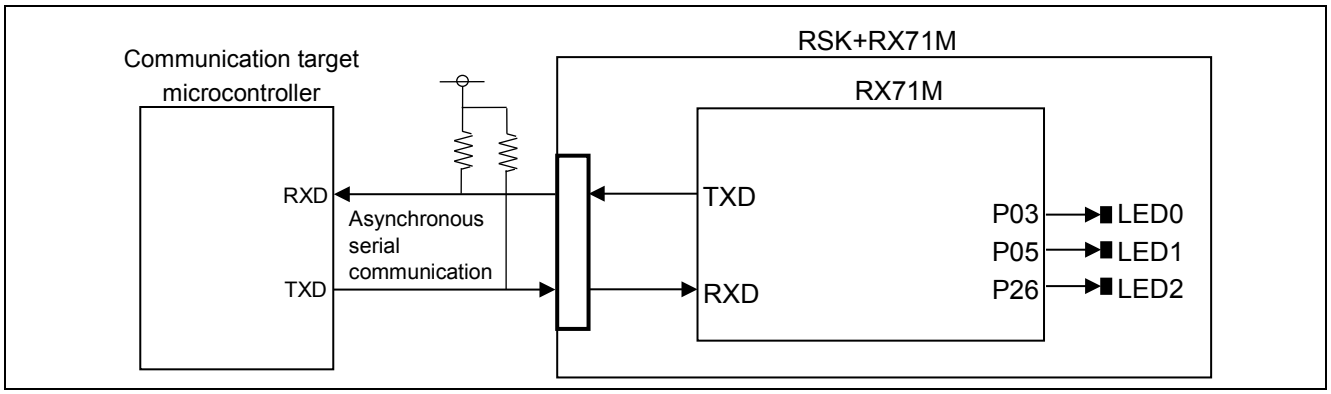


Figure 2.31 Asynchronous Transmit/Receive Connection Example

Note

On the Renesas Starter Kit+ for RX71M in the initial state the pins used for SCIFA8 in the connection example are connected for use with the external bus, so make appropriate modifications to the board as necessary.

List of Related Registers

Table 2.88 shows the interrupt-related registers used in the SH7216 Group setting example broken down by source.

Table 2.88 SH7216 Group Interrupt-Related Registers (SCIF and INTC)

			SCIF		INTC
Item	Vector No.	Name	Interrupt Enable	Status	Priority Level
Setting register	—	—	SCIF3.SCSCR	SCIF3.SCFSR	IPR14
Setting position of each source					
Transmit-end (no interrupt)	—	—	—	TEND	—
Break detection	252	BRI3	REIE, RIE	BRK	Bits 0 to 3
Receive error (overrun)				SCIF3.SCLSR.ORER (Register differs.)	
Receive error (framing error)	253	ERI3	REIE, RIE	ER, FER	
Receive error (parity error)				ER, PER	
Receive data-ready	254	RXI3	RIE	DR	
Receive FIFO data-full				RDF	
Transmit FIFO data-empty	255	TXI3	TIE	TDFE	

Table 2.89 and Table 2.90 show the interrupt-related registers used in the RX71M setting example broken down by source. On the RX71M some SCIFA-related interrupts are assigned as group AL0 interrupts.

Table 2.89 RX71M Interrupt-Related Registers (SCIFA)

Item	Name	Interrupt Enable	Status
Setting register	—	SCIFA8.SCR	SCIFA8.FSR
Setting position of each source (SCIFA8)			
Transmit-end	TEIF8	TEIE	TEND
Break detection	BRIF8	REIE, RIE	BRK
Receive error (overrun)			SCIFA8.LSR.ORER (Register differs.)
Receive error (framing error)	ERIF8	REIE, RIE	ER, FER
Receive error (parity error)			ER, PER
Receive data-ready	DRIF8	RIE	DR
Receive FIFO data-full	RXIF8	RIE	RDF
Transmit FIFO data-empty	TXIF8	TIE	TDFE

Table 2.90 RX71M Interrupt-Related Registers (ICUA)

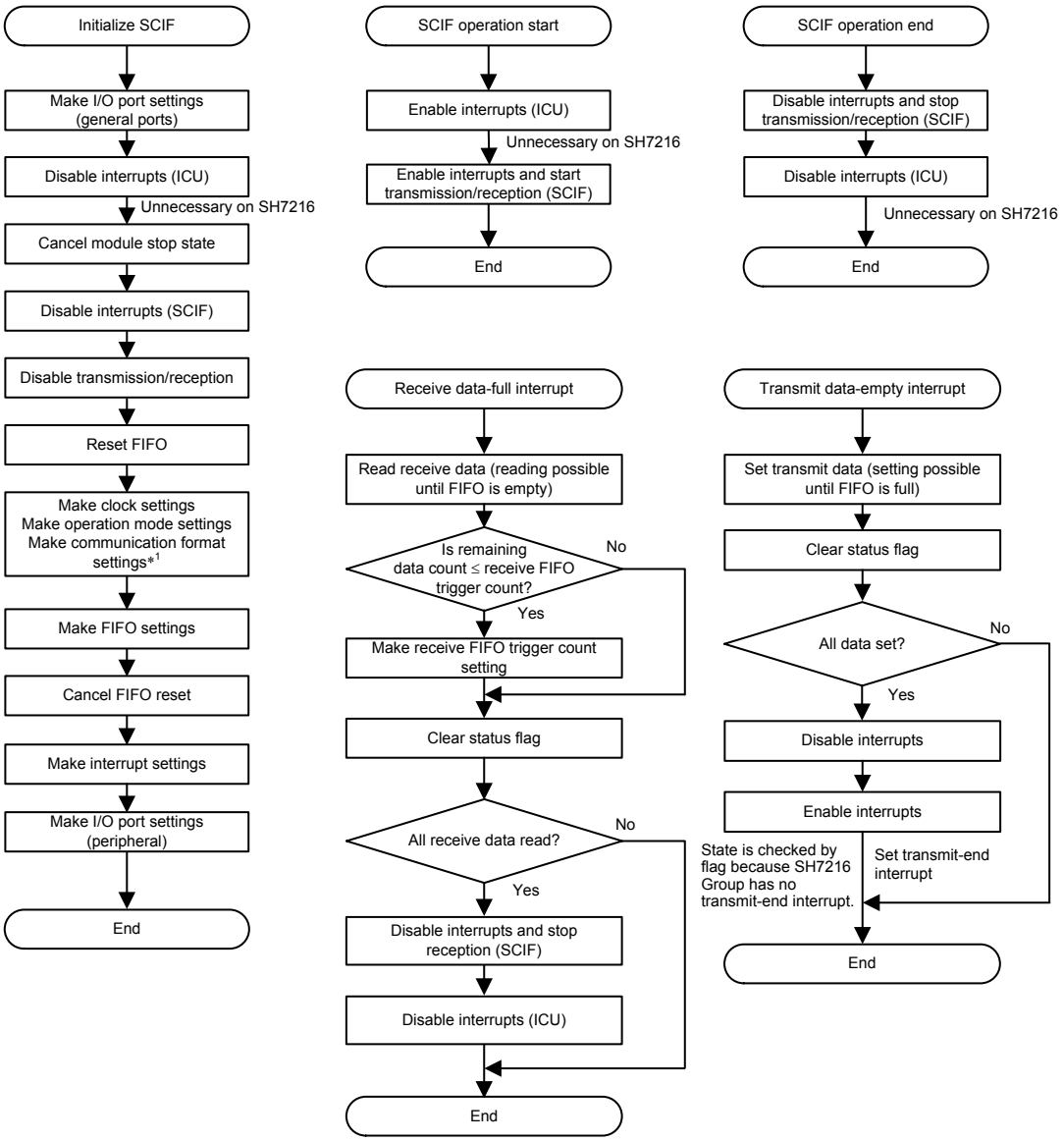
Item	Vector No.	Name	Interrupt Enable		Status		Priority Level
Setting register	—	—	IERm	GENAL0	IRr	GRPAL0	IPRr
Setting position of each source (SCIFA8)							
Transmit-end	112	TEIF8	IER0E.IEN0 (group AL0)	EN0	IR112 (group AL0)	IS0	IPR112 (group AL0)
Break detection		BRIF8		EN2		IS2	
Receive error (overrun)							
Receive error (framing error)		ERIF8		EN1		IS1	
Receive error (parity error)							
Receive data-ready		DRIF8		EN3		IS3	
Receive FIFO data-full	100	RXIF8	IER0C.IEN4	—	IR100	—	IPR100
Transmit FIFO data-empty	101	TXIF8	IER0C.IEN5	—	IR101	—	IPR101
				↑		↑	
				Group AL0 interrupt settings			

When making settings to the ICUA interrupt-related registers on the RX71M, iodefine.h can be used to make settings as follows. GROUPAL0 defines settings for group AL0 interrupts.

- IERm : IEN (SCIFA8 or ICUA interrupt name), IEN (ICU or GROUPAL0)
- IPRr : IPR (SCIFA8 or ICUA interrupt name), IPR (ICU or GROUPAL0)
- IRr : IR (SCIFA8 or ICUA interrupt name), IR (ICU or GROUPAL0)
- GENAL0 : EN (SCIFA8 or ICUA interrupt name)
- GRPAL0 : IS (SCIFA8 or ICUA interrupt name)

Processing Flowcharts

Figure 2.32 shows example flowcharts of processing using the SCIF. The names of the processing steps shown in the flowcharts correspond to the names in the setting examples.



Note 1. Bit rate correction takes place during RX71M asynchronous operation, so it is necessary to wait one bit period after setting the corrected value.

Figure 2.32 Example Flowcharts of SCIF Processing

Setting Examples

Setting examples for asynchronous transmit/receive are presented below. The names of the processing steps in these setting examples correspond to the names in the flowcharts. Refer to the flowcharts for the processing procedure.

Table 2.91 SCIF Initialization Setting Example (Asynchronous Transmit/Receive)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make I/O port settings (general ports).	Output value setting (output 1) PE.DR.B5 = 1b (PE5) Pin direction settings PFC.PEIORL.B5 = 1b (output/PE5) PFC.PEIORL.B6 = 0b (input/PE6) Pin mode settings (port) PFC.PECRL2.PE5MD = 000b (PE5) PFC.PECRL2.PE6MD = 000b (PE6)	Output value setting (output 1) PORTC.PODR.B7 = 1b (PC7) Pin direction settings PORTC.PDR.B7 = 1b (output/PC7) PORTC.PDR.B6 = 0b (input/PC6) Pin mode settings (general) PORTC.PMR.B7 = 0b (PC7) PORTC.PMR.B6 = 0b (PC6)
Disable interrupts (ICU).	—	Disabling of ICU interrupts IEN(SCIFA8, RXIF8) = 0b IEN(SCIFA8, TXIF8) = 0b IEN(ICU, GROUPAL0) = 0b (group AL0) Disabling interrupts in group AL0 EN(SCIFA8, ERIF8) = 0b EN(SCIFA8, BRIF8) = 0b EN(SCIFA8, DRIF8) = 0b EN(SCIFA8, TEIF8) = 0b
Cancel module stop state.	STB.CR4._SCIF3 = 0b	SYSTEM.PRCR = A502h SYSTEM.MSTPCRC.MSTPC27 = 0b SYSTEM.PRCR = A500h
Disable interrupts (SCIF).	Disabling of SCIF interrupts SCIF3.SCSCR.RIE = 0b (RXI3, ERI3, BRI3) SCIF3.SCSCR.REIE = 0b (ERI3, BRI3) SCIF3.SCSCR.TIE = 0b (TXI3)	Disabling of SCIF interrupts SCIFA8.SCR.RIE = 0b (RXIF8, DRIF8, ERIF8, BRIF8) SCIFA8.SCR.REIE = 0b (ERIF8, BRIF8) SCIFA8.SCR.TIE = 0b (TXIF8) SCIFA8.SCR.TEIE = 0b (TEIF8)
Disable transmission/reception.	SCIF3.SCSCR.TE, RE = 0b	SCIFA8.SCR.TE, RE = 0b
Reset FIFO.	SCIF3.SCFCR.TFRST, RFRST = 1b	SCIFA8.FCR.TFRST, RFRST = 1b
Make clock settings. Make operation mode settings. Make communication format settings.	Clock settings SCIF3.SCSCR.CKE = 00b SCIF3.SCSMR.CKS = 00b SCIF3.SCSEMR.ABCS = 0b Operation mode setting SCIF3.SCSMR.CA = 0b (asynchronous) Communication format settings SCIF3.SCSMR._PE = 0b SCIF3.SCSMR.STOP = 0b SCIF3.SCSMR.CHR = 0b	Clock settings SCIFA8.SCR.CKE = 00b SCIFA8.SMR.CKS = 00b SCIFA8.SEMR.ABCS0 = 0b SCIFA8.SEMR.BGDM = 0 (no doubling) Operation mode setting SCIFA8.SMR.CM = 0b (asynchronous) Communication format settings SCIFA8.SMR.PE = 0b SCIFA8.SMR.STOP = 0b SCIFA8.SMR.CHR = 0b SCIFA8.SEMR.DIR = 0b (LSB-first)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make clock settings. Make operation mode settings. Make communication format settings.	Bit rate setting SCIF3.SCBRR = 40 (38,400 bps @50 MHz)	TXD pin settings before transmit start (high output) SCIFA8.SPTR.SPB2IO = 1 SCIFA8.SPTR.SPB2DT = 1 Bit rate settings SCIFA8.SEMR.MDDRS = 0b (BRR register access enabled) SCIFA8.BRR = 97 (38,400 bps@120 MHz)
Make FIFO settings.	FIFO data count trigger settings SCIF3.SCFCR.TTRG = 00b (trigger: 8) SCIF3.SCFCR.RTRG = 10b (trigger: 8)	FIFO threshold settings SCIFA8.FTCR.TTRGS = 0b (FCR register used) SCIFA8.FTCR.RTRGS = 0b (FCR register used) SCIFA8.FCR.TTRG = 00b (threshold: 8) SCIFA8.FCR.RTRG = 10b (threshold: 8)
Cancel FIFO reset.	SCIF3.SCFCR.TFRST, RFRST = 0b	SCIFA8.FCR.TFRST, RFRST = 0b
Make interrupt settings.	Priority setting (level 5) INTC.IPR14._SCIF3 = 5 Clearing of SCIF status flag* SCIF3.SCFSR.BRK = 0b (BRI3) SCIF3.SCFSR.ER = 0b (ERI3) SCIF3.SCFSR.DR = 0b (RXI3) SCIF3.SCFSR.RDF = 0b (RXI3) SCIF3.SCLSR.ORER = 0b (BRI3) * Clear to 0 after reading value as 1.	Priority settings (level 5) IPR(SCIFA8, RXIF8) = 5 IPR(SCIFA8, TXIF8) = 5 IPR(ICU, GROUPAL0) = 5 (group AL0) Clearing of ICU status IR(SCIFA8, RXIF8) = 0b IR(SCIFA8, TXIF8) = 0b IR(ICU, GROUPAL0) = 0b (group AL0) Clearing of SCIF status flag* SCIFA8.FSR.BRK = 0b (BRIF8) SCIFA8.FSR.ER = 0b (ERIF8) SCIFA8.FSR.DR = 0b (DRIF8) SCIFA8.FSR.RDF = 0b (RXIF8) SCIFA8.LSR.ORER = 0b (BRIF8) * Clear to 0 after reading value as 1.
Make I/O port settings (peripheral).	 Pin mode settings PFC.PECRL2.PE5MD = 110b (TXD3) PFC.PECRL2.PE6MD = 110b (RXD3)	Cancellation of register protection MPC.PWPR.B0WI = 0b MPC.PWPR.PFSWE = 1b Pin peripheral function selection MPC.PC7PFS.PSEL = 001010b (TXD8) MPC.PC6PFS.PSEL = 001010b (RXD8) Register protection settings MPC.PWPR.PFSWE = 0b MPC.PWPR.B0WI = 1b Pin mode settings (peripheral) PORTC.PMR.B7 = 1b (TXD8) PORTC.PMR.B6 = 1b (RXD8)

Table 2.92 SCIF Operation Start Setting Example

Processing	SH7216 Group Setting Example	RX71M Setting Example
Enable interrupts (ICU). • Except for transmit-end interrupt		Enabling interrupts in group AL0 EN(SCIFA8, ERIF8) = 1b EN(SCIFA8, BRIF8) = 1b Enabling of ICU interrupts IEN(SCIFA8, RXIF8) = 1b IEN(SCIFA8, TXIF8) = 1b IEN(ICU, GROUPAL0) = 1b (group AL0)
Enable interrupts and start transmit/receive operation (SCIF) • Except for transmit-end interrupt	SCIF3.SCSCR.TE, RE, TIE, RIE = 1b	SCIFA8.SCR.TE, RE, TIE, RIE = 1b

Table 2.93 SCIF Operation End Setting Examples

Processing	SH7216 Group Setting Example	RX71M Setting Example
Disable interrupts and stop transmit/receive operation (SCIF). • All interrupts	SCIF3.SCSCR.TE, RE, TIE, RIE, REIE = 0b	SCIFA8.SCR.TE, RE, TIE, RIE, REIE, TEIE = 0b
Disable interrupts (ICU). • All interrupts	—	Disabling of ICU interrupts IEN(SCIFA8, RXIF8) = 0b IEN(SCIFA8, TXIF8) = 0b IEN(ICU, GROUPAL0) = 0b (group AL0) Disabling interrupts in group AL0 EN(SCIFA8, ERIF8) = 0b EN(SCIFA8, BRIF8) = 0b EN(SCIFA8, DRIF8) = 0b EN(SCIFA8, TEIF8) = 0b

The sample code does not specify any particular interrupt handling. The setting examples below apply to a portion of the processing involved in interrupt handling.

Table 2.94 Setting Examples in SCIF Receive Data-Full Interrupt Handling

Processing	SH7216 Group Setting Example	RX71M Setting Example
Read receive data.	Reading value of SCIF3.SCFRDR The value of the receive data count specified in SCIF3.SCFDR.R can be read repeatedly.	Reading value of SCIFA8.FRDR The value of the receive data count specified in SCIFA8.FDR.R can be read repeatedly.
Make receive FIFO trigger count setting.	SCIF3.SCFCR.RTRG = 00b (trigger: 1)	SCIFA8.FCR.RTRG = 00b (threshold: 1)
Clear status flag.	SCIF3.SCFSR.RDF = 0b (RXI3) Clear to 0 after reading value as 1.	SCIFA8.FSR.RDF = 0b (RXIF8) Clear to 0 after reading value as 1.
Disable interrupts and stop receive operation (SCIF). • Receive data-full interrupt • Receive error interrupt	SCIF3.SCSCR.RE, RIE = 0b	SCIFA8.SCR.RE, RIE = 0b
Disable interrupts (ICU) • Receive data-full interrupt • Receive error interrupt	—	Disabling of ICU interrupts IEN(SCIFA8, RXIF8) = 0b Disabling interrupts in group AL0 EN(SCIFA8, ERIF8) = 0b EN(SCIFA8, BRIF8) = 0b Group AL0 is used by TEIF8, so not necessary to disable.

Table 2.95 Setting Examples in SCIF Transmit Data-Empty Interrupt Handling

Processing	SH7216 Group Setting Example	RX71M Setting Example
Set transmit data.	Setting the value of SCIF3.SCFTDR Transmit data can be set repeatedly until SCIF3.SCFDR.T = 10h.	Setting the value of SCIFA8.FTDR Transmit data can be set repeatedly until SCIFA8.FDR.T = 10h.
Clear status flag.	SCIF3.SCFSR.TDFE = 0b (TXI3) Clear to 0 after reading value as 1.	SCIFA8.FSR.TDFE = 0b (TXIF8) Clear to 0 after reading value as 1.
Disable interrupts • Transmit data-empty interrupt	 Disabling of SCIF interrupts SCIF3.SCSCR.TIE = 0b (TXI3)	Disabling of ICU interrupts IEN(SCIFA8, TXIF8) = 0b Disabling of SCIF interrupts SCIFA8.SCR.TIE = 0b (TXIF8)
Enable interrupts • Transmit end interrupt	— State is checked by flag because there is no transmit-end interrupt.	Enabling of SCIF interrupts SCIFA8.SCR.TEIE = 1b (TEIF8) Enabling interrupts in group AL0 EN(SCIFA8, TEIF8) = 1b Group AL0 is used by ERIF8, so not necessary to enable.

2.11.6 Clock-Synchronous Master Transmit Setting Examples

Setting examples for clock-synchronous master transmit operation using the serial communication interface with FIFO on the SH7216 Group and RX71M are presented below.

Operational Overview

- Clock-synchronous master transmit operation is performed.
- Transmission is activated by a transmit FIFO data-empty interrupt.
- After transmission of all the data finishes, SCIF operation ends.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by "SH7216:" in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.96 Clock-Synchronous Master Transmit Operation Specifications

Item		Description	Remarks
SCIF channel		SCIFA8	SH7216: SCIF3
Clock		PCLKA/4 (PCLKA = 120 MHz)	SH7216: Pϕ = 50 MHz
Communication mode		Clock-synchronous serial communication	
Operating mode		Master transmit	
Transfer speed		100 kbps	
Data length		8 bits (fixed)	
Bit order		LSB-first	
Transmit FIFO data count trigger		8	
Clock source		Internal clock	
Transmit data		128 bytes (value from 1 to 128)	
Noise cancellation		Not used.	
Interrupts		Transmit-end (RX71M only) Transmit FIFO data-empty	Priority: level 5
Pins used	SCK	PC5/SCK8	SH7216: PE4/SCK3
	TXD	PC7/TXD8	SH7216: PE5/TXD3
	LED0	P03/general	Lights when transmission is possible.
	LED1	P05/general	Lights when transmission ends.

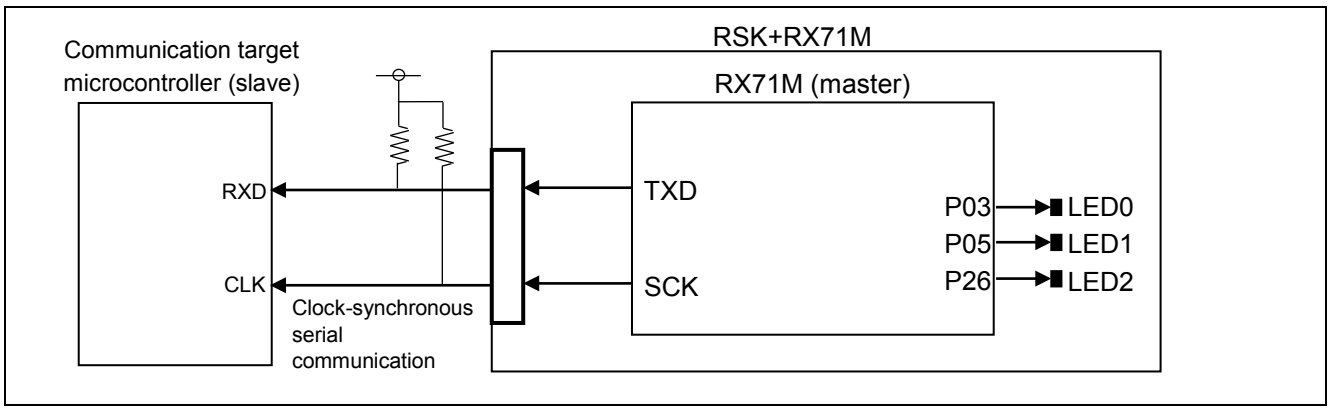


Figure 2.33 Clock-Synchronous Master Transmit Connection Example

Note

On the Renesas Starter Kit+ for RX71M in the initial state the pins used for SCIFA8 in the connection example are connected for use with the external bus, so make appropriate modifications to the board as necessary.

List of Related Registers

For the interrupt-related registers used in the setting examples, refer to the description of registers related to asynchronous operation in 2.11.5, Asynchronous Transmit/Receive Setting Examples.

Processing Flowcharts

For example flowcharts of processing using the SCIF, refer to the example flowcharts of asynchronous processing in 2.11.5, Asynchronous Transmit/Receive Setting Examples. In the clock-synchronous master transmit setting examples only transmit operation takes place, so reception-related processing is not necessary.

Setting Examples

Setting examples for clock-synchronous master transmit are presented below. The names of the processing steps in these setting examples correspond to the names in the flowcharts. Refer to the flowcharts for the processing procedure.

Table 2.97 SCIF Initialization Setting Example (Clock-Synchronous Master Transmit)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make I/O port settings (general ports).	Output value settings (output 1) PE.DR.B4 = 1b (PE4) PE.DR.B5 = 1b (PE5) Pin direction settings PFC.PEIORL.B4 = 1b (output/PE4) PFC.PEIORL.B5 = 1b (output/PE5) Pin mode settings (port) PFC.PECRL2.PE4MD = 000b (PE4) PFC.PECRL2.PE5MD = 000b (PE5)	Output value settings (output 1) PORTC.PODR.B5 = 1b (PC5) PORTC.PODR.B7 = 1b (PC7) Pin direction settings PORTC.PDR.B5 = 1b (output/PC5) PORTC.PDR.B7 = 1b (output/PC7) Pin mode settings (general) PORTC.PMR.B5 = 0b (PC5) PORTC.PMR.B7 = 0b (PC7)
Disable interrupt (ICU).	—	Disabling of ICU interrupts IEN(SCIFA8, RXIF8) = 0b IEN(SCIFA8, TXIF8) = 0b IEN(ICU, GROUPAL0) = 0b (group AL0) Disabling interrupts in group AL0 EN(SCIFA8, ERIF8) = 0b EN(SCIFA8, BRIF8) = 0b EN(SCIFA8, DRIF8) = 0b EN(SCIFA8, TEIF8) = 0b
Cancel module stop state.	STB.CR4._SCIF3 = 0b	SYSTEM.PRCR = A502h SYSTEM.MSTPCRC.MSTPC27 = 0b SYSTEM.PRCR = A500h
Disable interrupt (SCIF).	Disabling of SCIF interrupts SCIF3.SCSCR.RIE = 0b (RXI3, ERI3, BRI3) SCIF3.SCSCR.REIE = 0b (ERI3, BRI3) SCIF3.SCSCR.TIE = 0b (TXI3)	Disabling of SCIF interrupts SCIFA8.SCR.RIE = 0b (RXIF8, DRIF8, ERIF8, BRIF8) SCIFA8.SCR.REIE = 0b (ERIF8, RIF8) SCIFA8.SCR.TIE = 0b (TXIF8) SCIFA8.SCR.TEIE = 0b (TEIF8)
Disable transmission/ reception.	SCIF3.SCSCR.TE, RE = 0b	SCIFA8.SCR.TE, RE = 0b
Reset FIFO.	SCIF3.SCFCR.TFRST, RFRST = 1b	SCIFA8.FCR.TFRST, RFRST = 1b

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make clock settings. Make operation mode settings. Make communication format settings.	Clock settings SCIF3.SCSCR.CKE = 00b SCIF3.SCSMR.CKS = 00b Operation mode setting SCIF3.SCSMR.CA = 1b (clock-synchronous)	Clock settings SCIFA8.SCR.CKE = 00b SCIFA8.SMR.CKS = 01b Operation mode setting SCIFA8.SMR.CM = 1b (clock-synchronous) Communication format settings SCIFA8.SEMR.DIR = 0b (LSB-first) TXD pin settings before transmit start (high output) SCIFA8.SPTR.SPB2IO = 1 SCIFA8.SPTR.SPB2DT = 1
	Bit rate setting SCIF3.SCBRR = 124 (100 kbps @50 MHz)	Bit rate settings SCIFA8.SEMR.MDDRS = 0b (BRR register access enabled) SCIFA8.BRR = 74 (100 kbps @120 MHz)
Make FIFO settings.	FIFO data count trigger setting SCIF3.SCFCR.TTRG = 00b (trigger: 8)	FIFO threshold settings SCIFA8.FTCR.TTRGS = 0b (FCR register used) SCIFA8.FCR.TTRG = 00b (threshold: 8)
Cancel FIFO reset.	SCIF3.SCFCR.TFRST, RFRST = 0b	SCIFA8.FCR.TFRST, RFRST = 0b
Make interrupt settings.	Priority setting (level 5) INTC.IPR14._SCIF3 = 5	Priority settings (level 5) IPR(SCIFA8, TXIF8) = 5 IPR(ICU, GROUPAL0) = 5 (group AL0) Clearing of ICU status IR(SCIFA8, TXIF8) = 0b IR(ICU, GROUPAL0) = 0b (group AL0)
Make I/O port settings (peripheral).		Cancellation of register protection MPC.PWPR.B0WI = 0b MPC.PWPR.PFSWE = 1b Pin peripheral function selection MPC.PC5PFS.PSEL = 001010b (SCK8) MPC.PC7PFS.PSEL = 001010b (TXD8) Register protection settings MPC.PWPR.PFSWE = 0b MPC.PWPR.B0WI = 1b
	Pin mode settings PFC.PECRL2.PE4MD = 110b (SCK3) PFC.PECRL2.PE5MD = 110b (TXD3)	Pin mode settings (peripheral) PORTC.PMR.B5 = 1b (SCK8) PORTC.PMR.B7 = 1b (TXD8)

Refer to the setting examples for asynchronous processing in 2.11.5, Asynchronous Transmit/Receive Setting Examples, except regarding initialization of clock-synchronous master transmit operation. In the clock-synchronous master transmit setting examples only transmit operation takes place, so reception-related processing is not necessary.

2.11.7 Clock-Synchronous Slave Reception Setting Examples

Setting examples for clock-synchronous slave reception using the serial communication interface with FIFO on the SH7216 Group and RX71M are presented below.

Operational Overview

- Clock-synchronous slave reception is performed.
- Reception is activated by a receive FIFO data-full interrupt, and the receive data is stored in the RAM.
- After reception of all the data finishes, SCIF operation ends.
- SCIF operation ends when an error is detected.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by "SH7216:" in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.98 Clock-Synchronous Slave Receive Operation Specifications

Item		Description	Remarks
SCIF channel		SCIFA8	SH7216: SCIF3
Communication mode		Clock-synchronous serial communication	
Operating mode		Slave receive	
Data length		8 bits (fixed)	
Bit order		LSB-first	
Receive FIFO data count trigger		8	
Clock source		External clock	
Noise cancellation		Not used.	
Interrupts		Receive FIFO data-full Receive error Break detection	Priority level: 5
Pins used	SCK	PC5/SCK8	SH7216: PE4/SCK3
	RXD	PC6/RXD8	SH7216: PE6/RXD3
	LED0	P03/general	Lights when reception is possible.
	LED1	P05/general	Lights when reception ends.
	LED2	P26/general	Lights when error detected.

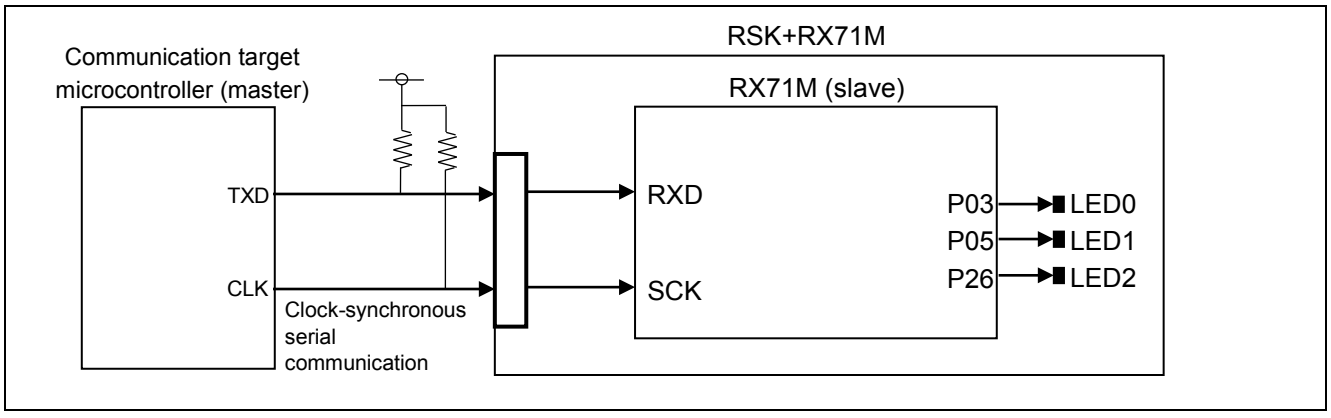


Figure 2.34 Clock-Synchronous Slave Receive Connection Example

Note

On the Renesas Starter Kit+ for RX71M in the initial state the pins used for SCIFA8 in the connection example are connected for use with the external bus, so make appropriate modifications to the board as necessary.

List of Related Registers

For the interrupt-related registers used in the setting examples, refer to the description of registers related to asynchronous operation in 2.11.5, Asynchronous Transmit/Receive Setting Examples.

Processing Flowcharts

For example flowcharts of processing using the SCI, refer to the example flowcharts of asynchronous processing in 2.11.5, Asynchronous Transmit/Receive Setting Examples. In the clock-synchronous slave receive setting examples only receive operation takes place, so transmission-related processing is not necessary.

Setting Examples

Setting examples for clock-synchronous slave receive are presented below. The names of the processing steps in these setting examples correspond to the names in the flowcharts. Refer to the flowcharts for the processing procedure.

Table 2.99 SCIF Initialization Setting Example (Clock-Synchronous Slave Receive)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make I/O port settings (general ports).	Pin direction settings PFC.PEIORL.B4 = 0b (input/PE4) PFC.PEIORL.B6 = 0b (input/PE6) Pin mode settings (port) PFC.PECRL2.PE4MD = 000b (PE4) PFC.PECRL2.PE6MD = 000b (PE6)	Pin direction settings PORTC.PDR.B5 = 0b (input/PC5) PORTC.PDR.B6 = 0b (input/PC6) Pin mode settings (general) PORTC.PMR.B5 = 0b (PC5) PORTC.PMR.B6 = 0b (PC6)
Disable interrupt (ICU).	—	Disabling of ICU interrupts IEN(SCIFA8, RXIF8) = 0b IEN(SCIFA8, TXIF8) = 0b IEN(ICU, GROUPAL0) = 0b (group AL0) Disabling interrupts in group AL0 EN(SCIFA8, ERIF8) = 0b EN(SCIFA8, BRIF8) = 0b EN(SCIFA8, DRIF8) = 0b EN(SCIFA8, TEIF8) = 0b
Cancel module stop state.	STB.CR4._SCIF3 = 0b	SYSTEM.PRCR = A502h SYSTEM.MSTPCRC.MSTPC27 = 0b SYSTEM.PRCR = A500h
Disable interrupt (SCIF).	Disabling of SCIF interrupts SCIF3.SCSCR.RIE = 0b (RXI3, ERI3, BRI3) SCIF3.SCSCR.REIE = 0b (ERI3, BRI3) SCIF3.SCSCR.TIE = 0b (TXI3)	Disabling of SCIF interrupts SCIFA8.SCR.RIE = 0b (RXIF8, DRIF8, ERIF8, BRIF8) SCIFA8.SCR.REIE = 0b (ERIF8, BRIF8) SCIFA8.SCR.TIE = 0b (TXIF8) SCIFA8.SCR.TEIE = 0b (TEIF8)
Disable transmission/ reception.	SCIF3.SCSCR.TE, RE = 0b	SCIFA8.SCR.TE, RE=0b
Reset FIFO.	SCIF3.SCFCR.TFRST, RFRST = 1b	SCIFA8.FCR.TFRST, RFRST = 1b

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make clock settings. Make operation mode settings. Make communication format settings.	Clock settings SCIF3.SCSCR.CKE = 10b SCIF3.SCSMR.CKS = 00b Operation mode setting SCIF3.SCSMR.CA = 1b (clock-synchronous)	Clock settings SCIFA8.SCR.CKE = 10b SCIFA8.SMR.CKS = 00b Operation mode setting SCIFA8.SMR.CM = 1b (clock-synchronous) Communication format setting SCIFA8.SEMR.DIR = 0b (LSB-first)
Make FIFO settings.	Data count trigger setting SCIF3.SCFCR.RTRG = 10b (trigger: 8)	FIFO threshold settings SCIFA8.FTCR.RTRGS = 0b (FCR register used) SCIFA8.FCR.RTRG = 10b (threshold: 8)
Cancel FIFO reset.	SCIF3.SCFCR.TFRST, RFRST = 0b	SCIFA8.FCR.TFRST, RFRST = 0b
Make interrupt settings.	Priority setting (level 5) INTC.IPR14._SCIF3 = 5 Clearing of SCIF status flag* SCIF3.SCFSR.BRK = 0b (BRI3) SCIF3.SCFSR.ER = 0b (ERI3) SCIF3.SCFSR.DR = 0b (RXI3) SCIF3.SCFSR.RDF = 0b (RXI3) SCIF3.SCLSR.ORER = 0b (BRI3) * Clear to 0 after reading value as 1.	Priority settings (level 5) IPR(SCIFA8, RXIF8) = 5 IPR(ICU, GROUPAL0) = 5 (group AL0) Clearing of ICU status IR(SCIFA8, RXIF8) = 0b IR(ICU, GROUPAL0) = 0b (group AL0) Clearing of SCIF status flag* SCIFA8.FSR.BRK = 0b (BRIF8) SCIFA8.FSR.ER = 0b (ERIF8) SCIFA8.FSR.DR = 0b (DRIF8) SCIFA8.FSR.RDF = 0b (RXIF8) SCIFA8.LSR.ORER = 0b (BRIF8) * Clear to 0 after reading value as 1.
Make I/O port settings (peripheral).	 Pin mode settings PFC.PECRL2.PE4MD = 110b (SCK3) PFC.PECRL2.PE6MD = 110b (RXD3)	Cancellation of register protection MPC.PWPR.B0WI = 0b MPC.PWPR.PFSWE = 1b Pin peripheral function selection MPC.PC5PFS.PSEL = 001010b (SCK8) MPC.PC6PFS.PSEL = 001010b (RXD8) Register protection settings MPC.PWPR.PFSWE = 0b MPC.PWPR.B0WI = 1b Pin mode settings (peripheral) PORTC.PMR.B5 = 1b (SCK8) PORTC.PMR.B6 = 1b (RXD8)

Refer to the setting examples for asynchronous processing in 2.11.5, Asynchronous Transmit/Receive Setting Examples, except regarding initialization of clock-synchronous slave receive operation. In the clock-synchronous slave receive setting examples only receive operation takes place, so transmission-related processing is not necessary.

2.12 Serial Peripheral Interface (RSPI)

2.12.1 Comparison of Specifications

Serial peripheral interface functionality is provided on the SH7216 Group by the RSPI and on the RX71M by the RSPIa.

Table 2.100 presents a comparison of the specifications of the SH7216 Group and RX71M.

Table 2.100 Comparison of SH7216 Group and RX71M Specifications (RSPI)

Item	SH7216 Group (RSPI)	RX71M (RSPIa)
Number of channels	1 channel	2 channels
Clock sources	Peripheral clock (Pϕ) External clock (RSPCK)	Peripheral module clock (PCLKA) External clock (RSPCK)
Transmit/receive data length	8 to 16, 20, 24, or 32 bits	
Transfer operation	SPI (4-wire method) Clock-synchronous communication (3-wire method)	
Data format	Selectable between MSB-first and LSB-first	
Clock phase/polarity	Variable	
SSL polarity	Variable	
Operating modes	• Master transmit mode • Master receive mode • Slave transmit mode • Slave receive mode	
Communication operating mode	Full-duplex communication	Selectable between full duplex and transmit only
Multi-master support	Yes	
Sequence control	Sequence length: 4	Sequence length: 8
Loopback mode	Data inverted	Ability to select data inversion
DTC/DMAC activation	DTC/DMAC activation supported	
Interrupt sources	• Transmit buffer-empty • Receive buffer-full • Overrun error • Mode fault error	• Transmit buffer-empty • Receive buffer-full • RSPI idle • Overrun error • Parity error • Mode fault error
Other	—	• Event link • Parity bit addition

2.12.2 Register Comparison

Table 2.101 is a comparative listing of the registers on the SH7216 Group and RX71M.

Guide to Symbols in "Changes" Column of Table

◎: Register with same bit assignments on SH7216 Group and RX71M

△: Register with different bit assignments on SH7216 Group and RX71M

—: Register not present on SH7216 Group or RX71M

Table 2.101 SH7216 Group and RX71M Register Comparison (RSPI)

SH7216 Group (RSPI)	RX71M (RSPIa)*1	Changes
RSPI control register (SPCR)	RSPI control register (RSPIn.SPCR)	◎
RSPI pin control register (SPPCR)*2	RSPI pin control register (RSPIn.SPPCR)	△
RSPI command registers 0 to 3 (SPCMD0 to SPCMD3)	RSPI command registers 0 to 7 (RSPIn.SPCMD0 to RSPIn.SPCMD7)	◎
RSPI bit rate register (SPBR)	RSPI bit rate register (RSPIn.SPBR)	◎
RSPI status register (SPSR)	RSPI status register (RSPIn.SPSR)	△
RSPI data register (SPDR)	RSPI data register (RSPIn.SPDR)	△
RSPI data control register (SPDCR)	RSPI data control register (RSPIn.SPDCR)	◎
RSPI slave select polarity register (SSLP)	RSPI slave select polarity register (RSPIn.SSLP)	◎
RSPI sequence control register (SPSCR)	RSPI sequence control register (RSPIn. SPSCR)	△
RSPI sequence status register (SPSSR)	RSPI sequence status register (RSPIn.SPSSR)	△
SPI slave select negation delay register (SSLND)	RSPI slave select negation delay register (RSPIn.SSLND)	◎
RSPI clock delay register (SPCKD)	RSPI clock delay register (RSPIn.SPCKD)	◎
RSPI next-access delay register (SPND)	RSPI next-access delay register (RSPIn.SPND)	◎
—	RSPI control register 2 (RSPIn.SPCR2)	—

Note 1. RSPIn n: 0 or 1

Note 2. RSPI output pin mode setting is accomplished using the I/O port function on the RX71M.

2.12.3 Interrupts

On both the SH7216 Group and RX71M the receive buffer-full and transmit buffer-empty interrupts can be used to activate the DTC and DMAC.

On the RX71M an interrupt request generated while the receive buffer-full or transmit buffer-empty interrupt status flag (IRn.IR) is set to 1 is stored internally by the module, and after the interrupt status flag (IRn.IR) is cleared to 0 it is once again set to 1 by the stored request.

On the RX71M some interrupts are assigned to group interrupt AL0. The interrupt controller's interrupt status flag (IRn.IR) is cleared automatically when the corresponding interrupt is accepted. The group AL0 interrupt status flag (GRPAL0.ISn) is cleared automatically when the corresponding bit in the module's status register is cleared.

Refer to 1.9, Interrupt Handling for information about interrupts.

2.12.4 Module Stop

The RSPIa of the RX71M, like the SSU of the SH7216 Group, is set to the module-stop state after a reset, and no clock is supplied.

Refer to 2.21, Low Power Consumption Function for information on the module-stop state.

2.12.5 Setting Example for Master Transmission/Reception Using SPI Operation

Setting examples for master transmission and reception through SPI operation (4-wire method) using the serial peripheral interface of the SH7216 Group and RX71M are presented below.

Operational Overview

- The SPI is used to perform master transmission and reception.
- Transmission is activated by a transmit data-empty interrupt.
- Reception is activated by a receive data-full interrupt, receive data is stored in the RAM.
- After transmission and reception of all the data finishes, RSPI operation ends.
- RSPI operation ends when an error is detected.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by "SH7216:" in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.102 SPI Master Transmission and Reception Specifications

Item		Description	Remarks
RSPI channel		RSPI0	SH7216: RSPI
Clock		PCLKA = 120 MHz	SH7216: Pϕ = 50 MHz
Communication mode		SPI operation (4-wire method)	
Operating mode		Full-duplex master transmission and reception	Single-master
Transfer speed		2.5 Mbps (Base bit rate divided by 4)	SH7216: Base bit rate divided by 2
Bit length		8 bits	
Bit order		MSB-first	
RSPCK phase		Data change at odd edges Data sampling at even edges	
RSPCK polarity		RSPCK high in idle state	
RSPCK delay		1RSPCK	
SSL polarity		Active-low	
SSL assert signal		SSLA0	
SSL negate operation		All SSL signals negated when transfer finishes	
SSL negation delay		1 RSPCK	
MOSI value during SSL gate period		Fixed high output	
Next access delay		1 RSPCK + 2 PCLK	
Command count		1	No sequence control
Frame count		1	
Trabsmit/receive data		128 bytes (value from 1 to 128)	
Interrupts		All interrupts used.	Priority level: 5
Pins used	SSL	PC4/SSLA0-A	SH7216: PA2/SSL0
	RSPCK	PC5/RSPCKA-A	SH7216: PA5/RSPCK
	MOSI	PC6/MOSIA-A	SH7216: PA4/MOSI
	MISO	PC7/MISOA-A	SH7216: PA3/MISO
	LED0	P03/general	Lights when transmission/reception is possible.
	LED1	P05/general	Lights when transmission/reception ends.
	LED2	P26/general	Lights when error detected.

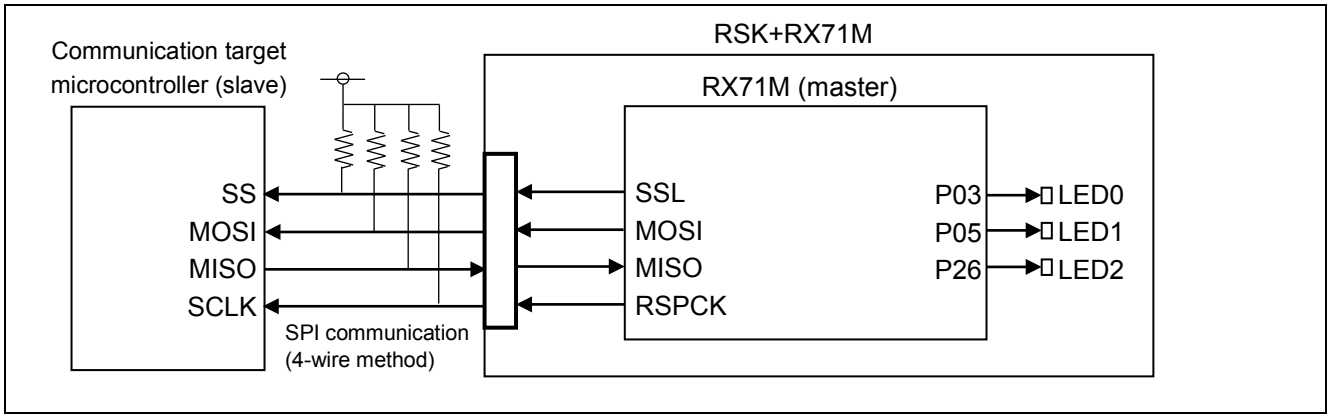


Figure 2.35 SPI Master Transmission and Reception Connection Example

Note

On the Renesas Starter Kit+ for RX71M in the initial state the pins used for RSPI0 in the connection example are connected to the Ethernet-PHY, so make appropriate modifications to the board as necessary.

List of Related Registers

Table 2.103 shows the interrupt-related registers used in the SH7216 Group setting example broken down by source.

Table 2.103 SH7216 Group Interrupt-Related Registers (RSPI and INTC)

			RSPI		INTC
Item	**Vector No.**	**Name**	**Interrupt Enable**	**Status**	**Priority Level**
Setting register	—	—	RSPI.SPCR	RSPI.SPSR	IPR17
Setting position of each source					
Receive buffer-full	234	SPRI	SPRIE	SPRF	Bits 12 to 15
Transmit buffer-empty	235	SPTI	SPTIE	SPTEF	
Overrun	233	SPEI	SPEIE	OVRF	
Mode fault				MODF	
RSPI idle (no interrupt)	—	—	—	MIDLE	—

Table 2.104 and Table 2.105 show the interrupt-related registers used in the RX71M setting example broken down by source. On the RX71M some RSPIa-related interrupts are assigned as group AL0 interrupts.

Table 2.104 RX71M Interrupt-Related Registers (RSPIa)

Item	**Name**	**Interrupt Enable**	**Status**
Setting register	—	RSPI0.SPCR	RSPI0.SPSR
Setting position of each source (RSPI0)			
Receive buffer-full	SPRI	RSPI0.SPCR.SPRIE	SPRF
Transmit buffer-empty	SPTI	RSPI0.SPCR.SPTIE	SPTEF
Overrun	SPEI	RSPI0.SPCR.SPEIE	OVRF
Mode fault			MODF
Parity error			PERF
RSPI idle	SPII	RSPI0.SPCR2.SPIIE (Register differs.)	IDLNF

Table 2.105 RX71M Interrupt-Related Registers (ICUA)

Item	Vector No.	Name	Interrupt Enable		Status		Priority Level
Setting register	—	—	IERm	GENAL0	IRr	GRPAL0	IPRr
Setting position of each source (RSPI0)							
Receive buffer-full	38	SPRI0	IER04.IEN6	—	IR038	—	IPR038
Transmit buffer-empty	39	SPTI0	IER04.IEN7	—	IR039	—	IPR039
Overrun	112	SPEI0	IER0E.IEN0 (group AL0)	EN17	IR112 (group AL0)	IS17	IPR112 (group AL0)
Mode fault							
Parity error							
RSPI idle		SPII0		EN16		IS16	
				↑		↑	
				Group AL0 interrupt settings			

When making settings to the ICUA interrupt-related registers on the RX71M, iodefine.h can be used to make settings as follows. GROUPAL0 defines settings for group AL0 interrupts.

- IERm : IEN (RSPI0 or ICUA interrupt name), IEN (ICU or GROUPAL0)
- IPRr : IPR (RSPI0 or ICUA interrupt name), IPR (ICU or GROUPAL0)
- IRr : IR (RSPI0 or ICUA interrupt name), IR (ICU or GROUPAL0)
- GENAL0 : EN (RSPI0 or ICUA interrupt name)
- GRPAL0 : IS (RSPI0 or ICUA interrupt name)

Processing Flowcharts

Figure 2.36 shows example flowcharts of processing using the RSPI. The names of the processing steps shown in the flowcharts correspond to the names in the setting examples.

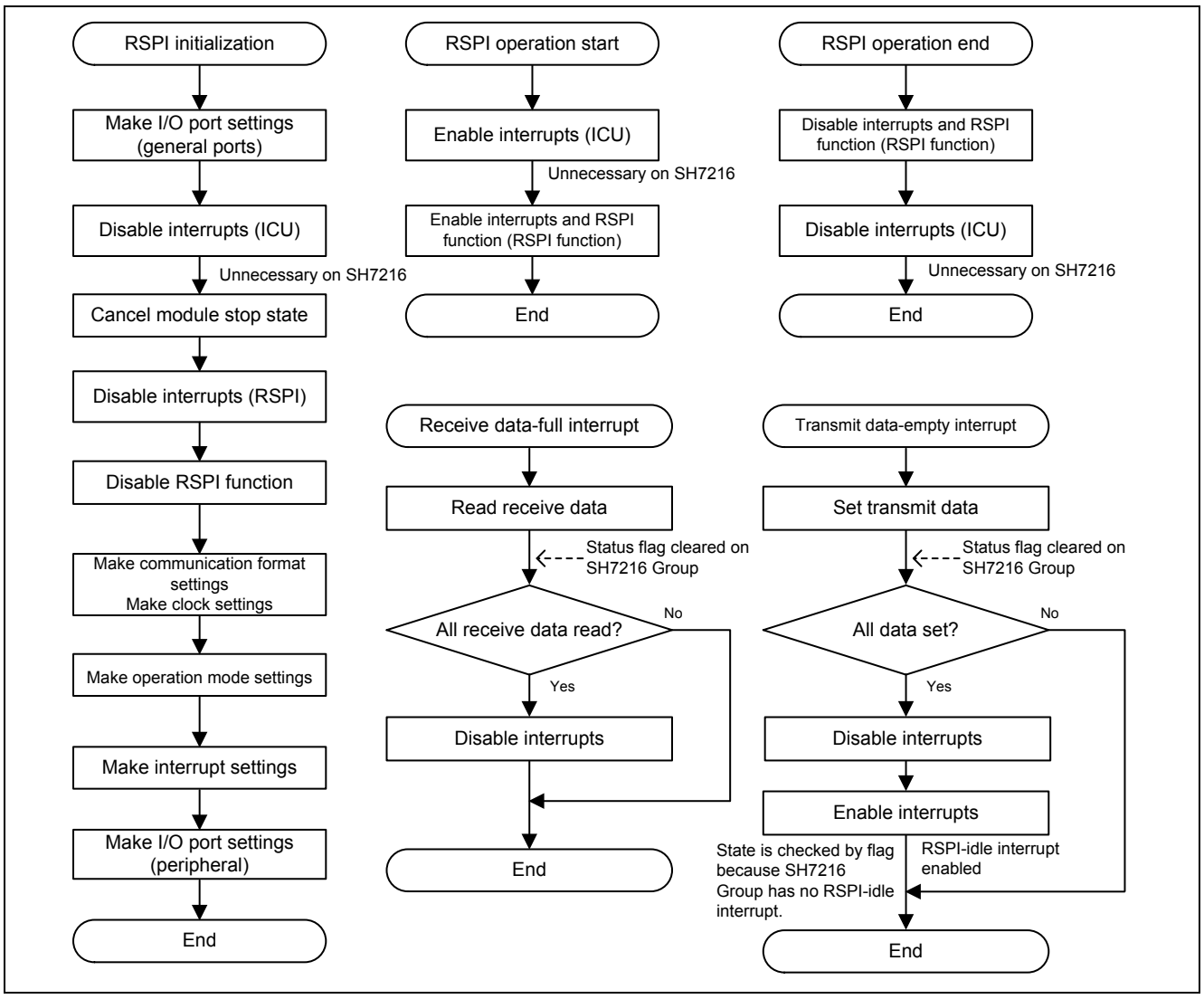


Figure 2.36 Example Flowcharts of RSPI Processing

Setting Examples

Setting examples for SPI master transmission and reception are presented below. The names of the processing steps in these setting examples correspond to the names in the flowcharts. Refer to the flowcharts for the processing procedure.

Table 2.106 RSPI Initialization Setting Examples (SPI Transmission and Reception)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make I/O port settings (general ports).	Output value settings (output 1) PA.DR.B2 = 1b (PA2) PA.DR.B5 = 1b (PA5) PA.DR.B4 = 1b (PA4) Pin direction settings PFC.PAIORL.B2 = 1b (output/PA2) PFC.PAIORL.B5 = 1b (output/PA5) PFC.PAIORL.B4 = 1b (output/PA4) PFC.PAIORL.B3 = 0b (input/PA3) Pin mode settings (port) PFC.PACRL1.PA2MD = 000b (PA2) PFC.PACRL2.PA5MD = 000b (PA5) PFC.PACRL2.PA4MD = 000b (PA4) PFC.PACRL1.PA3MD = 000b (PA3)	Output value settings (output 1) PORTC.PODR.B4 = 1b (PC4) PORTC.PODR.B5 = 1b (PC5) PORTC.PODR.B6 = 1b (PC6) Pin direction settings PORTC.PDR.B4 = 1b (output/PC4) PORTC.PDR.B5 = 1b (output/PC5) PORTC.PDR.B6 = 1b (output/PC6) PORTC.PDR.B7 = 0b (input/PC7) Pin mode settings (general) PORTC.PMR.B4 = 0b (PC4) PORTC.PMR.B5 = 0b (PC5) PORTC.PMR.B6 = 0b (PC6) PORTC.PMR.B7 = 0b (PC7)
Disable interrupts (ICU).	—	Disabling of ICU interrupts IEN(RSPI0, SPRI0) = 0b IEN(RSPI0, SPTI0) = 0b IEN(ICU, GROUPAL0) = 0b (group AL0) Disabling interrupts in group AL0 EN(RSPI0, SPII0) = 0b EN(RSPI0, SPEI0) = 0b
Cancel module stop state.	STB.CR5._RSPI = 0b	SYSTEM.PRCR = A502h SYSTEM.MSTPCRB.MSTPB17 = 0b SYSTEM.PRCR = A500h
Disable interrupts (RSPI).	Disabling of RSPI interrupts RSPI.SPCR.SPRIE = 0b (SPRI) RSPI.SPCR.SPTIE = 0b (SPTI) RSPI.SPCR.SPEIE = 0b (SPEI) RSPI.SPCR.MODFEN = 0b (for MODF)	Disabling of RSPI interrupts RSPI0.SPCR.SPRIE = 0b (SPRI0) RSPI0.SPCR.SPTIE = 0b (SPTI0) RSPI0.SPCR.SPEIE = 0b (SPEI0) RSPI0.SPCR2.SPIIE = 0b (SPII0) RSPI0.SPCR.MODFEN = 0b (for MODF)
Disable RSPI function.	RSPI.SPCR.SPE = 0b	RSPI0.SPCR.SPE = 0b
Make communication format settings.	SSL polarity setting RSPI.SSLP.SSL0P = 0b RSPI pin control RSPI.SPPCR.SPOM = 0b (CMOS output) RSPI.SPPCR.MOIFV = 1b RSPI.SPPCR.MOIFE = 1b Clock settings (2.5 Mbps @50 MHz/20) RSPI.SPCMD0.BRDV = 01b RSPI.SPBR = 4 Frame count and access size settings RSPI.SPDCR.SPFC = 00b RSPI.SPDCR.SPLW = 0b	SSL polarity setting RSPI0.SSLP.SSL0P = 0b RSPI pin control RSPI0.SPPCR.MOIFV = 1b RSPI0.SPPCR.MOIFE = 1b Clock settings (2.5 Mbps @120 MHz/48) RSPI0.SPCMD0.BRDV = 10b RSPI0.SPBR = 5 Frame count and access size settings RSPI0.SPDCR.SPFC = 00b RSPI0.SPDCR.SPLW = 0b

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make communication format settings.		Parity setting RSPI0.SPCR2.SPPE = 0b (no parity)
	Sequence setting RSPI.SPSCR.SPSLN = 00b	Sequence setting RSPI0.SPSCR.SPSLN = 000b
	Command settings RSPI.SPCMD0.SPB = 0111b RSPI.SPCMD0.LSBF = 0b RSPI.SPCMD0.CPHA = 1b RSPI.SPCMD0.CPOL = 1b RSPI.SPCMD0.SSLA = 000b RSPI.SPCMD0.SSLKP = 0b RSPI.SPCMD0.SPNDEN = 0b RSPI.SPCMD0.SLNDEN = 0b RSPI.SPCMD0.SCKDEN = 0b	Command settings RSPI0.SPCMD0.SPB = 0111b RSPI0.SPCMD0.LSBF = 0b RSPI0.SPCMD0.CPHA = 1b RSPI0.SPCMD0.CPOL = 1b RSPI0.SPCMD0.SSLA = 000b RSPI0.SPCMD0.SSLKP = 0b RSPI0.SPCMD0.SPNDEN = 0b RSPI0.SPCMD0.SLNDEN = 0b RSPI0.SPCMD0.SCKDEN = 0b
Make operation mode settings.	RSPI.SPCR.SPMS = 0b RSPI.SPCR.MSTR = 1b	RSPI0.SPCR.SPMS = 0b RSPI0.SPCR.MSTR = 1b RSPI0.SPCR.TXMD = 0b (full-duplex) Dummy read of RSPI0.SPCR value
Make interrupt settings.	Priority setting (level 5) INTC.IPR17._RSPI = 5	Priority settings (level 5) IPR(RSPI0, SPRI0) = 5 IPR(RSPI0, SPTI0) = 5 IPR(ICU, GROUPAL0) = 5(group AL0) Clearing of SCIF status flag IR(RSPI0, SPRI0) = 0b IR(RSPI0, SPTI0) = 0b IR(ICU, GROUPAL0) = 0b (group AL0)
	Clearing of RSPI status flag* RSPI.SPSR.MODF = 0b (SPEI) RSPI.SPSR.OVRF = 0b (SPEI) RSPI.SPSR.SPRF = 0b (SPRI) No need to clear the RSPI status flag because it is cleared when RSPI operation ends. However, SPRF, MODF, and OVRF are not initialized. * Clear to 0 after reading value as 1.	Clearing of RSPI status flag* RSPI0.SPSR.MODF = 0b (SPEI0) RSPI0.SPSR.OVRF = 0b (SPEI0) No need to clear the RSPI status flag because it is cleared when RSPI operation ends. However, MODF and OVRF are not initialized. * Clear to 0 after reading value as 1.
Make I/O port settings (peripheral).		Cancellation of register protection MPC.PWPR.B0WI = 0b MPC.PWPR.PFSWE = 1b Pin peripheral function selection MPC.PC4PFS.PSEL = 001101b (SSL) MPC.PC5PFS.PSEL = 001101b (RSPCK) MPC.PC6PFS.PSEL = 001101b (MOSI) MPC.PC7PFS.PSEL = 001101b (MISO) Register protection settings MPC.PWPR.PFSWE = 0b MPC.PWPR.B0WI = 1b
	Pin mode settings PFC.PACRL1.PA2MD = 101b (SSL) PFC.PACRL2.PA5MD = 101b (RSPCK) PFC.PACRL2.PA4MD = 101b (MOSI) PFC.PACRL1.PA3MD = 101b (MISO)	Pin mode settings (peripheral) PORTC.PMR.B4 = 1b (SSL) PORTC.PMR.B5 = 1b (RSPCK) PORTC.PMR.B6 = 1b (MOSI) PORTC.PMR.B7 = 1b (MISO)

Table 2.107 RSPI Operation Start Setting Example

Processing	SH7216 Group Setting Example	RX71M Setting Example
Enable interrupts (ICU). • Except for RSPI-idle interrupt	—	Enabling interrupts in group AL0 EN(RSPI0, SPEI0) = 1b Enabling of ICU interrupts IEN(RSPI0, SPRI0) = 1b IEN(RSPI0, SPTI0) = 1b IEN(ICU, GROUPAL0) = 1b (group AL0)
Enable interrupts and RSPI function (RSPI function). • Except for RSPI-idle interrupt	RSPI.SPCR.SPE, SPTIE, SPRIE, SPEIE = 1b	RSPI0.SPCR.SPE, SPTIE, SPRIE, SPEIE = 1b

Table 2.108 RSPI Operation End Setting Examples

Processing	SH7216 Group Setting Example	RX71M Setting Example
Disable interrupts and RSPI function (RSPI function). • All interrupts	RSPI.SPCR.SPE, SPTIE, SPRIE, SPEIE = 0b	RSPI0.SPCR.SPE, SPTIE, SPRIE, SPEIE = 0b RSPI0.SPCR2.SPIIE = 0b
Disable interrupts (ICU). • All interrupts	—	Disabling of ICU interrupts IEN(RSPI0, SPRI0) = 0b IEN(RSPI0, SPTI0) = 0b IEN(ICU, GROUPAL0) = 0b (group AL0) Disabling interrupts in group AL0 EN(RSPI0, SPII0) = 0b EN(RSPI0, SPEI0) = 0b

The sample code does not specify any particular interrupt handling. The setting examples below apply to a portion of the processing involved in interrupt handling.

Table 2.109 Setting Examples in RSPI Receive Data-Full Interrupt Handling

Processing	SH7216 Group Setting Example	RX71M Setting Example
Read receive data.	Reading value of RSPI.SPDR	Reading value of RSPI0.SPDR
Clear status flag.	RSPI.SPSR.SPRF = 0b Clear to 0 after reading value as 1.	—
Disable interrupts. • Receive data-full interrupt • Receive error interrupt	Disabling of RSPI interrupts RSPI.SPCR.SPRIE = 0b (SPRI) RSPI.SPCR.SPEIE = 0b (SPEI)	Disabling of ICU interrupts IEN(RSPI0, SPRI0) = 0b Disabling interrupts in group AL0 EN(RSPI0, SPEI0) = 0b Disabling of RSPI interrupts RSPI0.SPCR.SPRIE = 0b RSPI0.SPCR.SPEIE = 0b Group AL0 is used by SPII0, so not necessary to disable.

Table 2.110 Setting Examples in RSPI Transmit Data-Empty Interrupt Handling

Processing	SH7216 Group Setting Example	RX71M Setting Example
Set transmit data.	Setting the value of RSPI.SPDR	Setting the value of RSPI0.SPDR
Clear status flag.	RSPI.SPSR.SPTEF = 0b Clear to 0 after reading value as 1.	—
Disable interrupts. • Transmit data-empty interrupt	Disabling of RSPI interrupts RSPI.SPCR.SPTIE = 0b	Disabling of ICU interrupts IEN(RSPI0, SPTI0) = 0b Disabling of RSPI interrupts RSPI0.SPCR.SPTIE = 0b
Enable interrupts. • RSPI idle interrupt	— There is no RSPI-idle interrupt, so the flag state is checked.	Enabling of RSPI interrupts RSPI0.SPCR2.SPIIE = 1b (SPII0) Enabling interrupts in group AL0 EN(RSPI0, SPII0) = 1b Group AL0 is used by SPEI0, so not necessary to enable.

2.12.6 Clock-Synchronous Master Transmission Setting Example

Setting examples for clock-synchronous (3-wire method) master transmission using the serial peripheral interface of the SH7216 Group and RX71M are presented below.

Operational Overview

- Clock-synchronous master transmission is performed.
- Transmission is activated by a transmit data-empty interrupt.
- After transmission of all the data finishes, RSPI operation ends.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by "SH7216:" in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.111 Clock-Synchronous Master Transmission Specifications

Item		Description	Remarks
RSPI channel		RSPI0	SH7216: RSPI
Clock		PCLKA = 120 MHz	SH7216: Pϕ = 50 MHz
Communication mode		Clock-synchronous operation (3-wire method)	
Operating mode		Transmit operation only	
Transfer speed		2.5 Mbps	Base bit rate dividing ratios RX71M: Division by 4 SH7216: Division by 2
Data length		8 bits	
Bit order		MSB-first	
RSPCK phase		Data change at odd edges Data sampling at even edges	
RSPCK polarity		RSPCK high in idle state	
RSPCK delay		1 RSPCK	
Next access delay		1 RSPCK + 2 PCLK	
Command count		1	No sequence control
Frame count		1	
Transmit data		128 bytes (value from 1 to 128)	
Interrupts		All interrupts used.	Priority level: 5
Pins used	RSPCK	PC5/RSPCKA-A	SH7216: PA5/RSPCK
	MOSI	PC6/MOSIA-A	SH7216: PA4/MOSI
	LED0	P03/general	Lights when transmission is possible.
	LED1	P05/general	Lights when transmission ends.

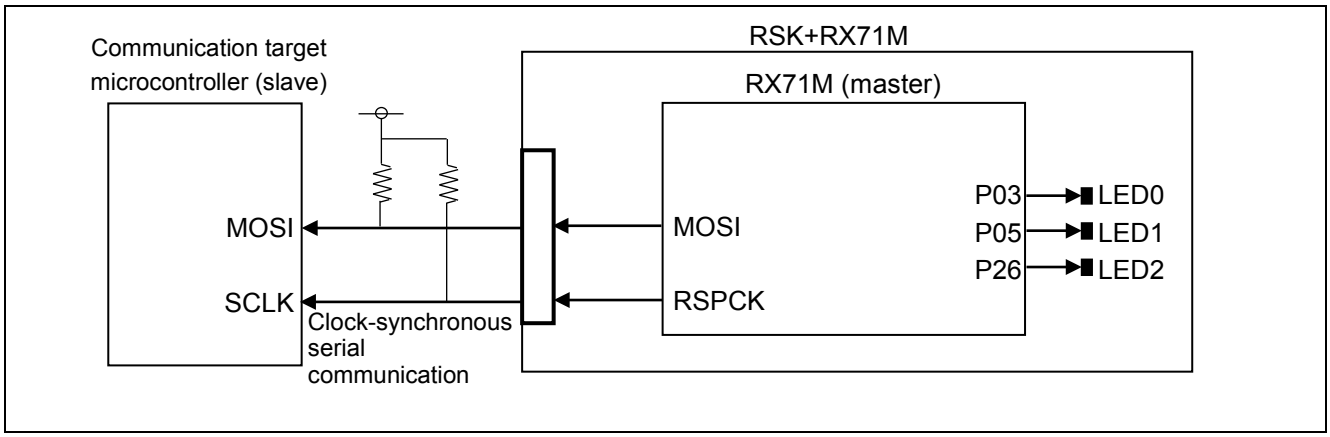


Figure 2.37 Clock-Synchronous Master Transmission Connection Example

Note

On the Renesas Starter Kit+ for RX71M in the initial state the pins used for RSPI0 in the connection example are connected to the Ethernet-PHY, so make appropriate modifications to the board as necessary.

List of Related Registers

For details of the interrupt-related registers used in the setting examples, refer to the list of registers related to SPI operation in 2.12.5, Setting Example for Master Transmission/Reception Using SPI Operation. The clock-synchronous master transmission setting examples cover transmission only, so processing related to reception is unnecessary.

Processing Flowcharts

For example flowcharts of processing using the RSPI, refer to the example flowcharts of SPI operation in 2.12.5, Setting Example for Master Transmission/Reception Using SPI Operation. The clock-synchronous master transmission setting examples cover transmission only, so processing related to reception is unnecessary.

Setting Examples

Setting examples for clock-synchronous master transmission are presented below. The names of the processing steps in these setting examples correspond to the names in the flowcharts. Refer to the flowcharts for the processing procedure.

Table 2.112 RSPI Initialization Setting Examples (Clock-Synchronous Master Transmission)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make I/O port settings (general ports).	Output value settings (output 1) PA.DR.B5 = 1b (PA5) PA.DR.B4 = 1b (PA4) Pin direction settings PFC.PAIORL.B5 = 1b (output/PA5) PFC.PAIORL.B4 = 1b (output/PA4) Pin mode settings (port) PFC.PACRL2.PA5MD = 000b (PA5) PFC.PACRL2.PA4MD = 000b (PA4)	Output value settings (output 1) PORTC.PODR.B5 = 1b (PC5) PORTC.PODR.B6 = 1b (PC6) Pin direction settings PORTC.PDR.B5 = 1b (output/PC5) PORTC.PDR.B6 = 1b (output/PC6) Pin mode settings (general) PORTC.PMR.B5 = 0b (PC5) PORTC.PMR.B6 = 0b (PC6)
Disable interrupts (ICU).	—	Disabling of ICU interrupts IEN(RSPI0, SPRI0) = 0b IEN(RSPI0, SPTI0) = 0b IEN(ICU, GROUPAL0) = 0b (group AL0) Disabling interrupts in group AL0 EN(RSPI0, SPII0) = 0b EN(RSPI0, SPEI0) = 0b
Cancel module stop state.	STB.CR5._RSPI = 0b	SYSTEM.PRCR = A502h SYSTEM.MSTPCRB.MSTPB17 = 0b SYSTEM.PRCR = A500h
Disable interrupts (RSPI).	Disabling of RSPI interrupts RSPI.SPCR.SPRIE = 0b (SPRI) RSPI.SPCR.SPTIE = 0b (SPTI) RSPI.SPCR.SPEIE = 0b (SPEI) RSPI.SPCR.MODFEN = 0b (for MODF)	Disabling of RSPI interrupts RSPI0.SPCR.SPRIE = 0b (SPRI0) RSPI0.SPCR.SPTIE = 0b (SPTI0) RSPI0.SPCR.SPEIE = 0b (SPEI0) RSPI0.SPCR2.SPIIE = 0b (SPII0) RSPI0.SPCR.MODFEN = 0b (for MODF)
Disable RSPI function.	RSPI.SPCR.SPE = 0b	RSPI0.SPCR.SPE = 0b
Make communication format settings.	RSPI pin control RSPI.SPPCR.SPOM = 0b (CMOS output) RSPI.SPPCR.MOIFV = 1b RSPI.SPPCR.MOIFE = 1b Clock settings (2.5 Mbps @50 MHz/20) RSPI.SPCMD0.BRDV = 01b RSPI.SPBR = 4 Frame count and access size settings RSPI.SPDCR.SPFC = 00b RSPI.SPDCR.SPLW = 0b Sequence setting RSPI.SPSCR.SPSLN = 00b	RSPI pin control RSPI0.SPPCR.MOIFV = 1b RSPI0.SPPCR.MOIFE = 1b Clock settings (2.5 Mbps @120 MHz/48) RSPI0.SPCMD0.BRDV = 10b RSPI0.SPBR = 5 Frame count and access size settings RSPI0.SPDCR.SPFC = 00b RSPI0.SPDCR.SPLW = 0b Parity setting RSPI0.SPCR2.SPPE = 0b (no parity) Sequence setting RSPI0.SPSCR.SPSLN = 000b

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make communication format settings.	Command settings RSPI.SPCMD0.SPB = 0111b RSPI.SPCMD0.LSBF = 0b RSPI.SPCMD0.CPHA = 1b RSPI.SPCMD0.CPOL = 1b RSPI.SPCMD0.SPNDEN = 0b RSPI.SPCMD0.SCKDEN = 0b	Command settings RSPI0.SPCMD0.SPB = 0111b RSPI0.SPCMD0.LSBF = 0b RSPI0.SPCMD0.CPHA = 1b RSPI0.SPCMD0.CPOL = 1b RSPI0.SPCMD0.SPNDEN = 0b RSPI0.SPCMD0.SCKDEN = 0b
Make operation mode settings.	RSPI.SPCR.SPMS = 1b RSPI.SPCR.MSTR = 1b	RSPI0.SPCR.SPMS = 1b RSPI0.SPCR.MSTR = 1b RSPI0.SPCR.TXMD = 1b (transmit only) Dummy read of RSPI0.SPCR value
Make interrupt settings.	Priority setting (level 5) INTC.IPR17._RSPI = 5 Clearing of RSPI status flag* RSPI.SPSR.SPRF = 0b (SPRI) No need to clear the RSPI status flag because it is cleared when RSPI operation ends. However, SPRF is not initialized. * Clear to 0 after reading value as 1.	Priority settings (level 5) IPR(RSPI0, SPTI0) = 5 IPR(ICU, GROUPAL0) = 5 (group AL0) Clearing of SCIF status flag IR(RSPI0, SPTI0) = 0b IR(ICU, GROUPAL0) = 0b (group AL0)
Make I/O port settings (peripheral).	 Pin mode settings PFC.PACRL2.PA5MD = 101b (RSPCK) PFC.PACRL2.PA4MD = 101b (MOSI)	Cancellation of register protection MPC.PWPR.B0WI = 0b MPC.PWPR.PFSWE = 1b Pin peripheral function selection MPC.PC5PFS.PSEL = 001101b (RSPCK) MPC.PC6PFS.PSEL = 001101b (MOSI) Register protection settings MPC.PWPR.PFSWE = 0b MPC.PWPR.B0WI = 1b Pin mode settings (peripheral) PORTC.PMR.B5 = 1b (RSPCK) PORTC.PMR.B6 = 1b (MOSI)

For setting examples for SPI operation other than clock-synchronous master transmission, refer to the setting examples in 2.12.5, Setting Example for Master Transmission/Reception Using SPI Operation. The clock-synchronous master transmission setting examples cover transmission only, so processing related to reception is unnecessary.

2.12.7 Clock-Synchronous Slave Reception Setting Example

Setting examples for clock-synchronous (3-wire method) slave reception using the serial peripheral interface of the SH7216 Group and RX71M are presented below.

Operational Overview

- Clock-synchronous slave reception is performed.
- Reception is activated by a receive data-full interrupt, receive data is stored in the RAM.
- After reception of all the data finishes, RSPI operation ends.
- RSPI operation ends when an error is detected.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by "SH7216:" in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.113 Clock-Synchronous Slave Reception Specifications

Item		Description	Remarks
RSPI channel		RSPI0	SH7216: RSPI
Clock		PCLKA = 120 MHz	SH7216: Pϕ = 50 MHz
Communication mode		Clock-synchronous operation (3-wire method)	
Operating mode		Full-duplex slave receive dummy reception is performed.	
Data length		8 bits	
Bit order		MSB-first	
RSPCK phase		Data change at odd edges Data sampling at even edges	
RSPCK polarity		RSPCK high in idle state	
Frame count		1	
Receive data		128 bytes	
Interrupts		All interrupts used.	Priority: level 5
Pins used	RSPCK	PC5/RSPCKA-A	SH7216: PA5/RSPCK
	MOSI	PC6/MOSIA-A	SH7216: PA4/MOSI
	LED0	P03/general	Lights when reception is possible.
	LED1	P05/general	Lights when reception ends.
	LED2	P73/general	Lights when error detected.

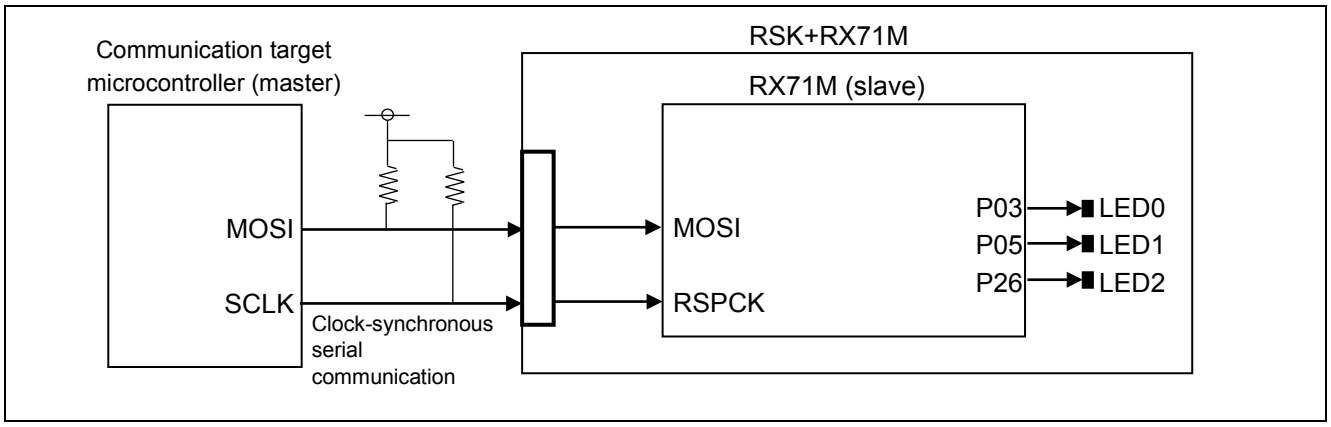


Figure 2.38 Clock-Synchronous Slave Reception Connection Example

Note

On the Renesas Starter Kit+ for RX71M in the initial state the pins used for RSPI0 in the connection example are connected to the Ethernet-PHY, so make appropriate modifications to the board as necessary.

List of Related Registers

For details of the interrupt-related registers used in the setting examples, refer to the list of registers related to SPI operation in 2.12.5, Setting Example for Master Transmission/Reception Using SPI Operation. Full-duplex operation is used in the clock-synchronous slave reception setting examples, so dummy reception processing is necessary.

Processing Flowcharts

For example flowcharts of processing using the RSPI, refer to the example flowcharts of SPI operation in 2.12.5, Setting Example for Master Transmission/Reception Using SPI Operation. Full-duplex operation is used in the clock-synchronous slave reception setting examples, so dummy reception processing is necessary.

Setting Examples

Setting examples for clock-synchronous slave reception are presented below. The names of the processing steps in these setting examples correspond to the names in the flowcharts. Refer to the flowcharts for the processing procedure.

Table 2.114 RSPI Initialization Setting Examples (Clock-Synchronous Slave Reception)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make I/O port settings (general ports).	Pin direction settings PFC.PAIORL.B5 = 0b (input/PA5) PFC.PAIORL.B4 = 0b (input/PA4) Pin mode settings (port) PFC.PACRL2.PA5MD = 000b (PA5) PFC.PACRL2.PA4MD = 000b (PA4)	Pin direction settings PORTC.PDR.B5 = 0b (input/PC5) PORTC.PDR.B6 = 0b (input/PC6) Pin mode settings (general) PORTC.PMR.B5 = 0b (PC5) PORTC.PMR.B6 = 0b (PC6)
Disable interrupts (ICU).	—	Disabling of ICU interrupts IEN(RSPI0, SPRI0) = 0b IEN(RSPI0, SPTI0) = 0b IEN(ICU, GROUPAL0) = 0b (group AL0) Disabling interrupts in group AL0 EN(RSPI0, SPII0) = 0b EN(RSPI0, SPEI0) = 0b
Cancel module stop state.	STB.CR5._RSPI = 0b	SYSTEM.PRCR = A502h SYSTEM.MSTPCRB.MSTPB17 = 0b SYSTEM.PRCR = A500h
Disable interrupts (RSPI).	Disabling of RSPI interrupts RSPI.SPCR.SPRIE = 0b (SPRI) RSPI.SPCR.SPTIE = 0b (SPTI) RSPI.SPCR.SPEIE = 0b (SPEI) RSPI.SPCR.MODFEN = 0b (for MODF)	Disabling of RSPI interrupts RSPI0.SPCR.SPRIE = 0b (SPRI0) RSPI0.SPCR.SPTIE = 0b (SPTI0) RSPI0.SPCR.SPEIE = 0b (SPEI0) RSPI0.SPCR2.SPIIE = 0b (SPII0) RSPI0.SPCR.MODFEN = 0b (for MODF)
Disable RSPI function.	RSPI.SPCR.SPE = 0b	RSPI0.SPCR.SPE = 0b
Make communication format settings.	Frame count and access size settings RSPI.SPDCR.SPFC = 00b RSPI.SPDCR.SPLW = 0b Sequence setting RSPI.SPSCR.SPSLN = 00b Command settings RSPI.SPCMD0.SPB = 0111b RSPI.SPCMD0.LSBF = 0b RSPI.SPCMD0.CPHA = 1b RSPI.SPCMD0.CPOL = 1b RSPI.SPCMD0.SPNDEN = 0b RSPI.SPCMD0.SCKDEN = 0b	Frame count and access size settings RSPI0.SPDCR.SPFC = 00b RSPI0.SPDCR.SPLW = 0b Parity setting RSPI0.SPCR2.SPPE = 0b (no parity) Sequence setting RSPI0.SPSCR.SPSLN = 000b Command settings RSPI0.SPCMD0.SPB = 0111b RSPI0.SPCMD0.LSBF = 0b RSPI0.SPCMD0.CPHA = 1b RSPI0.SPCMD0.CPOL = 1b RSPI0.SPCMD0.SPNDEN = 0b RSPI0.SPCMD0.SCKDEN = 0b
Make operation mode settings.	RSPI.SPCR.SPMS = 1b RSPI.SPCR.MSTR = 0b	RSPI0.SPCR.SPMS = 1b RSPI0.SPCR.MSTR = 0b RSPI0.SPCR.TXMD = 0b (full-duplex operation) Dummy read of RSPI0.SPCR value

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make interrupt settings.	Priority setting (level 5) INTC.IPR17._RSPI = 5	Priority settings (level 5) IPR(RSPI0, SPRI0) = 5 IPR(RSPI0, SPTI0) = 5 IPR(ICU, GROUPAL0) = 5 (group AL0) Clearing of SCIF status flag IR(RSPI0, SPRI0) = 0b IR(RSPI0, SPTI0) = 0b IR(ICU, GROUPAL0) = 0b (group AL0)
	Clearing of RSPI status flag* RSPI.SPSR.MODF = 0b (SPEI) RSPI.SPSR.OVRF = 0b (SPEI) RSPI.SPSR.SPRF = 0b (SPRI)	Clearing of RSPI status flag* RSPI0.SPSR.MODF = 0b (SPEI0) RSPI0.SPSR.OVRF = 0b (SPEI0)
	No need to clear the RSPI status flag because it is cleared when RSPI operation ends. However, SPRF, MODF, and OVRF are not initialized. * Clear to 0 after reading value as 1.	No need to clear the RSPI status flag because it is cleared when RSPI operation ends. However, MODF and OVRF are not initialized. * Clear to 0 after reading value as 1.
Make I/O port settings (peripheral).		Cancellation of register protection MPC.PWPR.B0WI = 0b MPC.PWPR.PFSWE = 1b Pin peripheral function selection MPC.PC5PFS.PSEL = 001101b (RSPCK) MPC.PC6PFS.PSEL = 001101b (MOSI) Register protection settings MPC.PWPR.PFSWE = 0b MPC.PWPR.B0WI = 1b
	Pin mode settings PFC.PACRL2.PA5MD = 101b (RSPCK) PFC.PACRL2.PA4MD = 101b (MOSI)	Pin mode settings (peripheral) PORTC.PMR.B5 = 1b (RSPCK) PORTC.PMR.B6 = 1b (MOSI)

For setting examples for SPI operation other than clock-synchronous slave reception, refer to the setting examples in 2.12.5, Setting Example for Master Transmission/Reception Using SPI Operation.

Full-duplex operation is used in the clock-synchronous slave reception setting examples, so dummy reception processing is necessary.

2.13 I²C Bus Interface (IIC)

2.13.1 Comparison of Specifications

I²C bus interface functionality is provided on the SH7216 Group by the IIC3 and on the RX71M by the RIICa, which supports communication operation compliant with SMBus (ver. 2.0).

Table 2.115 is a comparative listing of the specifications of the SH7216 Group and RX71M.

Table 2.115 Comparison of SH7216 Group and RX71M Specifications (IIC)

Item		SH7216 Group (IIC3)	RX71M (RIICa)
Number of channels		1 channel	2 channels
Clock source		Peripheral clock (Pϕ)	Peripheral module clock (PCLKB)
Communication format		• I²C bus format • Clock-synchronous serial format*1	• I²C bus format • SMBus format
Data transfer		Fixed at MSB-first Selectable between MSB-first and LSB-first for clock-synchronous serial format	Fixed at MSB-first
I²C bus format (SMBus)	Operating modes	• Master transmit mode • Master receive mode • Slave transmit mode • Slave receive mode	
	Start condition/ stop condition	Automatically generated	
	Address detection	• 7-bit slave addresses	• 3 types of 7- or 10-bit slave addresses • General call address • Device ID address • SMBus host address
	DTC/DMAC activation	DTC activation supported	DTC/DMAC activation supported
	Interrupt sources	• Arbitration lost • NACK detection • Stop condition detection • Receive data-full • Transmit data-empty • Transmit end	• Arbitration lost detection • NACK detection • Timeout detection • Start condition detection • Stop condition detection • Receive data-full • Transmit data-empty • Transmit end
	Multi-master support	Bit synchronization circuit Ability to specify a transfer rate at least 1/1.8 times the fastest transfer rate of another master	SCL synchronization circuit
Noise cancellation		Ability to specify the noise cancellation width for the SCL and SDA pins Up to 3-stage latch circuit	Ability to enable digital noise filter and specify the noise cancellation width for the SCL and SDA pins Up to 5-stage latch circuit
Other		—	• Event link • SCL clock duty ratio setting • SDA output delay function • SCL auto low-hold function • Bus hang-up support

Note 1. The RIICa on the RX71M does not support clock-synchronous serial format, but the clock-synchronous communication format of the SCIg and SCIh can be used as a substitute.

2.13.2 Register Comparison

Table 2.116 is a comparative listing of the registers on the SH7216 Group and RX71M.

Guide to Symbols in "Changes" Column of Table

◎: Register with same bit assignments on SH7216 Group and RX71M

△: Register with different bit assignments on SH7216 Group and RX71M

—: Register not present on SH7216 Group or RX71M

Table 2.116 SH7216 Group and RX71M Register Comparison (IIC)

SH7216 Group (IIC3)	RX71M (RIICa)*1	Changes
I²C bus control register 1 (ICCR1) I²C bus control register 2 (ICCR2)	I²C bus control register 1 (RIICn.ICCR1) I²C bus control register 2 (RIICn.ICCR2)	△
I²C bus mode register (ICMR)	I²C bus mode register 1 (RIICn.ICMR1)	△
I²C bus interrupt enable register (ICIER)	I²C bus mode register 3 (RIICn.ICMR3)*2 I²C bus interrupt enable register (RIICn.ICIER) I²C bus function enable register (RIICn.ICFER)	△
I²C bus status register (ICSR)	I²C bus status register 1 (RIICn.ICSR1) I²C bus status register 2 (RIICn.ICSR2)	△
Slave address register (SAR)	Slave address register Ly (RIICn.SARLy) (y = 0 to 2) I²C bus mode register 3 (RIICn.ICMR3)*2	△
I²C bus transmit data register (ICDRT)	I²C bus transmit data register (RIICn.ICDRT)	◎
I²C bus receive data register (ICDRR)	I²C bus receive data register (RIICn.ICDRR)	◎
I²C bus shift register (ICDRS)	I²C bus shift register (ICDRS)	◎
NF2CYC register (NF2CYC)	I²C bus mode register 3 (RIICn.ICMR3)*2	△
—	I²C bus mode register 2 (RIICn.ICMR2)	—
	Slave address register Uy (RIICn.SARUy) (y = 0 to 2)	
	I²C bus bit rate low-level register (RIICn.ICBRL)	
	I²C bus bit rate high-level register (RIICn.ICBRH)	
	I²C bus status enable register (RIICn.ICSER)	

Note 1. RIICn, n: 0 or 2

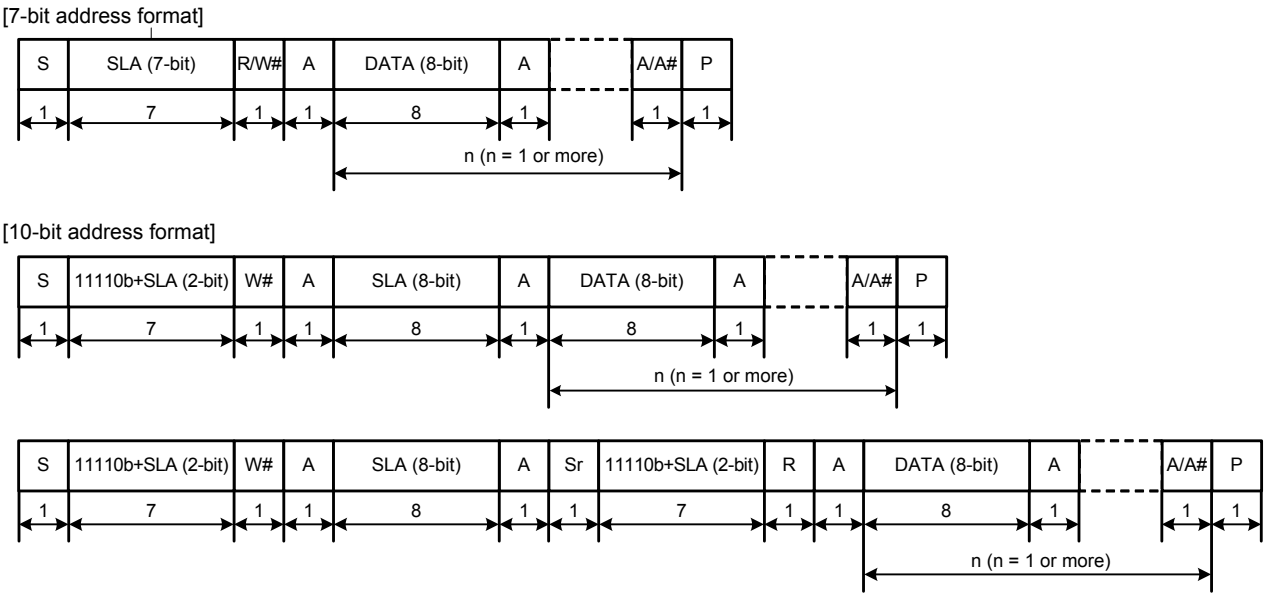
Note 2. The functions of some registers on the SH7216 Group are divided among multiple registers on the RX71M.

2.13.3 Address Detection

The SH7216 Group can detect 7-bit slave addresses of a single type.

The RX71M can detect three types of slave addresses, as well as general call addresses, device ID addresses, and SMBus host addresses. In addition, the slave address bit count can be specified as either 7-bit or 10-bit.

Figure 2.39 shows the RX71M I²C bus format.



n: Transfer data byte count
S: Start condition
SLA: Slave address
R/W#: Transmit/receive direction
A: Acknowledge
A#: Not acknowledge
Sr: Restart condition
DATA: Transmit/receive data
P: Stop condition

Figure 2.39 RX71M I²C Bus Format

2.13.4 Arbitration Lost Detection

In addition to the ordinary arbitration lost detection function stipulated in the I²C bus specification, the RX71M provides functions for prevention of issuance of overlapping start requests, arbitration lost detection during NACK transmission, and arbitration lost detection during slave receive operation.

2.13.5 Bus Hang-up

If synchronization of the master device and slave device on the I²C bus is disrupted due to noise or the like, a bus hang-up may occur where the SCL line or SDA line becomes fixed at a single level.

To deal with bus hang-ups, the RX71M provides a timeout detection function that monitors the SCL line to detect bus hang-up states and, to recover from bus hang-up states caused by disrupted synchronization, an SCL clock additional output function, an RIIC reset function, and an internal reset function.

2.13.6 SCL Clock

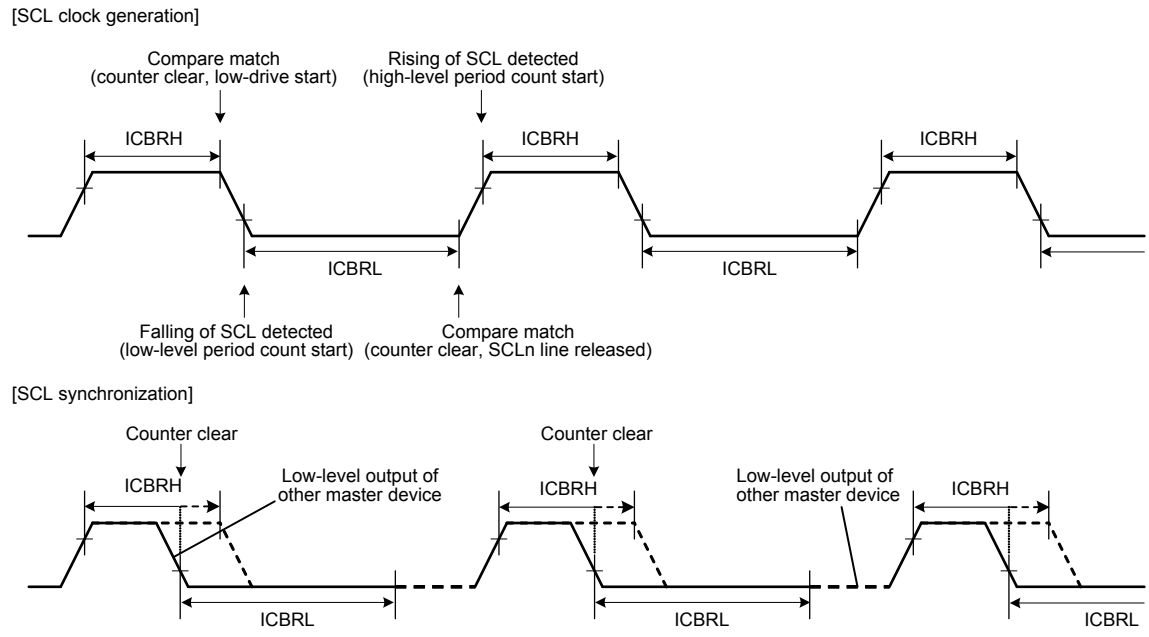
Under the I²C bus format transmission and reception of data are synchronized with the SCL clock output by the master device.

When operating in master mode, the SCL clock transfer rate on the SH7216 Group is determined by the peripheral clock division ratio setting in I²C bus control register 1 (ICCR1). On the RX71M the SCL transfer rate and duty ratio are determined by the SCL clock high-level period setting in the I²C bus bit rate high-level register (ICBRH) and the SCL clock low-level setting in the I²C bus bit rate low-level register (ICBRL).

The RX71M provides a transmit data accidental transmission prevention function, a NACK receive transfer cutoff function, and a receive data loss prevention function. The SCL line is automatically held low when certain conditions are met.

When the I²C bus format is used in a multi-master configuration, conflicts can arise between the SCL clock that that of the other master device. This is why the SH7216 Group is provided with a bit synchronization circuit, and the RX71M with an SCL synchronization circuit, that monitors the SCLn line in master mode and generates the SCL clock with bit-by-bit synchronization.

Figure 2.40 illustrates SCL clock generation and SCL synchronization on the RX71M.



ICBRH: I²C-bus bit rate high-level register (SCL clock high-level period counter)
ICBRL: I²C-bus bit rate low-level register (SCL clock low-level period counter)

Figure 2.40 SCL Clock Generation and SCL Synchronization

2.13.7 Noise Cancellation

In addition to a setting for the noise cancellation width, on the RX71M it is possible to enable or disable the digital noise filter circuit by making a setting in the I²C-bus function enable register (ICFER).

2.13.8 Interrupts

On both the SH7216 Group and RX71M the receive data-full and transmit data-empty interrupts can be used to activate the DTC and the DMAC.

On the RX71M when a receive data-full or transmit data-empty interrupt occurs while the corresponding interrupt status flag (IRn.IR) is set to 1, the interrupt request is also stored internally by the module, and after the interrupt status flag (IRn.IR) is cleared to 0 it is reset to 1 by the stored request.

On the RX71M some interrupts are assigned to group interrupt BL1. The interrupt controller's interrupt status flag (IRn.IR) is cleared automatically when the corresponding interrupt is accepted. The group BL1 interrupt status flag (GRPBL1.ISn) is cleared automatically when the corresponding bit in the module's status register is cleared.

Table 2.117 and Table 2.118 list interrupt sources for the SH7216 Group and RX71M.

Refer to 1.9, Interrupt Handling for information about interrupts.

Table 2.117 SH7216 Group IIC3 Interrupt Sources (I^2C Bus Format)

Priority	Interrupt Source	Activation by Interrupt
High	Stop condition detection	Not possible
↑	NACK detection Arbitration lost/overrun error	
	Receive data-full	DMAC and DTC activation possible
	Transmit data-empty	
Low	Transmit end	Not possible

Table 2.118 RX71M RIICa Interrupt Sources

Priority	Interrupt Source		Activation by Interrupt
High	Communication error/ event occurrence	Arbitration lost	Not possible
↑		NACK detection	
		Timeout	
		Start condition detection	
		Stop condition detection	
	Receive data-full		DMAC and DTC activation possible
	Transmit data-empty		
Low	Transmit end		Not possible

2.13.9 Module Stop

As on the SH7216 Group, the RIICa of the RX71M is set to the module-stop state after a reset, and no clock is supplied.

Refer to 2.21, Low Power Consumption Function for information on the module-stop state.

2.13.10 Setting Example for Master Transmission/Reception

Setting examples for master transmission and master reception using the I²C bus interface of the SH7216 Group and RX71M are presented below.

Operational Overview

- The IIC is used to perform master transmission and reception in sequence.
- Master transmission:
  - Operation is started by a start condition issuance request.
  - Transmission of the slave address and data transmission are activated by a transmit data-empty interrupt.
  - A stop condition is issued at the end of the transmit data, and a transition to the standby state takes place.
  - Reception of a NACK during transmission, other than at the end of the transmit data, results in an error.
- Master reception:
  - Operation is started by a start condition issuance request.
  - Transmission of the slave address is activated by a transmit data-empty interrupt.
  - Reception is activated by a receive data-full interrupt, and the receive data is stored in the RAM.
  - A NACK is transmitted at the end of the receive data, a stop condition is issued, and a transition to the standby state takes place.
  - Reception of a NACK targeted to the slave address results in an error.
- IIC operation ends after master transmission and master reception finish.
- IIC operation ends when an error is detected.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by "SH7216:" in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.119 IIC Master Transmission and Reception Specifications

Item		Description	Remarks
IIC channel		RIIC2	SH7216: 1 channel only
Clock		PCLKB = 60 MHz	SH7216: Pϕ = 50 MHz
Communication mode		I²C bus	
Operating mode		Master transmission and master reception	
Transfer speed		400 kbps	SH7216: 403 kbps
Number of data bits		9 bits (including ACK)	
Wait between data and ACK		Wait at end of receive data only	
ACK detection		Transfer halted when ACK = 1 received	
Slave address format		7-bit address	
Slave address		50h	
Noise cancellation		3-stage latch circuit (2 filter stages)	
Transmit data		32 bytes (value from 1 to 32)	
Interrupts		Interrupts are used (except for start condition detection).	Priority level: 5
Pins used	SCL	P16/SCL2-DS	SH7216: PB12/SCL
	SDA	P17/SDA2-DS	SH7216: PB13/SDA
	LED0	P03/general	Lights when transmission/ reception is possible.
	LED1	P05/general	Lights when transmission ends.
	LED2	P26/general	Lights when reception ends.
	LED3	P27/general	Lights when error detected.

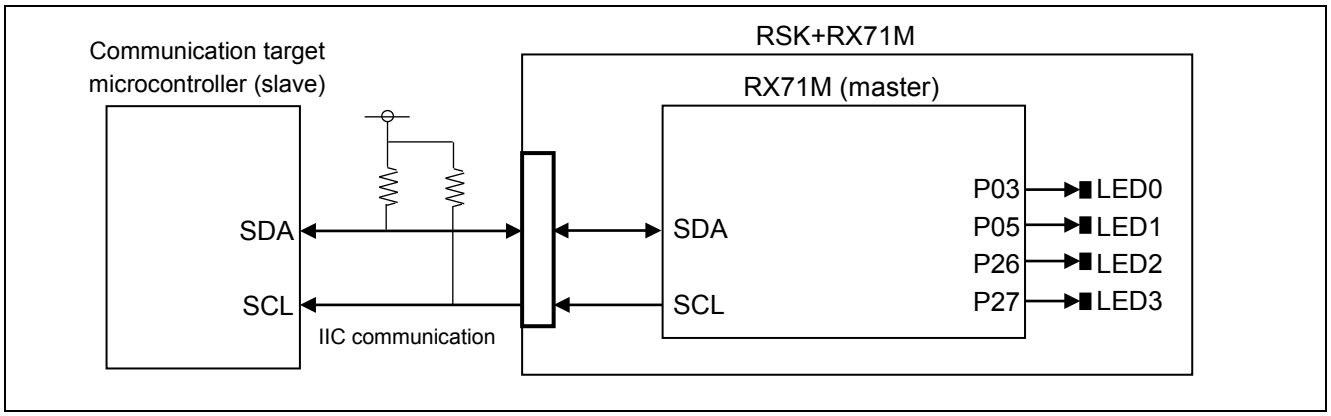


Figure 2.41 IIC Master Transmission and Reception Connection Example

Note

On the Renesas Starter Kit+ for RX71M in the initial state the pins used for RIIC2 in the connection example are connected for use with the USB and EEPROM, so make appropriate modifications to the board as necessary.

List of Related Registers

Table 2.120 shows the IIC3 interrupt-related registers used in the SH7216 Group setting example broken down by source.

Table 2.120 SH7216 Group Interrupt-Related Registers (IIIC3 and INTC)

			IIC3		INTC
Item	**Vector No.**	**Name**	**Interrupt Enable**	**Status**	**Priority Level**
Setting register	—	—	IIC3.ICIER	IIC3.ICSR	IPR13
Setting position of each source					
Stop condition detection	228	STPI	STIE	STOP	Bits 4 to 7
NACK detection	229	NAKI	NAKIE	NACKF	
Arbitration lost/overrun error				AL/OVE	
Transmit end	232	TEI	TEIE	TEND	
Receive data-full	230	RXI	RIE	RDRF	
Transmit data-empty	231	TXI	TIE	TDRE	

Table 2.121 and Table 2.122 show the interrupt-related registers used in the RX71M setting example broken down by source. On the RX71M some RIICa-related interrupts are assigned as group BL1 interrupts.

Table 2.121 RX71M Interrupt-Related Registers (RIICa)

Item	Name	Interrupt Enable	Status
Setting register	—	RIIC2.ICIER	RIIC2.ICSR2
Setting position of each source (RIIC2)			
Start condition detection	STI2	STIE	START
Stop condition detection	SPI2	SPIE	STOP
NACK detection	NAKI2	NAKIE	NACKF
Arbitration lost	ALI2	ALIE	AL
Timeout	TMOI2	TMOIE	TMOF
Transmit end	TEI2	TEIE	TEND
Receive data-full	RXI2	RIE	RDRF
Transmit data-empty	TXI2	TIE	TDRE

Table 2.122 RX71M RIIC2 Interrupt-Related Registers (ICUA)

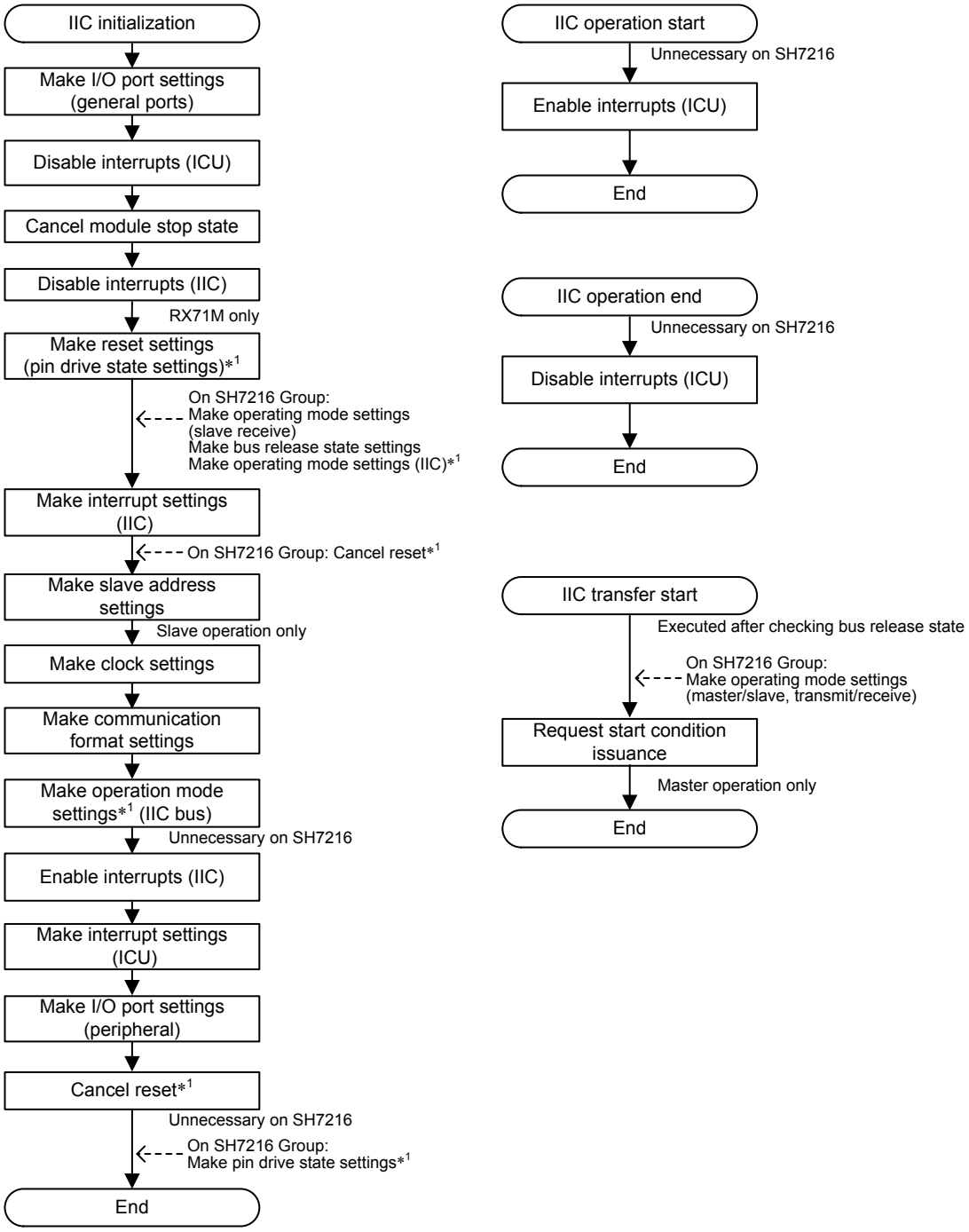
Item	Vector No.	Name	Interrupt Enable		Status		Priority Level
Setting register	—	—	IERm	GENBL1	IRr	GRPBL1	IPRr
Setting position of each source (RIIC2)							
Start condition detection	111	EEI2	IER0D.IEN7 (group BL1)	EN16	IR111 (group BL1)	IS16	IPR111 (group BL1)
Stop condition detection							
NACK detection							
Arbitration lost							
Timeout							
Transmit end		TEI2		EN15		IS15	
Receive data-full	54	RXI2	IER06.IEN6	—	IR054	—	IPR054
Transmit data-empty	55	TXI2	IER06.IEN7	—	IR055	—	IPR055
				↑		↑	
				Group BL1 interrupt settings			

When making settings to the ICUA interrupt-related registers on the RX71M, iodefine.h can be used to make settings as follows. GROUPBL1 defines settings for group BL1 interrupts.

- IERm : IEN (RIIC2 or ICUA interrupt name), IEN (ICU or GROUPBL1)
- IPRr : IPR (RIIC2 or ICUA interrupt name), IPR (ICU or GROUPBL1)
- IRr : IR (RIIC2 or ICUA interrupt name), IR (ICU or GROUPBL1)
- GENBL1 : EN (RIIC2 or ICUA interrupt name)
- GRPBL1 : IS (RIIC2 or ICUA interrupt name)

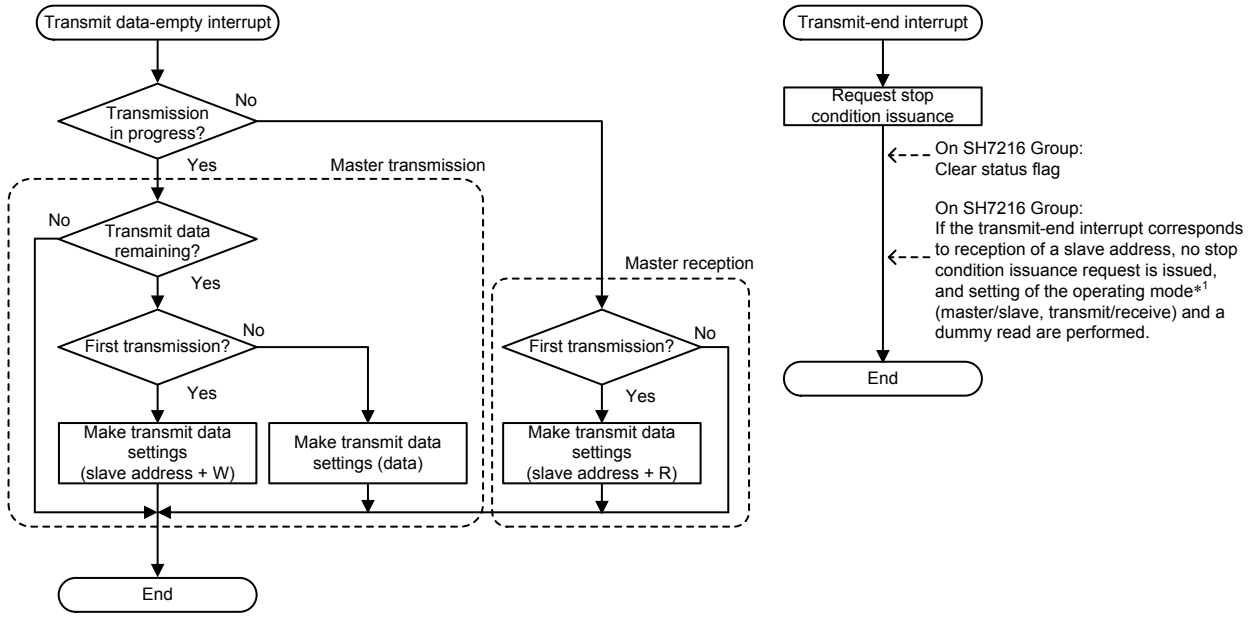
Processing Flowcharts

Figure 2.42 to Figure 2.44 show example flowcharts of processing using the IIC. The names of the processing steps shown in the flowcharts correspond to the names in the setting examples.



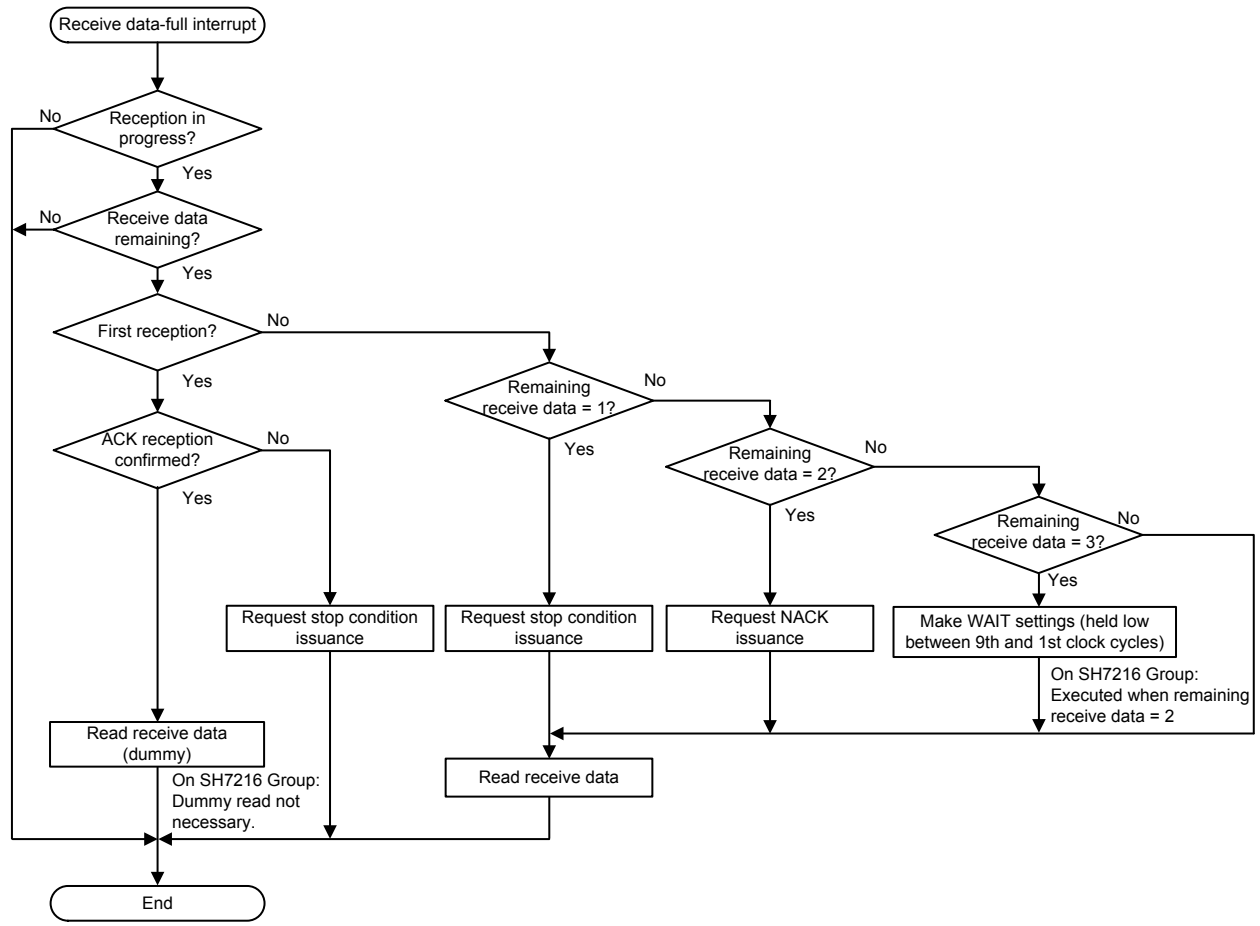
Note 1. The setting sequence differs on the SH7216 Group and on the RX71M.

Figure 2.42 Example Flowcharts of IIC Processing



Note 1. The method of changing the operating mode differs on the SH7216 Group and on the RX71M.
On SH7216 Group:
For reception, it is necessary to change to master reception mode by using a transmit-end interrupt after setting of the transmit data (slave address + R/W). It is also necessary to change to slave reception mode when a stop condition is detected.
On RX71M:
For reception, the change to master reception mode occurs automatically after setting of the transmit data (slave address + R/W). Also, the change to slave reception mode occurs automatically when a stop condition is detected.

Figure 2.43 Example Flowcharts of IIC Transmit Data-Empty and Transmit-End Interrupt Handling (Master Operation)



Note: Processing when the receive data count is 3 or less is omitted from the flowchart above.

Figure 2.44 Example Flowchart of IIC Receive Data-Full Interrupt Handling (Master Operation)

Setting Examples

Setting examples for master transmission and master reception are presented below. The names of the processing steps in these setting examples correspond to the names in the flowcharts. Refer to the flowcharts for the processing procedure.

Table 2.123 IIC Initialization Setting Examples

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make I/O port settings (general ports).	Output value settings (output 1) PB.DR.B12 = 1b (PB12) PB.DR.B13 = 1b (PB13) Pin direction settings PFC.PBIORL.B12 = 1b (output/PB12) PFC.PBIORL.B13 = 1b (output/PB13) Pin mode settings (port) PFC.PBCRL4.PB12MD = 000b (PB12) PFC.PBCRL4.PB13MD = 000b (PB13)	Output value settings (output 1) PORT1.PODR.B6 = 1b (P16) PORT1.PODR.B7 = 1b (P17) Pin direction settings PORT1.PDR.B6 = 1b (output/P16) PORT1.PDR.B7 = 1b (output/P17) Pin mode settings (general) PORT1.PMR.B6 = 0b (P16) PORT1.PMR.B7 = 0b (P17)
Disable interrupt (ICU).	—	Disabling of ICU interrupts IEN(RIIC2, RXI2) = 0b IEN(RIIC2, TXI2) = 0b IEN(ICU, GROUPBL1) = 0b (group BL1) Disabling interrupts in group BL1 EN(RIIC2, TEI2) = 0b EN(RIIC2, EEI2) = 0b
Cancel module stop state.	STB.CR3._IIC3 = 0b	SYSTEM.PRCR = A502h SYSTEM.MSTPCRC.MSTPC17 = 0b SYSTEM.PRCR = A500h
Disable interrupt (IIC).	IIC3.ICIER.TEIE = 0b (TEI) IIC3.ICIER.RIE = 0b (RXI) IIC3.ICIER.TIE = 0b (TXI) IIC3.ICIER.STIE = 0b (SPTI) IIC3.ICIER.NAKIE = 0b (NAKI)	It is not necessary to disable RIIC interrupts because they are all initialized by an RIIC reset.
Make reset settings (pin drive state settings).*2	IIC3.ICCR1.ICE = 0b (pin non-drive state) IIC3.ICCR2.IICRST = 1b (IIC reset)	Reset of RIIC RIIC2.ICCR1.ICE = 0b (pin non-drive state) RIIC2.ICCR1.IICRST = 1b RIIC internal reset RIIC2.ICCR1.ICE = 1b (pin drive state)
Make operating mode settings (slave receive).	IIC3.ICCR1.MST = 0b IIC3.ICCR1.TRS = 0b	—
Make bus release state settings.	Setting of register state after checking bus release state IIC3.SAR.FS = 1b (IIC3.ICCR2.BBSY cleared to 0 by selecting clock-synchronous operation)	—
Make interrupt settings (IIC).	IIC3.ICSR = 0* * Clear to 0 after reading value as 1.	— Not necessary because all interrupts are initialized by an RIIC reset.

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make slave address settings.*1	IIC3.SAR.SVA = 1010000b	RIIC2.SARL0.SVA = 1010000b RIIC2.SARU0.FS = 0b (7 bits) RIIC2.ICSER.SAR0E = 1b (SARL0 enabled) RIIC2.ICSER.GCAE = 0b (general call address detection disabled)
Make clock settings.	IIC3.ICCR1.CKS = 0111b (403 kbps @50 MHz/124)	Clock setting RIIC2.ICMR1.CKS = 010b Bit rate settings (400 kbps @60 MHz/4) RIIC2.ICBRL.BRL = 19 RIIC2.ICBRH.BRH = 8
Make communication format settings.		Timeout detection settings RIIC2.ICMR2.TMOS = 0b (long mode) RIIC2.ICMR2.TMOL = 1b RIIC2.ICMR2.TMOH = 1b RIIC2.ICFER.TMOE = 1b (enabled) Arbitration lost detection settings RIIC2.ICFER.MALE = 1b (master) RIIC2.ICFER.NALE = 1b (NACK) RIIC2.ICFER.SALE = 1b (slave)
	Enabling of transfer halt when NACK received IIC3.ICIER.ACKE = 1b	Enabling of transfer halt when NACK received RIIC2.ICFER.NACKE = 1b
	Noise cancellation setting IIC3.NF2CYC.NF2CYC = 1b (2 cycles)	Digital filter settings RIIC2.ICMR3.NF = 01b (2-stage filter) RIIC2.ICFER.NFE = 1b (enabled)
Make operation mode settings*2 (IIC bus).	IIC3.SAR.FS = 0b (IIC bus)	RIIC2.ICMR3.SMBS = 0b (IIC bus)
Enable interrupts (IIC).	IIC3.ICIER.TEIE = 1b (TEI) IIC3.ICIER.RIE = 1b (RXI) IIC3.ICIER.TIE = 1b (TXI) IIC3.ICIER.STIE = 1b (SPTI) IIC3.ICIER.NAKIE = 1b (NAKI)	RIIC2.ICIER.TEIE = 1b (TEI2) RIIC2.ICIER.RIE = 1b (RXI2) RIIC2.ICIER.TIE = 1b (TXI2) RIIC2.ICIER.SPIE = 1b (SPI2) RIIC2.ICIER.NAKIE = 1b (NAKI2) RIIC2.ICIER.ALIE = 1b (ALI2) RIIC2.ICIER.TMOIE = 1b (TMOI)
Make interrupt settings (ICU).	Priority setting (level 5) INTC.IPR13._IIC3 = 5	Priority settings (level 5) IPR(RIIC2, RXI2) = 5 IPR(RIIC2, TXI2) = 5 IPR(ICU, GROUPBL1) = 5 (group BL1) Clearing of ICU status IR(RIIC2, RXI2) = 0b IR(RIIC2, TXI2) = 0b IR(ICU, GROUPBL1) = 0b (group BL1)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make I/O port settings (peripheral).		Cancellation of register protection MPC.PWPR.B0WI = 0b MPC.PWPR.PFSWE = 1b Pin peripheral function selection MPC.P16PFS.PSEL = 001111b (SCL2-DS) MPC.P17PFS.PSEL = 001111b (SDA2-DS) Register protection settings MPC.PWPR.PFSWE = 0b MPC.PWPR.B0WI = 1b
	Pin mode settings PFC.PBCRL4.PB12MD = 110b (SCL) PFC.PBCRL4.PB13MD = 110b (SDA)	Pin mode settings (peripheral) PORT1.PMR.B6 = 1b (SCL2-DS) PORT1.PMR.B7 = 1b (SDA2-DS)
Cancel reset.*2	IIC3.ICCR2.IICRST = 0b	RIIC2.ICCR1.IICRST = 0b
Make pin drive state setting.*2	IIC3.ICCR1.ICE = 1b	—

Note 1. Slave address settings are necessary for slave operation.

Note 2. The setting sequence differs on the SH7216 Group and on the RX71M. Refer to the example flowchart for details.

Table 2.124 IIC Operation Start Setting Example

Processing	SH7216 Group Setting Example	RX71M Setting Example
Enable interrupts (ICU).	—	Enabling interrupts in group BL1 EN(RIIC2, TEI2) = 1b EN(RIIC2, EEI2) = 1b Enabling of ICU interrupts IEN(RIIC2, RXI2) = 1b IEN(RIIC2, TXI2) = 1b IEN(ICU, GROUPBL1) = 1b (group BL1)

Table 2.125 IIC Operation End Setting Examples

Processing	SH7216 Group Setting Example	RX71M Setting Example
Disable interrupt (ICU).	—	Disabling of ICU interrupts IEN(RIIC2, RXI2) = 0b IEN(RIIC2, TXI2) = 0b IEN(ICU, GROUPBL1) = 0b (group BL1) Disabling interrupts in group BL1 EN(RIIC2, TEI2) = 0b EN(RIIC2, EEI2) = 0b

Table 2.126 IIC Transfer Operation Start Setting Example (Master)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Check bus release state.	Checking that IIC3.ICCR2.BBSY = 0b	Checking that RIIC2.ICCR2.BBSY = 0b
Make operating mode settings (master/slave, transmit/receive).	IIC3.ICCR1.MST = 1 (master) IIC3.ICCR1.TRS = 1 (transmit)	—
Request start condition issuance.	IIC3.ICCR2.BBSY = 1b IIC3.ICCR2.SCP = 0b	RIIC2.ICCR2.ST = 1b

The sample code does not specify any particular interrupt handling. The setting examples below apply to a portion of the processing involved in interrupt handling.

Table 2.127 Setting Examples in IIC Receive Data-Full Interrupt Handling

Processing	SH7216 Group Setting Example	RX71M Setting Example
Confirm ACK reception.	Checking that IIC3.ICIER.ACKBR = 0b	Checking that RIIC2.ICMR3.ACKBR = 0b
Read receive data.	Reading value of IIC3.ICDRR	Reading value of RIIC2.ICDRR
Request stop condition issuance.	IIC3.ICSR.STOP = 0b (flag cleared)* IIC3.ICCR2.BBSY = 0b IIC3.ICCR2.SCP = 0b * Clear to 0 after reading value as 1.	RIIC2.ICSR2.STOP = 0b (flag cleared)* RIIC2.ICCR2.SP = 1b * Clear to 0 after reading value as 1.
Request NACK issuance.	IIC3.ICIER.ACKBT = 1b	RIIC2.ICMR3.ACKWP = 1b RIIC2.ICMR3.ACKBT = 1b RIIC2.ICMR3.ACKWP = 0b
Make WAIT setting.	IIC3.ICCR1.RCVD = 1b	RIIC2.ICMR3.WAIT = 1b

Table 2.128 Setting Examples in IIC Transmit Data-Empty Interrupt Handling

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make transmit data settings (slave address + W).	Setting of value in IIC3.ICDRT Setting value: Upper 7 bits: Slave address Lowest bit: 0	Setting of value in RIIC2.ICDRT Setting value: Upper 7 bits: Slave address Lowest bit: 0
Make transmit data settings (slave address + R).	Setting of value in IIC3.ICDRT Setting value: Upper 7 bits: Slave address Lowest bit: 1	Setting of value in RIIC2.ICDRT Setting value: Upper 7 bits: Slave address Lowest bit: 1
Make transmit data settings (data).	Setting of value in IIC3.ICDRT	Setting of value in RIIC2.ICDRT

2.13.11 Setting Example for Slave Transmission/Reception

Setting examples for slave transmission and reception using the I²C bus interface of the SH7216 Group and RX71M are presented below.

Operational Overview

- Slave transmission and reception using the IIC operate continuously.
- Slave transmission:
  - Operation starts when the slave address matches and the value of the received R/W# is 1.
  - Transmission of data is activated by a transmit data-empty interrupt.
  - Transition to the standby state occurs when a NACK is received or a stop condition is detected.
- Slave reception:
  - Operation starts when the slave address matches and the value of the received R/W# is 0.
  - Reception is activated by a receive data-full interrupt, and the receive data is stored in the RAM .
  - Transition to the standby state occurs when a stop condition is detected.
- IIC operation ends when an error is detected.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by “SH7216:” in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.129 IIC Slave Transmission and Reception Specifications

Item		Description	Remarks
IIC channel		RIIC2	
Communication mode		I²C bus	
Operating mode		Slave transmission and slave reception	
Number of data bits		9 bits (including ACK)	
Wait between data and ACK		No	
ACK detection		Transfer does not stop when ACK = 1 is received.	
Slave address format		7-bit address	
Slave address		50h (general call addresses not supported)	
Noise cancellation		3-stage latch circuit (2 filter stages)	
Transmit data		Repetition of values from 1 to 32	1 is returned in standby state.
Interrupts		Interrupts are used (excluding transmit-end and start condition detection)	Priority: level 5
Pins used	SCL	P16/SCL2-DS	SH7216: PB12/SCL
	SDA	P17/SDA2-DS	SH7216: PB13/SDA
	LED0	P03/general	Lights when transmission/ reception is possible.
	LED1	P05/general	Lights during transmission.
	LED2	P26/general	Lights during reception.
	LED3	P27/general	Lights when error detected.

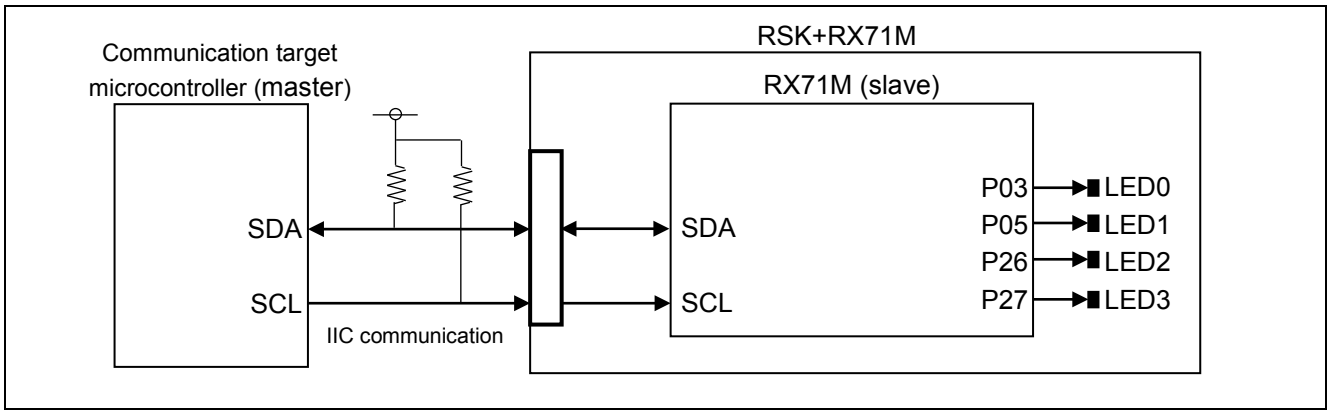


Figure 2.45 IIC Slave Transmission and Reception Connection Example

Note

On the Renesas Starter Kit+ for RX71M in the initial state the pins used for RIIC2 in the connection example are connected for use with the USB and EEPROM, so make appropriate modifications to the board as necessary.

List of Related Registers

For the interrupt-related registers used in these setting examples, refer to the list of registers related to master transmission and reception in 2.13.10, Setting Example for Master Transmission/Reception.

Processing Flowcharts

Example flowcharts of processing for slave transmission and slave transmission using the IIC are shown below. For flowcharts of initialization processing, refer to the flowcharts of initialization processing in 2.13.10, Setting Example for Master Transmission/Reception.

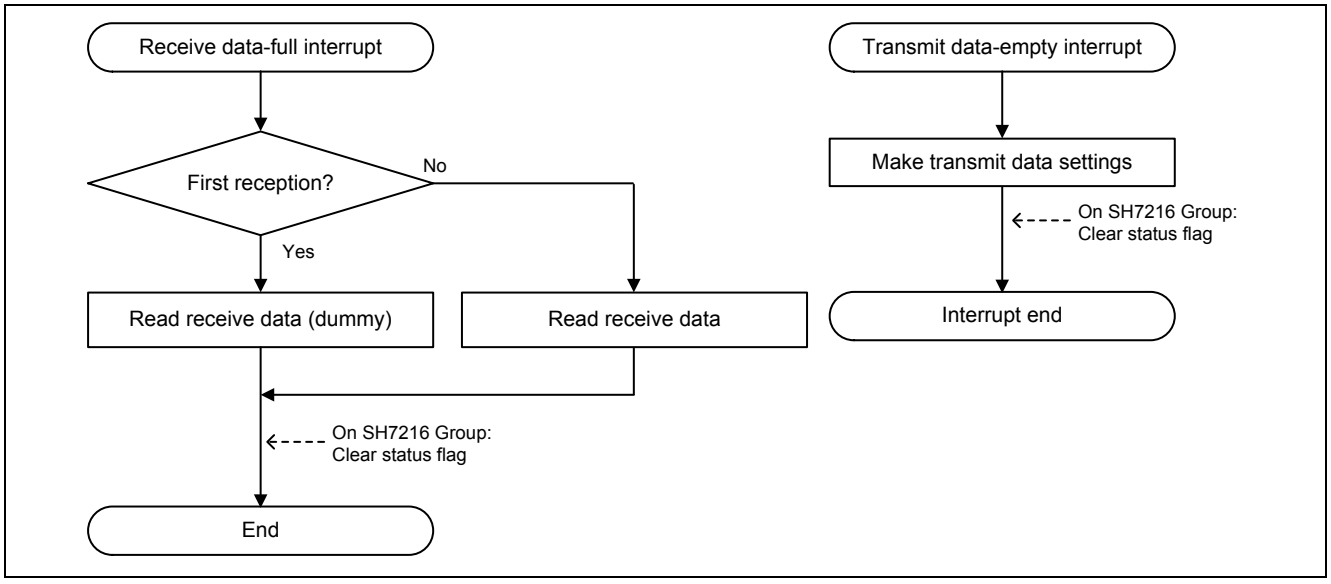


Figure 2.46 Example Flowcharts of IIC interrupt Handling (Slave Operation)

Setting Examples

Setting examples for slave transmission and slave reception are presented below. For initialization, operation start, and operation end setting examples, refer to the master transmission and transmission setting examples in 2.13.10, Setting Example for Master Transmission/Reception. No settings are necessary for interrupts not used in slave transmission and slave reception.

The names of the processing steps in these setting examples correspond to the names in the flowcharts. Refer to the flowcharts for the processing procedure.

Table 2.130 IIC Transfer Operation Start Setting Example (Slave)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Check bus release state.	Checking that ICCR2.BBSY = 0b	Checking that RIIC2.ICCR2.BBSY = 0b
Make operating mode settings (master/slave, transmit/receive).	ICCR1.MST = 0 (slave mode) ICCR1.TRS = 1 (transmit mode)	—

The sample code does not specify any particular interrupt handling. For interrupt handling setting examples, refer to the setting examples in 2.13.10, Setting Example for Master Transmission/Reception. The processing sequences differ, but setting examples for slave transmission and reception are presented.

2.14 A/D Converter (ADC)

2.14.1 Comparison of Specifications

A/D converter functionality is provided on the SH7216 Group by the ADC and on the RX71M by the 12-bit A/D converter (S12ADC).

Table 2.131 is a comparative specifications of the SH7216 Group and RX71M.

Table 2.131 Comparison of SH7216 Group and RX71M Specifications (ADC)

Item	SH7216 Group (ADC)	RX71M (S12ADC)
Number of input channels	8 channels (4 channels × 2)	Unit 0 (S12AD): 8 channels Unit 1 (S12AD1): 21 channels + 1 extension
Clock source	AD clock (Aϕ)	S12AD: Peripheral module clock (PCLKC) S12AD1: Peripheral module clock (PCLKD)
Resolution	12 bits	Max. 12 bits (selectable among 8, 10, and 12 bits)
A/D conversion method	Successive approximation	Successive approximation
Conversion speed	1.0 μs per channel (AD clock: 25 MHz)	0.48 μs per channel (12-bit conversion mode, A/D converter clock: 60 MHz)
Conversion modes	• Single-cycle scan mode • Continuous scan mode	• Single scan mode • Continuous scan mode • Group scan mode
A/D conversion start conditions	• Software trigger • Synchronous trigger (MTU2, MTU2S) • Asynchronous trigger ($\overline{\text{ADTRG}}$ pin)	• Software trigger • Synchronous trigger (MTU, GPT, TMR, TPU, ELC) • Asynchronous trigger (ADTRG0# pin, ADTRG1# pin)
Operations linked to A/D conversion-end interrupt	• CPU interrupt generation • DMAC or DTC activation	• CPU interrupt generation • DMAC or DTC activation
Conversion targets	• AN pin	• AN pin • Internal reference voltage (S12AD1) • Temperature sensor (S12AD1)
DTC/DMAC activation	DTC/DMAC activation supported	DTC/DMAC activation supported
Interrupt sources	• A/D conversion end	• A/D conversion end • Digital compare
Other	• Sample and hold function • Channel-specific Sample-and-hold function (module 0) • A/D data register auto-clear function	• Event link • Sample and hold function • Channel-specific Sample-and-hold function (S12AD) • Variable sampling state count function • A/D converter self-diagnostic function • Selectable between A/D-converted value addition mode or average mode • Analog input disconnection detection assist function • Double trigger mode • 12-/10-/8-bit conversion switching • A/D data register auto-clear function • Extended analog input function • Comparison function (ability to select window function)

2.14.2 Input Channels

On the SH7216 Group the ADC comprises two modules, each of which has four analog input channels. On the RX71M the S12ADC comprises two units, S12AD and S12AD1, one with eight channels and the other with 21 channels. As on the SH7216 Group, on the RX71M each unit incorporates an A/D converter. Simultaneous operation is possible, but continuous scan operation spanning the two units is not supported.

Figure 2.47 compares the A/D converter configurations of the SH7216 Group and RX71M.

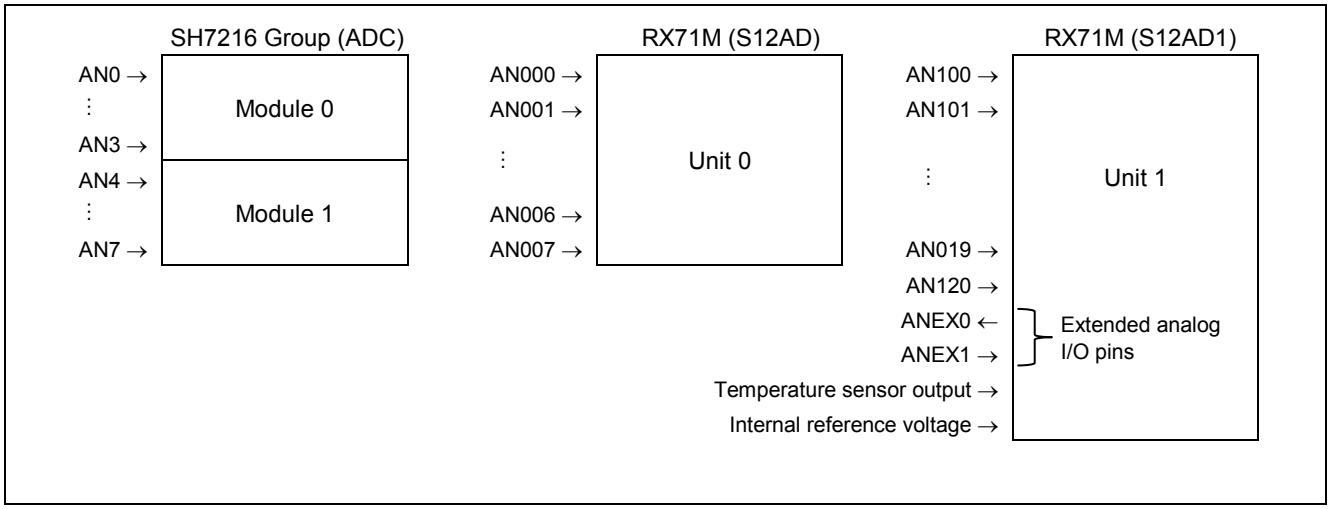


Figure 2.47 Comparison of A/D Converter Configurations

2.14.3 Scanning Sequence

Table 2.132 lists the scanning sequence when all channels are specified.

Table 2.132 A/D Converter Scanning Sequence

Microcontroller	A/D Converter	Conversion Sequence
SH7216 Group	ADC (module 0)	AN0 ⇒ AN1 ⇒ AN2 ⇒ AN3
	ADC (module 1)	AN4 ⇒ AN5 ⇒ AN6 ⇒ AN7
RX71M	S12AD	AN0 ⇒ AN1 ⇒ omitted ⇒ AN6 ⇒ AN7 ⇒ Temperature sensor output ⇒ Internal reference voltage It is possible to select group A priority control for group scan operation.
	S12AD1	AN100 ⇒ AN101 ⇒ omitted ⇒ AN119 ⇒ AN120 ⇒Temperature sensor output ⇒ Internal reference voltage It is possible to select group A priority control for group scan operation.

2.14.4 Operating Modes

Table 2.133 lists correspondences between the operating modes of the SH7216 Group and RX71M.

Table 2.133 Correspondences between A/D Converter Operating Modes

SH7216 Group	RX71M
Single-cycle scan	Single scan mode
Continuous scan	Continuous scan mode
—	Group scan mode When the specified synchronous trigger occurs, A/D conversion is performed once each on the multiple channels specified for each group. After A/D conversion completes for each group, an interrupt is generated if interrupts have been enabled.

2.14.5 Interrupts

ADC interrupts can be used to activate the DTC and DMAC on both the SH7216 Group and the RX71M.

On the RX71M the S12ADC interrupts are assigned to group interrupt BL1 and to software configurable interrupt B. The group BL1 interrupt status flag (GRPBL1.ISn) is cleared automatically when the corresponding bit in the module's status register is cleared. The software configurable interrupt B status flag (PIBRk.PIRn) is not cleared automatically, but there is no effect on the generation of interrupt requests.

Refer to 1.9, Interrupt Handling for information about interrupts.

2.14.6 Module Stop

As on the SH7216 Group, the S12ADC of the RX71M is set to the module-stop state after a reset, and no clock is supplied.

Refer to 2.21, Low Power Consumption Function for information on the module-stop state.

2.14.7 Setting Examples for A/D Conversion in Continuous Scan Mode

Setting examples are presented below for A/D conversion in continuous scan mode using the A/D converter module on the SH7216 Group and RX71M.

Operational Overview

- A/D conversion of analog input on three pins is performed continuously.
- A/D conversion is activated by a software trigger.
- Conversion data for the three pins is stored in the RAM when an A/D conversion-end interrupt occurs, and then operation continues.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by "SH7216:" in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.134 Operation Specifications for A/D Conversion in Continuous Scan Mode

Item		Description	Remarks
ADC channels		AN001, AN002, AN003 (unit 0)	SH7216: AN1, AN2, AN3
Operating mode		Continuous scan mode	
Sampling time		150 cycle (2.5 µs @60 MHz)	SH7216: 3 µs @50 MHz
Conversion accuracy		12-bit accuracy	
Conversion start trigger and cycle		Software trigger (repeating conversion after start)	
Extended analog input		Not used.	
Data alignment		Flush-left	
Interrupts		A/D conversion-end interrupt	Priority level: 5
Pins used	AN001	P41/AN001	SH7216: PF1/AN1
	AN002	P42/AN002	SH7216: PF2/AN2
	AN003	P43/AN003	SH7216: PF3/AN3
	LED0	P03/general	Lights when A/D conversion starts.

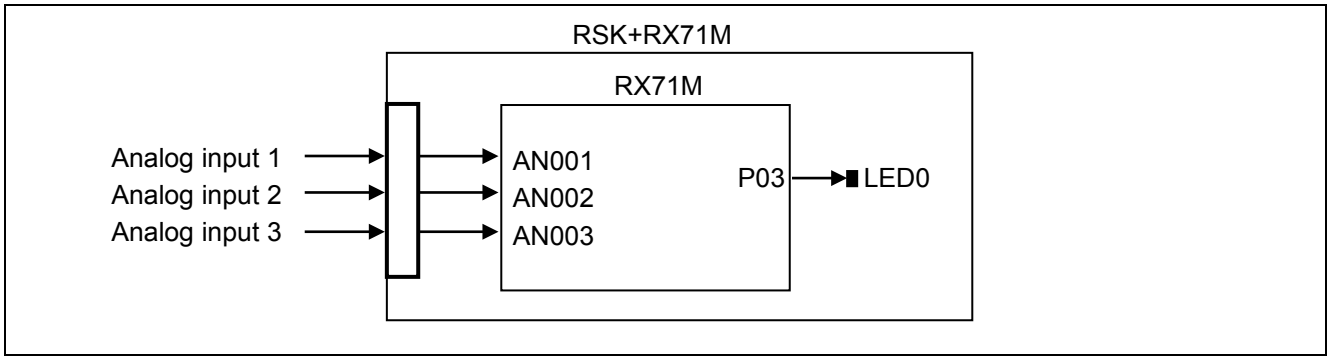


Figure 2.48 Operation Specifications for A/D Conversion in Continuous Scan Mode

Note

On the Renesas Starter Kit+ for RX71M in the initial state the pins used in the connection example are connected for use with the LCD, so make appropriate modifications to the board as necessary.

List of Related Registers

Table 2.135 shows the interrupt-related registers used in the SH7216 Group setting example broken down by source.

Table 2.135 SH7216 Group Interrupt-Related Registers (ADC and INTC)

			ADC		INTC
Item	Vector No.	Name	Interrupt Enable	Status	Priority Level
Setting register	—	—	ADC0.ADCR	ADC0.ADSR	IPR05
Setting position of each source					
A/D conversion-end	92	ADI0	ADIE	ADF	Bits 4 to 7

Table 2.136 and Table 2.137 show the interrupt-related registers used in the RX71M setting example broken down by source. On the RX71M the A/D conversion-end interrupt is assigned to software configurable interrupt B. The setting example uses vector 128 of software configurable interrupt B, as shown below:

- SLIBXR128 = 64: ADIE assigned to vector 128 of software configurable interrupt B.

Table 2.136 RX71M Interrupt-Related Registers (S12ADC)

Item	Name	Interrupt Enable	Status
Setting register	—	S12AD.ADCSR	—
Setting position of each source (S12AD)			
A/D conversion-end	S12ADI	ADIE	—

Table 2.137 RX71M Interrupt-Related Registers (ICUA)

Item	Vector No.	Name	Interrupt Enable	Status	Priority Level	Software Configurable Interrupt Source Selection
Setting register	—	—	IERm	IRr	IPRr	SLIBXRn
Setting position of each source (S12AD)						
A/D conversion-end	128	S12ADI	IER10.IEN0	IR128	IPR128	SLIBXR128 = 64

↑
Software configurable interrupt settings

When making settings to the ICUA interrupt-related registers on the RX71M, iodefine.h can be used to make settings as follows. The notation “xx” represents a vector number assigned to a software configurable interrupt.

- IERm : IEN (PERIB or INTBxx)
- IPRr : IPR (PERIB or INTBxx)
- IRr : IR (PERIB or INTBxx)

Processing Flowcharts

Figure 2.49 shows example flowcharts of processing using the ADC. The names of the processing steps shown in the flowcharts correspond to the names in the setting examples.

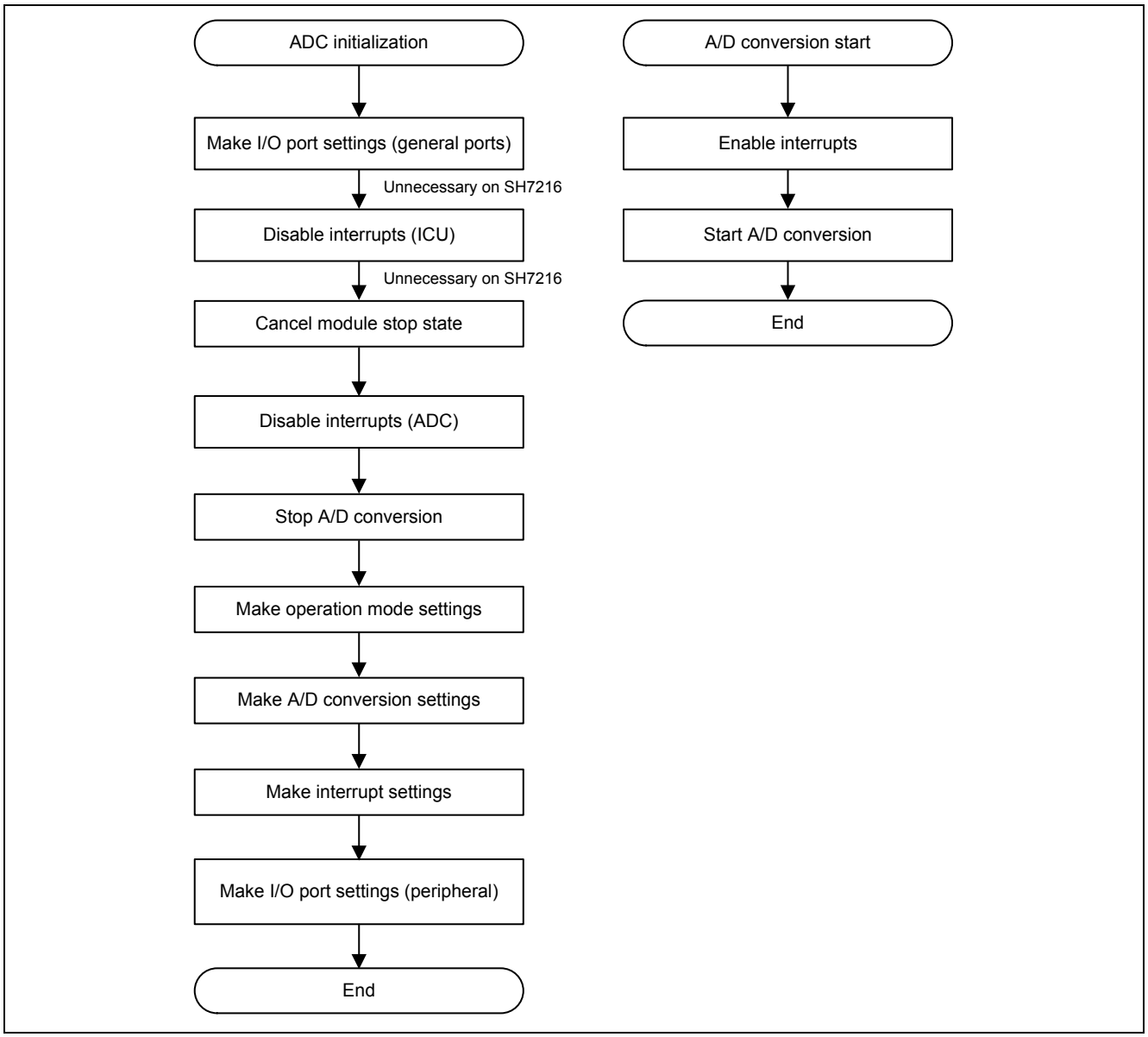


Figure 2.49 Example Flowcharts of ADC Processing

Setting Examples

Initial setting examples for A/D conversion in continuous scan mode are presented below. The names of the processing steps in these setting examples correspond to the names in the flowcharts.

Table 2.138 ADC Initialization Setting Examples

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make I/O port settings (general ports).	—	Pin direction settings PORT4.PDR.B1 = 0b (input/P41) PORT4.PDR.B2 = 0b (input/P42) PORT4.PDR.B3 = 0b (input/P43) Pin mode settings (general) PORT4.PMR.B1 = 0b (P41) PORT4.PMR.B2 = 0b (P42) PORT4.PMR.B3 = 0b (P43)
Disable interrupt (ICU).	—	Disabling of ICU interrupts IEN(PERIB, INTB128) = 0b
Cancel module stop state.	STB.CR3._ADC0 = 0b	SYSTEM.PRCR = A502h SYSTEM.MSTPCRA.MSTPA17 = 0b SYSTEM.PRCR = A500h
Disable interrupt (ADC).	ADC0.ADCR.ADIE = 0b (ADI0)	S12AD.ADCSR.ADIE = 0b (S12ADI)
Stop A/D conversion.	ADC0.ADCR.ADST = 0b	S12AD.ADCSR.ADST = 0b
Make operation mode settings.	ADC0.ADCR.ADCS = 1b	S12AD.ADCSR.ADCS = 10b
Make A/D conversion settings.	Channel setting ADC0.ADANSR = 0Eh	Channel setting S12AD.ADANSA0 = 000Eh Sampling time settings S12AD.ADSSTR1 = 150 S12AD.ADSSTR2 = 150 S12AD.ADSSTR3 = 150 Register format setting S12AD.ADCER.ADRFMT = 0b Accuracy setting S12AD.ADCER.ADPRC = 00b
Make interrupt settings.	Priority setting (level 5) INTC.IPR05._AD0 = 5(ADI0) Clearing of ADC status* ADC0.ADSR.ADF = 0b (ADI0) * Clear to 0 after reading value as 1.	Software configurable interrupt vector assignments ICU.SLIBXR128 = 64 (S12ADI) Protecting the software configurable interrupt source select register ICU.SLIPRCR.WPRC = 1b*1 Reading value of ICU.SLIPRCR.WPRC Priority setting (level 5) IPR(PERIB, INTB128) = 5(S12ADI) Clearing of ICU status IR(PERIB, INTB128) = 0b (S12ADI)

Processing	SH7216 Group Setting Example	RX71M Setting Example
Make I/O port settings (analog).	—	Cancellation of register protection MPC.PWPR.B0WI = 0b MPC.PWPR.PFSWE = 1b Pin peripheral function selection MPC.P41PFS.ASEL = 1b (P41/analog pin) MPC.P42PFS.ASEL = 1b (P42/analog pin) MPC.P43PFS.ASEL = 1b (P43/analog pin) Register protection settings MPC.PWPR.PFSWE = 0b MPC.PWPR.B0WI = 1b

Note 1. Once ICU.SLIPRCR.WPRC is set to 1, it cannot be cleared to 0 by software.

Table 2.139 A/D Conversion Start Setting Examples

Processing	SH7216 Group Setting Example	RX71M Setting Example
Enable interrupts.	Enabling of ADC interrupts ADC0.ADCR.ADIE = 1b (ADI0)	Enabling of ADC interrupts S12AD.ADCSR.ADIE = 1b (S12ADI) Enabling of ICU interrupts IEN(PERIB, INTB128) = 1b
Start counting.	ADC0.ADCR.ADST = 1b	S12AD.ADCSR.ADST = 1b

2.15 CAN

2.15.1 Comparison of Specifications

Controller area network functionality is provided on the SH7216 Group by the RCAN-ET module and on the RX71M by the CAN module (CAN).

Table 2.140 is a comparative specifications of the SH7216 Group and RX71M.

Table 2.140 Comparison of SH7216 Group and RX71M Specifications (CAN)

Item	SH7216 Group (RCAN-ET)	RX71M (CAN)
Number of channels	1 channel	3 channels
Protocol	Support for CAN standard 2.0B ISO-11898 compliant bit timing	ISO 11898-1 compliant
Clock source	Peripheral bus clock (Pϕ) 20 to 50 MHz	Peripheral module clock (PCLKB) or CAN clock (CANMCLK)
Bit rate	Max. 1 Mbps	Max. 1 Mbps
Mailboxes per channel	(Equivalent to normal mailbox mode on RX71M) Transmit/receive: 15 Receive: 1	Normal mailbox mode • Transmit/receive: 32 FIFO mailbox mode • Transmit/receive: 24 • Transmit: 4-stage FIFO • Receive: 4-stage FIFO
Supported ID selection	• Both standard ID and extended ID	• Standard ID • Extended ID • Both standard ID and extended ID
Test functions	• Listen-only mode • Self-test mode 1 (external) • Self-test mode 2 (internal) • Write error counter • Error-passive mode	• Listen-only mode • Self-test mode 0 (external) • Self-test mode 1 (internal)
DTC/DMAC activation	DTC/DMAC activation supported	No
Interrupt source	• Data frame receive • Remote frame receive • Message transmit/transmit cancel • 2 error systems	• Receive-end • Transmit-end • Receive FIFO • Transmit FIFO • Error
Other	• HCAN2-compatible ID rearrangement • Auto-wakeup from CAN sleep mode • Auto-transmit of data frames • Acceptance filter	• Time stamp function • One-shot receive • Mailbox search support • Channel search support • Acceptance filter support

2.15.2 Mailboxes

The SH7216 Group has 16 mailboxes, each comprising 18 bytes. Figure 2.50 shows the mailbox configuration on the SH7216 Group.

- Mailbox 0: Receive-only mailbox
- Mailboxes 1 to 15: Transmit/receive mailboxes

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDE	RTR	0	STDID[10:0]											EXTID[17:16]	
EXTID[15:0]															
IDE_ LAFM	0	0	STDID_LAFM[10:0]											EXTID_ LAFM[17:16]	
EXTID_LAFM[15:0]															
MSG_DATA_0								MSG_DATA_1							
MSG_DATA_2								MSG_DATA_3							
MSG_DATA_4								MSG_DATA_5							
MSG_DATA_6								MSG_DATA_7							
0	0	NMC	ATX*1	DART*1	MBC[2:0]			0	0	0	0	DLC[3:0]			

Note 1. ATX and DART are not supported by mailbox 0.

Figure 2.50 SH7216 Group Mailbox Configuration

The SH71M has 32 mailboxes, each comprising 16 bytes. Figure 2.51 shows the mailbox configuration on the RX71M.

Normal mailbox mode

- Mailboxes 0 to 31: Transmit/receive mailboxes

FIFO mailbox mode

- Mailboxes 0 to 23: Transmit/receive mailboxes
- Mailboxes 24 to 27: Mailboxes for transmit FIFO
- Mailboxes 28 to 31: Mailboxes for receive FIFO

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDE	RTR	—	SID[10:0]											EID[17:16]	
EID[15:0]															
—	—	—	—	—	—	—	—	—	—	—	—	DLC[3:0]			
DATA_0								DATA_1							
DATA_2								DATA_3							
DATA_4								DATA_5							
DATA_6								DATA_7							
TSH[7:0]								TSL[7:0]							

Figure 2.51 RX71M Mailbox Configuration

Some mailbox items on the SH7216 Group are accomplished on the RX71M by means of register settings. Table 2.141 compares mailbox settings on the SH7216 Group and the RX71M.

Guide to Symbols in "Changes" Column of Table

◎: Same setting on SH7216 Group and RX71M

△: Different setting on SH7216 Group and RX71M

—: Register not present on SH7216 Group or RX71M

Table 2.141 Comparison of Mailbox Settings on SH7216 Group and RX71M

SH7216 Group*1	RX71M*2	Changes
MB[x].CONTROL0.IDE	MBj.IDE	◎
MB[x].CONTROL0.RTR	MBj.RTR	◎
MB[x].CONTROL0.STDID[10:0]	MBj.SID[10:0]	◎
MB[x].CONTROL0.EXTID[17:0]	MBj.EID[17:0]	◎
MB[x].LAFM.IDE_LAFM	—	—
MB[x].LAFM.STDID_LAFM[10:0]	MKRk.SID[10:0], MKIVLR register	△
MB[x].LAFM.EXTID_LAFM[17:0]	MKRk.EID[17:0], MKIVLR register	△
MB[x].MSG_DATA[0 to 7].MSG_DATA_0 to 7	MBj.DATA0 to MBj.DATA7	◎
MB[x].CONTROL1.NMC	CTLR.MLM (channel unit)	△
MB[x].CONTROL1.ATX	—	—
MB[x].CONTROL1.DART	MCTLj.ONESHOT	△
MB[x].CONTROL1.MBC[2:0]	MCTLj.RECREQ, MCTLj.TRMREQ	△
MB[x].CONTROL1.DLC[3:0]	MBj.DLC[3:0]	◎
—	MBj.TSL[7:0] MBj.TSH[7:0]	—

Note 1. x: 0 to 15

Note 2. j: 0 to 31, k: 0 to 7

2.15.3 Acceptance Filtering

Both the SH7216 Group and RX71M support acceptance filtering, which enables mailboxes to accept messages with multiple receive IDs.

Whereas on the SH7216 Group acceptance filter settings are made to the local acceptance filter mask (LAFM) in each mailbox, on the RX71M they are made to mask register k (MKRk) and the mask invalid register (MKIVLR).

Table 2.142 Acceptance Filter Setting Specifications

Item	SH7216 Group	RX71M
Target	IDE Standard ID Extended ID	Standard ID Extended ID
Mask settings	Local acceptance filter mask (LAFM): Individual mask settings for each mailbox 0: Compare target ID bit. 1: Do not compare target ID bit.	Mask register k (MKRk): Individual mask settings for four mailboxes 0: Compare target ID bit. 1: Do not compare target ID bit. Mask invalid register (MKIVLR): Mask enable/disable settings for each mailbox

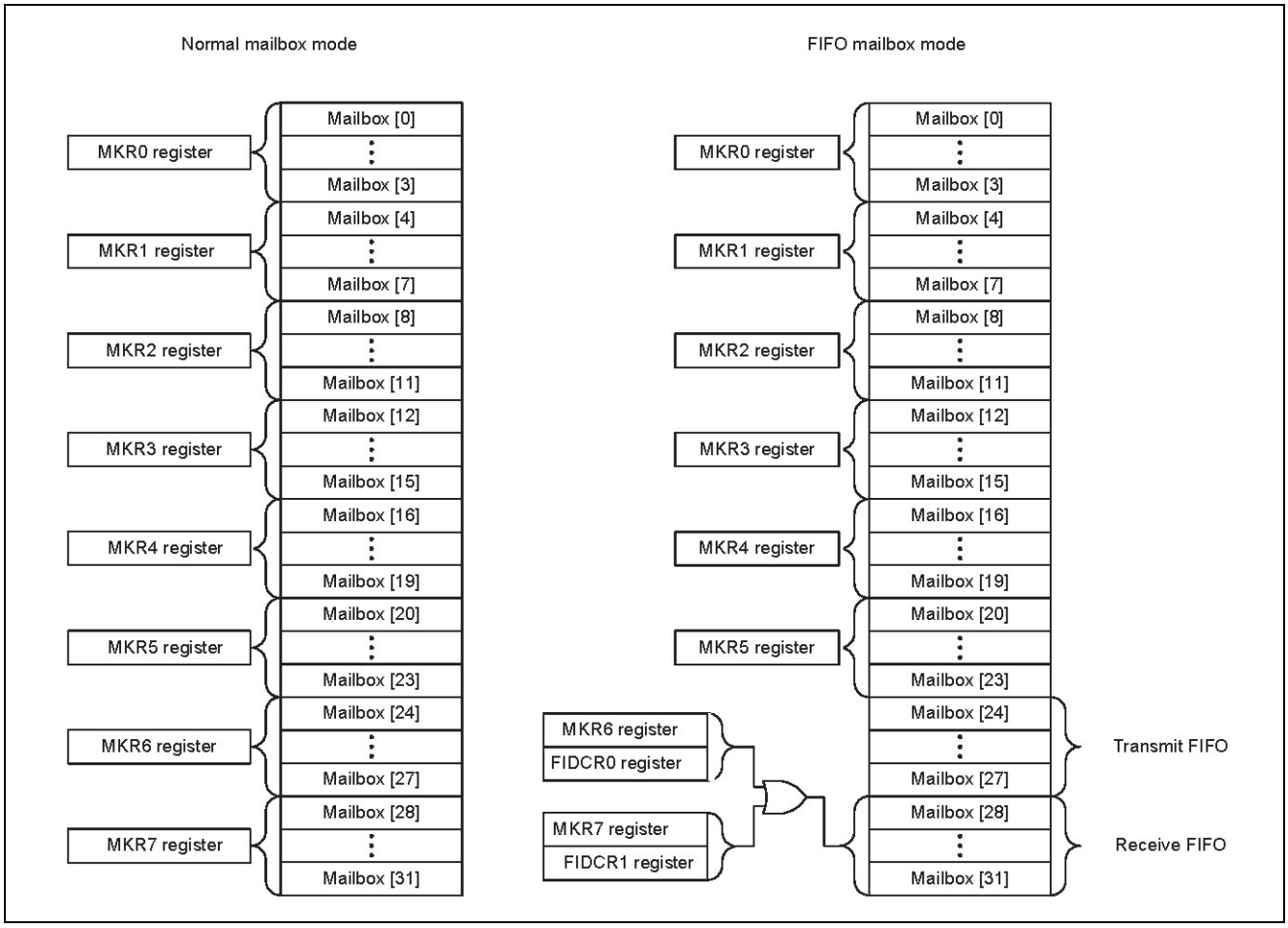


Figure 2.52 Correspondence of Mask Registers and Mailboxes on RX71M

2.15.4 Transmission Priority

On both the SH7216 Group and the RX71M it is possible to select the priority of message transmission.

The mailbox numbers and their priority when mailbox number priority mode is selected differ on the SH7216 Group and on the RX71M. Table 2.143 shows the transmission priority specifications of the SH7216 Group and the RX71M.

Table 2.143 Transmission Priority Specifications

Item	SH7216 Group	RX71M
ID priority	The message with the arbitration field having the lowest digital value has the highest priority (ISO 11898-1 compliant).	
Mailbox number priority	The highest mailbox number has the highest priority Mailbox 15 to mailbox 1	The lowest mailbox number has the highest priority Mailbox 0 to mailbox 31

2.15.5 Mode Transitions

Whereas on the SH7216 Group a transition to configuration mode occurs after a hardware reset, on the RX71M a transition to CAN sleep mode occurs.

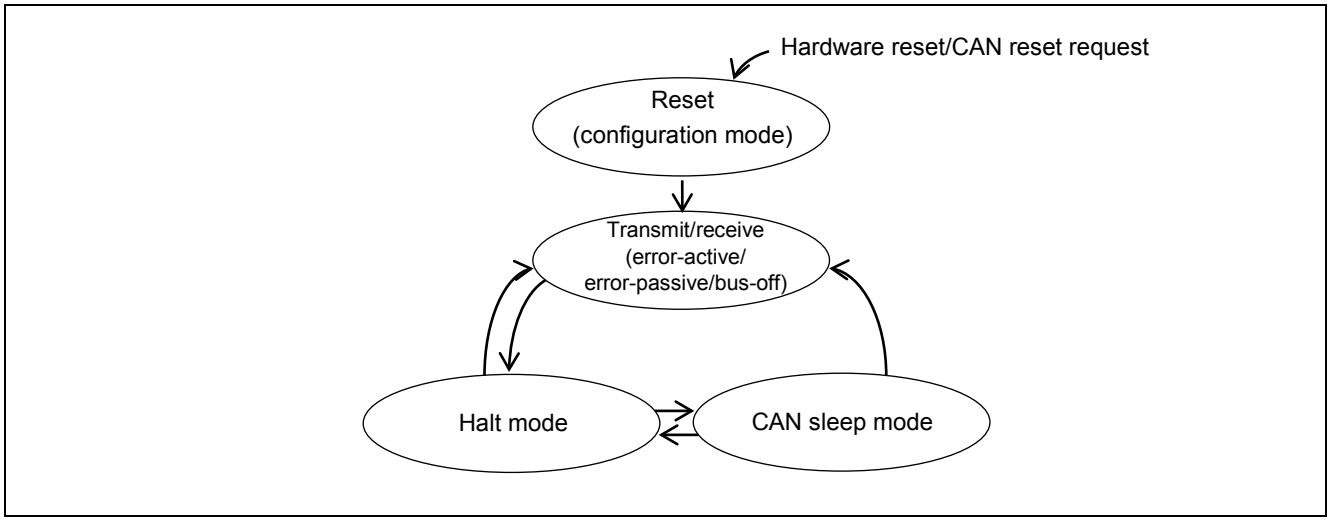


Figure 2.53 SH7216 Group State Transitions

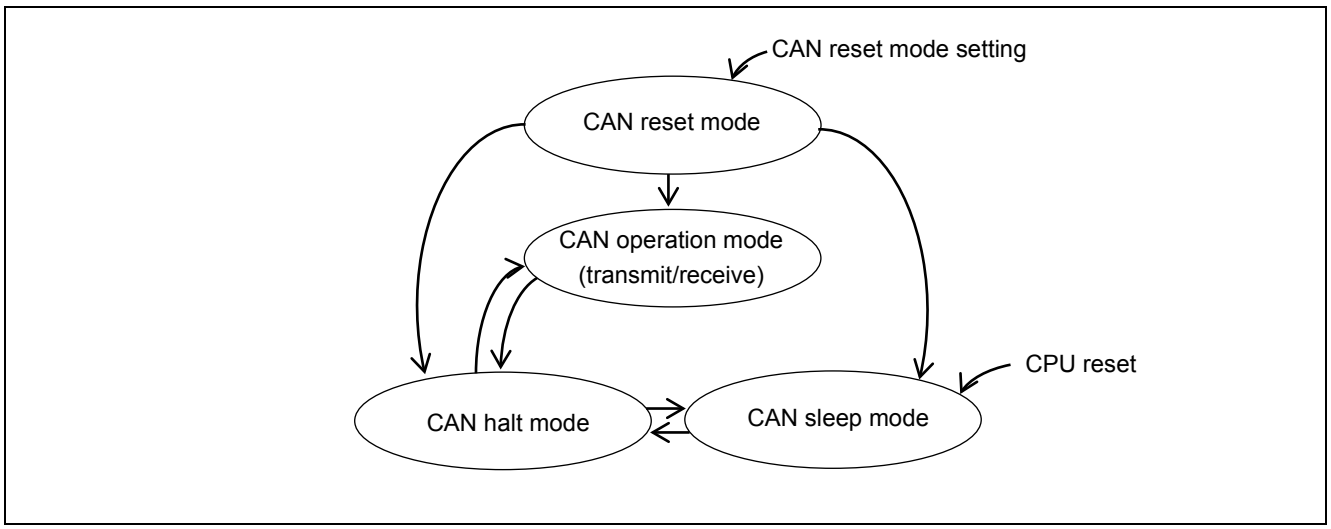


Figure 2.54 RX71M State Transitions

2.15.6 Interrupts

Whereas on the SH7216 Group a data frame receive or remote frame receive interrupt can be used to activate the DTC and the DMA, on the RX71M no CAN interrupts can be used to activate the DTC and the DMAC.

On the RX71M CAN interrupts are assigned to the group BE0 interrupt and to software configurable interrupt B. The group BE0 interrupt status flag (GRPBE0.ISn) is cleared when 1 is written to the interrupt source clear bit (GCRBE0.CLRn). The software configurable interrupt B status flag (PIBRk.PIRn) is not cleared automatically, but it has no effect on the generation of interrupt requests even if left uncleared.

Refer to 1.9, Interrupt Handling for information about interrupts.

2.15.7 Module Stop

As on the SH7216 Group, the CAN of the RX71M is set to the module-stop state after a reset, and no clock is supplied.

Refer to 2.21, Low Power Consumption Function for information on the module-stop state.

2.16 USB

2.16.1 Comparison of Specifications

Support for USB 2.0 is provided on the SH7216 Group by the USB function module (USB) and on the RX71M by the USB 2.0 FS host/function module (USBb), which supports full-speed and low-speed transfer as defined in Universal Serial Bus (USB) Specification 2.0, and by the USB 2.0 high-speed host/function module (USBAa), which also supports high-speed mode.

Table 2.144 is a comparative specifications of the SH7216 Group and RX71M.

Table 2.144 Comparison of SH7216 Group and RX71M Specifications (USB)

	SH7216 Group	RX71M	
Item	**USB**	**USBb**	**USBAa**
Controller functions	• Function controller function	• Host controller function • Function controller function • On-The-Go (OTG)	
Clock source	USB clock (Uϕ)	Peripheral module clock (PCLKB) USB clock (UCLK)	Peripheral module clock (PCLKA, PCLKB) USB clock (UCLK) USBA clock (USBMCLK)*1
Transfer speed	• Full-speed mode	• Low-speed mode*2 • Full-speed mode	• Low-speed mode*2 • Full-speed mode • High-speed mode
Communication data transfer types	• Control transfer • Bulk transfer • Interrupt transfer	• Control transfer • Bulk transfer • Interrupt transfer • Isochronous transfer	
Power modes	• Self-power mode	• Self-power mode • Bus-power mode	
Endpoints/pipes	Endpoints: Up to 10	Pipes: Up to 10 The endpoint numbers assigned to pipes 1 to 9 are selectable.	
DTC/DMAC activation	DTC/DMAC activation supported	DTC/DMAC activation supported	
Other	• D+ line pull-up control pin (PUPD) • Ability to switch to low-power mode when USB cable is disconnected or internal clock for protocol processing stops	• Incorporation into MCU of D+/D– line pull-up and pull-down resistors • Support for following functions when function controller is selected: — Control transfer stage management function — Device state management function — SOF recovery function	

Note 1. USBMCLK is supplied to the PLL incorporated into the USB-PHY of the USBA, and the PHY clock generation method is selectable.

Note 2. Supported by host controller only.

2.17 Ethernet Controller (EtherC)

2.17.1 Comparison of Specifications

Ethernet controller functionality with support for the Ethernet and IEEE 802.3 MAC layer protocol standards is provided on the SH7216 Group by the EtherC and on the RX71M by the ETHERC.

Direct memory access controller functionality for the Ethernet controller is provided on the SH7216 Group by the E-DMAC and on the RX71M by the EDMACa.

The RX71M also integrates a PTP module for the Ethernet controller (EPTPCa) to handle synchronization between devices.

Table 2.145 and Table 2.146 are comparative specifications of the SH7216 Group and RX71M.

Table 2.145 Comparison of SH7216 Group and RX71M Specifications (EtherC)

Item	SH7216 Group (EtherC)	RX71M (ETHERC)
Number of input channels	1 channel	2 channels
Protocol	Flow control compliant with IEEE 802.3x	Flow control compliant with IEEE 802.3x
Data transmission/ reception	Frame transmission/reception compliant with Ethernet/IEEE 802.3	Frame transmission/reception compliant with Ethernet/IEEE 802.3
Transfer speed	10 Mbps 100 Mbps	10 Mbps 100 Mbps
Communication mode	Full-duplex communication Half-duplex communication	Full-duplex communication Half-duplex communication
Inerface	MII compliant with IEEE 802.3u	MII and RMII compliant with IEEE 802.3u
Other	Magic Packet™*1 detection Wake-On-LAN (WOL) signal output	Magic Packet™*1 detection Wake-On-LAN (WOL) signal output

Note 1. Magic Packet is a trademark of Advanced Micro Devices, Inc.

Table 2.146 Comparison of SH7216 Group and RX71M Specifications (E-DMAC)

Item	SH7216 Group (E-DMAC)	RX71M (EDMACAa)
Number of channels	1 channel: EtherC	2 channels: ETHERC 1 channel: EPTPCa
Data transfer	Transmission/reception control using descriptors	Transmission/reception control using descriptors
Transfer methods	• Single frame transmission/reception • Multi-buffer transmission/reception	• Single-buffer frame transmission/ reception • Multi-buffer frame transmission/reception
Transfer unit	Block transfer (32-byte units)	Block transfer (32-byte units)
Other	• Reflection in descriptor of transmit/ receive frame status	• Reflection in descriptor of transmit/ receive frame status • Insertion of padding in receive data

2.18 Compare Match Timer (CMT)

2.18.1 Comparison of Specifications

Compare match timer functionality is provided on the SH7216 Group by the CMT and on the RX71M by the CMT, which has a 16-bit timer, and the CMTW, which has a 32-bit timer.

The RX71M includes all the CMT functionality of the SH7216 Group (backward compatibility). Table 2.147 provides a comparative listing of the specifications of the SH7216 Group and RX71M.

Table 2.147 Comparison of SH7216 Group and RX71M Specifications (CMT)

	SH7216 Group	RX71M	
Item	**CMT**	**CMT**	**CMTW**
Number of units (channels)	1 unit (total 2 channels)	2 units (total 4 channels)	2 units (total 2 channels)
Clock source	Internal clock (Pϕ)	Peripheral module clock (PCLKB)	Peripheral module clock (PCLKB)
Clock frequency division ratio	Pϕ/8, 32, 128, 512	PCLKB/8, 32, 128, 512	PCLKB/8, 32, 128, 512
Count operation	16-bit up-counter	16-bit up-counter	Max. 32-bit up-counter (selectable between 16 and 32 bits)
DTC/DMAC activation	DTC/DMAC activation supported	DTC/DMAC activation supported	DTC/DMAC activation supported
Interrupt sources	• Compare match	• Compare match	• Compare match • Input compare • Output compare
Other	—	Event link	Event link

2.18.2 Register Comparison

The CMT of the RX71M does not have interrupt flags, but equivalent processing can be accomplished by using the interrupt controller.

Table 2.148 and Table 2.149 are a comparative listing of the registers on the SH7216 Group and RX71M.

Guide to Symbols in "Changes" Column of Table

◎: Register with same bit assignments on SH7216 Group and RX71M

△: Register with different bit assignments on SH7216 Group and RX71M

—: Register not present on SH7216 Group or RX71M

Table 2.148 SH7216 Group and RX71M Register Comparison (CMT)

SH7216 Group (CMT)*1	RX71M (CMT)*2	Changes
Compare match timer start register (CMSTR)	Compare match timer start register 0 (CMSTR0) Compare match timer start register 1 (CMSTR1)	◎
Compare match timer control/ status register n (CMCSR_n)	Compare match timer control register (CMTm.CMCR)	△
Compare match counter n (CMCNT_n)	Compare match timer counter (CMTm.CMCNT)	◎
Compare match constant register n (CMCOR_n)	Compare match constant register (CMTm.CMCOR)	◎

Note 1. CMT n: 0 or 1
Note 2. CMT m: 0 to 3

Table 2.149 SH7216 Group and RX71M Register Comparison (CMTW)

SH7216 Group (CMT)*1	RX71M (CMTW)*2	Changes
Compare match timer start register (CMSTR)	Timer start register (CMTWm.CMWSTR)	△
Compare match timer control/ status register n (CMCSR_n)	Timer control register (CMTWm.CMWCR)	△
Compare match counter n (CMCNT_n)	Timer counter (CMTWm.CMWCNT)	△
Compare match constant register n (CMCOR_n)	Compare match constant register (CMTWm.CMWCOR)	△
—	Timer I/O control register (CMTWm.CMWIOR)	—
	Input capture registers 0 and 1 (CMTWm.CMWICR0 and CMTWm.CMWICR1)	
	Output compare registers 0 and 1 (CMTWm.CMWOCR0 and CMTWm.CMWOCR1)	

Note 1. CMT n: 0 or 1
Note 2. CMTW m: 0 or 1

2.18.3 Interrupts

CMT interrupts can be used to activate the DTC and the DMAC on both the SH7216 Group and the RX71M.

On the RX71M some of the CMT and CMTW interrupts are assigned to software configurable interrupt B. The interrupt controller's interrupt status flag (IRn.IR) is cleared automatically when the corresponding interrupt is accepted. The software configurable interrupt B status flag (PIBRk.PIRn) is not cleared automatically, but there is no effect on the generation of interrupt requests.

Refer to 1.9, Interrupt Handling for information about interrupts.

2.18.4 Module Stop

As on the SH7216 Group, the CMT of the RX71M is set to the module-stop state after a reset, and no clock is supplied.

Refer to 2.21, Low Power Consumption Function for information on the module-stop state.

2.18.5 Compare Match Timer Setting Example

Setting examples for flashing the LED at regular intervals by using the compare match timer on the SH7216 Group and the RX71M are presented below.

Operational Overview

- Continuous operation is performed in which LED1 is inverted when a compare match interrupt occurs.

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by "SH7216:" in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.150 Compare Match Timer Operation Specifications

Item	Description	Remarks
CMT channel	CMT0	
Count clock	PCLKB/512 (PCLKB = 60 MHz)	SH7216: Pϕ/512 (ϕ = 50 MHz)
Count value (CMCOR)	0xE4E1 (0.5s @60 MHz/512)	SH7216: 0xBEBB (0.5s @50 MHz/512)
Interrupts	Compare match interrupt	Priority level: 5
Pins used LED1	P05/general	Turns on and off at 0.5-second intervals.

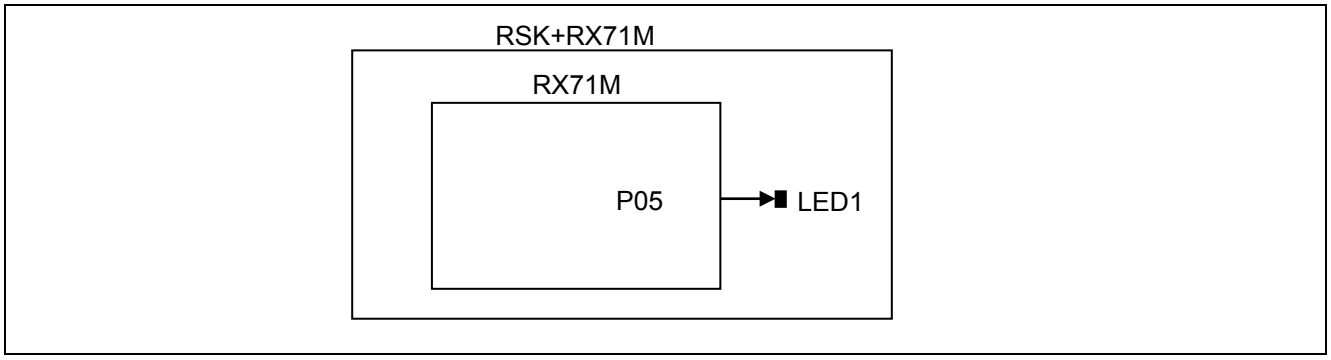


Figure 2.55 Compare Match Timer Connection Example

List of Related Registers

Table 2.151 shows the interrupt-related registers used in the SH7216 Group setting example broken down by source.

Table 2.151 SH7216 Group Interrupt-Related Registers (CMT and INTC)

			CMT		INTC
Item	**Vector No.**	**Name**	**Interrupt Enable**	**Status**	**Priority Level**
Setting register	—	—	CMT0.CMCSR	CMT0.CMCSR	IPR08
Setting position of each source (CMT0)					
Compare match	140	CMI0	CMIE	CMF	Bits 12 to 15

Table 2.152 and Table 2.153 show the interrupt-related registers used in the RX71M setting example broken down by source.

Table 2.152 RX71M Interrupt-Related Registers (CMT)

Item	Name	Interrupt Enable	Status
Setting register	—	CMT0.CMCR	—
Setting position of each source (CMT0)			
Compare match	CMI0	CMIE	—

Table 2.153 RX71M Interrupt-Related Registers (ICUA)

Item	Vector No.	Name	Interrupt Enable	Status	Priority Level
Setting register	—	—	IERm	IRr	IPRr
Setting position of each source (CMT0)					
Compare match	28	CMI0	IER03.IEN4	IR028	IPR004

When making settings to the ICUA interrupt-related registers on the RX71M, iodefine.h can be used to make settings as follows.

- IERm : IEN (CMT0 or ICUA interrupt name)
- IPRr : IPR (CMT0 or ICUA interrupt name)
- IRr : IR (CMT0 or ICUA interrupt name)

Processing Flowcharts

Figure 2.56 shows example flowcharts of processing using the CMT. The names of the processing steps shown in the flowcharts correspond to the names in the setting examples.

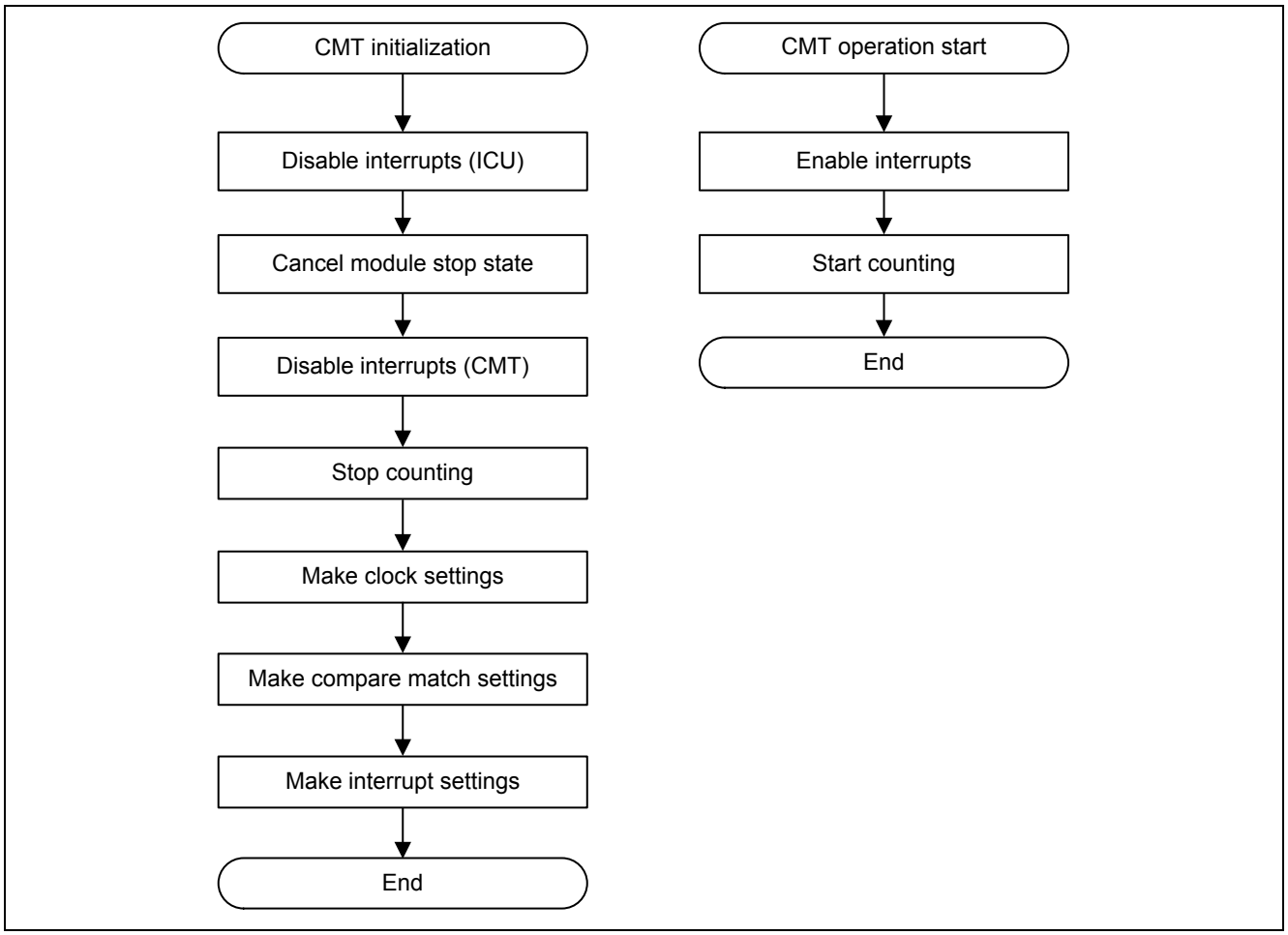


Figure 2.56 Example Flowcharts of CMT Processing

Setting Examples

Compare match timer setting examples are presented below. The names of the processing steps in these setting examples correspond to the names in the flowcharts.

Table 2.154 CMT Initialization Setting Examples

Procedure	SH7216 Group Setting Example	RX71M Setting Example
Disable interrupts (ICU).	—	Disabling of ICU interrupts IEN(CMT0, CMI0) = 0b
Cancel module stop state.	STB.CR4._CMT = 0b	SYSTEM.PRCR = A502h SYSTEM.MSTPCRA.MSTPA15 = 0b SYSTEM.PRCR = A500h
Disable interrupts (CMT).	Disabling of ICU interrupts CMT0.CMCSR.CMIE = 0b (CMI0)	Disabling of ICU interrupts CMT0.CMCR.CMIE = 0b (CMI0)
Stop counting.	CMT.CMSTR.STR0 = 0b	CMT.CMSTR0.STR0 = 0b
Make clock settings.	CMT0.CMCSR.CKS = 11b	CMT0.CMCR.CKS = 11b
Make compare match settings.	Cycle setting CMT0.CMCOR = BEBBh Clearing the counter CMT0.CMCNT = 0	Cycle setting CMT0.CMCOR = E4E1h Clearing the counter CMT0.CMCNT = 0
Make interrupt settings.	Priority setting (level 5) INTC.IPR08._CMT0 = 5 Clearing of CMT status flag* CMT0.CMCSR.CMF = 0b (CMI0) * Clear to 0 after reading value as 1.	Priority setting (level 5) IPR(CMT0, CMI0) = 5 Clearing of SCIF status flag IR(CMT0, CMI0) = 0b

Table 2.155 CMT Operation Start Setting Example

Processing	SH7216 Group Setting Example	RX71M Setting Example
Enable interrupts.	Enabling of CMT interrupts CMT0.CMCSR.CMIE = 1b (CMI0)	Enabling of CMT interrupts CMT0.CMCR.CMIE = 1b (CMI0) Enabling of ICU interrupts IEN(CMT0, CMI0) = 1b
Start counting.	CMT.CMSTR.STR0 = 1b	CMT.CMSTR0.STR0 = 1b

2.19 Code Flash Memory

2.19.1 Comparison of Specifications

Table 2.156 is a comparative listing of the specifications of the SH7216 Group and RX71M.

Table 2.156 Comparison of SH7216 Group and RX71M Specifications (Code Flash Memory)

Item	SH7216 Group	RX71M
Size	User MAT: Max. 1 MB User boot MAT: 32 KB	User area: Max. 4 MB User boot area: 32 KB Option-setting area: 256 bytes
Block size × block count	1 MB products • 128 KB × 3 blocks • 64 KB × 8 blocks • 8 KB × 8 blocks 768 KB product • 128 KB × 1 block • 64 KB × 9 blocks • 8 KB × 8 blocks 512 KB product • 64 KB × 7 blocks • 8 KB × 8 blocks	4 MB products • 32 KB × 126 blocks • 8 KB × 8 blocks 3 MB products • 32 KB × 94 blocks • 8 KB × 8 blocks 2.5 MB products • 32 KB × 78 blocks • 8 KB × 8 blocks 2 MB products • 32 KB × 62 blocks • 8 KB × 8 blocks
Write unit	256 bytes	256 bytes
Erase unit	User MAT • Writer mode: Erasure of entire area • Other than writer mode: Block units User boot MAT: Erasure of entire area	User area • Writer mode: Erasure of entire area • Other than writer mode: Block units User boot area: Erasure of entire area Option-setting area: Erasure of entire area
Write/erase count	1,000 times	1,000 times
Programming modes	On-board programming • Boot mode • USB boot mode • User boot mode • User programming mode Off-board programming • Writer mode	On-board programming • Boot mode (SCI interface) • Boot mode (USB interface) • User boot mode • Programming by a routine for code flash memory programming within a user program Off-board programming • Programming with flash writer
Other	• Automatic bit rate matching • Protect mode • Suspend/resume function • BGO function (ability to run programs not assigned to code flash memory while programming or erasing of code flash memory is in progress) • ROM caching for faster operation	• Automatic bit rate matching • Protection function (prevention of unintentional overwriting) • Suspend/resume function • BGO function (ability to read code flash memory while programming of code flash memory is in progress) • Acceleration by advanced fetch unit (AFU) • Security function (prevention of unauthorized modification/reading) • TM function (prevention of unauthorized reading) • 12-byte unique ID

On the RX71M FACI commands can be used to program the code flash memory. Refer to the following application note for details:

RX64M Group, RX71M Group Flash Memory User's Manual: Hardware Interface (R01UH0435EJ0110)

2.20 Data Flash Memory

2.20.1 Comparison of Specifications

Table 2.157 is a comparative listing of the specifications of the SH7216 Group and RX71M.

Table 2.157 Comparison of SH7216 Group and RX71M Specifications (Data Flash Memory)

Item	SH7216 Group	RX71M
Size	Data MAT: 32 KB	Data area: 64 KB
Block size × block count	8 KB × 4 blocks	64 KB × 1,024 blocks
Write unit	Boot mode : 256 bytes Other than boot mode: 8-byte or 128-byte units	4 bytes
Erase unit	Block units	64 bytes
Write/erase count	30,000 times	100,000 times
Programming modes	On-board programming • Boot mode • USB boot mode • User boot mode • User mode/user programming mode	On-board programming • Boot mode (SCI interface) • Boot mode (USB interface) • User boot mode • Programming by a routine for data flash memory programming within a user program
Other	• Automatic bit rate matching • Protect mode • Suspend/resume function • BGO function (ability to run programs from code flash memory while programming or erasing of data flash memory is in progress) • Blank checking function	• Automatic bit rate matching • Protection function (prevention of unintentional overwriting) • Suspend/resume function • BGO function (ability to read code flash memory while programming of data flash memory is in progress) • Blank checking function • Security function (prevention of unauthorized modification/reading) • 12-byte unique ID

On the RX71M FACI commands can be used to program the code flash memory. Refer to the following application note for details:

RX64M Group, RX71M Group Flash Memory User's Manual: Hardware Interface (R01UH0435EJ0110)

2.21 Low Power Consumption Function

2.21.1 Comparison of Mode Specifications

Table 2.158 and Table 2.159 summarize the methods for transitioning to and canceling the various low-power states on the SH7216 Group and RX71M, and list the operating states of the clock, CPU, and on-chip modules.

Table 2.158 SH7216 Group Low-Power States

Transition and Cancelation Methods, and Operating States	Sleep Mode	Module Standby Function	Software Standby Mode
Transition method	Control register + instruction	Control register	Control register + instruction
Cancelation method other than reset	Interrupt DMA address error	Control register	Interrupt
Clock	Operating	Operating	Stopped
CPU	Stopped	Operating	Stopped
On-chip peripheral modules	Operating	Specified modules stopped	Stopped

Table 2.159 RX71M Low-Power States

Transition and Cancelation Methods, and Operating States	Sleep Mode	All-Module Clock Stop Mode	Software Standby Mode	Deep Software Standby Mode
Transition method	Control register + instruction	Control register + instruction	Control register + instruction	Control register + instruction
Cancelation method other than reset	Interrupt	Interrupt	Interrupt	Interrupt
Main clock oscillator, sub-clock oscillator	Operation possible	Operation possible	Operation possible	Operation possible
High-speed on-chip oscillator, low-speed on-chip oscillator	Operation possible	Operation possible	Stopped	Stopped
IWDT dedicated on-chip oscillator	Operation possible	Operation possible	Operation possible	Stopped (settings undetermined)
PLL	Operation possible	Operation possible	Stopped	Stopped
CPU	Stopped (settings retained)	Stopped (settings retained)	Stopped (settings retained)	Stopped (settings undetermined)
RAM	Operation possible (settings retained)	Stopped (settings retained)	Stopped (settings retained)	Stopped (settings undetermined)
Standby RAM	Operation possible (settings retained)	Stopped (settings retained)	Stopped (settings retained)	Stopped (settings retained/ undetermined)*1
Flash memory	Operation possible	Stopped (settings retained)	Stopped (settings retained)	Stopped (settings retained)
USBFS host/function module (USBb)	Operation possible	Stopped	Stopped	Stopped (settings retained/ undetermined)*1
USBFS host/function module (USBA)	Operation possible	Stopped	Stopped	Stopped (settings retained/ undetermined)*1

Transition and Cancelation Methods, and Operating States	Sleep Mode	All-Module Clock Stop Mode	Software Standby Mode	Deep Software Standby Mode
Watchdog timer (WDT)	Stopped (settings retained)	Stopped (settings retained)	Stopped (settings retained)	Stopped (settings undetermined)
Independent watchdog timer (IWDT)	Operation possible	Operation possible	Operation possible	Stopped (settings undetermined)
Realtime clock (RTC)	Operation possible	Operation possible	Operation possible	Operation possible
8-bit timer (TMR)	Operation possible	Operation possible	Stopped (settings retained)	Stopped (settings undetermined)
Voltage detection circuit (LVDA)	Operation possible	Operation possible	Operation possible	Operation possible
Power-on reset circuit	Operation possible	Operation possible	Operation possible	Operation possible
Peripheral modules	Operation possible	Stopped (settings retained)	Stopped (settings retained)	Stopped (settings undetermined)
I/O ports	Operation possible	Settings retained	Settings retained	Settings retained

Stopped (settings retained):
State in which the values of the internal registers are retained and the internal state is operation suspended.

Stopped (settings undetermined):
State in which the values of the internal registers are undetermined and the internal state is power-off.

Note 1. Either “settings retained” or “settings undetermined” may be selected by means of a control register setting

2.21.2 Mode Transitions

Figure 2.57 diagrams the transitions between the modes of the RX71M, and Table 2.160 lists transition conditions.

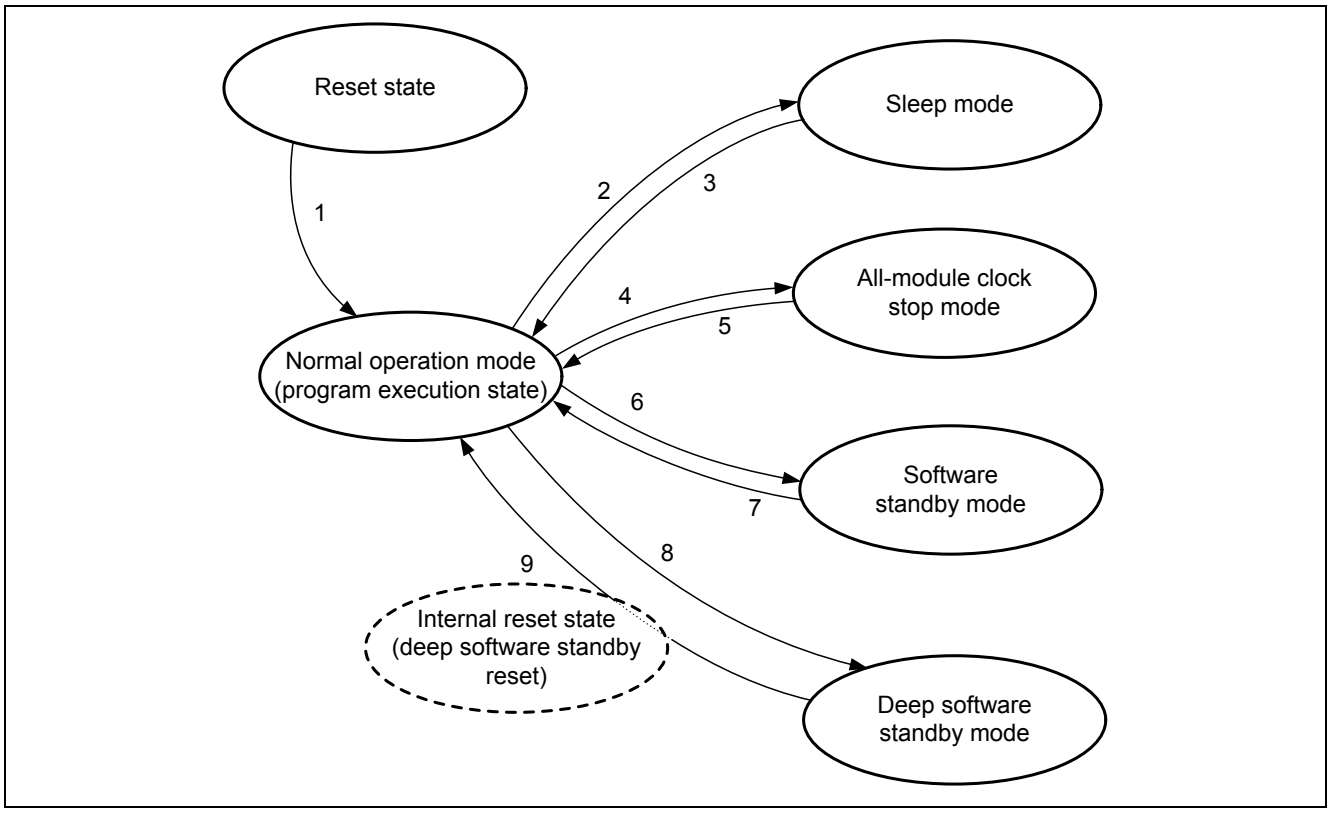


Figure 2.57 RX71M Mode Transitions

Table 2.160 List of RX71M Mode Transitions and Events

No.	Event	Transition Condition (The following conditions are specified before the event.)
1	RES# pin = high	—
2	WAIT instruction executed	SBYCR.SSBY = 0
3	All interrupts	—
4	WAIT instruction executed	SBYCR.SSBY = “0”, MSTPCRA.ACSE = “1”, MSTPCRA = “FFFF FF[C-F]Fh” MSTPCRB = “FFFF FFFFh”, MSTPCRC[31:16] = “FFFFh”, MSTPCRD = “FFFF FFFFh”
5	External and peripheral interrupts	External pin interrupts (NMI, IRQ0 to IRQ15) Peripheral function interrupts (8-bit timer, RTC alarm, RTC cycle, IWDT, USB suspend/resume, voltage monitor 1, voltage monitor 2, oscillation stop detection)*1
6	WAIT instruction executed	SBYCR.SSBY = 1, DPSBYCR.DPSBY = 0
7	External and peripheral interrupts	External pin interrupts (NMI, IRQ0 to IRQ15) Peripheral function interrupts (RTC alarm, RTC cycle, IWDT, USB suspend/resume, voltage monitor 1, voltage monitor 2)*1
8	WAIT instruction executed	SBYCR.SSBY = 1, DPSBYCR.DPSBY = 1
9	External and peripheral interrupts	Some pins used as external pin interrupt sources (NMI, IRQ0-DS to IRQ15-DS, SCL2-DS, SDA2-DS, CRX1-DS), peripheral function interrupts (RTC alarm, RTC cycle, USB suspend/resume, voltage monitor 1, voltage monitor 2)*1 After one of the above interrupts occurs the internal reset state lasts for a specified duration, after which the internal reset and deep software standby mode are canceled at the same time, and the CPU operates in normal operation mode using the LOCO (recovery after a reset).

Note 1. Each interrupt has detailed conditions. For descriptions, see the User’s Manual: Hardware.

2.21.3 Module-Stop State

On the SH7216 Group the modules other than the RAM and ROM are set to the module-stop state after a reset, and no clock is supplied.

On the RX71M the modules other than the DMACAa, EXDMACa, DTCa, RAM, ECCRAM, and standby RAM are set to the module-stop state after a reset, and no clock is supplied. The DTCa and DMACAa share a common module-stop setting bit (MSTPCRA.MSTPA28), so module-stop control is applied to them both simultaneously. The EXDMACa has an independent module-stop setting bit (MSTPCRA.MSTPA29), so it can be controlled individually.

As on the SH7216 Group, on the RX71M it is necessary to cancel the module-stop state before using any module that enters the module-stop state after a reset.

When changing module-stop state settings on the RX71M it is necessary to turn off register write protection in the protect register (PRCR) before accessing the module-stop control register (MSTPCRn).

Table 2.161 lists the clock supply state after a reset of each module.

Table 2.161 Clock Supply State after Reset

Function Name*1	SH7216 Group	RX71M*2
RAM	Clock supplied (operating)	Clock supplied (operating)
User break controller (UBC)	No clock supplied	—
Data transfer controller (DTC)		Clock supplied (operating)
Direct memory access controller (DMAC)		
Multi-function timer pulse unit (MTU)		No clock supplied
Serial communication interface (SCI, SCIF)		
Serial peripheral interface (RSPI)		
I^2C bus interface (IIC)		
A/D converter (ADC)		
Compare match timer (CMT)		

Note 1. The function names listed are those for the SH7216 Group.

Note 2. On the RX71M there are other modules affected by the module-stop function in addition to those listed in the table.

2.21.4 Write Protection

The RX71M has a register write protection function to protect important registers from being overwritten if program runaway occurs. The low power consumption function-related registers are protected by this function.

If necessary, set protect bit 1 (PRCR.PRC1) to 1 to enable writes before writing to these registers.

2.21.5 Low-Power-Consumption Mode Transition Setting Example

A mode transition setting example using the RX71M is shown below. A mode transition setting example using the low-power-consumption functions of the SH7216 Group and the RX71M is presented below. In the example described here an IRQ9 interrupt is generated by pressing SW3. This operation requires that SW3 be connected to the IRQ9-DS pin.

Operational Overview

- Pulse output is produced by the MTU and TMR to confirm the mode transition status.
- Mode transition occurs when an interrupt is generated by pressing SW3 (IRQ9-DS pin input).
- At release from deep software standby mode, a reset is generated and processing ends.

The mode transition sequence when SW3 is pressed and the operations of the output pins are listed below. The numbers in the No. column of the table correspond to those in Figure 2.57 and Table 2.160.

Table 2.162 Low-Power-Consumption Mode Transition Sequence

No.	SW3 Pressed	State Transition	LED2	LED3	MTU Pin	TMR Pin
1	—	RES pin ⇒ normal operation mode	Flashing	Off	Pulse output	Pulse output
2	1st	SLEEP mode	Stop (sustained)		Pulse output	Pulse output
3	2nd	Normal operation mode	Flashing		Pulse output	Pulse output
4	3rd	All-module clock stop mode	Stop (sustained)		Stop (sustained)	Pulse output
5	4th	Normal operation mode	Flashing		Pulse output	Pulse output
6	5th	Software standby mode	Stop (sustained)		Stop (sustained)	Stop (sustained)
7	6th	Normal operation mode	Flashing		Pulse output	Pulse output
8	7th	Deep software standby mode	Stop (undefined)		Stop (undefined)	Stop (undefined)
9	8th	Deep software standby mode ⇒ normal operation mode	Off	On	Stop	Stop

Operation specifications and connection examples are shown below. Where there are differences between the operation specifications of the SH7216 Group and the RX71M, the SH7216 Group operation specifications are indicated by “SH7216:” in the Remarks column. LED-related information is provided for the RX71M only.

The register names in the setting examples are those when using iodefine.h. For information on the operating environment, see 3.1, Operating Environment.

Table 2.163 Low-Power-Consumption Mode Transition Operation Specifications

Item			Description	Remarks
Processor mode			Supervisor mode	
TMR	TMR channel		TMR0, TMR1	
	Count clock		PCLKB/1 (PCLKB = 60 MHz)	
	Operating mode		16-bit counter	2-channel cascade connection
	Counter clearing source		Compare match A	
	Compare match value		0x752F (0.5 ms @60 MHz)	
	TMR pin operation		Output toggled by compare match A.	
	Interrupts		Not used.	
MTU			See Table 2.48.	Waveform output based on compare match
IRQ			See Table 2.19.	No noise cancellation when returning to normal mode
Pins used		TMO	P22/TMO0	Pulse output (TMR)
		MTIOC	PC2/MTIOC4B	Pulse output (MTU)
		IRQ9-DS	P41/general	Mode transition trigger (IRQ9-DS) Board reconfiguration required when connecting SW3.
		LED2	P26/general	Flashes in normal mode.
		LED3	P27/general	Lights when reset occurs at return from deep software standby mode.

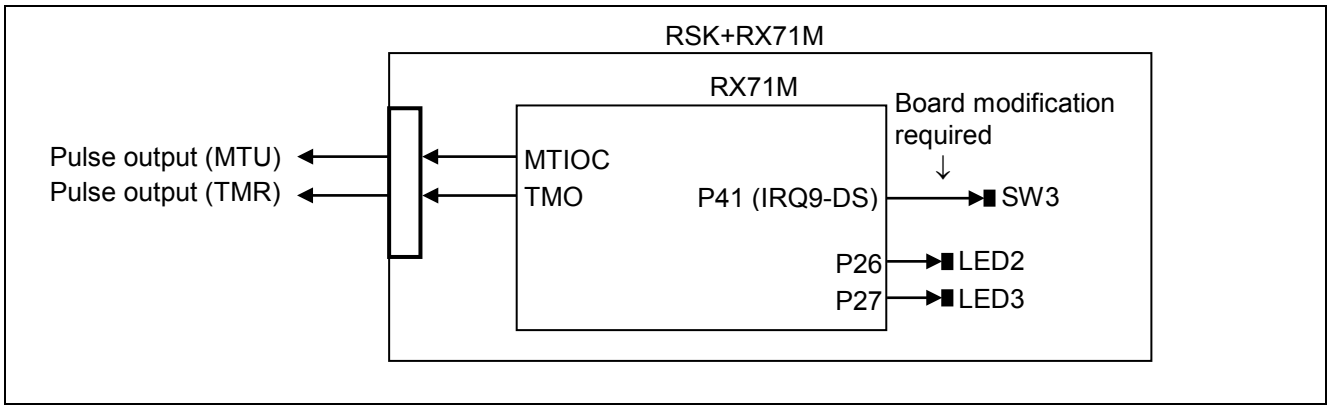


Figure 2.58 Low-Power-Consumption Mode Transition Connection Example

Notes

On the Renesas Starter Kit+ for RX71M in the initial state the SW3 and IRQ9-DS pins are not connected. The IRQ9-DS pin is connected for use with the LCD, so modify the board to connect it to the SW3.

On the Renesas Starter Kit+ for RX71M in the initial state the pins used for MTU4 in the connection example are connected for use with the Ethernet-PHY, and the pins used for TMR0 and TMR1 are connected for use with the PDC, so make appropriate modifications to the board as necessary.

Processing Flowcharts

Figure 2.59 is an example flowchart of low-power-consumption mode transition processing.

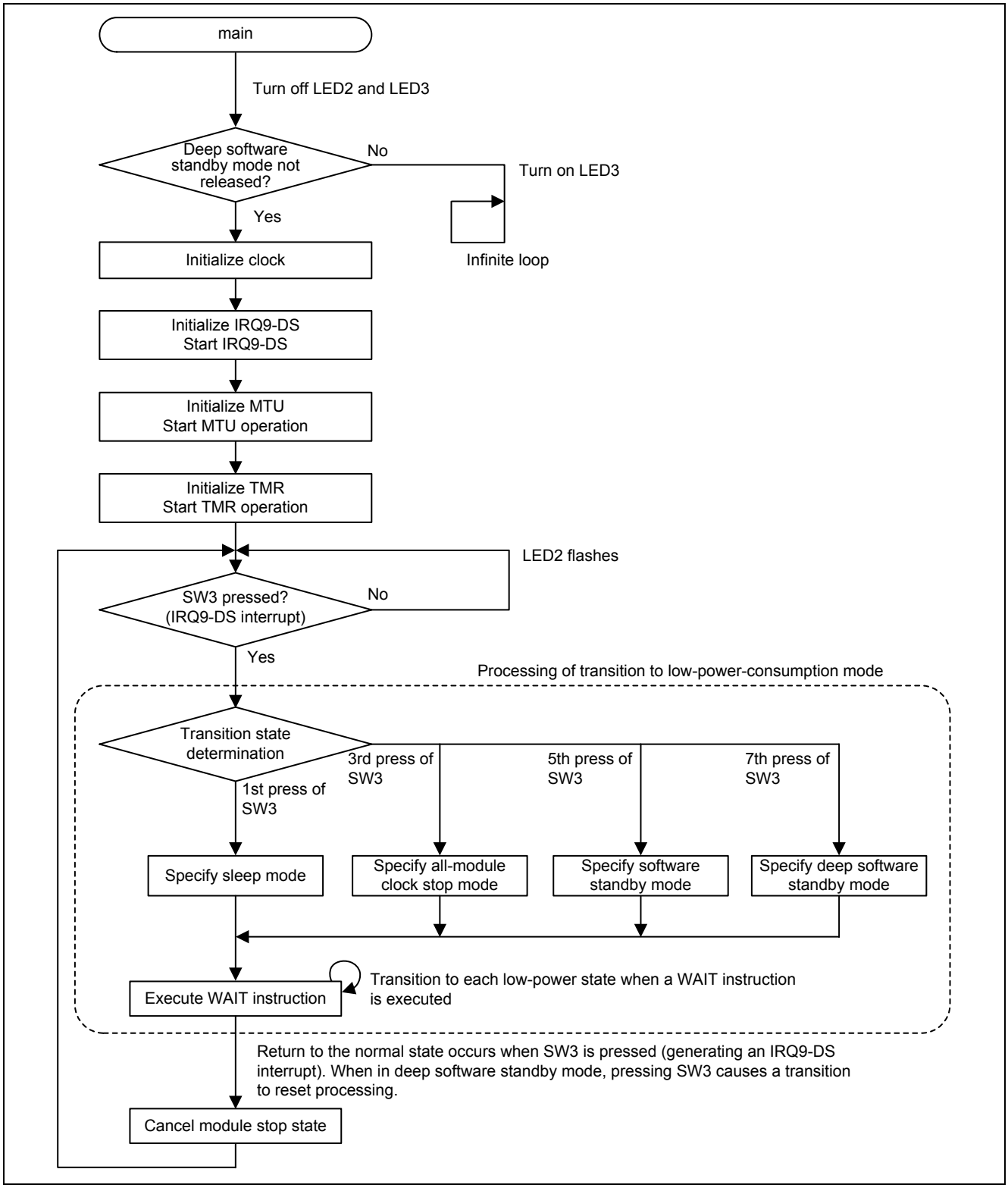


Figure 2.59 Example Flowchart of Low-Power-Consumption Mode Transition Processing

Setting Examples

Low-power-consumption mode transition setting examples are presented below. For MTU, LED, and SW3 setting examples, refer to the sections listed in Table 2.164.

Table 2.164 Setting Example References

Item	Reference
MTU-related setting examples	Section 2.7.5, Compare Match Pulse Output Setting Examples
LED-related setting examples	Section 2.2.3, General I/O Setting Examples
SW3 (IRQ9-DS)-related setting examples	Section 2.4.1, IRQ Usage Example

Table 2.165 TMR Compare Match Initialization Setting Example

Processing	Setting Example
Make I/O port settings (general ports).	Output value setting (output 0) PORT2.PODR.B2 = 0b (P22) Pin direction setting PORT2.PDR.B2 = 1b (output/P22) Pin mode setting (general) PORT2.PMR.B2 = 0b (P22)
Cancel module stop state.	SYSTEM.PRCR = A502h SYSTEM.MSTPCRA.MSTPA5 = 0b SYSTEM.PRCR = A500h
Stop counting (clock input disabled).	TMR0.TCCR = 00h TMR1.TCCR = 0Fh
Make count operation settings	Clearing the counter TMR0.TCNT = 0 TMR1.TCNT = 0 Counter clear source setting TMR0.TCR.CCLR = 01b Pin operation setting TMR0.TCSR.OSA = 11b Cycle setting TMR01.TCORA = 752Fh
Make I/O port settings (peripheral)	Cancellation of register protection MPC.PWPR.B0WI = 0b MPC.PWPR.PFSWE = 1b Pin peripheral function selection MPC.P22PFS.PSEL = 000101b (TMO0) Register protection settings MPC.PWPR.PFSWE = 0b MPC.PWPR.B0WI = 1b Pin mode setting (peripheral) PORT2.PMR.B2 = 1b (TMO0)

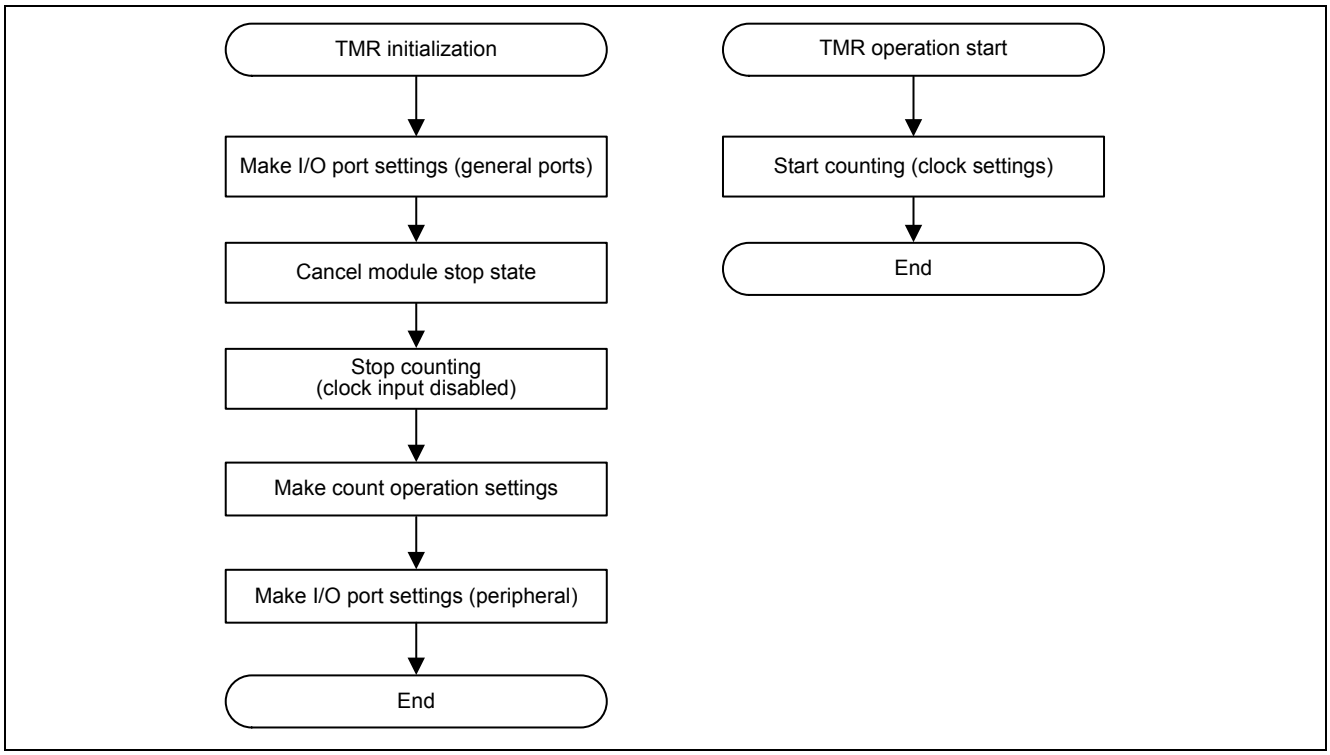


Figure 2.60 Example Flowcharts of TMR Processing

Table 2.166 TMR Operation Start Setting Example

Processing	Reference
Start counting (clock settings).	TMR0.TCCR.CSS = 11b TMR1.TCCR.CKS = 000b TMR1.TCCR.CSS = 01b

Table 2.167 Sleep Mode Setting Example

Processing	Setting Example
Disable protection.	SYSTEM.PRCR = A502h
Disable all-module clock stop mode.	SYSTEM.MSTPCRA.ACSE = 0b
Specify mode setting after WAIT instruction.	Transition to sleep mode or all-module clock stop mode SYSTEM.SBYCR.SSBY = 0b
Make protection setting.	SYSTEM.PRCR = A500h

Table 2.168 All-Module Clock Stop Mode Setting Example

Processing	Setting Example
Disable protection.	SYSTEM.PRCR = A502h
Enable all-module clock stop mode.	SYSTEM.MSTPCRA.ACSE = 1b
Specify mode setting after WAIT instruction.	Transition to sleep mode or all-module clock stop mode SYSTEM.SBYCR.SSBY = 0b
Make module stop state settings.	SYSTEM.MSTPCRA = FFFFFFDFh SYSTEM.MSTPCRB = FFFFFFFFh SYSTEM.MSTPCRC = FFFF0000h SYSTEM.MSTPCRD = FFFFFFFFh
Make protection setting.	SYSTEM.PRCR = A500h

Table 2.169 Software Standby Mode Setting Example

Processing	Setting Example
Disable protection.	SYSTEM.PRCR = A502h
Make settings for software standby mode and deep software standby mode.	Setting address bus and bus control signals to high-impedance SYSTEM.SBYCR.OPE = 0b
Disable deep software standby mode.	SYSTEM.DPSBYCR.DPSBY = 0b
Specify mode setting after WAIT instruction.	Transitioning to software standby mode SYSTEM.SBYCR.SSBY = 1b
Make protection setting.	SYSTEM.PRCR = A500h

Table 2.170 Deep Software Standby Mode Setting Example

Processing	Setting Example
Disable protection.	SYSTEM.PRCR = A502h
Make settings for software standby mode and deep software standby mode.	Setting address bus and bus control signals to high-impedance SYSTEM.SBYCR.OPE = 0b
Make settings for deep software standby mode.	No supply of power to standby RAM and USB resume detection blocks SYSTEM.DPSBYCR.DEEPCUT = 01b Canceling I/O port retention at time of release SYSTEM.DPSBYCR.IOKEEP = 0b
Make settings for canceling deep software standby mode by IRQn-DS pin.	Generation of cancelation request at falling edge of IRQ9-DS pin SYSTEM.DPSIEGR1.DIRQ9EG = 0b Enabling cancelation by IRQ9-DS pin SYSTEM.DPSIER1.DIRQ9E = 1b Reading value of SYSTEM.DPSIER1.DIRQ9E (elapsing of 6 or more cycles) Clearing of cancelation request status flag by IRQ9-DS pin* SYSTEM.DPSIFR1.DIRQ9F = 0b * Clear to 0 after reading value as 1.
Enable deep software standby mode.	SYSTEM.DPSBYCR.DPSBY = 1b
Specify mode setting after WAIT instruction.	Transitioning to deep software standby mode SYSTEM.SBYCR.SSBY = 1b
Make protection setting.	SYSTEM.PRCR = A500h

3. Sample Code

3.1 Operating Environment

The sample code associated with this application note has been confirmed to run in the conditions listed in Table 3.1.

The SH7216 Group setting examples apply to the operating frequencies listed in Table 3.2.

Table 3.1 Operating Environment (RX71M)

Item	Description
Microcontroller used	R5F571MLCDFC (RX71M Group)
Operating frequency	• Main clock: 24 MHz • Sub clock: 32.768 kHz • PLL: 240 MHz (main clock divided by 1 and multiplied by 10) • HOCO: Stopped • System clock (ICLK): 120 MHz (PLL divided by 2) • Peripheral module clock A(PCLKA): 120 MHz (PLL divided by 2) • Peripheral module clock B(PCLKB): 60 MHz (PLL divided by 4) • Peripheral module clock C(PCLKC): 60 MHz (PLL divided by 4) • Peripheral module clock D(PCLKD): 60 MHz (PLL divided by 4) • External bus clock (BCLK): 60 MHz (PLL divided by 4) • SDRAM clock (SDCLK): 60 MHz (PLL divided by 4) • USB clock (UCLK): 48 MHz (PLL divided by 5)
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics Corporation e^2 studio V6.1.0
C compiler	Renesas Electronics Corporation C/C++ Compiler Package for RX Family (V.2.07.00)
CPU series (type)	RX700 (RX71M)
Optimization	Optimization level: 2 (general optimization) Optimization method: Optimization prioritizing code size
iodefine.h version	1.0A
Endian order	Little endian
Operating mode	Single-chip mode (on-chip ROM enabled extended mode only when using SDRAM)
Processor mode	Supervisor mode
Board used	Renesas Starter Kit for RX71M (Product type: R0K50571MC000BE)

Table 3.2 Operating Frequency (SH7216)

Item	Description
Operating frequency	Input clock (XTAL): 12.5 MHz PLL: 200 MHz (input clock multiplied by 16) Internal clock (Iϕ): 100 MHz (PLL divided by 2) Bus clock (Bϕ): 50 MHz (PLL divided by 4) Peripheral clock (Pϕ): 50 MHz (PLL divided by 4) MTU2S clock (Mϕ): 50 MHz (PLL divided by 4) AD clock (Aϕ): 50 MHz (PLL divided by 4)

3.2 Sample Code Configuration

The configuration of the sample code is shown below.

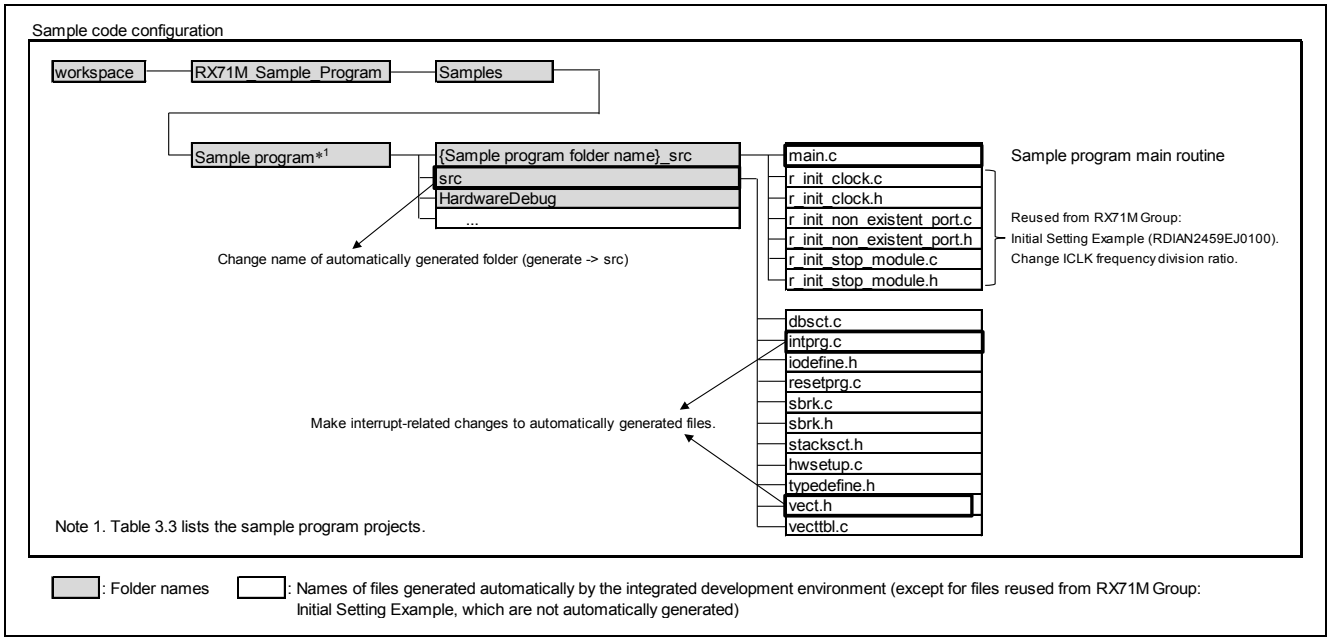


Figure 3.1 Sample Code Configuration

Initial Settings

The initial setting function of this application note reuses the sample code from RX71M Group: Initial Setting Example, Rev. 1.00. This revision was current when this application note was produced.

Note that the RX71M Group support drivers and middleware (Firmware Integration Technology) and a sample code generation tool (Code Generator) that can be used to shorten the amount of time required for development.

Items Requiring Changes in Automatically Generated Files

The file main.c specifies interrupt declarations, vector registrations, and interrupt handlers. Portions of the automatically generated files interrupt_handlers.c and vect.h duplicate settings and code in main.c, so they have been modified as follows:

- interrupt_handlers.c: Interrupt handlers that are specified in main.c have been commented out.
- vect.h: The interrupt function declarations and vector registrations in vect.h have been commented out.

Table 3.3 List of Sample Code Projects

Sample Project Name	Related Items
BSC_sdram_read_write	2.3.3
DTC_normal_transfer_mode	2.5.7
DMA_normal_transfer_mode	2.6.10
MTU_compare_match	2.7.5
MTU_input_capture	2.7.6
SCI_asynchronous_interrupt	2.10.6
SCI_asynchronous_polling	
SCI_sync_master_transmit_int	2.10.7
SCI_sync_master_transmit_pol	
SCI_sync_slave_receive_int	2.10.8
SCI_sync_slave_receive_pol	
SCIF_asynchronous_interrupt	2.11.5
SCIF_sync_master_transmit_int	2.11.6
SCIF_sync_slave_receive_int	2.11.7
SPI_4wire_master_transceiver	2.12.5
SPI_3wire_master_transmit	2.12.6
SPI_3wire_slave_receive	2.12.7
IIC_master_transceiver	2.13.10
IIC_slave_transceiver	2.13.11
AD_continuous_scan_multi_ch	2.14.7
CMT_compare_match	2.18.5
Low_power_consumption_mode	2.21.5

4. Reference Documents

4.1 Reference Documents

Section 4.1 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 4.1 Reference Documents

SH7214 Group, SH7216 Group User's Manual: Hardware (R01UH0230EJ0400)
SH-2A, SH2A-FPU User's Manual: Software (R01US0031EJ)
RX71M Group User's Manual: Hardware (R01UH0493EJ0110)
RX64M Group, RX71M Group Flash Memory User's Manual: Hardware Interface (R01UH0435EJ0110)
RX Family RXv2 Instruction Set Architecture User's Manual: Software (R01US0071EJ0100)
RX71M Group Renesas Starter Kit+ User's Manual (R20UT3217EG0100)
Renesas Starter Kit+ for RX71M CPU Board Schematics (R20UT3216EG0100)
RX71M Group Initial Settings Example (R01AN2459EJ0100)

Consistency with Technical Updates

This application note reflects the contents of the following technical update:

TN-RX*-A127A/E

Adds function of “RX64M Group, RX7lM Group User’s Manual: Hardware”

Page 2746 of 2923 (RX7IM Group)

Specifications of code flash memory and data flash memory add unique ID.

Website and Support

Renesas Electronics Website

http://www.renesas.com/

Inquiries

http://www.renesas.com/contact/

Revision History

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Feb. 27, 2018	—	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. 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the 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 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 7.0