
RX74M Group

Initial Settings Example

Introduction

This application note describes the settings of the RX74M products that must be made in accord with the conditions of usage specified in the header files that become applicable after a reset. These settings include clock settings, stopping any peripheral modules that are still operating after the reset, and the setting of bits corresponding to absent port pins.

Target Device

- Products of the RX74M Group with 320, 224, 176, 144, or 100 pins and ROM capacity of 512 Kbytes to 1 Mbyte

When applying this application note to another type of microcontroller, modify the settings to suit the specifications of that microcontroller and sufficiently evaluate the results.

Contents

1. Specifications	3
1.1 Project Description	3
1.2 Stopping Peripheral Modules that Are Still Operating After a Reset.....	3
1.3 Settings of Bits Corresponding to Absent Port Pins.....	3
1.4 Usage Note.....	3
1.5 Clock Settings.....	4
1.5.1 Overview.....	4
1.5.2 Clock Specifications in the Sample Code	4
1.5.3 Clock Selection.....	5
2. Conditions under Which Operation Was Verified	6
3. Software	7
3.1 Stopping Peripheral Modules that Are Still Operating After a Reset.....	7
3.2 Setting of Bits Corresponding to Absent Port Pins.....	7
3.2.1 Overview of Processing.....	7
3.2.2 Selecting the Number of Pins.....	7
3.3 Clock Settings.....	8
3.3.1 Clock Setting Procedure.....	8
3.3.2 Sub-clock Oscillation Stabilization Time	9
3.4 Section Configuration	9
3.5 File Composition.....	10
3.6 Option-Setting Memory	10
3.7 Constants	11
3.8 Functions	17
3.9 Function Specifications	18
3.10 Flowcharts	21
3.10.1 Main Processing.....	21
3.10.2 Stopping Peripheral Modules that Are Still Operating After a Reset	21
3.10.3 Initialization of the Bits Corresponding to Absent Port Pins.....	22
3.10.4 Clock Initialization.....	23
3.10.5 Main Clock Oscillation Setting.....	26
3.10.6 HOCO Clock Oscillation Setting.....	27
3.10.7 PLL1 Clock Oscillation Setting	29
3.10.8 Sub-clock Oscillation Setting.....	30
3.10.9 Initialization of the RTC	31
3.10.10Setting up Software Waiting with the Use of CMT0	32
4. Importing a Project	33
5. Sample Code.....	34
6. Reference Documents.....	34
Revision History	35

1. Specifications

The sample code provides processing to stop peripheral modules that are operating after a reset, configure ports that are absent on the given device, and make clock settings. This application note is on the assumption that processing is to proceed when power is supplied (cold start).

1.1 Project Description

This application note includes the project "r01an8015_rx74m" for the EK-RX74M evaluation kit. This project contains files that were automatically generated by the e² studio. The settings of this project are adapted for the 320-pin device with a ROM capacity of 1 Mbyte mounted on the EK-RX74M. If another device is to be used, change the project settings as required. Refer to the following URL for details.
<https://en-support.renesas.com/knowledgeBase/18696526>

1.2 Stopping Peripheral Modules that Are Still Operating After a Reset

Some peripheral modules start operating immediately after power is supplied, and some may have the module stop function disabled. Stopping these modules requires provision of the following processing:

- Processing to stop the functioning of the DMAC, DTC, and RAM modules

Note that the sample code does not execute the above processing. To execute this processing, change the constants as required. For details, see Table 3.6.

1.3 Settings of Bits Corresponding to Absent Port Pins

The specified values must be set for the bits in the port direction registers corresponding to absent port pins. In the sample code for this application note, the initial values set are those for the port direction registers of a 320-pin device. Change the relevant constants to values that are appropriate for the product you will be using. For details, see section 3.2 and Table 3.8 to Table 3.12 in section 3.7.

1.4 Usage Note

In the sample code, the CPU clock (CPUCLK0) is set to 600 MHz by default. Note, however, that for the products in packages with 100, 144, and 176 (in QFP) pins, the maximum frequency of the CPU clock (CPUCLK0) is 480 MHz. Accordingly, when you apply the sample code to these products, be sure to set the operating frequency of the CPU clock (CPUCLK0) to a frequency no higher than 480 MHz. Do not set VSCR_1 as the target voltage for VDD in a product where the CPU clock (CPUCLK0) is to be run at its maximum frequency of 480 MHz.

1.5 Clock Settings

1.5.1 Overview

Clock settings are made in the following order:

- (1) Setting the sub-clock
- (2) Setting the main clock
- (3) Setting the HOCO clock
- (4) Setting the PLL1 clock
- (5) Switching the system clock

In this application note, the settings of clocks can be switched by changing constants defined in `r_init_clock.h`.

In the sample code, the PLL1 clock is selected as the system clock. Select the clock to be used by changing the constants as required. For details, see section 1.5.3, Clock Selection.

1.5.2 Clock Specifications in the Sample Code

Table 1.1 lists the clock specifications assumed in the sample code. We calculated the oscillation stabilization time and other parameters in accord with the specifications of the resonators listed in this table.

Table 1.1 Clock Specifications Assumed in the Sample Code

Clock	Oscillation Frequency	Oscillation Stabilization Time	Remarks
Main clock resonator	24 MHz	2 ms ^(Note 2)	Crystal
Sub-clock resonator	32.768 kHz ^(Note 1)	1.5 s ^(Note 2)	For use with loads having low capacitive components
PLL1 clock	600 MHz	— ^(Note 3)	—
HOCO clock	48 MHz ^(Note 1)	— ^(Note 3)	—

Note 1. Oscillation is stopped in the sample code.

Note 2. The time required before the oscillation of the resonator becomes stable depends on the conditions, such as the wiring pattern and oscillation parameters, of the actual system. To determine the appropriate oscillation stabilization time, ask the manufacturer of the crystal or ceramic resonator to evaluate the user system.

Note 3. Refer to section 69, Electrical Characteristics, in the User's Manual: Hardware.

1.5.3 Clock Selection

In the sample code, the clock source of the system clock and whether to enable oscillation of each clock can be selected by changing constants defined in `r_init_clock.h`. See Table 3.5 and Table 3.6, User-Changeable Constants Used in the Sample Code. Table 1.2 lists examples of the clock selection. Table 1.3 lists examples of the RTC selection.

Table 1.2 Examples of the Clock Selection

No.		1	2	3	4
System clock		PLL	Main clock	HOCO	Sub-clock
PLL1 clock		Enabled	Disabled	Disabled	Disabled
Main clock		Enabled	Enabled	Disabled	Disabled
HOCO clock		Disabled	Disabled	Enabled	Disabled
Sub-clock		Disabled ^(Note 1)	Disabled ^(Note 1)	Enabled ^(Note 1, Note 2)	Enabled ^(Note 1)
Constants	SEL_SYSCLK	CLK_PLL	CLK_MAIN	CLK_HOCO	CLK_SUB
	SEL_PLL	B_USE_PLL_MAIN	B_NOT_USE	B_NOT_USE	B_NOT_USE
	SEL_MAIN	B_USE	B_USE	B_NOT_USE	B_NOT_USE
	SEL_HOCO	B_NOT_USE	B_NOT_USE	B_USE	B_NOT_USE
	SEL_SUB	B_NOT_USE ^(Note 1)	B_NOT_USE ^(Note 1)	B_USE ^(Note 1, Note 2)	B_USE ^(Note 1)
	SEL_SRAM_WAIT	MEMWAIT_ON	MEMWAIT_OFF	MEMWAIT_OFF	MEMWAIT_OFF

Note 1. If the sub-clock is not to be used for the system clock, input to the clock frequency accuracy measurement circuit (CAC), or realtime clock (RTC), select disabled (B_NOT_USE).

Note 2. When the FLL function is to be used, select enabled (B_USE).

Table 1.3 Examples of the RTC Selection

No.	1	2	3
SEL_RTC	B_NOT_USE	B_USE	B_USE
SEL_RTC_CLK	B_RTC_CLK_LOCO	B_RTC_CLK_SUB ^(Note 1)	B_RTC_CLK_LOCO
SEL_RTC_CMD	B_RTC_CMD_CLD	B_RTC_CMD_CLD	B_RTC_CMD_CLD

Note 1. Even when the setting for SEL_SUB is B_NOT_USE, the sub-clock can be made to oscillate as long as this condition in column "2" is satisfied.

2. Conditions under Which Operation Was Verified

For the sample code covered by this application note, operation was verified under the specific conditions listed below. Table 2.1 lists the conditions under which operation of r01an8015_rx74m was verified.

Table 2.1 Conditions under Which Operation of r01an8015_rx74m Was Verified

Item		Contents
MCU used		R5K574MCB4AE (RX74M Group)
Operating frequency	The PLL1 is selected as the system clock	• Main clock: 24 MHz • Sub-clock: 32.768 kHz • PLLP: 600 MHz (main clock $\times 100 \times 1/4$) • PLLQ: 480 MHz (main clock $\times 100 \times 1/5$) • PLLL: 480 MHz (main clock $\times 100 \times 1/5$) • CPU clock (CPUCLK0): 600 MHz (PLL $\times 1/1$) • System clock (ICLK): 300 MHz (PLL $\times 1/2$) • Peripheral module clock A (PCLKA): 150 MHz (PLL $\times 1/4$) • Peripheral module clock B (PCLKB): 75 MHz (PLL $\times 1/8$) • Peripheral module clock C (PCLKC): 75 MHz (PLL $\times 1/8$) • Peripheral module clock D (PCLKD): 150 MHz (PLL $\times 1/4$) • Peripheral module clock E (PCLKE): 300 MHz (PLL $\times 1/2$)
	The main clock is selected as the system clock	• Main clock: 24 MHz • CPU clock (CPUCLK0): 24 MHz (main clock $\times 1/1$) • System clock (ICLK): 24 MHz (main clock $\times 1/1$) • Peripheral module clock A (PCLKA): 24 MHz (main clock $\times 1/1$) • Peripheral module clock B (PCLKB): 24 MHz (main clock $\times 1/1$) • Peripheral module clock C (PCLKC): 24 MHz (main clock $\times 1/1$) • Peripheral module clock D (PCLKD): 24 MHz (main clock $\times 1/1$) • Peripheral module clock E (PCLKE): 24 MHz (main clock $\times 1/1$)
	The HOCO is selected as the system clock	• Main clock: Disabled • HOCO: 48 MHz • Sub-clock: 32.768 kHz • CPU clock (CPUCLK0): 48 MHz (HOCO $\times 1/1$) • System clock (ICLK): 48 MHz (HOCO $\times 1/1$) • Peripheral module clock A (PCLKA): 48 MHz (HOCO $\times 1/1$) • Peripheral module clock B (PCLKB): 48 MHz (HOCO $\times 1/1$) • Peripheral module clock C (PCLKC): 48 MHz (HOCO $\times 1/1$) • Peripheral module clock D (PCLKD): 48 MHz (HOCO $\times 1/1$) • Peripheral module clock E (PCLKE): 48 MHz (HOCO $\times 1/1$)
Operating voltage		3.3 V
Integrated development environment		Renesas Electronics e ² studio Version 2026-07
C compiler		Renesas Electronics C/C++ Compiler Package for RX Family V.3.08
		Compiler option The default settings in the integrated development environment are used.
Version of iodef.h		V1.00
Endian		Little endian, big endian
Operating mode		Single-chip mode
Processor mode		Supervisor mode
Sample code version		Version 1.00
Board used		Evaluation kit for the RX74M MCU Group (Product No.: RTK5EK74Mxxxxxxxxx)

3. Software

In the sample code, peripheral functions operating after a reset are stopped, settings of bits corresponding to absent port pins are made, and then clock settings are made.

3.1 Stopping Peripheral Modules that Are Still Operating After a Reset

This processing is to stop peripheral modules that are operating after a reset. Only the peripheral modules listed in Table 3.1 are released from the module-stop state after a reset. To place a module in the module-stop state, set the module-stop bit to 1. Placing modules in the module-stop state reduces power consumption.

In the sample code, the constants of the form "MSTP_STATE_target module name" in the file `r_init_stop_module.h` are set to "0 (MODULE_STOP_DISABLE)" so the target modules are not placed in the module-stop state. In the case of modules with states that should be changed to "module-stop" in your system, set the corresponding constants to "1 (MODULE_STOP_ENABLE)".

Table 3.1 lists the peripheral modules that are not automatically placed in the module-stop state after a reset.

Table 3.1 List of the Peripheral Modules that Are Not Automatically Placed in the Module-Stop State After a Reset

Peripheral Module	Module Stop Setting Bit	Value After Reset	Setting When Not Using the Module
DMAC/DTC	MSTPCRA.MSTPA22 bit	0 (Release from the module-stop state)	1 (Transition to the module-stop state)
EXDMAC	MSTPCRA.MSTPA29 bit		
SRAM0	MSTPCRA.MSTPA0 bit		
SRAM1	MSTPCRA.MSTPA1 bit		

3.2 Setting of Bits Corresponding to Absent Port Pins

3.2.1 Overview of Processing

Bits of the PDR registers corresponding to port pins that are absent in the given device should be set to 0 (input) or 1 (output). If data are to be written in byte units to a PDR register that includes bits corresponding to absent port pins after the call of the `R_INIT_Port_Initialize` function from `main.c`, also make sure that the direction control bits for those port pins are set as indicated by the table corresponding to the number of pins on the package in 22.6.9, Initialization of the Port Direction Register (PDR), in the User's Manual: Hardware. When writing to a PODR register in byte units, set the port output data storage bits for absent pins with the output setting to 0.

3.2.2 Selecting the Number of Pins

In the sample code, use of the 320-pin product is set (`PIN_SIZE = 320`). This application note also supports the products with 320, 224, 176, 144, and 100 pins. To use a device other than a 320-pin product, set the number of pins for `PIN_SIZE` in `r_init_port_initialize.h`.

3.3 Clock Settings

3.3.1 Clock Setting Procedure

Table 3.2 lists the steps of the procedure for setting the clocks. It also describes each step of the processing and lists the settings configured in the sample code. The sample code enables the main clock and PLL1, and disables the HOCO.

Table 3.2 Procedure for Clock Settings

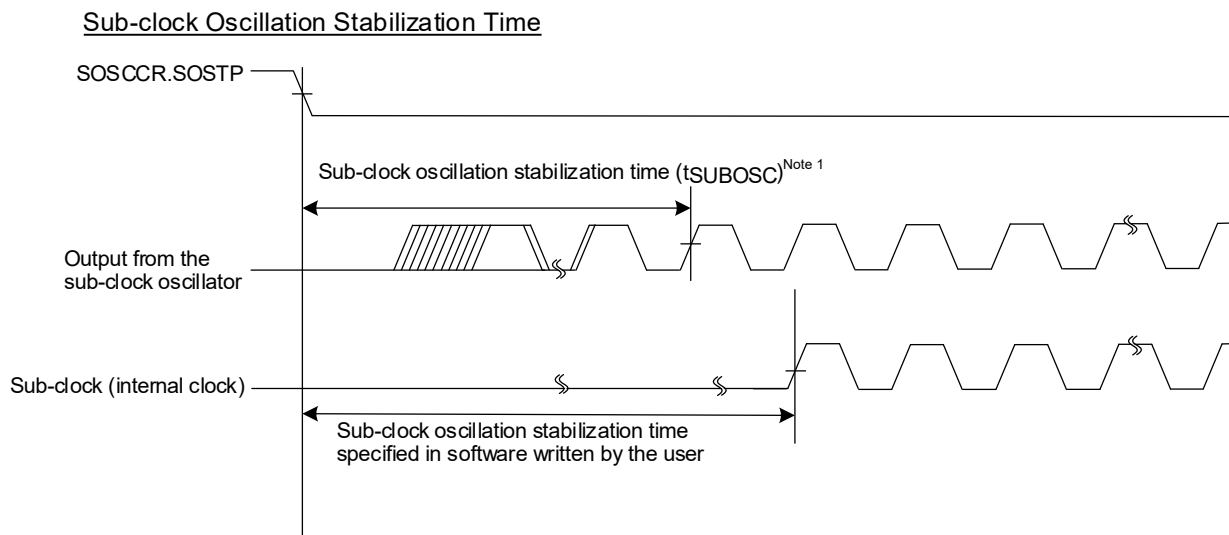
Step	Processing	Description		Setting in Sample Code
1	Setting the sub-clock (Note 1)	If not to be used	Initialize the sub-clock control circuit.	The sub-clock is not used.
		If to be used	Initialize the sub-clock control circuit and then make the sub-clock start oscillating. After that, waiting until oscillation becomes stable by software proceeds. ^(Note 2)	
2	Setting the main clock (Note 1)	If not to be used	No setting is required.	The main clock is used.
		If to be used	Set the driving ability of the main clock in the MOFCR register, set the waiting time until the output of the main clock is supplied as the internal clock in the MOSCWTCR register, and then make the main clock start oscillating. After that, waiting until oscillation becomes stable proceeds.	
3	Setting the HOCO clock (Note 1)	If not to be used	No setting is required.	The HOCO clock is not used.
		If to be used	Make the HOCO clock start oscillating. After that, waiting until oscillation becomes stable proceeds.	
4	Setting the PLL1 clock (Note 1)	If not to be used	No setting is required.	The PLL1 clock is used.
		If to be used	Set the PLL1 input division ratio and frequency multiplication factor, and then start the PLL1 clock oscillating. After that, waiting until oscillation becomes stable proceeds.	
6	Setting the clock frequency division ratios	Set the clock division ratios.		• CPUCLK0: $\times 1/1$ • ICLK and PCLKE: $\times 1/2$ • PCLKA and PCLKD: $\times 1/4$ • PCLKB and PCLKC: $\times 1/8$
7	Switching the system clock	Switch the system clock in accord with the user system.		Switches to the PLL1 clock.

Note 1. Whether to use a clock can be selected by changing the setting of the corresponding constant in `r_init_clock.h`. See section 3.7 for constants.

Note 2. For the sub-clock oscillation stabilization time, see section 3.3.2, Sub-clock Oscillation Stabilization Time.

3.3.2 Sub-clock Oscillation Stabilization Time

Figure 3.1 is an overview of the sub-clock oscillation stabilization time (t_{SUBOSC}). Set t_{SUBOSC} to the value recommended by the resonator manufacturer. Use software to cause a wait for at least the oscillation stabilization time (t_{SUBOSC}). In the sample code, the sub-clock oscillation stabilization time is set to 1.5 seconds, so the waiting time caused by software becomes approximately 1.5 seconds.



Note 1. Consult the resonator manufacturer to determine the resonator stabilization time for your system. The oscillation stabilization time defines the condition required for the resonator to start stable oscillation and does not denote an operating condition for the microcontroller.

Figure 3.1 Sub-clock Oscillation Stabilization Time

3.4 Section Configuration

Table 3.3 lists the information of the section changed in the sample code.

For details about how to add, change, or delete sections, refer to the latest version of the RX Family CC-RX Compiler User's Manual.

Table 3.3 Information of the Section Changed in the Sample Code (r01an8015_rx74m)

Section Name	Type	Address	Description
End_of_RAM	Addition	8217 FFFCh ^(Note 1)	End address of the on-chip RAM

Note 1. The capacity of the on-chip RAM depends on the product. The section should suit the on-chip RAM of the product to be used.

3.5 File Composition

Table 3.4 lists the files used in the sample code. Files automatically generated by the integrated development environment are not included in this table.

Table 3.4 Files Used in the Sample Code

File Name	Outline	Remarks
main.c	Main processing	
r_init_stop_module.c	Stopping the peripheral modules that are still operating after a reset	
r_init_stop_module.h	Header file for r_init_stop_module.c	
r_init_port_initialize.c	Initialization of the bits corresponding to absent port pins	
r_init_port_initialize.h	Header file for r_init_port_initialize.c	
r_init_clock.c	Initialization of clocks	
r_init_clock.h	Header file for r_init_clock.c	

3.6 Option-Setting Memory

The sample code does not change the settings of the option-setting memory from FFFF FFFF. Set a value that suites the user system as required.

3.7 Constants

Table 3.5 and Table 3.6 list the user-changeable constants used in the sample code. Table 3.7 lists the non-user-changeable constants used in the sample code. Table 3.8 to Table 3.12 list the constants that may vary with the product package.

Table 3.5 User-Changeable Constants Used in the Sample Code (1/2)

Constant Name	Setting Value	Description
SEL_MAIN ^(Note 1)	B_USE	Enabling or disabling the main clock: B_USE: Enable (enables oscillation of the main clock) B_NOT_USE: Disable (disables the main clock)
MAIN_CLOCK_HZ ^(Note 1)	24,000,000L	Oscillation frequency of the resonator for the main clock (Hz)
REG_MOMCR ^(Note 1)	1Ah	Setting of the driving ability of the main clock oscillator (setting of the MOMCR register)
REG_MOSCWTCR ^(Note 1)	05h	Setting of the main clock oscillator wait control register
SEL_HOCO ^(Note 1)	B_NOT_USE	Enabling or disabling the HOCO clock: B_USE: Enable (enables oscillation of the HOCO clock) B_NOT_USE: Disable (disables the HOCO clock)
REG_HOCO_FREQ ^(Note 1)	FREQ_48MHz	Selection of the HOCO clock frequency: FREQ_16MHZ: 16 MHz FREQ_18MHZ: 18 MHz FREQ_20MHZ: 20 MHz FREQ_32MHZ: 32 MHz FREQ_48MHZ: 48 MHz
SEL_HOCO_FLL ^(Note 1, Note 2)	B_NOT_USE	Enabling or disabling the FLL function: B_USE: Enable (enables the FLL function) B_NOT_USE: Disable (disables the FLL function)
SEL_PLL1 ^(Note 1)	B_USE_PLL_MAIN	Enabling or disabling the PLL1 clock: B_USE_PLL_MAIN: Enable (main clock) B_USE_PLL_HOCO: Enable (HOCO) B_NOT_USE: Disable (disables the PLL clock)
REG_PLLCCR ^(Note 1)	00006300h	Settings of the PLL1 clock input frequency division ratio and frequency multiplication factor (setting of the PLLCCR register)
REG_PLLCCR2 ^(Note 1)	0443h	Setting of the PLL1 clock output frequency division ratio (setting of the PLLCCR2 register)
SEL_SUB ^(Note 1)	B_NOT_USE	Enabling the sub-clock as the system clock B_USE: Enable B_NOT_USE: Disable
SEL_RTC ^(Note 1)	B_NOT_USE	Enabling or disabling the RTC B_USE: Enable B_NOT_USE: Disable
SEL_RTC_CLK ^(Note 1)	B_RTC_CLK_LOCO	Selection of the RTC count source: B_RTC_CLK_LOCO: LOCO B_RTC_CLK_SUB: Sub-clock
SEL_RTC_CMD ^(Note 1)	B_RTC_CMD_CLD	Selection of the RTC counting mode: B_RTC_CMD_CLD: Calendar count mode B_RTC_CMD_BIN: Binary count mode

Note 1. Change the setting value in r_init_clock.h in accord with the user system.

Note 2. Start the sub-clock oscillator before using the FLL function.

Table 3.6 User-Changeable Constants Used in the Sample Code (2/2)

Constant Name	Setting Value	Description
SUB_CLOCK_HZ ^(Note 1)	32,768L	Oscillation frequency of the resonator for the sub-clock (Hz)
WAIT_TIME_FOR_SUB_OSCILLATION ^(Note 1, Note 5)	1,500,000,000L	Oscillation stabilization time of the sub-clock (ns)
SEL_SYSCLK ^(Note 1)	CLK_PLL	Selection of the system clock source: CLK_HOCO: HOCO CLK_MOCO: MOCO CLK_MAIN: Main clock CLK_SUB: Sub-clock CLK_PLL: PLL clock
REG_SCKDIVCR ^(Note 1)	2112 2332h	Internal clock frequency division ratio (setting of the SCKDIVCR register)
REG_SCKDIVCR2 ^(Note 1)	1000h	Internal clock frequency division ratio (setting of the SCKDIVCR2 register)
REG_SDCKOCR ^(Note 1)	00h	Selection of the SDRAM clock (setting of the SDCKOCR register)
FREQ_MRAM_MRCLK ^(Note 1)	00000012Ch	Frequency of the extra MRAM: 300 MHz
FREQ_MRAM_MRPCLK ^(Note 1)	000000096h	Frequency of the extra MRAM: 150 MHz
FOR_CMT0_TIME_PLL ^(Note 1)	427	Interval (in ns) for counting by the oscillation stabilization timer (CMT0; with PCLKB = 75 MHz (max.) divided by 32)
SEL_SRAM_WAIT ^(Note 1)	MEMWAIT_ON	Selection of whether or not to insert SRAM waiting cycles MEMWAIT_ON: Waiting cycles to be used MEMWAIT_OFF: Waiting cycles not to be used
REG_VSCR ^(Note 2)	VDD_VSCR_1	Selection of the VDD target voltage VDD_VSCR_1: VSCR_1 VDD_VSCR_2: VSCR_2
MSTP_STATE_DMACDTC ^(Note 3)	MODULE_STOP_DISABLE	Selection of the module-stop state for DMAC and DTC: MODULE_STOP_DISABLE: Operating MODULE_STOP_ENABLE: Stopped
MSTP_STATE_EXDMAC ^(Note 3)	MODULE_STOP_DISABLE	Selection of the module-stop state for EXDMAC: MODULE_STOP_DISABLE: Operating MODULE_STOP_ENABLE: Stopped
MSTP_STATE_SRAM0 ^(Note 3)	MODULE_STOP_DISABLE	Selection of the module-stop state for SRAM0: MODULE_STOP_DISABLE: Operating MODULE_STOP_ENABLE: Stopped
MSTP_STATE_SRAM1 ^(Note 3)	MODULE_STOP_DISABLE	Selection of the module-stop state for SRAM1: MODULE_STOP_DISABLE: Operating MODULE_STOP_ENABLE: Stopped
PIN_SIZE ^(Note 4)	320	Number of pins for the product used

Note 1. Change the setting value in `r_init_clock.h` in accord with the user system.

Note 2. Change the setting value in `r_init_clock.h` in accord with the maximum frequency of the device in use.

Note 3. Change the setting value in `r_init_stop_module.h` in accord with the user system.

Note 4. Change the setting value in `r_init_port_initialize.h` in accord with the user system. Changing the settings of the bits corresponding to the absent port pins in the given device is also required. See section 3.2 for details.

Note 5. The sub-clock oscillation stabilization time depends on the resonator. Set the value in accord with section 3.3.2, Sub-clock Oscillation Stabilization Time.

Table 3.7 Non-User-Changeable Constants Used in the Sample Code

Constant Name	Setting Value	Description
B_NOT_USE	0	Disable
B_USE	1	Enable
B_USE_PLL_MAIN	2	The PLL clock is used. (Clock source: Main clock)
B_USE_PLL_HOCO	3	The PLL clock is used. (Clock source: HOCO)
B_RTC_CLK_SUB	0	The sub-clock is used as the clock source for the RTC.
B_RTC_CLK_LOCO	1	The LOCO is used as the clock source for the RTC.
B_RTC_CMD_CLD	0	Calendar counting is used as the RTC counting mode.
B_RTC_CMD_BIN	1	Binary counting is used as the RTC counting mode.
FREQ_16MHZ	0	HOCO frequency: 16 MHz
FREQ_18MHZ	1	HOCO frequency: 18 MHz
FREQ_20MHZ	2	HOCO frequency: 20 MHz
FREQ_32MHZ	3	HOCO frequency: 32 MHz
FREQ_48MHZ	4	HOCO frequency: 48 MHz
CLK_MAIN	03h	Clock source: Main clock
CLK_PLL	05h	Clock source: PLL
CLK_HOCO	00h	Clock source: HOCO
CLK_SUB	04h	Clock source: Sub-clock
CLK_MOCO	01h	Clock source: MOCO
SUB_CLOCK_CYCLE	1000000L/SUB_CLOCK_HZ	Cycle of the sub-clock (μ s)
LOCO_CLOCK_CYCLE	1000000L/LOCO_CLOCK_HZ	Cycle of the LOCO clock (μ s)
LOCO_CLOCK_HZ	32768L	Frequency of the LOCO clock (Hz)
FOR_CMT0_TIME	4000	Interval (in ns) for counting by the oscillation stabilization timer (CMT0; with MOCO = 8 MHz (max.) divided by 32)
FREQ_MRAM_DEFAULT	000000008h	Frequency of the extra MRAM: 8 MHz
REG_MRICKL	(FREQ_MRAM_MRICKL) (0x1E000000h)	Setting of the frequency of the MRAM code bus clock (setting of the respective register for MRICKL)
REG_MRPCLK	(FREQ_MRAM_MRPCLK) (0xE1000000h)	Setting of the frequency of the MRAM peripheral bus clock (setting of the respective register for MRPCLK)
MEMWAIT_ON	01h	SRAM waiting cycles to be used.
MEMWAIT_OFF	00h	SRAM waiting cycles not to be used.
VDD_VSCR_1	01h	VDD target voltage: VSCR_1
VDD_VSCR_2	02h	VDD target voltage: VSCR_2
MODULE_STOP_ENABLE	1	Transition to the module-stop state is made.
MODULE_STOP_DISABLE	0	Release from the module-stop state

Table 3.8 Constants for 320-Pin Products (PIN_SIZE = 320)

Constant Name	Setting Value	Description
DEF_P0PDR	0000h	Value set in the port direction register for port P0
DEF_P1PDR	0000h	Value set in the port direction register for port P1
DEF_P2PDR	0000h	Value set in the port direction register for port P2
DEF_P3PDR	0000h	Value set in the port direction register for port P3
DEF_P4PDR	0000h	Value set in the port direction register for port P4
DEF_P5PDR	0000h	Value set in the port direction register for port P5
DEF_P6PDR	0000h	Value set in the port direction register for port P6
DEF_P7PDR	0000h	Value set in the port direction register for port P7
DEF_P8PDR	0000h	Value set in the port direction register for port P8
DEF_P9PDR	0000h	Value set in the port direction register for port P9
DEF_PAPDR	0000h	Value set in the port direction register for port PA
DEF_PBPDR	0000h	Value set in the port direction register for port PB
DEF_PCPDR	0000h	Value set in the port direction register for port PC
DEF_PDPDR	0000h	Value set in the port direction register for port PD
DEF_PEPDR	0000h	Value set in the port direction register for port PE
DEF_PFPDR	0000h	Value set in the port direction register for port PF
DEF_PGPDR	0000h	Value set in the port direction register for port PG

Table 3.9 Constants for 224-Pin Products (PIN_SIZE = 224)

Constant Name	Setting Value	Description
DEF_P0PDR	FF00h	Value set in the port direction register for port P0
DEF_P1PDR	0000h	Value set in the port direction register for port P1
DEF_P2PDR	0000h	Value set in the port direction register for port P2
DEF_P3PDR	0000h	Value set in the port direction register for port P3
DEF_P4PDR	3F00h	Value set in the port direction register for port P4
DEF_P5PDR	FF00h	Value set in the port direction register for port P5
DEF_P6PDR	0000h	Value set in the port direction register for port P6
DEF_P7PDR	FF00h	Value set in the port direction register for port P7
DEF_P8PDR	0000h	Value set in the port direction register for port P8
DEF_P9PDR	0000h	Value set in the port direction register for port P9
DEF_PAPDR	0000h	Value set in the port direction register for port PA
DEF_PBPDR	0000h	Value set in the port direction register for port PB
DEF_PCPDR	0000h	Value set in the port direction register for port PC
DEF_PDPDR	0000h	Value set in the port direction register for port PD
DEF_PEPDR	0000h	Value set in the port direction register for port PE
DEF_PFPDR	0000h	Value set in the port direction register for port PF
DEF_PGPDR	FF00h	Value set in the port direction register for port PG

Table 3.10 Constants for 176-Pin Products (PIN_SIZE = 176)

Constant Name	Setting Value	Description
DEF_P0PDR	FF00h	Value set in the port direction register for port P0
DEF_P1PDR	FF00h	Value set in the port direction register for port P1
DEF_P2PDR	FF00h	Value set in the port direction register for port P2
DEF_P3PDR	0000h	Value set in the port direction register for port P3
DEF_P4PDR	3F00h	Value set in the port direction register for port P4
DEF_P5PDR	FFF0h	Value set in the port direction register for port P5
DEF_P6PDR	0000h	Value set in the port direction register for port P6
DEF_P7PDR	FF00h	Value set in the port direction register for port P7
DEF_P8PDR	00C0h	Value set in the port direction register for port P8
DEF_P9PDR	00F0h	Value set in the port direction register for port P9
DEF_PAPDR	0000h	Value set in the port direction register for port PA
DEF_PBPDR	0000h	Value set in the port direction register for port PB
DEF_PCPDR	0000h	Value set in the port direction register for port PC
DEF_PDPDR	0000h	Value set in the port direction register for port PD
DEF_PEPDR	0000h	Value set in the port direction register for port PE
DEF_PFPDR	03FCh	Value set in the port direction register for port PF
DEF_PGPDR	FFF8h	Value set in the port direction register for port PG

Table 3.11 Constants for 144-Pin Products (PIN_SIZE = 144)

Constant Name	Setting Value	Description
DEF_P0PDR	FF00h	Value set in the port direction register for port P0
DEF_P1PDR	FF00h	Value set in the port direction register for port P1
DEF_P2PDR	FF00h	Value set in the port direction register for port P2
DEF_P3PDR	0000h	Value set in the port direction register for port P3
DEF_P4PDR	3FC0h	Value set in the port direction register for port P4
DEF_P5PDR	FFFFh	Value set in the port direction register for port P5
DEF_P6PDR	00C0h	Value set in the port direction register for port P6
DEF_P7PDR	FFF8h	Value set in the port direction register for port P7
DEF_P8PDR	00F0h	Value set in the port direction register for port P8
DEF_P9PDR	00FCh	Value set in the port direction register for port P9
DEF_PAPDR	0000h	Value set in the port direction register for port PA
DEF_PBPDR	0000h	Value set in the port direction register for port PB
DEF_PCPDR	0000h	Value set in the port direction register for port PC
DEF_PDPDR	0000h	Value set in the port direction register for port PD
DEF_PEPDR	0000h	Value set in the port direction register for port PE
DEF_PFPDR	03FCh	Value set in the port direction register for port PF
DEF_PGPDR	FFFFh	Value set in the port direction register for port PG

Table 3.12 Constants for 100-Pin Products (PIN_SIZE = 100)

Constant Name	Setting Value	Description
DEF_P0PDR	FF01h	Value set in the port direction register for port P0
DEF_P1PDR	FF01h	Value set in the port direction register for port P1
DEF_P2PDR	FFFCh	Value set in the port direction register for port P2
DEF_P3PDR	0000h	Value set in the port direction register for port P3
DEF_P4PDR	BFEAh	Value set in the port direction register for port P4
DEF_P5PDR	FFFFh	Value set in the port direction register for port P5
DEF_P6PDR	00FCh	Value set in the port direction register for port P6
DEF_P7PDR	FFFFh	Value set in the port direction register for port P7
DEF_P8PDR	00FFh	Value set in the port direction register for port P8
DEF_P9PDR	00FFh	Value set in the port direction register for port P9
DEF_PAPDR	0000h	Value set in the port direction register for port PA
DEF_PBPDR	0080h	Value set in the port direction register for port PB
DEF_PCPDR	0080h	Value set in the port direction register for port PC
DEF_PDPDR	0000h	Value set in the port direction register for port PD
DEF_PEPDR	0080h	Value set in the port direction register for port PE
DEF_PFPDR	03FCh	Value set in the port direction register for port PF
DEF_PGPDR	FFFFh	Value set in the port direction register for port PG

3.8 Functions

Table 3.13 lists the functions.

Table 3.13 Functions

Function Name	Outline
main	Main processing
R_INIT_StopModule	Stopping peripheral modules that are still operating after a reset
R_INIT_Port_Initialize	Initialization of the bits corresponding to absent port pins
R_INIT_Clock	Clock initialization
CGC_oscillation_main	Main clock oscillation setting
CGC_oscillation_HOCO	HOCO clock oscillation setting
CGC_oscillation_PLL1	PLL1 clock oscillation setting
CGC_oscillation_sub	Sub-clock oscillation setting
init_rtc	The RTC is initialized before use.
cmt0_wait	Software waiting is controlled by using CMT0.

3.9 Function Specifications

The following tables list the specifications of sample code functions.

main	
Outline	Main processing
Header	None
Declaration	void main (void)
Description	Calls the functions that stop peripheral modules that are still operating after a reset, initialize the bits corresponding to absent port pins, and initialize clocks.
Arguments	None
Return Value	None
R_INIT_StopModule	
Outline	Stopping the peripheral modules that are still operating after a reset
Header	r_init_stop_module.h
Declaration	void R_INIT_StopModule (void)
Description	Configures the settings for modules to make transitions to the module-stop state.
Arguments	None
Return Value	None
Remarks	Transitions to the module-stop state do not proceed in the sample code.
R_INIT_Port_Initialize	
Outline	Initialization of the bits corresponding to absent port pins
Header	r_init_port_initialize.h
Declaration	void R_INIT_Port_Initialize (void)
Description	Initializes port direction register bits that correspond to absent port pins.
Arguments	None
Return Value	None
Remarks	In the sample code, use of the 320-pin product is set (PIN_SIZE = 320). If data are written in byte units to the PDR register after the call of this function, set the direction-control bits for the absent port pins to 1. Also, set the output data storage bits corresponding to port pins set as outputs to 0.
R_INIT_Clock	
Outline	Clock initialization
Header	r_init_clock.h
Declaration	void R_INIT_Clock (void)
Description	Initializes clocks.
Arguments	None
Return Value	None
Remarks	In the sample code, the PLL is selected as the system clock, and the sub-clock and RTC are not used.

CGC_oscillation_main	
Outline	Main clock oscillation setting
Header	r_init_clock.h
Declaration	void CGC_oscillation_main (void)
Description	Sets the driving ability of the main clock and sets the MOSCWTCR register, and then starts oscillation of the main clock. After that, waiting until oscillation of the main clock becomes stable proceeds.
Arguments	None
Return Value	None
CGC_oscillation_HOCO	
Outline	HOCO clock oscillation setting
Header	r_init_clock.h
Declaration	void CGC_oscillation_HOCO (void)
Description	Starts oscillation of the HOCO. After that, waiting until oscillation of the HOCO becomes stable proceeds.
Arguments	None
Return Value	None
CGC_oscillation_PLL1	
Outline	PLL1 clock oscillation setting
Header	r_init_clock.h
Declaration	void CGC_oscillation_PLL1 (void)
Description	Sets the PLL1 input frequency division ratio and frequency multiplication factor, and then starts oscillation of the PLL1 clock. After that, waiting until oscillation of the PLL1 clock becomes stable proceeds.
Arguments	None
Return Value	None
CGC_oscillation_sub	
Outline	Sub-clock oscillation setting
Header	r_init_clock.h
Declaration	void CGC_oscillation_sub (void)
Description	Sets the driving ability of the sub-clock and then starts oscillation of the sub-clock. After that, waiting until oscillation of the sub-clock becomes stable proceeds.
Arguments	None
Return Value	None
init_rtc	
Outline	RTC initialization
Header	None
Declaration	static void init_rtc (void)
Description	Initializes the RTC, i.e., software sets the clock supply and resets the RTC.
Arguments	None
Return Value	None

cmt0_wait	
Outline	Sets CMT control of software waiting
Header	None
Declaration	static void cmt0_wait(uint32_t cnt)
Description	Software waiting proceeds with the use of CMT0.
Arguments	uint32_t cnt: Waiting time $\text{cnt} = \text{Waiting time (ns)} \div \text{FOR_CMT0_TIME}^{(\text{Note 1})}$ $\text{cnt} = \text{Waiting time (ns)} \div \text{FOR_CMT0_TIME_PLL}^{(\text{Note 2})}$
Return Value	None
Remarks	Note 1. FOR_CMT0_TIME is calculated with MOCO = 8 MHz (max). The actual waiting time depends on the MOCO frequency. Note 2. FOR_CMT0_TIME_PLL is calculated with PCLKB = 75 MHz (max). The actual waiting time depends on the PCLKB frequency.

3.10 Flowcharts

3.10.1 Main Processing

Figure 3.2 is the flowchart of the main processing.

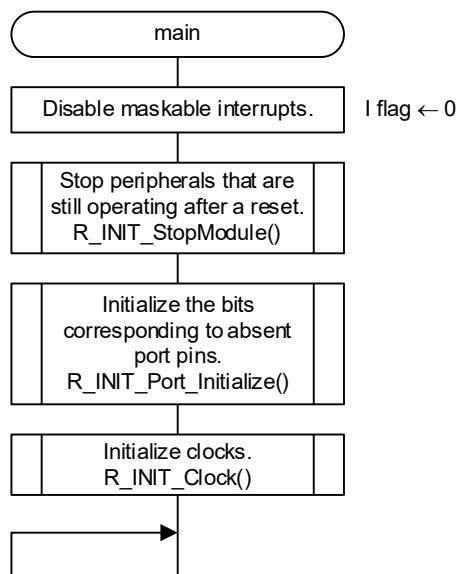
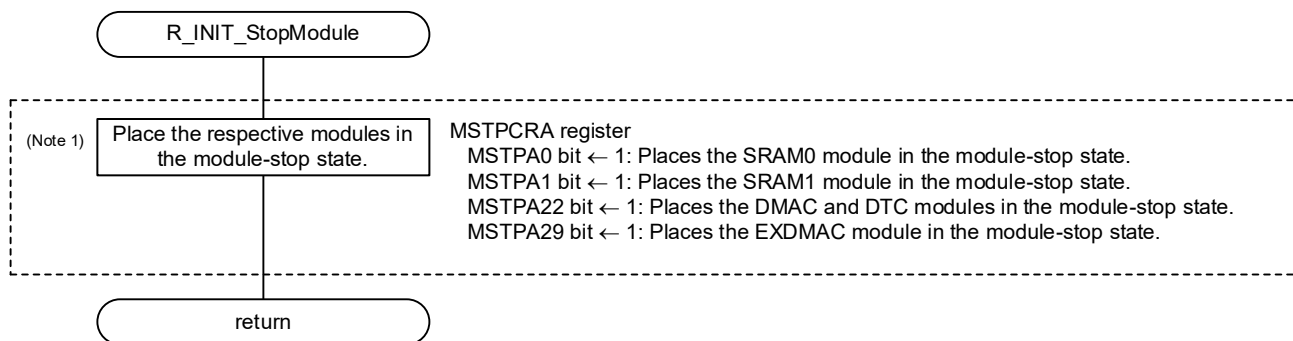


Figure 3.2 Main Processing

3.10.2 Stopping Peripheral Modules that Are Still Operating After a Reset

Figure 3.3 is the flowchart of stopping peripheral modules that are still operating after a reset.



Note 1. In the sample code, none of these modules is placed in the module-stop state. To place these modules in the module-stop state, set the constants of the form "#define MSTP_STATE_<target module name>" to 1.

Figure 3.3 Stopping Peripheral Modules that Are Still Operating After a Reset

3.10.3 Initialization of the Bits Corresponding to Absent Port Pins

Figure 3.4 is the flowchart of initialization of the bits corresponding to absent port pins.

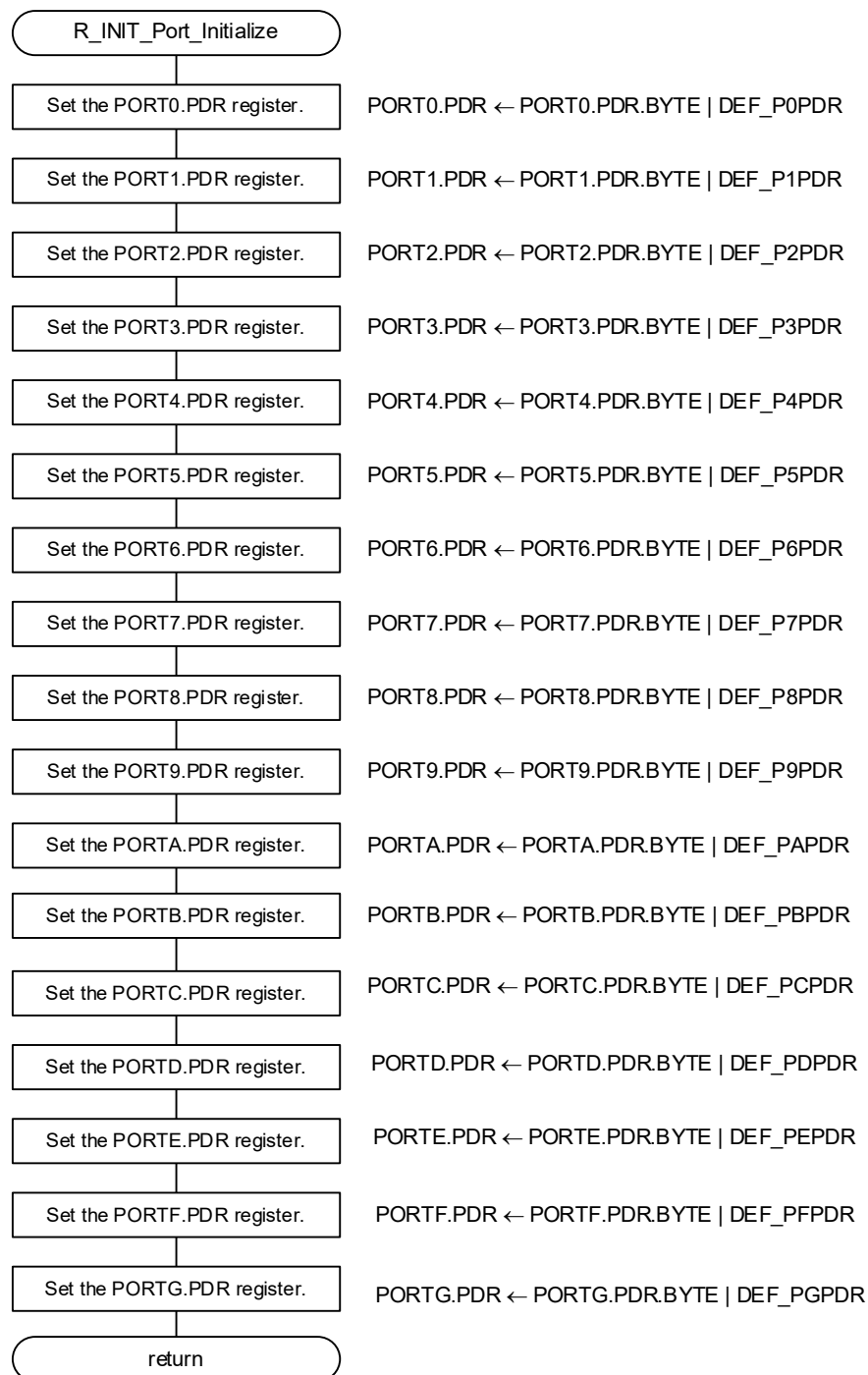
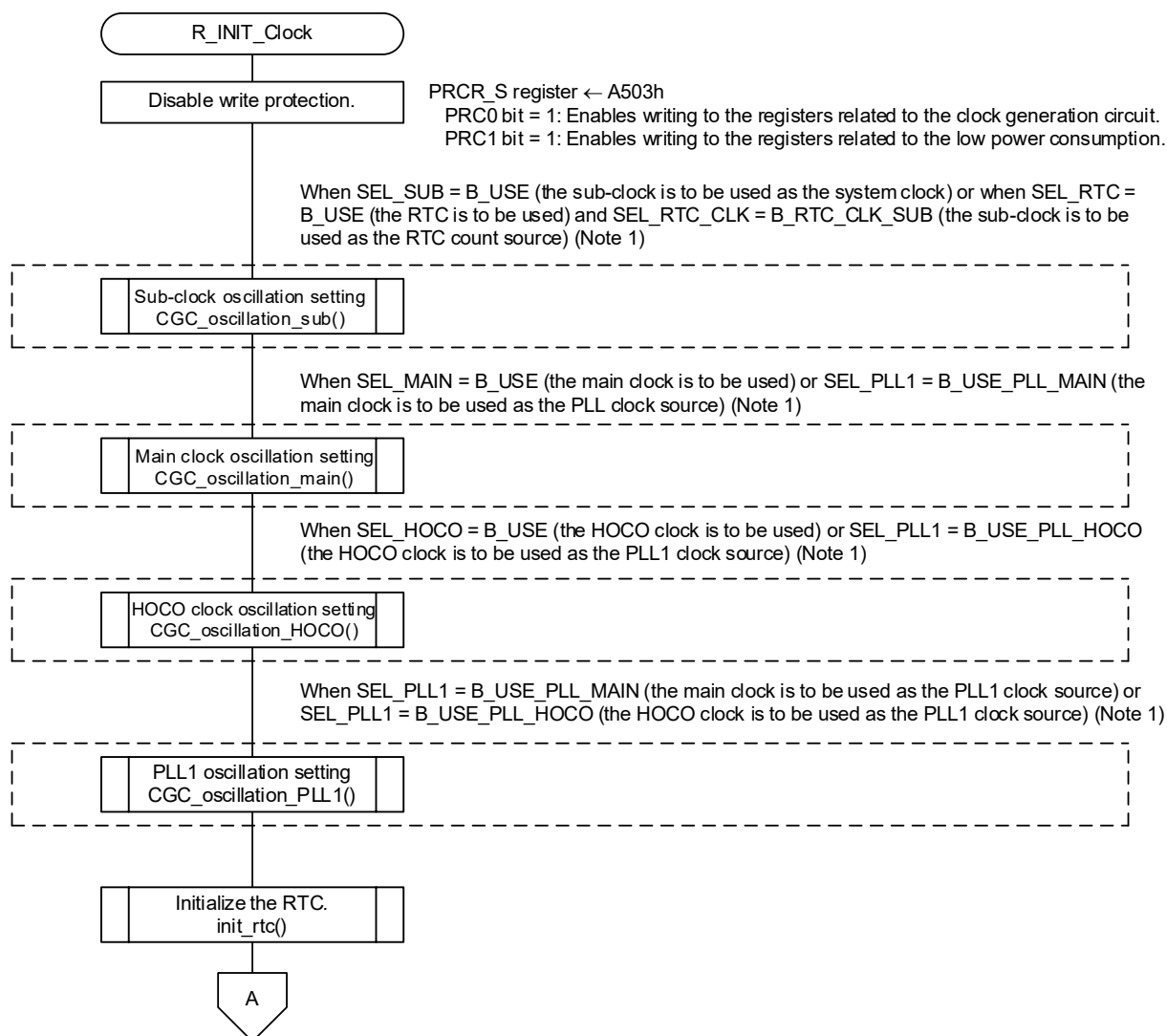


Figure 3.4 Initialization of the Bits Corresponding to Absent Port Pins

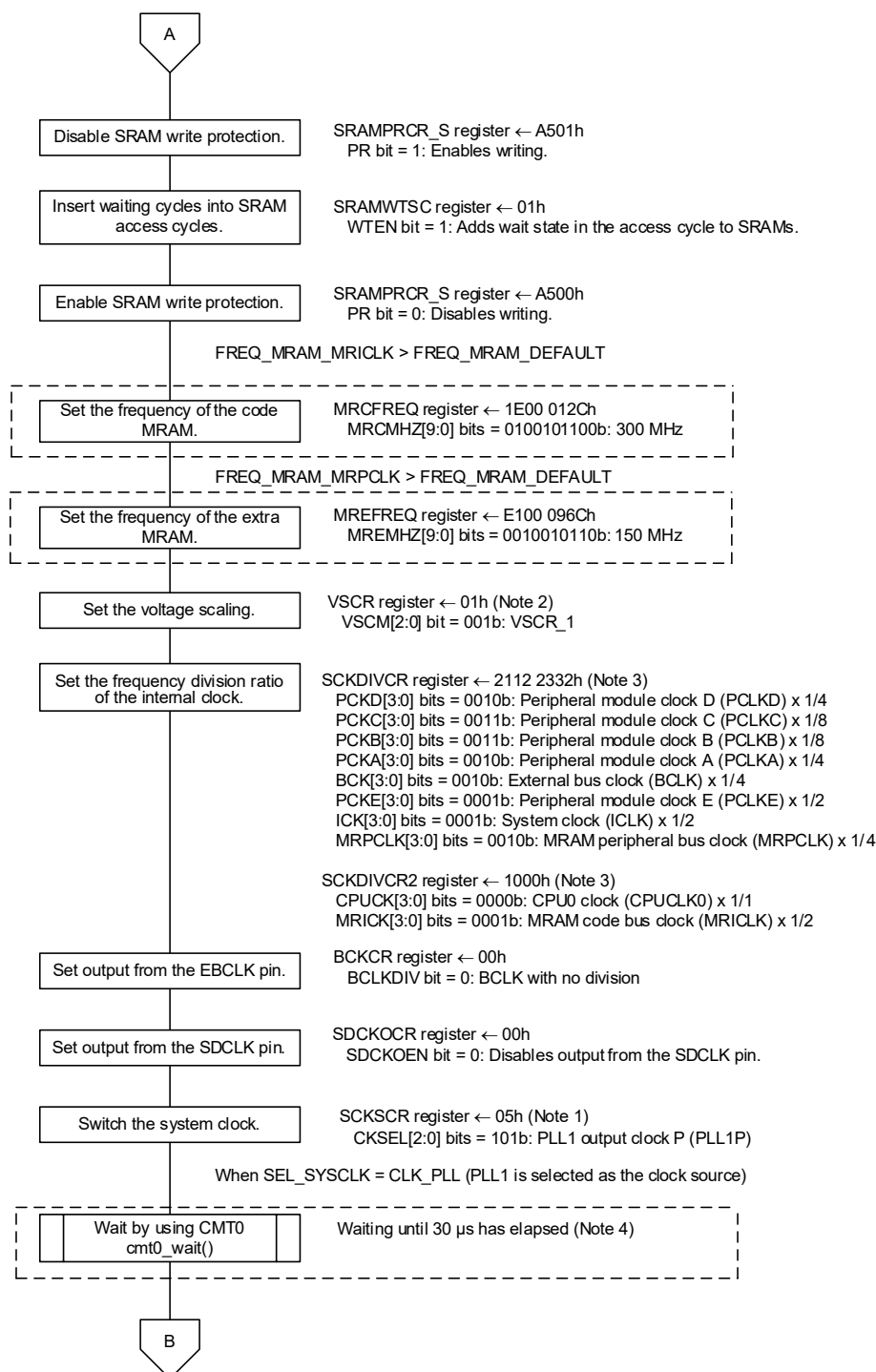
3.10.4 Clock Initialization

Figure 3.5 to Figure 3.7 are the flowcharts of clock initialization.



Note 1. Change the `SEL_MAIN`, `SEL_PLL1`, `SEL_HOCO`, `SEL_SUB`, and `SEL_RTC` constant settings in "r_init_clock.h" in accord with the user system. See Table 3.5 for details.

Figure 3.5 Clock Initialization (1/3)



Note 1. Change the SEL_SYSCCLK constant setting in "r_init_clock.h" in accord with the user system.
See Table 3.6 for details.

Note 2. Only set VSCR_1 as the target voltage for VDD if the maximum CPU clock frequency of the device is 600 MHz.

Note 3. The value to be set depends on the system clock source selected by the SEL_SYSCCLK constant in "r_init_clock.h".

Note 4. Set the FOR_CMT0_TIME_PLL constant in accord with the frequency of PCLKB to be specified.

Figure 3.6 Clock Initialization (2/3)

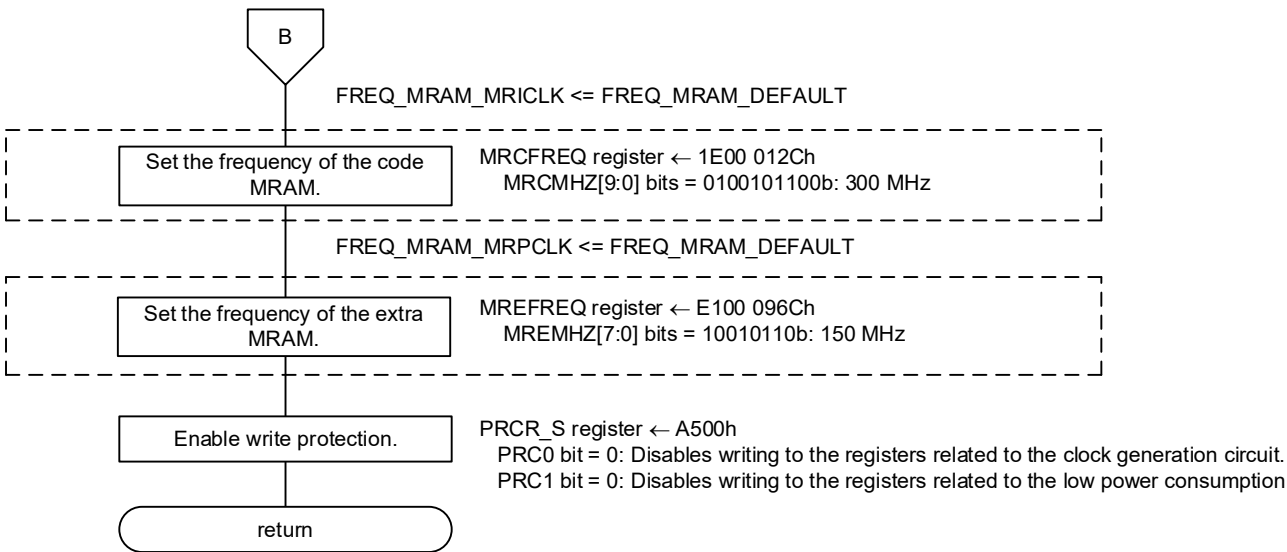
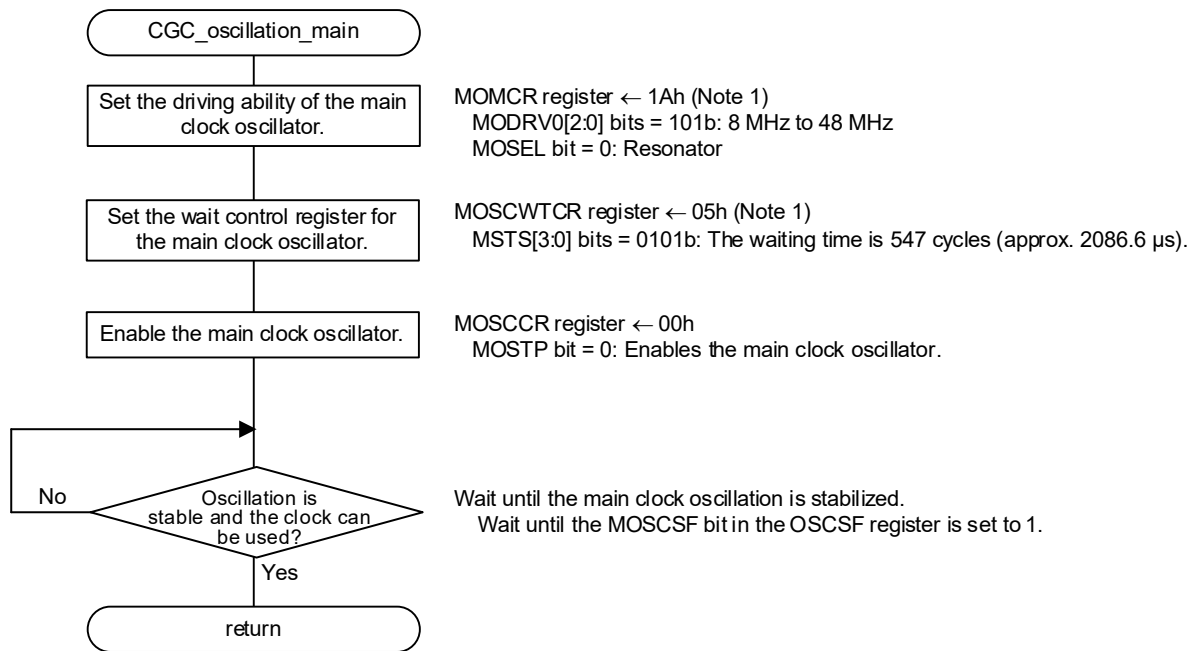


Figure 3.7 Clock Initialization (3/3)

3.10.5 Main Clock Oscillation Setting

Figure 3.8 is the flowchart of the main clock oscillation setting.

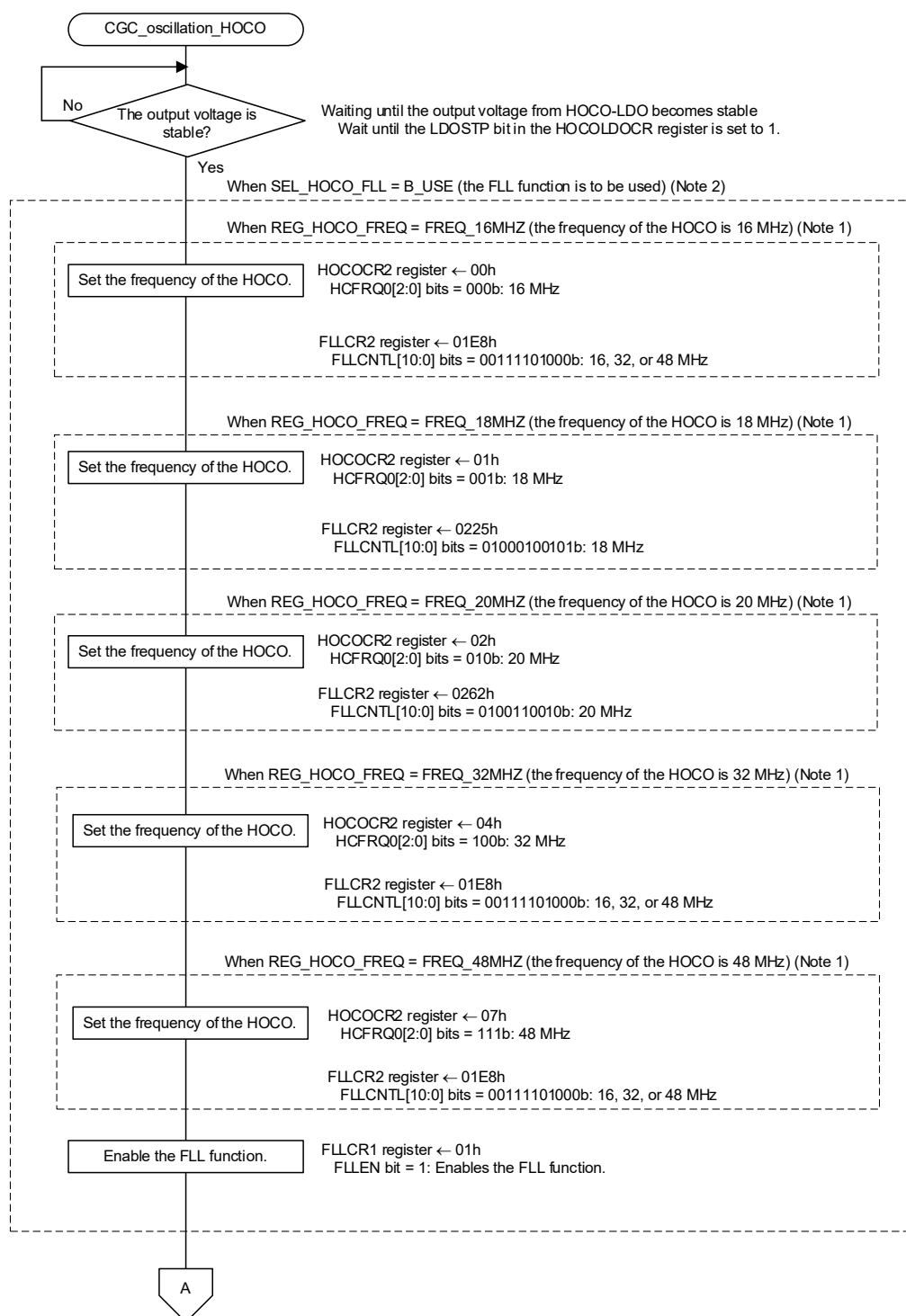


Note 1. Change the REG_MOMCR and REG_MOSCWTCR constant settings in "r_init_clock.h" in accord with the user system. See Table 3.5 for details.

Figure 3.8 Main Clock Oscillation Setting

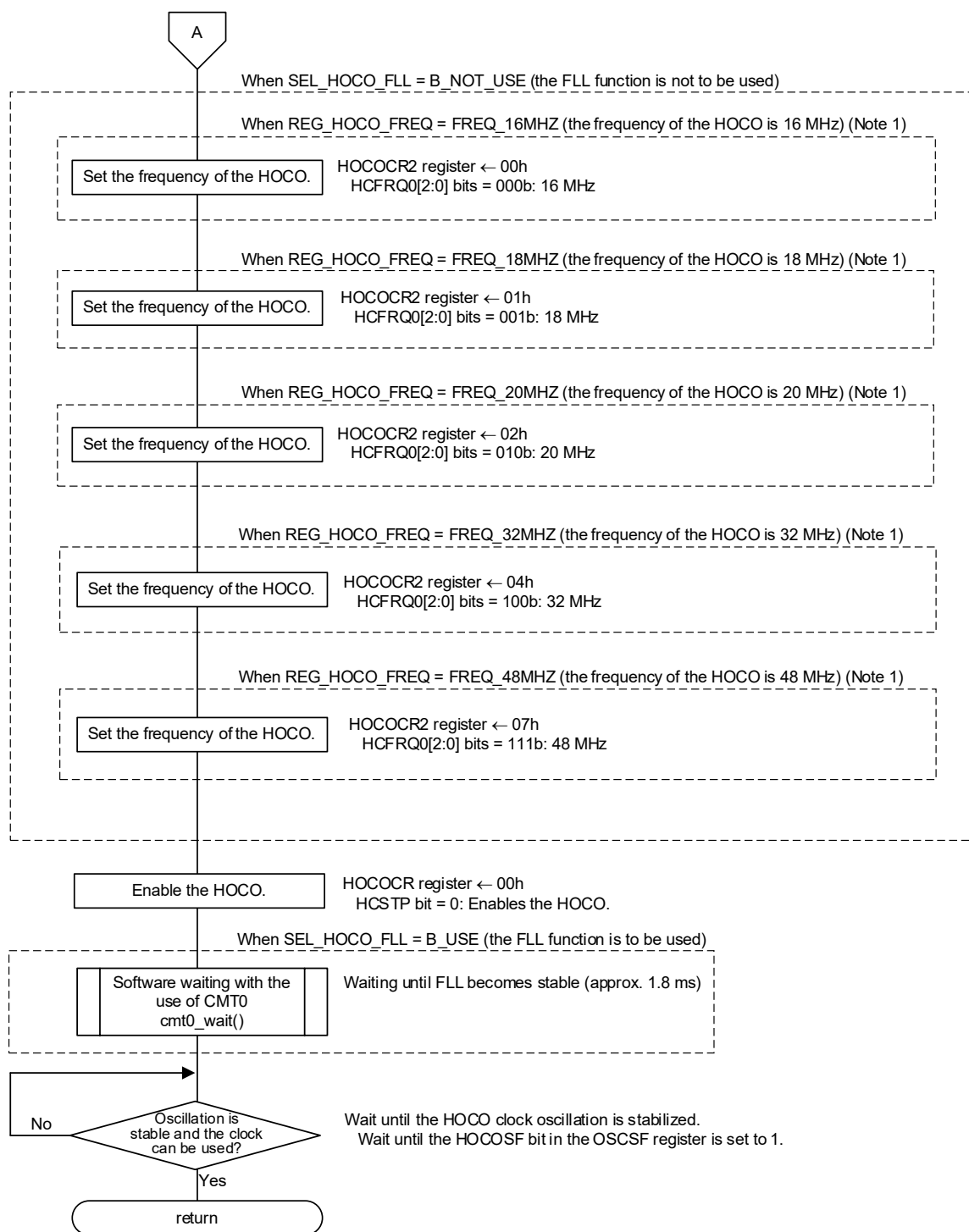
3.10.6 HOCO Clock Oscillation Setting

Figure 3.9 and Figure 3.10 are the flowcharts of the HOCO clock oscillation setting.



Note 1. Change the REG_HOCO_FREQ constant setting in "r_init_clock.h" in accord with the user system. See Table 3.5 for details.
Note 2. Start the sub-clock oscillator before using the FLL function.

Figure 3.9 HOCO Clock Oscillation Setting (1/2)

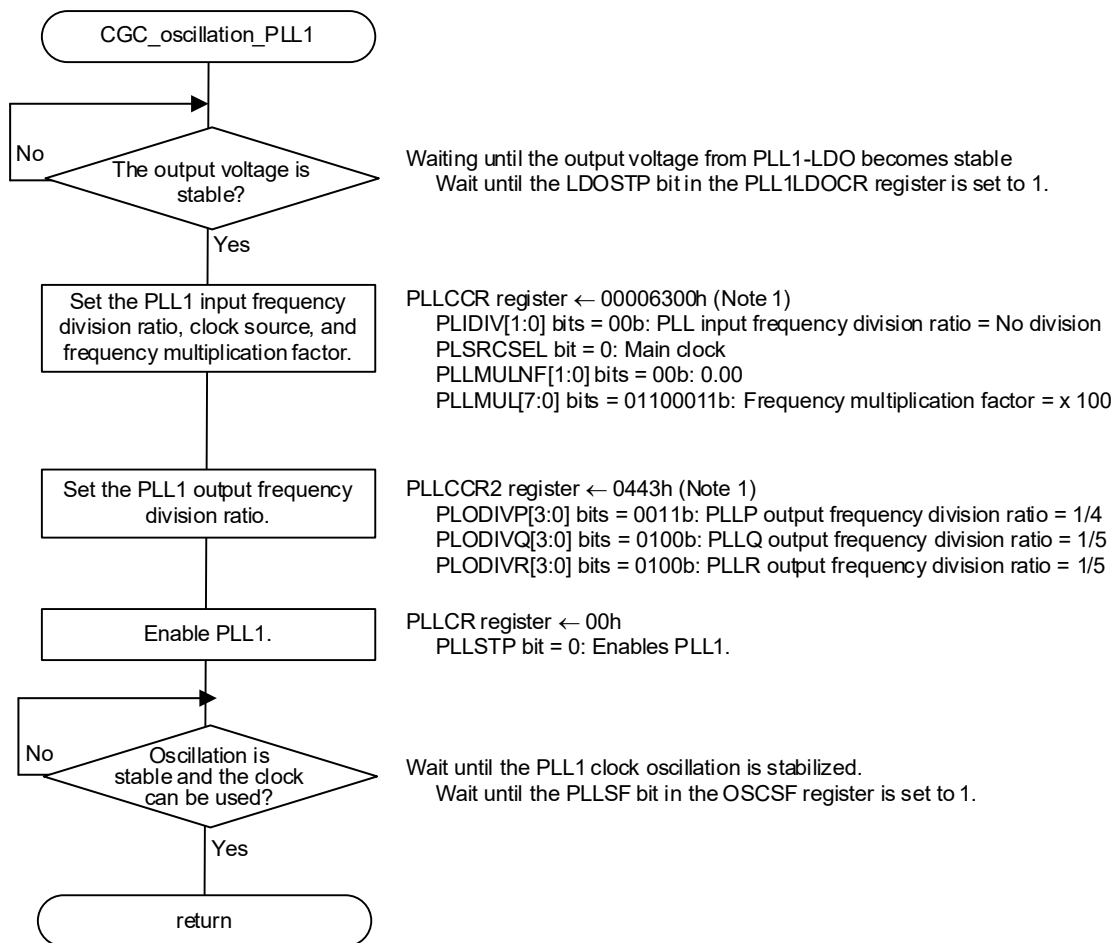


Note 1. Change the REG_HOCO_FREQ constant setting in "r_init_clock.h" in accord with the user system. See Table 3.5 for details.

Figure 3.10 HOCO Clock Oscillation Setting (2/2)

3.10.7 PLL1 Clock Oscillation Setting

Figure 3.11 is the flowchart of the PLL1 clock oscillation setting.

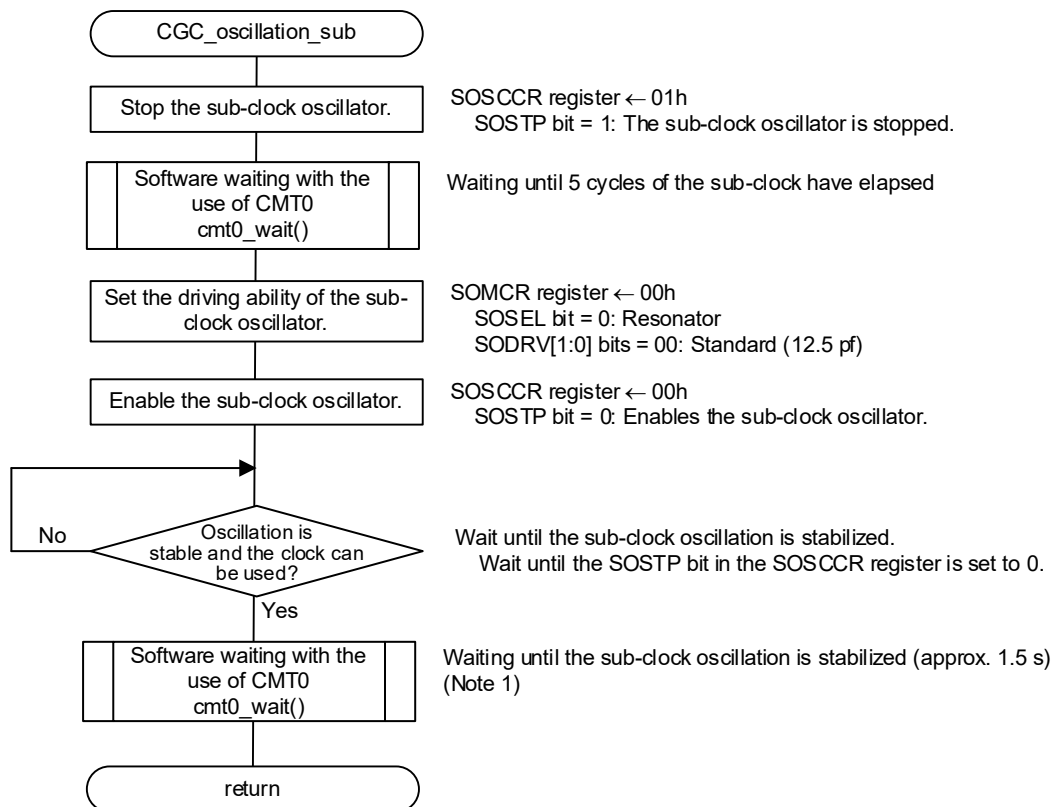


Note 1. Change the REG_PLLCCR and REG_PLLCCR2 constant settings in "r_init_clock.h" in accord with the user system. See Table 3.5 for details.

Figure 3.11 PLL1 Clock Oscillation Setting

3.10.8 Sub-clock Oscillation Setting

Figure 3.12 is the flowchart of the sub-clock oscillation setting.

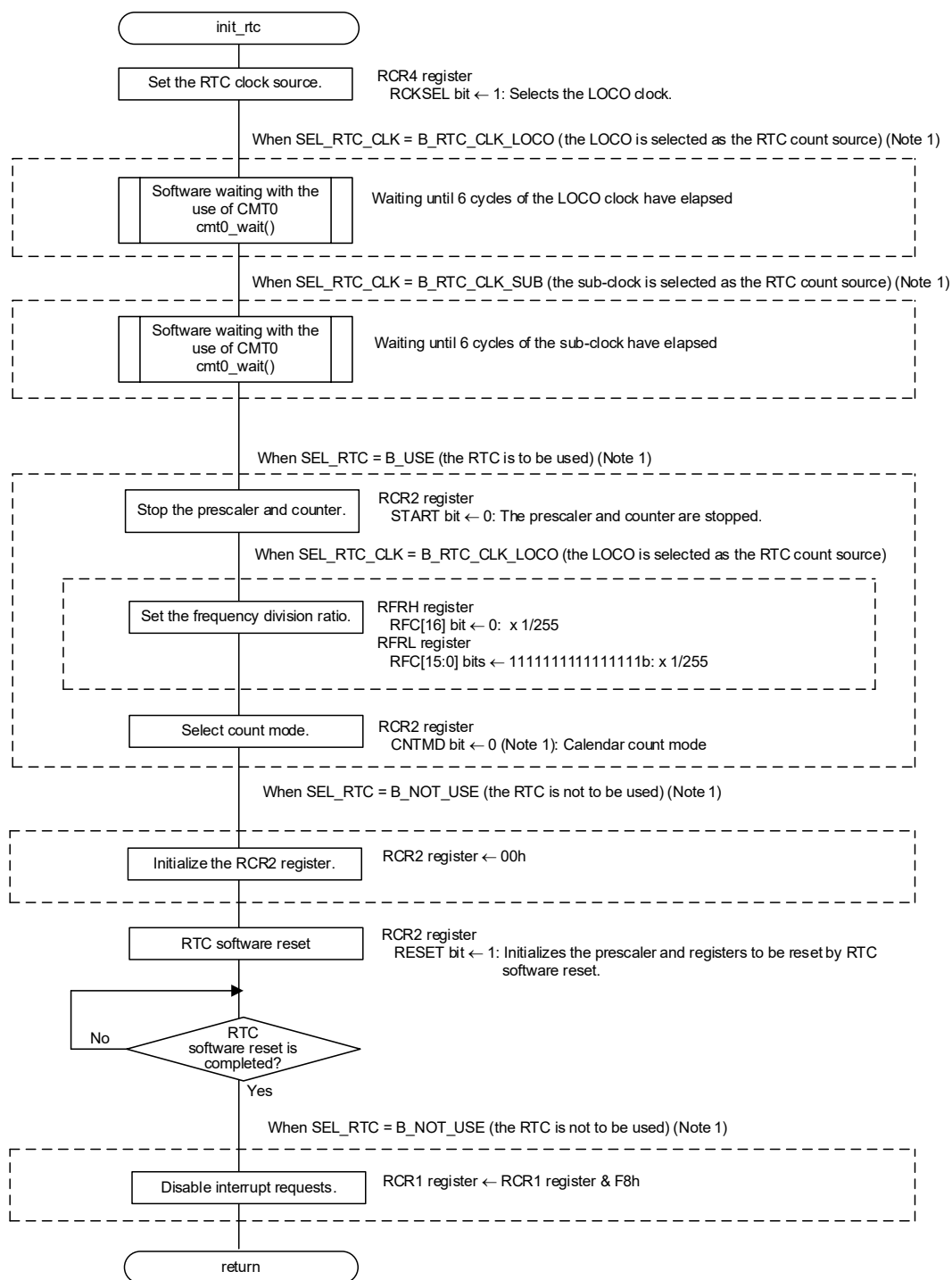


Note 1. Change the WAIT_TIME_FOR_SUB_OSCILLATION constant setting in "r_init_clock.h" in accord with the user system. See Table 3.6 for details.

Figure 3.12 Sub-clock Oscillation Setting

3.10.9 Initialization of the RTC

Figure 3.13 is the flowchart for initializing the RTC.



Note 1. Change the SEL_RTC and SEL_RTC_CLK constant settings in "r_init_clock.h" in accord with the user system. See Table 3.5 for details.

Figure 3.13 Initialization of the RTC before Use

3.10.10 Setting up Software Waiting with the Use of CMT0

Figure 3.14 is the flowchart for setting up software waiting with the use of CMT0.

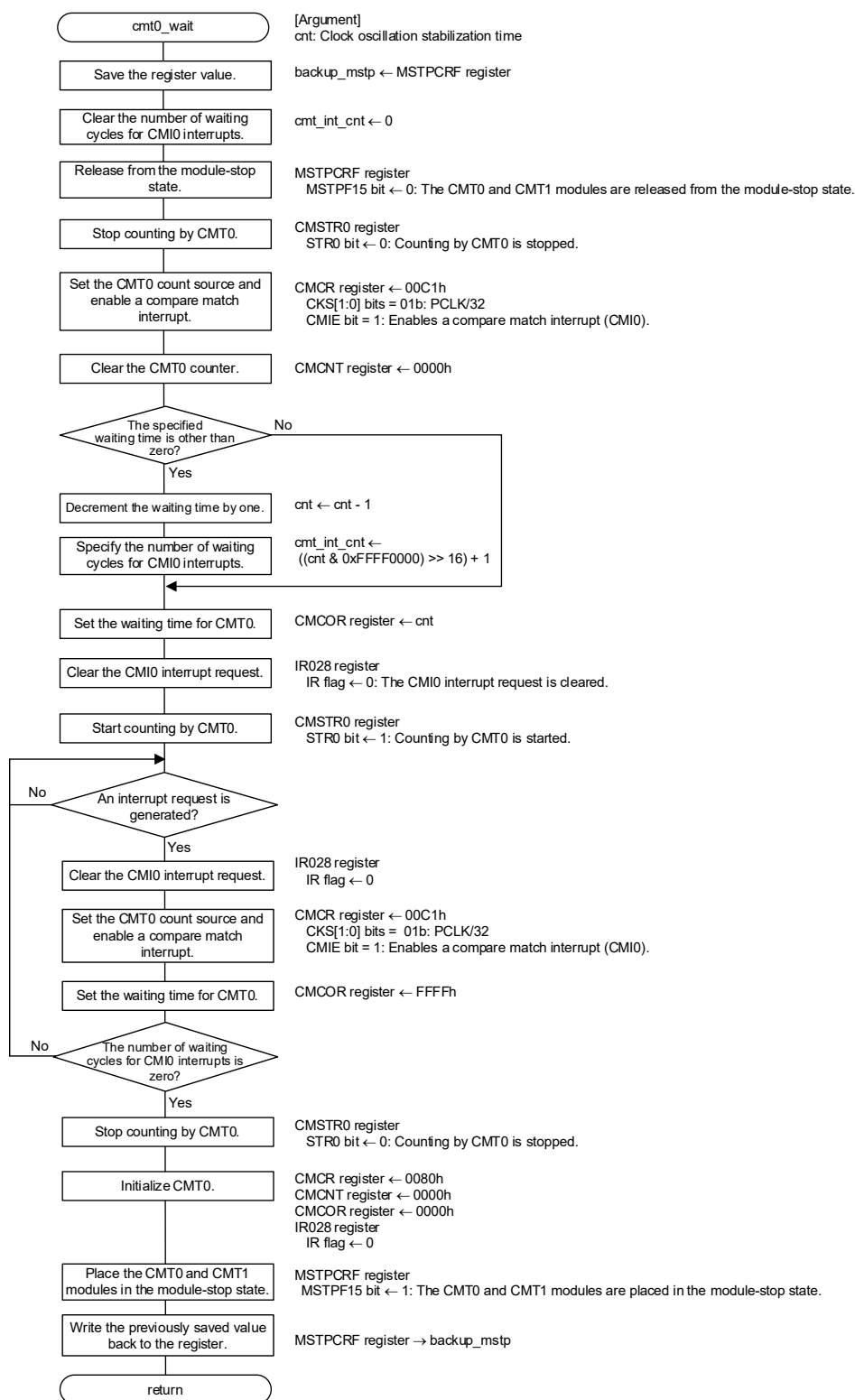


Figure 3.14 Setting up Software Waiting with the Use of CMT0

4. Importing a Project

The sample code is provided in the form of an e² studio project. Follow the steps below to import the file into the e² studio. After importing a project, confirm that the build settings and the debug settings are correct. The windows and dialog boxes shown in the following procedure may slightly differ from those that are actually displayed, depending on the version of e² studio you are using.

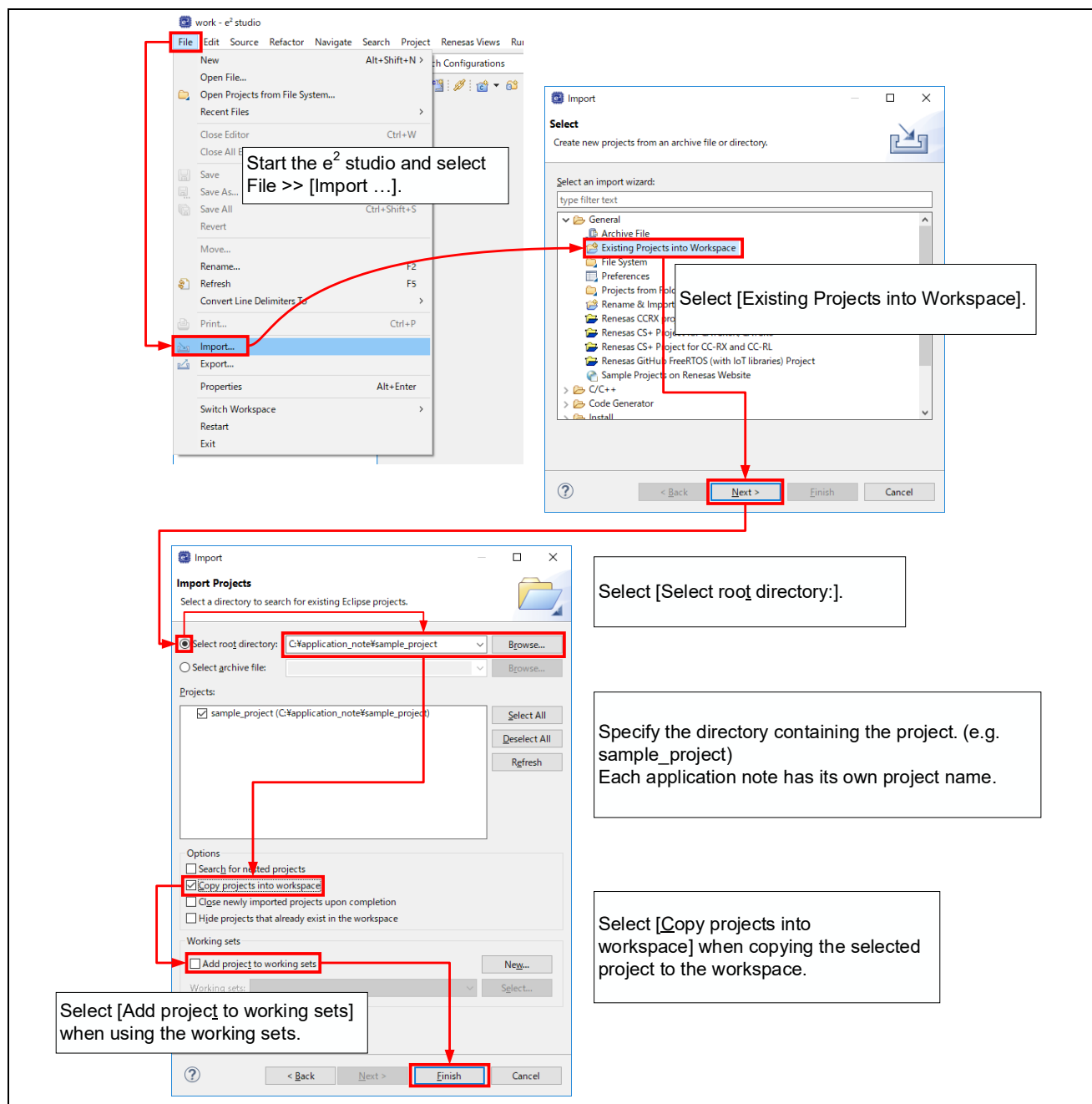


Figure 4.1 Importing a Project into the e² studio

5. Sample Code

Download the desired sample code from the Renesas Electronics website.

6. Reference Documents

User's Manual: Hardware

RX74M Group User's Manual: Hardware (R01UH1055)

(The latest version of this document is available at the Renesas Electronics website.)

User's Manual: Development environment

RX Family CC-RX Compiler User's Manual (R20UT3248)

(The latest version of this document is available at the Renesas Electronics website.)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jun.30,26	—	First edition issued

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

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

1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

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

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

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

6. Voltage application waveform at input pin

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

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

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Trademarks

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

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:

www.renesas.com/contact/.