RX210 Group

Initial Setting

Abstract

This document describes settings required after a reset such as clock settings, stop processing for active peripheral functions after a reset, or nonexistent port initialization according to the conditions selected in the header file.

Products

- RX210 Group 145-pin and 144-pin packages with a ROM size between 128 KB and 1 MB
- RX210 Group 100-pin package with a ROM size between 128 KB and 1 MB
- RX210 Group 80-pin package with a ROM size between 64 KB and 512 KB
- RX210 Group 69-pin package with a ROM size between 128 KB and 256 KB
- RX210 Group 64-pin package with a ROM size between 64 KB and 512 KB
- RX210 Group 48-pin package with a ROM size between 64 KB and 256 KB

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Contents

1. Specifications

In the sample code, peripheral functions operating after a reset are stopped, and nonexistent port and clock settings are configured. The application note assumes processing at power-on (cold start).

1.1 Stopping Peripheral Functions Operating after a Reset

Some peripheral functions operate at power-on, and the module-stop function is disabled for some. These include the DMAC, DTC, RAM0, and RAM1. Although the sample code includes processing for stopping these peripheral functions, it is not executed in the sample code. Change the constant as required to execute processing.

1.2 Configuring Nonexistent Ports

Ports which are not connected to pins must be set as output for products with less than 144 pins. In the sample code, initial values are set for 100-pin products. Change the value according to the product used.

1.3 Setting Clocks

1.3.1 Overview

The clock setting procedure is as follows:

1. Sub-clock setting
2. Main clock setting
3. PLL clock setting
4. HOCO clock setting
5. System clock switching

In this application note, the clock settings are switched by changing the constants defined in r_init_clock.h.

In the sample code, the PLL clock is used as the system clock without using the sub-clock. Change the constant to select the required clock setting.

1.3.2 Clock Specifications Used in the Sample Code

Table 1.1 lists the Clock Specifications Used in the Sample Code. Values such as the oscillation stabilization wait time are calculated using values listed in Table 1.1.

Table 1.2 lists the Peripheral Function and Its Application.

Table 1.1 Clock Specifications Used in the Sample Code

Clock	Oscillation Frequency	Oscillation Stabilization Time	Remarks
Crystal/ceramic resonator for the main clock	20 MHz	4.2 ms [2]	Crystal used
Crystal for the sub-clock	32.768 kHz [1]	1.3 sec. [2]	For low clock loads
PLL clock	100 MHz (main clock divided by 2 and multiplied by 10)	Maximum of 500 μs [3]	
HOCO clock	50 MHz [1]	Maximum of 175 μs [3]	

Notes:

1. Sub-clock oscillation is disabled in the sample code.
2. The oscillation stabilization time of a crystal/ceramic resonator differs depending on the wiring pattern, conditions of oscillation parameters, and other settings in the user system. Ask the crystal/ceramic resonator manufacturer to evaluate the user system and provide an appropriate oscillation stabilization time.
3. Refer to the Electrical Characteristics chapter in the User's Manual: Hardware.

Table 1.2 Peripheral Function and Its Application

Peripheral Function	Application
Compare match timer, channel 0 (CMT0)	Measuring the clock oscillation stabilization wait time [1]

Note:

1. When using OS, select a channel for a timer that is not being used by OS.

1.3.3 Selecting Clocks

In the sample code, users can select the system clock source, whether clocks to be oscillating or stopped, and other settings by changing constants defined in r_init_clock.h. Refer to Table 3.11 and Table 3.12 for constants that can be changed.

Table 1.3 lists Examples of Clock Selections. In the sample code, processing No. 1, which uses the PLL clock as the system clock without using the sub-clock, is selected.

Table 1.3 Examples of Clock Selections

No.		1	2	3	4	5	6
System clock		PLL	PLL	HOCO	HOCO	Main clock	Main clock
PLL clock		Oscillating	Oscillating	Stopped	Stopped	Stopped	Stopped
Main clock		Oscillating	Oscillating	Stopped	Stopped	Oscillating	Oscillating
HOCO clock		Stopped	Stopped	Oscillating (50 MHz)	Oscillating (50 MHz)	Stopped	Stopped
Sub-clock (2)		Stopped	Oscillating (RTC used)	Stopped	Oscillating (RTC used)	Stopped	Oscillating (RTC used)
Operating mode		High-speed operating mode	High-speed operating mode	High-speed operating mode	High-speed operating mode	Middle-speed operating mode 1A	Middle-speed operating mode 1A
Constants	SEL_SYSCLK	CLK_PLL	CLK_PLL	CLK_HOCO	CLK_HOCO	CLK_MAIN	CLK_MAIN
	SEL_PLL	B_USE	B_USE	B_NOT_USE	B_NOT_USE	B_NOT_USE	B_NOT_USE
	SEL_MAIN	B_USE	B_USE	B_NOT_USE	B_NOT_USE	B_USE	B_USE
	SEL_HOCO	B_NOT_USE	B_NOT_USE	B_USE	B_USE	B_NOT_USE	B_NOT_USE
	SEL_SUB (1)	B_NOT_USE	B_NOT_USE	B_NOT_USE	B_NOT_USE	B_NOT_USE	B_NOT_USE
	SEL_RTC (1)	B_NOT_USE	B_USE	B_NOT_USE	B_USE	B_NOT_USE	B_USE
	REG_OPCCR	OPCM_HIGH	OPCM_HIGH	OPCM_HIGH	OPCM_HIGH	OPCM_MID_1A	OPCM_MID_1A

Notes:

1. When using the sub-clock as the system clock, set the value of the SEL_SUB constant to B_USE (sub-clock used). When using the sub-clock as the RTC count source, set the value of the SEL_RTC constant to B_USE. When either SEL_SUB or SEL_RTC, or both are set to B_USE, the sub-clock operates.
2. The sub-clock oscillator is controlled by bits SOSCCR.SOSTP and RCR3.RTCEN. When the sub-clock is used as the system clock, it is controlled by the SOSCCR.SOSTP bit, and when the sub-clock is used as the RTC count source, it is controlled by the RCR3.RTCEN bit. Therefore the initial setting for the sub-clock differs depending on whether the sub-clock is used as the system clock or not. Also the sub-clock starts oscillating at power-on. Therefore processing to stop the sub-clock is performed even when the sub-clock is not used.

2. Operation Confirmation Conditions

The sample code accompanying this application note has been run and confirmed under the conditions below.

Table 2.1 Operation Confirmation Conditions

Item		Contents
MCU used		R5F5210BBDFP (RX210 Group)
Operating frequencies	When PLL is selected as the system clock (Nos. 1 and 2 in Table 1.3)	- Main clock: 20 MHz - Sub-clock: 32.768 kHz (stopped when the sub-clock is not used) - PLL: 100 MHz (main clock divided by 2 and multiplied by 10) - HOCO: Stopped - System clock (ICLK): 50 MHz (PLL divided by 2) - Peripheral module clock B (PCLKB): 25 MHz (PLL divided by 4) - Peripheral module clock D (PCLKD): 50 MHz (PLL divided by 2) - External bus clock (BCLK): 25 MHz (PLL divided by 4) - FlashIF clock (FCLK): 25 MHz (PLL divided by 4)
	When HOCO is selected as the system clock (Nos. 3 and 4 in Table 1.3)	- Main clock: Stopped - Sub-clock: 32.768 kHz (stopped when the sub-clock is not used) - PLL: Stopped - HOCO: 50 MHz - System clock (ICLK): 50 MHz (HOCO divided by 1) - Peripheral module clock B (PCLKB): 25 MHz (HOCO divided by 2) - Peripheral module clock D (PCLKD): 50 MHz (HOCO divided by 1) - External bus clock (BCLK): 25 MHz (HOCO divided by 2) - FlashIF clock (FCLK): 25 MHz (HOCO divided by 2)
	When the main clock is selected as the system clock (Nos. 5 and 6 in Table 1.3)	- Main clock: 20 MHz - Sub-clock: 32.768 kHz (stopped when the sub-clock is not used) - PLL: Stopped - HOCO: Stopped - System clock (ICLK): 20 MHz (main clock divided by 1) - Peripheral module clock B (PCLKB): 20 MHz (main clock divided by 1) - Peripheral module clock D (PCLKD): 20 MHz (main clock divided by 1) - External bus clock (BCLK): 20 MHz (main clock divided by 1) - FlashIF clock (FCLK): 20 MHz (main clock divided by 1)
Operating voltage		5.0 V
Integrated development environment		Renesas Electronics e^2 studio Version: 2021-01
C compiler		Renesas Electronics C/C++ Compiler Package for RX Family V 3.02
		Compiler option The integrated development environment default settings are used.
iodefine.h version		Version 1.4
Endian		Little endian
Operating mode		Single-chip mode
Processor mode		Supervisor mode
Sample code version		Version 2.21
Board used		Renesas Starter Kit for RX210 (product part no.: R0K505210C002BE)

3. Software

In the sample code, peripheral functions operating after a reset are stopped, nonexistent ports are configured, and then clock settings are configured.

3.1 Stop Processing for Active Peripheral Functions after a Reset

Peripheral functions that are operating after a reset are stopped in this processing.

The module-stop state is canceled after a reset only for modules listed in Table 3.1. To enter the module-stop state, set the module stop bit to 1 (transition to the module-stop state is made). Power consumption can be reduced by entering the module-stop state.

In the sample code, set the MSTP_STATE_“target module” constant to 0 (MODULE_STOP_DISABLE), so the target module does not enter the module-stop state. When the system requires a module to enter the module-stop state, set the constant in r_init_stop_module.h to 1 (MODULE_STOP_ENABLE).

Table 3.1 lists the Peripheral Modules whose Module-Stop States are Canceled after a Reset.

Table 3.1 Peripheral Modules whose Module-Stop States are Canceled after a Reset

Peripheral Module	Module Stop Bit	Value after a Reset	Value when not Using the Module
DMAC/DTC	MSTPCRA.MSTPA28 bit	0 (module-stop state is canceled)	1 (transition to the module-stop state is made)
RAM0	MSTPCRC.MSTPC0 bit		
RAM1	MSTPCRC.MSTPC1 bit		

3.2 Nonexistent Port Initialization

3.2.1 Overview

When using a product with less than 144 pins, set the corresponding bits of nonexistent ports in the PDR register to 1 (output). After the nonexistent port initialization function is called, when writing in byte units to the PDR registers or PODR registers which have nonexistent ports, set the corresponding bits for nonexistent ports as follows: set the I/O select bits in the PDR registers to 1 and set the output data store bits in the PODR registers to 0.

Table 3.2 and Table 3.3 list Nonexistent Ports.

Table 3.2 Nonexistent Ports (1/2)

Port Symbol	100-Pin Package	Number of Pins	80-Pin Package	Number of Pins
PORT0	P00 to P02	3	P00 to P02	3
PORT1	—	—	—	—
PORT2	—	—	P22 to P25	4
PORT3	—	—	P33	1
PORT4	—	—	—	—
PORT5	P56	1	P50 to P53, P56	4
PORT6	P60 to P67	8	P60 to P67	8
PORT7	P70 to P77	8	P70 to P77	8
PORT8	P80 to P83, P86, P87	6	P80 to P83, P86, P87	6
PORT9	P90 to P93	4	P90 to P93	4
PORTA	—	—	PA7	1
PORTB	—	—	—	—
PORTC	—	—	PC0, PC1	2
PORTD	—	—	PD3 to PD7	5
PORTE	—	—	PE6, PE7	2
PORTF	PF5	1	PF5	1
PORTH	—	—	—	—
PORTJ	PJ5	1	PJ3, PJ5	2
PORTK	PK2 to PK5	4	PK2 to PK5	4
PORTL	PL0, PL1	2	PL0, PL1	2

Table 3.3 Nonexistent Ports (2/2)

Port Symbol	69- and 64-Pin Packages	Number of Pins	48-Pin Package	Number of Pins
PORT0	P00 to P02, P07	4	P00 to P03, P05, P07	6
PORT1	P12, P13	2	P12, P13	2
PORT2	P20 to P25	6	P20 to P25	6
PORT3	P33, P34	2	P32 to P34	3
PORT4	P45, P47	2	P43 to P45, P47	4
PORT5	P50 to P53, P56	5	P50 to P56	7
PORT6	P60 to P67	8	P60 to P67	8
PORT7	P70 to P77	8	P70 to P77	8
PORT8	P80 to P83, P86, P87	6	P80 to P83, P86, P87	6
PORT9	P90 to P93	4	P90 to P93	4
PORTA	PA2, PA5, PA7	3	PA0, PA2, PA5, PA7	4
PORTB	PB2, PB4	2	PB2, PB4, PB6, PB7	4
PORTC	PC0, PC1	2	PC0 to PC3	4
PORTD	PD0 to PD7	8	PD0 to PD7	8
PORTE	PE6, PE7	2	PE0, PE5 to PE7	4
PORTF	PF5	1	PF5	1
PORTH	—	—	—	—
PORTJ	PJ1, PJ3, PJ5	3	PJ1, PJ3, PJ5	3
PORTK	PK2 to PK5	4	PK2 to PK5	4
PORTL	PL0, PL1	2	PL0, PL1	2

3.2.2 Selecting the Number of Pins

The number of pins in the sample code is set for the 100-pin package (PIN_SIZE=100). This application note covers 145-pin, 144-pin, 100-pin, 80-pin, 69-pin, 64-pin, and 48-pin packages. When using products with other than 100 pins, change PIN_SIZE in r_init_non_existent_port.h to the number of pins on the package.

3.3 Clock Settings

3.3.1 Clock Setting Procedure

Table 3.4 lists the Clock Setting with each processing and setting in the sample code. In the sample code, the main clock and PLL are operated, and HOCO and the sub-clock are stopped.

Table 3.4 Clock Setting Procedure

Step	Processing	Details		Setting in the Sample Code
1	VRCR register setting	Write 00h to the voltage regulator control register (VRCR)		The VRCR register is set.
2	Sub-clock setting [2]	Not used	Initialize the sub-clock control circuit.	Sub-clock is not used.
		Used	Initialize the sub-clock control circuit, set the driving ability, and set the SOSCWTCR register with a wait time until the sub-clock output is provided to the internal clock, and then enable the sub-clock oscillation. Then wait for the oscillation stabilization wait time [1] by software.	
3	Main clock setting [2]	Not used	No setting is required.	Main clock is used.
		Used	Set the main clock driving ability, set the MOSCWTCR register with a wait time until the main clock output is provided to the internal clock, and then enable the main clock oscillation. Then wait for the oscillation stabilization wait time [1] by software.	
4	PLL clock setting [2]	Not used	Turn off the PLL power supply. [3]	PLL clock is used.
		Used	Set the PLL input frequency division ratio and frequency multiplication factor, set the PLLWTCR register with a wait time until the PLL clock output is provided to the internal clock, and enable PLL clock oscillation. Then wait for the oscillation stabilization wait time [1] by software.	
5	HOCO clock setting [2]	Not used	Turn off the HOCO power supply.	HOCO clock is not used.
		Used	Set the HOCO frequency, set the HOCOWTCR2 register with a wait time until the HOCO clock output is provided to the internal clock, and then enable the HOCO clock oscillation. Then wait for the oscillation stabilization wait time [1] by software.	
6	Operating power control mode setting	Set the operating power control mode according to the operating frequency and operating voltage in the user system.		High-speed operating mode is set.
7	Clock division ratio setting	Change the clock division ratio.		- ICLK, PCLKD: Divided by 2 - PCLKB, BCLK, FCLK: Divided by 4 - BCLK: Stopped
8	System clock switching	Switch the system clock according to the user system.		Switched to the PLL clock.

Notes:

1. Refer to 3.3.2 Oscillation Stabilization Wait Time for Each Clock for details on the oscillation stabilization wait time.
2. When selecting each clock usage, change the appropriate constant in r_init_clock.h as required.
3. The PLL power control register (PLLPCR) is only included in chip version B. When chip version A or C is selected, processing to turn off the PLL power supply is not performed.

3.3.2 Oscillation Stabilization Wait Time for Each Clock

This section describes the wait control registers and oscillation stabilization wait times for the main clock, PLL clock, sub-clock, and HOCO clock.

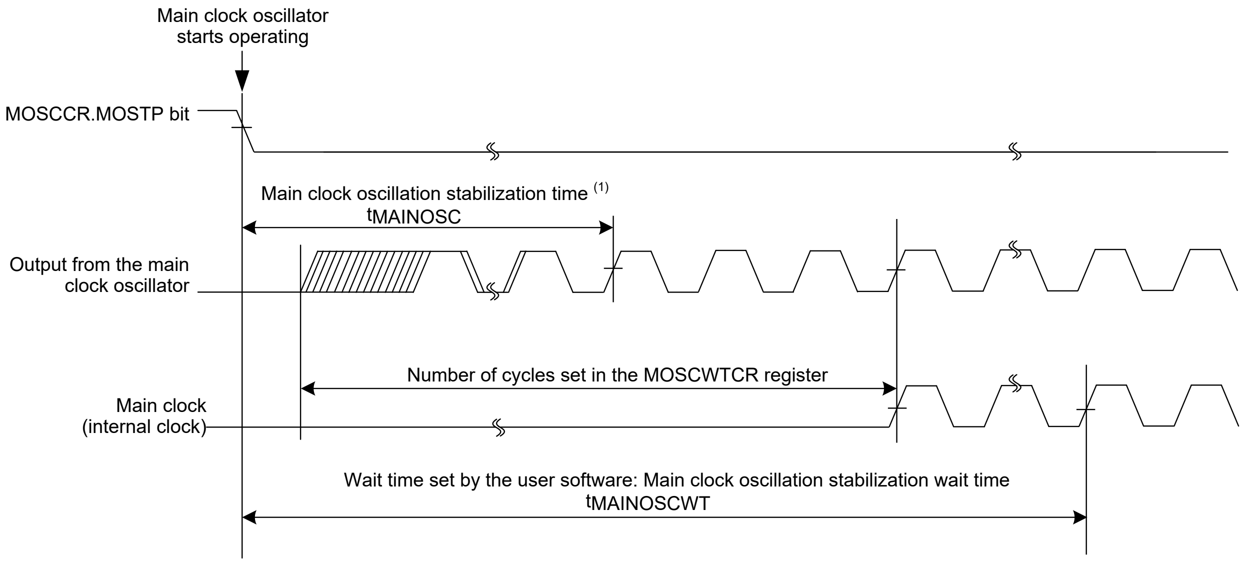
3.3.2.1 Main Clock Oscillation Stabilization Wait Time

Figure 3.1 shows the Main Clock Oscillation Stabilization Wait Time and Table 3.5 lists the Setting Value for the MOSCWTCR Register and Oscillation Stabilization Wait Time.

Set the main clock oscillator wait control register (MOSCWTCR) to a value greater than or equal to the main clock oscillation stabilization time (tMAINOSC) recommended by the crystal/ceramic resonator manufacturer. Set the main clock oscillation stabilization wait time (tMAINOSCWT) to a value greater than two times the number of cycles set in the MOSCWTCR register.

tMAINOSC used in the sample code is 4.2 ms, thus the setting value in the MOSCWTCR register is 0Dh (approximately 6.55 ms), and the setting value for tMAINOSCWT is approximately 13.1 ms.

Main Clock Oscillation Stabilization Wait Time


Main clock oscillator starts operating
MOSCCR.MOSTP bit
Main clock oscillation stabilization time (1)
tMAINOSC
Output from the main clock oscillator
Number of cycles set in the MOSCWTCR register
Main clock (internal clock)
Wait time set by the user software: Main clock oscillation stabilization wait time
tMAINOSCWT

Note:

1. The oscillation stabilization time of a crystal/ceramic resonator differs depending on the wiring pattern, conditions of oscillation parameters, and other settings in the user system. Ask the crystal/ceramic resonator manufacturer to evaluate the user system and provide an appropriate oscillation stabilization time.

Figure 3.1 Main Clock Oscillation Stabilization Wait Time

Table 3.5 Setting Value for the MOSCWTCR Register and Oscillation Stabilization Wait Time

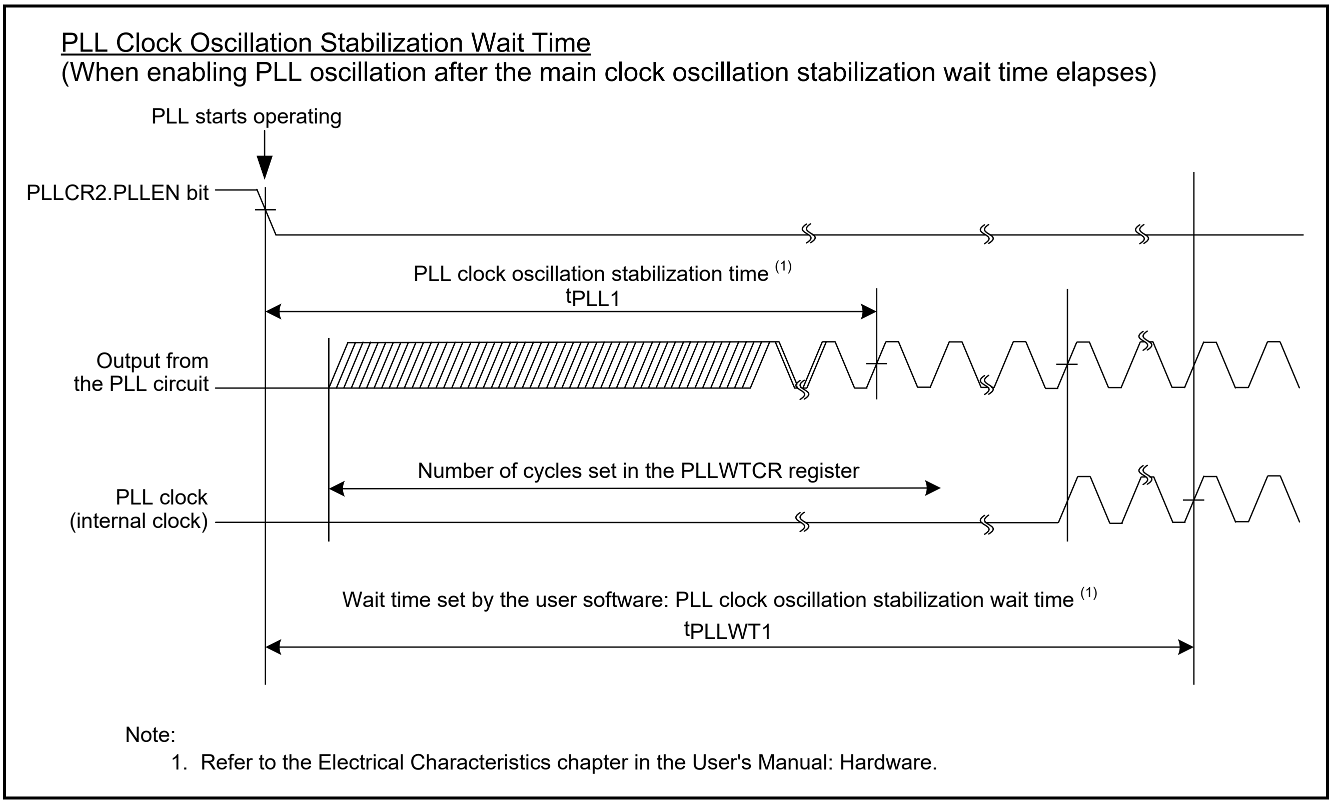
Setting Item	Condition of Setting Value	Setting Value in the Sample Code
MOSCWTCR.MSTS[4:0] bits	Value greater than or equal to tMAINOSC recommended by the crystal/ceramic resonator manufacturer	0Dh (approx. 6.5536 ms)
Oscillation stabilization wait time (tMAINOSCWT)	Value greater than or equal to two times the number of cycles set in the MOSCWTCR register.	Approx. 13.1072 ms

3.3.2.2 PLL Clock Oscillation Stabilization Wait Time (When enabling PLL oscillation after the main clock oscillation stabilization wait time elapses)

Figure 3.2 shows the PLL Clock Oscillation Stabilization Wait Time and Table 3.6 lists the Setting Value of the PLLWTCR Register and Oscillation Stabilization Wait Time.

Set the PLL wait control register (PLLWTCR) to a value greater than or equal to the PLL clock oscillation stabilization time (tPLL1 (max. 500 µs)). Set the PLL clock oscillation stabilization wait time (tPLLWT1) to a value greater than or equal to 1.5 ms.

tPLL1 is a maximum of 500 µs, thus the setting value in the PLLWTCR register is 09h (approximately 655.36 µs), and the setting value for tPLLWT1 is approximately 1.50 ms.


PLL Clock Oscillation Stabilization Wait Time
(When enabling PLL oscillation after the main clock oscillation stabilization wait time elapses)
PLL starts operating
PLLCR2.PLLEN bit
PLL clock oscillation stabilization time (1)
tPLL1
Output from the PLL circuit
Number of cycles set in the PLLWTCR register
PLL clock (internal clock)
Wait time set by the user software: PLL clock oscillation stabilization wait time (1)
tPLLWT1

Note:
1. Refer to the Electrical Characteristics chapter in the User's Manual: Hardware.

Figure 3.2 PLL Clock Oscillation Stabilization Wait Time

Table 3.6 Setting Value of the PLLWTCR Register and Oscillation Stabilization Wait Time

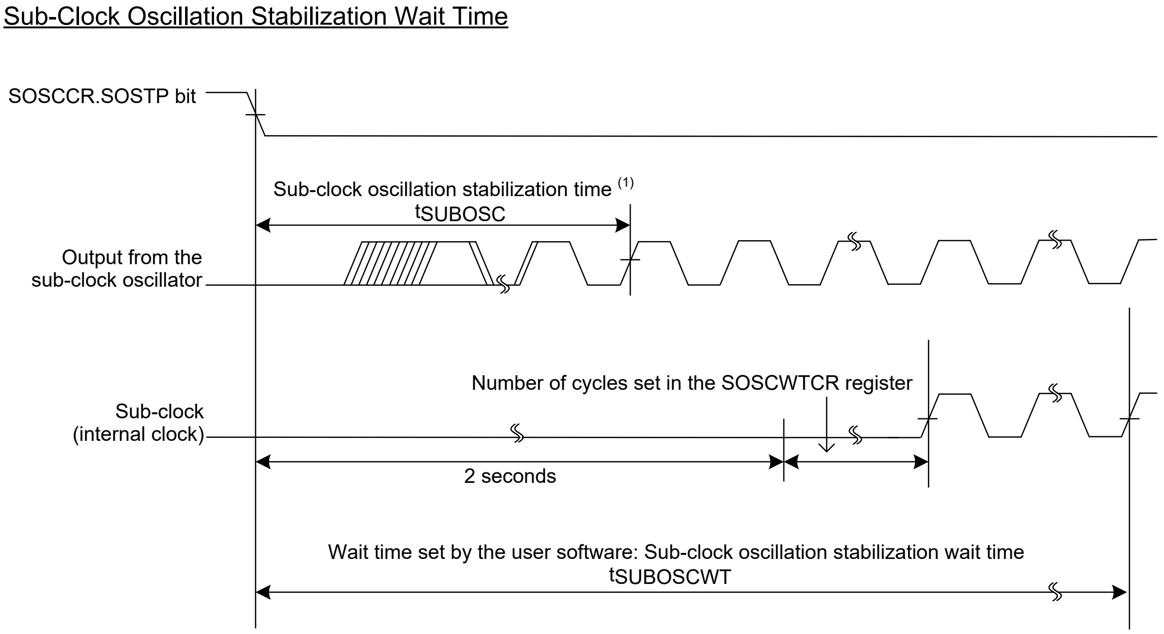
Setting Item	Condition of Setting Value	Setting Value in the Sample Code
PLLWTCR.PSTS[4:0] bits	Value greater than or equal to tPLL1 (max. 500 µs)	09h (approx. 655.36 µs)
Oscillation stabilization wait time (tPLLWT1)	Value greater than or equal to 1.5 ms	Approx. 1.50 ms

3.3.2.3 Sub-Clock Oscillation Stabilization Wait Time

Figure 3.3 shows the Sub-Clock Oscillation Stabilization Wait Time and Table 3.7 lists the Setting Value of the SOSCWTCR Register and Oscillation Stabilization Wait Time.

Set the sub-clock oscillator wait control register (SOSCWTCR) to a value greater than or equal to the sub-clock oscillation stabilization time (tSUBOSC) recommended by the crystal/ceramic resonator manufacturer. Set the sub-clock oscillation stabilization wait time (tSUBOSCWT) to a value greater than or equal to two times the value set in the SOSCWTCR register.

tSUBOSC used in the sample code is 1.3 seconds, thus the setting value in the SOSCWTCR register is 00h (2 sec. + approx. 61 µs), and the setting value for tSUBOSCWT is approximately 4 seconds.


Sub-Clock Oscillation Stabilization Wait Time
SOSCCR.SOSTP bit
Sub-clock oscillation stabilization time (1)
tSUBOSC
Output from the sub-clock oscillator
Number of cycles set in the SOSCWTCR register
Sub-clock (internal clock)
2 seconds
Wait time set by the user software: Sub-clock oscillation stabilization wait time
tSUBOSCWT

Notes:

1. Contact the crystal/ceramic resonator manufacturer to determine the oscillation stabilization time of a crystal/ceramic resonator for the user system. The oscillation stabilization time is not a condition for MCU operation but for a crystal/ceramic resonator to start oscillation.
2. Refer to the Electrical Characteristics chapter in the User's Manual: Hardware.

Figure 3.3 Sub-Clock Oscillation Stabilization Wait Time

Table 3.7 Setting Value of the SOSCWTCR Register and Oscillation Stabilization Wait Time

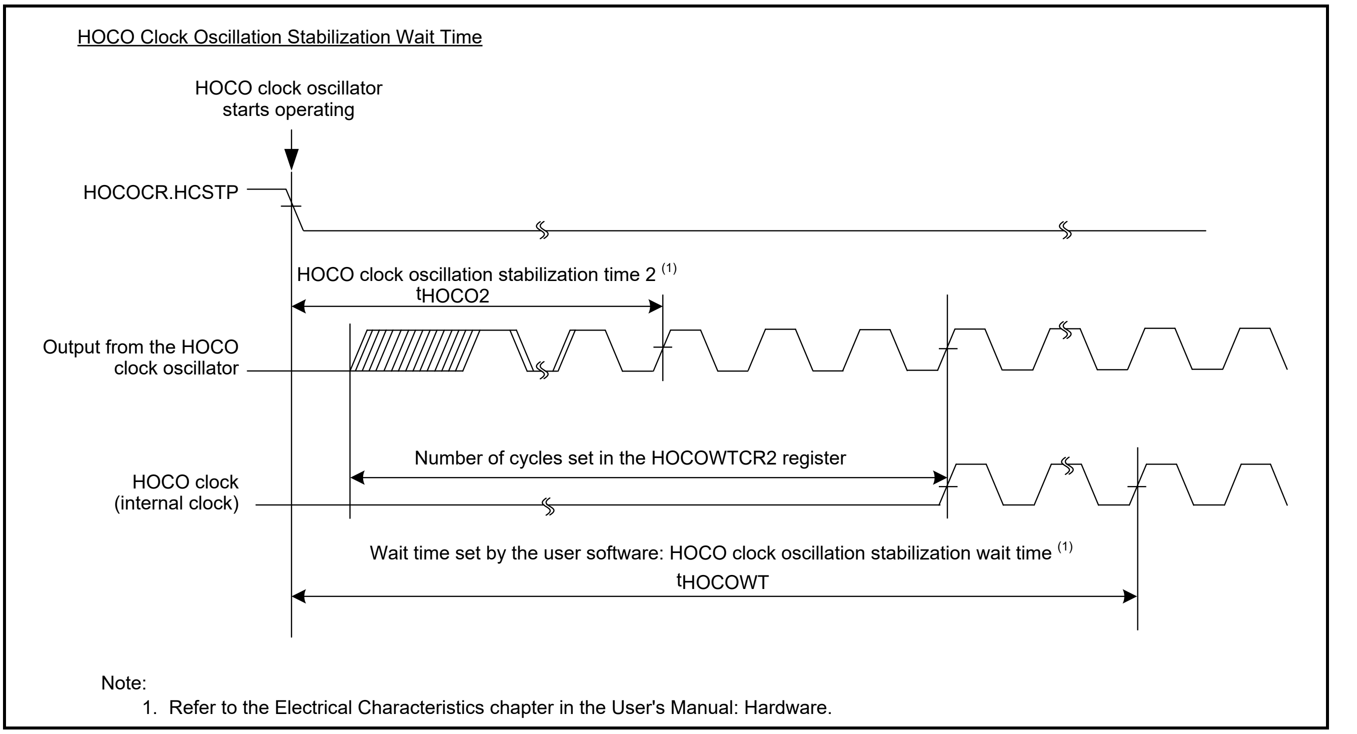
Setting Item	Condition of Setting Value	Setting Value in the Sample Code
SOSCWTCR.SSTS[4:0] bits	Value greater than or equal to tSUBOSC recommended by the crystal/ceramic resonator manufacturer.	00h (2 sec. + approx. 61 µs)
Oscillation stabilization wait time (tSUBOSCWT)	Value greater than or equal to two times the value set in the SOSCWTCR register	Approx. 4 sec.

3.3.2.4 HOCO Clock Oscillation Stabilization Wait Time

Figure 3.4 shows the HOCO Clock Oscillation Stabilization Wait Time and Table 3.8 lists the Setting Value of the HOCOWTCR2 Register and Oscillation Stabilization Wait Time.

Set 02h (7168 cycles) to the HOCO wait control register (HOCOWTCR2) when the HOCO clock oscillation frequency (fHOCO) is other than 50 MHz (32/36.864/40 MHz), and 03h (9216 cycles) when fHOCO is 50 MHz. Set the HOCO clock oscillation stabilization wait time (tHOCOWT) to a value greater than or equal to 350 µs.

fHOCO used in the sample code is 50 MHz, thus the setting value in the HOCOWTCR2 register is 03h (approximately 184.32 µs), and the setting value for tHOCOWT is approximately 350 µs.


HOCO Clock Oscillation Stabilization Wait Time
HOCO clock oscillator starts operating
HOCOCR.HCSTP
HOCO clock oscillation stabilization time 2 (1)
tHOCO2
Output from the HOCO clock oscillator
Number of cycles set in the HOCOWTCR2 register
HOCO clock (internal clock)
Wait time set by the user software: HOCO clock oscillation stabilization wait time (1)
tHOCOWT

Note:
1. Refer to the Electrical Characteristics chapter in the User's Manual: Hardware.

Figure 3.4 HOCO Clock Oscillation Stabilization Wait Time

Table 3.8 Setting Value of the HOCOWTCR2 Register and Oscillation Stabilization Wait Time

Setting Item	Condition of Setting Value	Setting Value in the Sample Code
HOCOWTCR2. HSTS2[3:0] bits	- When fHOCO is other than 50 MHz: 02h (7168 cycles) - When fHOCO is 50 MHz: 03h (9216 cycles)	03h (approx. 184.32 µs)
Oscillation stabilization wait time (tHOCOWT)	Value greater than or equal to 350 µs	Approx. 350 µs

3.4 File Composition

Table 3.9 lists the Files Used in the Sample Code. Files generated by the integrated development environment are not included in this table.

Table 3.9 Files Used in the Sample Code

File Name	Outline	Remarks
main.c	Main processing	
r_init_stop_module.c	Stop processing for active peripheral functions after a reset	
r_init_stop_module.h	Header file for r_init_stop_module.c	
r_init_non_existent_port.c	Nonexistent port initialization	
r_init_non_existent_port.h	Header file for r_init_non_existent_port.c	
r_init_clock.c	Clock initialization	
r_init_clock.h	Header file for r_init_clock.c	

3.5 Option-Setting Memory

Table 3.10 lists the Option-Setting Memory Configured in the Sample Code. When necessary, set a value suited to the user system.

Table 3.10 Option-Setting Memory Configured in the Sample Code

Symbol	Address	Setting Value	Contents
OFS0	FFFF FF8Fh to FFFF FF8Ch	FFFF FFFFh	The IWDT is stopped after a reset. The WDT is stopped after a reset.
OFS1	FFFF FF8Bh to FFFF FF88h	FFFF FFFFh	The voltage monitor 0 reset is disabled after a reset. HOCO oscillation is disabled after a reset.
MDES	FFFF FF83h to FFFF FF80h	FFFF FFFFh	Little endian

3.6 Constants

Table 3.11 and Table 3.12 list the constants used in the sample code, which can be changed by users. Table 3.13 lists the constants used in the sample code, which cannot be changed by users. Table 3.14 lists the Constants when a 145-Pin or 144-Pin Package is Used (PIN_SIZE=145 or PIN_SIZE=144), Table 3.15 lists the Constants when a 100-Pin Package is Used (PIN_SIZE=100), Table 3.16 lists the Constants when a 80-Pin Package is Used (PIN_SIZE=80), Table 3.17 lists the Constants when a 69-Pin or 64-Pin Package is Used (PIN_SIZE=69 or PIN_SIZE=64), and Table 3.18 lists the Constants when a 48-Pin Package is Used (PIN_SIZE=48).

Table 3.11 Constants Used in the Sample Code (1/2)
(Users can change the constants listed in this table.)

Constant Name	Setting Value	Contents
SEL_CHIP (1)	VERSION_A	Chip version selection: - VERSION_A: Chip version A - VERSION_B: Chip version B - VERSION_C: Chip version C
SEL_MAIN (1)	B_USE	Selection of the main clock operation: - B_USE: Used (main clock oscillating) - B_NOT_USE: Not used (main clock stopped)
MAIN_CLOCK_Hz (1)	20,000,000 L	Oscillation frequency of a crystal/ceramic resonator for the main clock (Hz)
REG_MOFCR (1)	30h	Setting for the driving ability of the main clock oscillator (set value in the MOFCR register)
REG_MOSCWTCR (1)	0Dh	Set value in the main clock wait control register
WAIT_TIME_FOR_MAIN_ OSCILLATION (1)	13,107,200 L	Main clock oscillation stabilization wait time (ns)
SEL_SUB (1, 2)	B_NOT_USE	Selection of the sub-clock usage for the system clock: - B_USE: Used - B_NOT_USE: Not used
SEL_RTC (1, 2)	B_NOT_USE	Selection of the sub-clock usage for the RTC count source: - B_USE: Used - B_NOT_USE: Not used
SUB_CLOCK_Hz (1)	32,768 L	Oscillation frequency of a crystal for the sub-clock (Hz)
REG_SOSCWTCR (1)	00h	Set value in the sub-clock wait control register
WAIT_TIME_FOR_SUB_ OSCILLATION (1)	4,000,000,000 L	Sub-clock oscillation stabilization wait time (ns)
REG_RCR3 (1)	CL_LOW	Selection of the sub-clock oscillator driving ability: - CL_STD: Standard clock loads - CL_LOW: Low clock loads
SEL_PLL (1)	B_USE	Selection of the PLL clock operation: - B_USE: Used (PLL clock oscillating) - B_NOT_USE: Not used (PLL clock stopped)
REG_PLLCR (1)	0901h	PLL input division ratio and frequency multiplication factor setting (set value in the PLLCR register)
REG_PLLWTCR (1)	09h	Set value in the PLL wait control register
WAIT_TIME_FOR_PLL_ OSCILLATION (1)	1,500,000 L	PLL clock oscillation stabilization wait time (ns)

Notes:

1. Change the setting value in r_init_clock.h according to the user system.
2. The sub-clock operation is set to be oscillating by setting B_USE (sub-clock used) to either of the SEL_SUB constant or SEL_RTC constant, or both.

Table 3.12 Constants Used in the Sample Code (2/2)
(Users can change the constants listed in this table.)

Constant Name	Setting Value	Contents
SEL_HOCO [1]	B_NOT_USE	Selection of the HOCO clock operation: - B_USE: Used (HOCO clock oscillating) - B_NOT_USE: Not used (HOCO clock stopped)
REG_HOCOCR2 [1]	FREQ_50MHz	Selection of the HOCO clock frequency - FREQ_32MHz: 32 MHz - FREQ_36MHz: 36.864 MHz - FREQ_40MHz: 40 MHz - FREQ_50MHz: 50 MHz
WAIT_TIME_FOR_HOCO_OSCILLATION [1]	350,000 L	HOCO clock oscillation stabilization wait time (ns)
SEL_SYSCLK [1]	CLK_PLL	Clock source selection for the system clock - CLK_PLL: PLL - CLK_HOCO: HOCO - CLK_MAIN: Main clock [4] - CLK_SUB: Sub-clock
REG_OPCCR [1]	OPCM_HIGH	Selection of the operating power control mode [8] - OPCM_HIGH: High-speed operating mode - OPCM_MID_1A: Middle-speed operating mode 1A - OPCM_MID_1B: Middle-speed operating mode 1B - OPCM_MID_2A: Middle-speed operating mode 2A [5] - OPCM_MID_2B: Middle-speed operating mode 2B [5] - OPCM_LOW_1: Low-speed operating mode 1 [6] - OPCM_LOW_2: Low-speed operating mode 2 [7]
MSTP_STATE_DMACDTC [2]	MODULE_STOP_DISABLE	Selection of the module-stop state for DMAC and DTC - MODULE_STOP_DISABLE: Module-stop state canceled - MODULE_STOP_ENABLE: Entering the module-stop state
MSTP_STATE_RAM0 [2]	MODULE_STOP_DISABLE	Selection of the module-stop state for RAM0 - MODULE_STOP_DISABLE: Operating - MODULE_STOP_ENABLE: Stopped
MSTP_STATE_RAM1 [2]	MODULE_STOP_DISABLE	Selection of the module-stop state for RAM1 - MODULE_STOP_DISABLE: Operating - MODULE_STOP_ENABLE: Stopped
PIN_SIZE [3]	100	Number of pins on the product used

Notes:

1. Change the setting value in r_init_clock.h according to the user system.
2. Change the setting value in r_init_stop_module.h according to the user system.
3. Change the setting value in r_init_non_existent_port.h according to the user system.
4. The main clock oscillator is not available as the clock source in the chip version A.
5. Middle-speed operating mode 2A and middle-speed operating mode 2B are only available in chip version B.
6. When PLL is set to be oscillating, low-speed operating mode 1 is not available.
7. When PLL or HOCO is set to be oscillating, low-speed operating mode 2 is not available.
8. The ranges of the operating frequency and operating voltage differ depending on operating modes. Refer to the User’s Manual: Hardware for details.

Table 3.13 Constants Used in the Sample Code
(Users cannot change the constants listed in this table.)

Constant Name	Setting Value	Contents
VERSION_A	1	Chip version A
VERSION_B	2	Chip version B
VERSION_C	3	Chip version C
B_NOT_USE	0	Not used
B_USE	1	Used
CL_LOW	02h	Sub-clock: Drive ability for low clock loads
CL_STD	0Ch	Sub-clock: Drive ability for standard clock loads
FREQ_32MHz	00h	HOCO frequency: 32 MHz
FREQ_36MHz	01h	HOCO frequency: 36.684 MHz
FREQ_40MHz	02h	HOCO frequency: 40 MHz
FREQ_50MHz	03h	HOCO frequency: 50 MHz
CLK_PLL	0400h	Clock source: PLL
CLK_HOCO	0100h	Clock source: HOCO
CLK_SUB	0300h	Clock source: Sub-clock
CLK_MAIN	0200h	Clock source: Main clock
REG_HOCOWTCR2 (1)	- 03h (when 50 MHz is selected) - 02h (when other than 50 MHz is selected)	Set value in the HOCO wait control register
REG_SCKCR (2)	- 2182 1211h (when PLL is selected) - 1081 1110h (when HOCO is selected) - 0080 1010h (other than above)	Setting for the internal clock division ratio and BCLK pin output control (set value in the SCKCR register)
OPCM_HIGH	00h	High-speed operating mode
OPCM_MID_1A	02h	Middle-speed operating mode 1A
OPCM_MID_1B	03h	Middle-speed operating mode 1B
OPCM_MID_2A	04h	Middle-speed operating mode 2A
OPCM_MID_2B	05h	Middle-speed operating mode 2B
OPCM_LOW_1	06h	Low-speed operating mode 1
OPCM_LOW_2	07h	Low-speed operating mode 2
MAIN_CLOCK_CYCLE	(1,000,000,000L / MAIN_CLOCK_Hz)	Main clock cycles (ns)
SUB_CLOCK_CYCLE	(1,000,000,000L / SUB_CLOCK_Hz)	Sub-clock cycles (ns)
FOR_CMT0_TIME	232727	Count cycles (ns) for the CMT0 timer to wait for the oscillation stabilization wait time: (1/LOCO) × 32, where LOCO = 137.5 kHz (max.), and 32 = PCLKB divided by 32
MODULE_STOP_ENABLE	1	Transition to the module stop-state is made
MODULE_STOP_DISABLE	0	Module stop-state is canceled

Notes:

1. The setting value varies depending on the HOCO frequency selected.
2. The setting value varies depending on the clock source of the system clock selected.

Table 3.14 Constants when a 145-Pin or 144-Pin Package is Used (PIN_SIZE=145 or PIN_SIZE=144)

Constant Name	Setting Value	Contents
DEF_P0PDR	0x00	Setting value for the port P0 direction register
DEF_P1PDR	0x00	Setting value for the port P1 direction register
DEF_P2PDR	0x00	Setting value for the port P2 direction register
DEF_P3PDR	0x00	Setting value for the port P3 direction register
DEF_P4PDR	0x00	Setting value for the port P4 direction register
DEF_P5PDR	0x00	Setting value for the port P5 direction register
DEF_P6PDR	0x00	Setting value for the port P6 direction register
DEF_P7PDR	0x00	Setting value for the port P7 direction register
DEF_P8PDR	0x00	Setting value for the port P8 direction register
DEF_P9PDR	0x00	Setting value for the port P9 direction register
DEF_PAPDR	0x00	Setting value for the port PA direction register
DEF_PBPDR	0x00	Setting value for the port PB direction register
DEF_PCPDR	0x00	Setting value for the port PC direction register
DEF_PDPDR	0x00	Setting value for the port PD direction register
DEF_PEPDR	0x00	Setting value for the port PE direction register
DEF_PFPDR	0x00	Setting value for the port PF direction register
DEF_PHPDR	0x00	Setting value for the port PH direction register
DEF_PJPDR	0x00	Setting value for the port PJ direction register
DEF_PKPDR	0x00	Setting value for the port PK direction register
DEF_PLPDR	0x00	Setting value for the port PL direction register

Table 3.15 Constants when a 100-Pin Package is Used (PIN_SIZE=100)

Constant Name	Setting Value	Contents
DEF_P0PDR	0x07	Setting value for the port P0 direction register
DEF_P1PDR	0x00	Setting value for the port P1 direction register
DEF_P2PDR	0x00	Setting value for the port P2 direction register
DEF_P3PDR	0x00	Setting value for the port P3 direction register
DEF_P4PDR	0x00	Setting value for the port P4 direction register
DEF_P5PDR	0x40	Setting value for the port P5 direction register
DEF_P6PDR	0xFF	Setting value for the port P6 direction register
DEF_P7PDR	0xFF	Setting value for the port P7 direction register
DEF_P8PDR	0xCF	Setting value for the port P8 direction register
DEF_P9PDR	0x0F	Setting value for the port P9 direction register
DEF_PAPDR	0x00	Setting value for the port PA direction register
DEF_PBPDR	0x00	Setting value for the port PB direction register
DEF_PCPDR	0x00	Setting value for the port PC direction register
DEF_PDPDR	0x00	Setting value for the port PD direction register
DEF_PEPDR	0x00	Setting value for the port PE direction register
DEF_PFPDR	0x20	Setting value for the port PF direction register
DEF_PHPDR	0x00	Setting value for the port PH direction register
DEF_PJPDR	0x20	Setting value for the port PJ direction register
DEF_PKPDR	0x3C	Setting value for the port PK direction register
DEF_PLPDR	0x03	Setting value for the port PL direction register

Table 3.16 Constants when a 80-Pin Package is Used (PIN_SIZE=80)

Constant Name	Setting Value	Contents
DEF_P0PDR	0x00	Setting value for the port P0 direction register
DEF_P1PDR	0x00	Setting value for the port P1 direction register
DEF_P2PDR	0x3C	Setting value for the port P2 direction register
DEF_P3PDR	0x08	Setting value for the port P3 direction register
DEF_P4PDR	0x00	Setting value for the port P4 direction register
DEF_P5PDR	0x0F	Setting value for the port P5 direction register
DEF_P6PDR	0x00	Setting value for the port P6 direction register
DEF_P7PDR	0x00	Setting value for the port P7 direction register
DEF_P8PDR	0x00	Setting value for the port P8 direction register
DEF_P9PDR	0x00	Setting value for the port P9 direction register
DEF_PAPDR	0x80	Setting value for the port PA direction register
DEF_PBPDR	0x00	Setting value for the port PB direction register
DEF_PCPDR	0x03	Setting value for the port PC direction register
DEF_PDPDR	0xF8	Setting value for the port PD direction register
DEF_PEPDR	0xC0	Setting value for the port PE direction register
DEF_PFPDR	0x00	Setting value for the port PF direction register
DEF_PHPDR	0x00	Setting value for the port PH direction register
DEF_PJPDR	0x08	Setting value for the port PJ direction register
DEF_PKPDR	0x00	Setting value for the port PK direction register
DEF_PLPDR	0x00	Setting value for the port PL direction register

Table 3.17 Constants when a 69-Pin or 64-Pin Package is Used (PIN_SIZE=69 or PIN_SIZE=64)

Constant Name	Setting Value	Contents
DEF_P0PDR	0x80	Setting value for the port P0 direction register
DEF_P1PDR	0x0C	Setting value for the port P1 direction register
DEF_P2PDR	0x3F	Setting value for the port P2 direction register
DEF_P3PDR	0x18	Setting value for the port P3 direction register
DEF_P4PDR	0xA0	Setting value for the port P4 direction register
DEF_P5PDR	0x0F	Setting value for the port P5 direction register
DEF_P6PDR	0x00	Setting value for the port P6 direction register
DEF_P7PDR	0x00	Setting value for the port P7 direction register
DEF_P8PDR	0x00	Setting value for the port P8 direction register
DEF_P9PDR	0x00	Setting value for the port P9 direction register
DEF_PAPDR	0xA4	Setting value for the port PA direction register
DEF_PBPDR	0x14	Setting value for the port PB direction register
DEF_PCPDR	0x03	Setting value for the port PC direction register
DEF_PDPDR	0xFF	Setting value for the port PD direction register
DEF_PEPDR	0xC0	Setting value for the port PE direction register
DEF_PFPDR	0x00	Setting value for the port PF direction register
DEF_PHPDR	0x00	Setting value for the port PH direction register
DEF_PJPDR	0x0A	Setting value for the port PJ direction register
DEF_PKPDR	0x00	Setting value for the port PK direction register
DEF_PLPDR	0x00	Setting value for the port PL direction register

Table 3.18 Constants when a 48-Pin Package is Used (PIN_SIZE=48)

Constant Name	Setting Value	Contents
DEF_P0PDR	0xA8	Setting value for the port P0 direction register
DEF_P1PDR	0x0C	Setting value for the port P1 direction register
DEF_P2PDR	0x3F	Setting value for the port P2 direction register
DEF_P3PDR	0x1C	Setting value for the port P3 direction register
DEF_P4PDR	0xB8	Setting value for the port P4 direction register
DEF_P5PDR	0x3F	Setting value for the port P5 direction register
DEF_P6PDR	0x00	Setting value for the port P6 direction register
DEF_P7PDR	0x00	Setting value for the port P7 direction register
DEF_P8PDR	0x00	Setting value for the port P8 direction register
DEF_P9PDR	0x00	Setting value for the port P9 direction register
DEF_PAPDR	0xA5	Setting value for the port PA direction register
DEF_PBPDR	0xD4	Setting value for the port PB direction register
DEF_PCPDR	0x0F	Setting value for the port PC direction register
DEF_PDPDR	0xFF	Setting value for the port PD direction register
DEF_PEPDR	0xE1	Setting value for the port PE direction register
DEF_PFPDR	0x00	Setting value for the port PF direction register
DEF_PHPDR	0x00	Setting value for the port PH direction register
DEF_PJPDR	0x0A	Setting value for the port PJ direction register
DEF_PKPDR	0x00	Setting value for the port PK direction register
DEF_PLPDR	0x00	Setting value for the port PL direction register

3.7 Functions

Table 3.19 lists the Functions Used in the Sample Code.

Table 3.19 Functions Used in the Sample Code

Function Name	Outline
main	Main processing
R_INIT_StopModule	Stop processing for active peripheral functions after a reset
R_INIT_NonExistentPort	Nonexistent port initialization
R_INIT_Clock	Clock initialization
CGC_oscillation_main	Main clock oscillation setting
CGC_oscillation_PLL	PLL clock oscillation setting
CGC_oscillation_HOCO	HOCO clock oscillation setting
CGC_oscillation_sub	Sub-clock oscillation setting
CGC_disable_subclk	Sub-clock stop setting
oscillation_subclk	Enabling sub-clock oscillation
no_use_subclk_as_sysclk	Processing when the sub-clock is not used as the system clock
resetting_wtcr_subclk	Resetting the sub-clock wait control register
enable_RTC	Initialization when using the RTC
cmt0_wait	Wait processing

3.8 Function Specifications

The following tables list the sample code function specifications.

main	
Outline	Main processing
Header	None
Declaration	void main(void)
Description	Call the following functions: Stop processing for active peripheral functions after a reset, nonexistent port initialization, and clock initialization.
Arguments	None
Return Value	None

R_INIT_StopModule	
Outline	Stop processing for active peripheral functions after a reset
Header	r_init_stop_module.h
Declaration	void R_INIT_StopModule(void)
Description	Configure the setting to enter the module-stop state.
Arguments	None
Return Value	None
Remarks	Transition to the module-stop state is not performed in the sample code.

R_INIT_NonExistentPort	
Outline	Nonexistent port initialization
Header	r_init_non_existent_port.h
Declaration	void R_INIT_NonExistentPort(void)
Description	Initialize port direction registers for ports that do not exist in products with less than 144 pins.
Arguments	None
Return Value	None
Remarks	The number of pins in the sample code is set for the 100-pin package (PIN_SIZE=100). After this function is called, when writing in byte units to the PDR registers or PODR registers which have nonexistent ports, set the corresponding bits for nonexistent ports as follows: set the I/O select bits in the PDR registers to 1 and set the output data store bits in the PODR registers to 0.

R_INIT_Clock	
Outline	Clock initialization
Header	r_init_clock.h
Declaration	void R_INIT_Clock(void)
Description	Initialize the clock.
Arguments	None
Return Value	None
Remarks	The sample code selects processing which uses PLL as the system clock without using the sub-clock.

CGC_oscillation_main	
Outline	Main clock oscillation setting
Header	r_init_clock.h
Declaration	void CGC_oscillation_main(void)
Description	Set the main clock driving ability, set the MOSCWTCR register, and enable main clock oscillation. Then wait for the main clock oscillation stabilization wait time by software.
Arguments	None
Return Value	None

CGC_oscillation_PLL	
Outline	PLL clock oscillation setting
Header	r_init_clock.h
Declaration	void CGC_oscillation_PLL(void)
Description	Set the PLL input frequency division ratio and frequency multiplication factor, set the PLLWTCR register, and enable PLL clock oscillation. Then wait for the PLL clock oscillation stabilization wait time by software.
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	Set the HOCO frequency, set the HOCOWTCR2 register, and enable HOCO oscillation. Then wait for the HOCO clock oscillation stabilization wait time by software.
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	Configure the setting when the sub-clock is used as either the system clock or the RTC count source, or both.
Arguments	None
Return Value	None

CGC_disable_subclk	
Outline	Sub-clock stop setting
Header	r_init_clock.h
Declaration	void CGC_disable_subclk(void)
Description	Configure the setting when the sub-clock is not used as the system clock or the RTC count source.
Arguments	None
Return Value	None

oscillation_subclk	
Outline	Enabling the sub-clock oscillation
Header	None
Declaration	static void oscillation_subclk(void)
Description	Configure the sub-clock oscillation.
Arguments	None
Return Value	None

no_use_subclk_as_sysclk	
Outline	Processing when the sub-clock is not used as the system clock
Header	None
Declaration	static void no_use_subclk_as_sysclk(void)
Description	Stop the sub-clock as the system clock when the sub-clock is used only as the RTC count source.
Arguments	None
Return Value	None

resetting_wtcr_ subclk	
Outline	Resetting the sub-clock wait control register
Header	None
Declaration	static void resetting_wtcr_subclk(void)
Description	Reset the wait control register when exiting from software standby mode. Set the minimum value to the wait control register.
Arguments	None
Return Value	None

enable_RTC	
Outline	Initialization when using the RTC
Header	None
Declaration	static void enable_RTC (void)
Description	Initialize the settings when using the RTC (setting for clock provision and RTC software reset).
Arguments	None
Return Value	None

cmt0_wait		
Outline	Wait processing	
Header	None	
Declaration	static void cmt0_wait(uint32_t cnt)	
Description	This function is used when waiting for the oscillation stabilization wait time.	
Arguments	uint32_t cnt:	Oscillation stabilization wait time cnt = oscillation stabilization wait time (ns) [1] ÷ FOR_CMT0_TIME [2]
Return Value	None	
Remarks	1. The oscillation stabilization wait time varies depending on the crystal/ceramic resonator. Set the value referring to 3.3.2 Oscillation Stabilization Wait Time for Each Clock. 2. The value of FOR_CMT0_TIME is calculated when LOCO is 137.5 kHz (max.). The actual wait time may differ depending on the LOCO frequency.	

3.9 Flowcharts

3.9.1 Main Processing

Figure 3.5 shows the Main Processing.

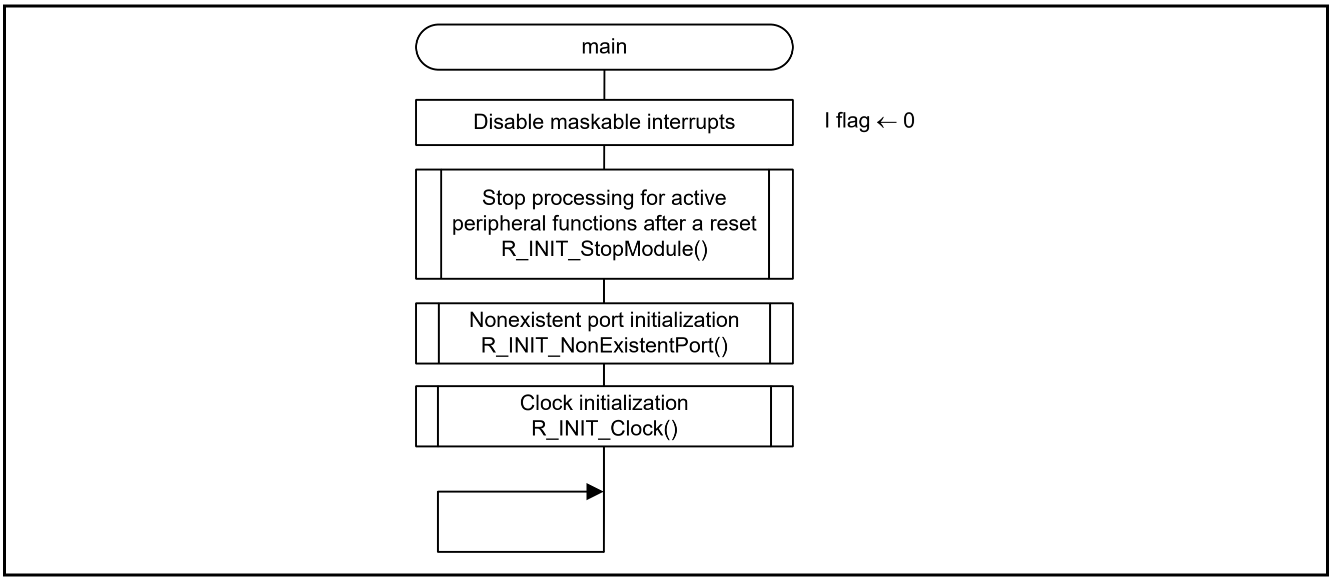
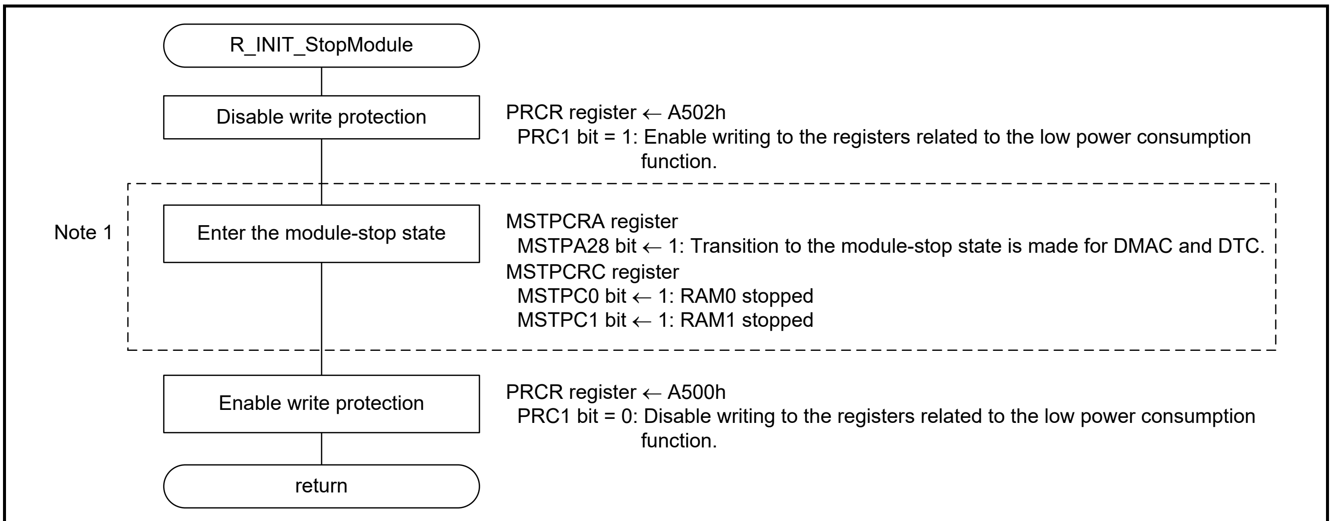

main
Disable maskable interrupts
I flag ← 0
Stop processing for active peripheral functions after a reset
R_INIT_StopModule()
Nonexistent port initialization
R_INIT_NonExistentPort()
Clock initialization
R_INIT_Clock()

Figure 3.5 Main Processing

3.9.2 Stop Processing for Active Peripheral Functions after a Reset

Figure 3.6 shows the Stop Processing for Active Peripheral Functions after a Reset.


R_INIT_StopModule
Disable write protection
PRCR register ← A502h
PRC1 bit = 1: Enable writing to the registers related to the low power consumption function.
Note 1
Enter the module-stop state
MSTPCRA register
MSTPA28 bit ← 1: Transition to the module-stop state is made for DMAC and DTC.
MSTPCRC register
MSTPC0 bit ← 1: RAM0 stopped
MSTPC1 bit ← 1: RAM1 stopped
Enable write protection
PRCR register ← A500h
PRC1 bit = 0: Disable writing to the registers related to the low power consumption function.
return

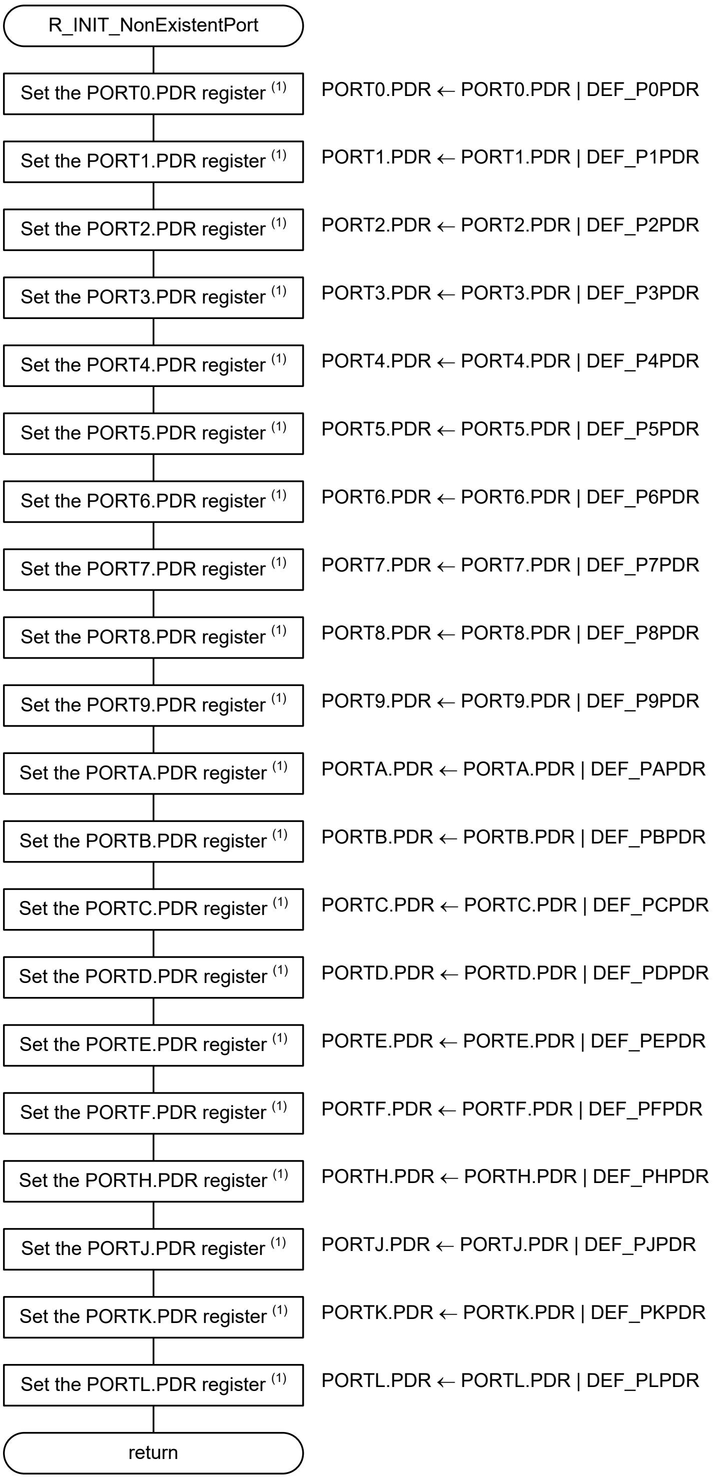
Note:

1. The module-stop state is canceled in the sample code. When entering the module-stop state for any peripheral functions, set the #define MSTP_STATE_"target module" constant to 1.

Figure 3.6 Stop Processing for Active Peripheral Functions after a Reset

3.9.3 Nonexistent Port Initialization

Figure 3.7 shows the Nonexistent Port Initialization.


R_INIT_NonExistentPort
Set the PORT0.PDR register (1)
PORT0.PDR ← PORT0.PDR | DEF_P0PDR
Set the PORT1.PDR register (1)
PORT1.PDR ← PORT1.PDR | DEF_P1PDR
Set the PORT2.PDR register (1)
PORT2.PDR ← PORT2.PDR | DEF_P2PDR
Set the PORT3.PDR register (1)
PORT3.PDR ← PORT3.PDR | DEF_P3PDR
Set the PORT4.PDR register (1)
PORT4.PDR ← PORT4.PDR | DEF_P4PDR
Set the PORT5.PDR register (1)
PORT5.PDR ← PORT5.PDR | DEF_P5PDR
Set the PORT6.PDR register (1)
PORT6.PDR ← PORT6.PDR | DEF_P6PDR
Set the PORT7.PDR register (1)
PORT7.PDR ← PORT7.PDR | DEF_P7PDR
Set the PORT8.PDR register (1)
PORT8.PDR ← PORT8.PDR | DEF_P8PDR
Set the PORT9.PDR register (1)
PORT9.PDR ← PORT9.PDR | DEF_P9PDR
Set the PORTA.PDR register (1)
PORTA.PDR ← PORTA.PDR | DEF_PAPDR
Set the PORTB.PDR register (1)
PORTB.PDR ← PORTB.PDR | DEF_PBPDR
Set the PORTC.PDR register (1)
PORTC.PDR ← PORTC.PDR | DEF_PCPDR
Set the PORTD.PDR register (1)
PORTD.PDR ← PORTD.PDR | DEF_PDPDR
Set the PORTE.PDR register (1)
PORTE.PDR ← PORTE.PDR | DEF_PEPDR
Set the PORTF.PDR register (1)
PORTF.PDR ← PORTF.PDR | DEF_PFPDR
Set the PORTH.PDR register (1)
PORTH.PDR ← PORTH.PDR | DEF_PHPDR
Set the PORTJ.PDR register (1)
PORTJ.PDR ← PORTJ.PDR | DEF_PJPDR
Set the PORTK.PDR register (1)
PORTK.PDR ← PORTK.PDR | DEF_PKPDR
Set the PORTL.PDR register (1)
PORTL.PDR ← PORTL.PDR | DEF_PLPDR
return

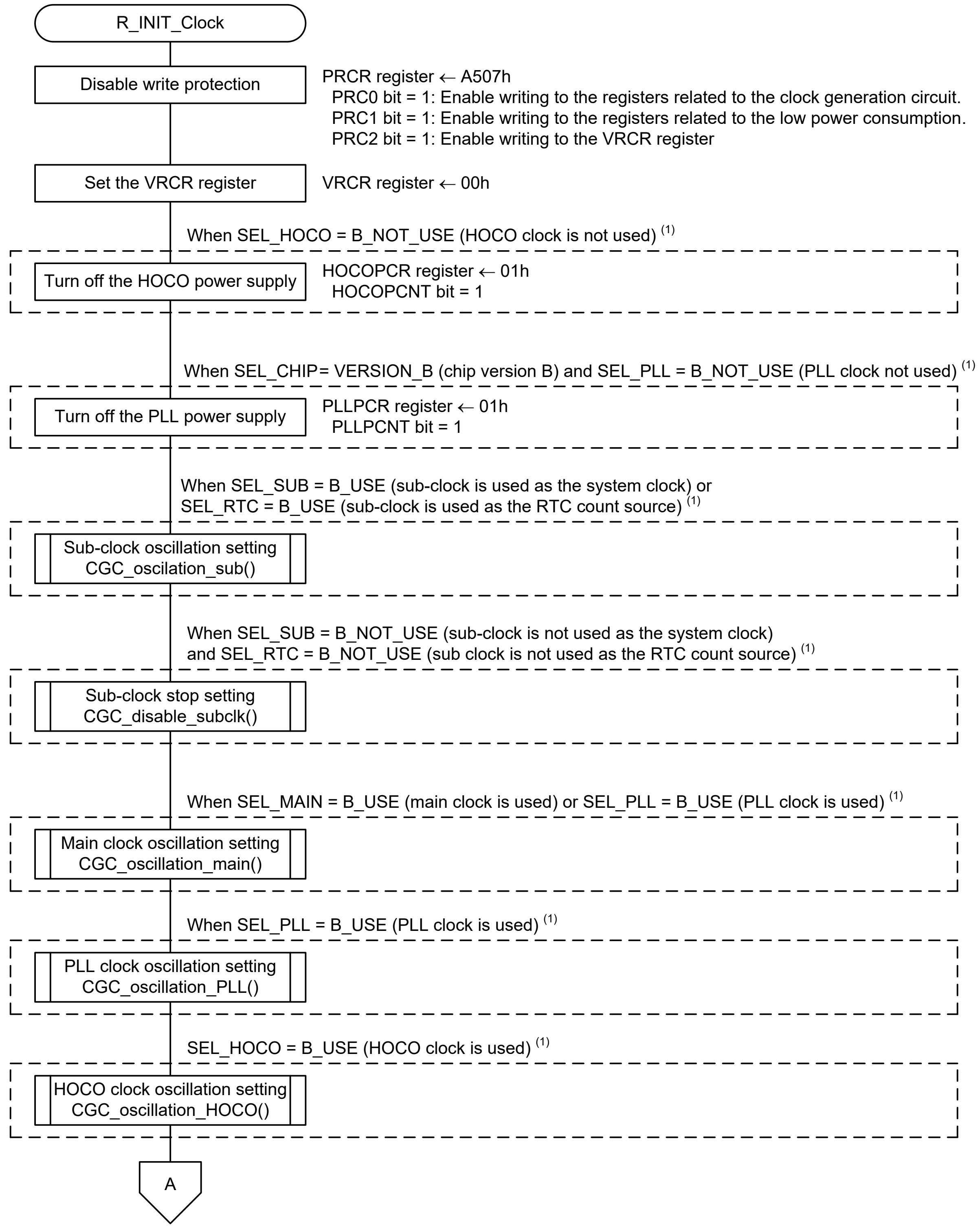
Note:

1. Port initialization processing is not executed for registers which have all pins (processing is skipped when compiling).

Figure 3.7 Nonexistent Port Initialization

3.9.4 Clock Initialization

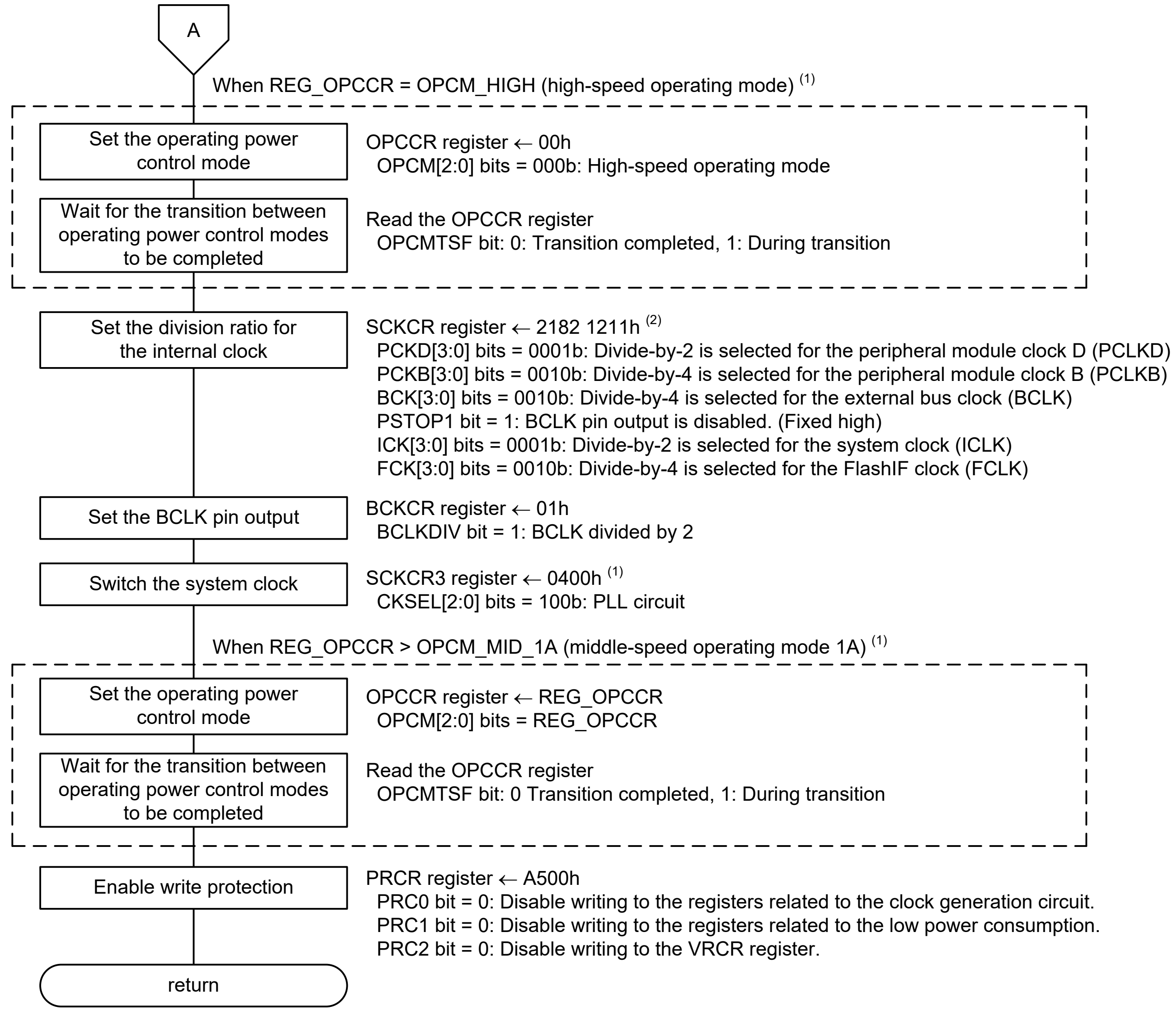
Figure 3.8 and Figure 3.9 show the clock initialization.


R_INIT_Clock
Disable write protection
PRCR register ← A507h
PRC0 bit = 1: Enable writing to the registers related to the clock generation circuit.
PRC1 bit = 1: Enable writing to the registers related to the low power consumption.
PRC2 bit = 1: Enable writing to the VRCR register
Set the VRCR register
VRCR register ← 00h
When SEL_HOCO = B_NOT_USE (HOCO clock is not used) (1)
Turn off the HOCO power supply
HOCOPCR register ← 01h
HOCOPCNT bit = 1
When SEL_CHIP= VERSION_B (chip version B) and SEL_PLL = B_NOT_USE (PLL clock not used) (1)
Turn off the PLL power supply
PLLPCR register ← 01h
PLLPCNT bit = 1
When SEL_SUB = B_USE (sub-clock is used as the system clock) or
SEL_RTC = B_USE (sub-clock is used as the RTC count source) (1)
Sub-clock oscillation setting
CGC_oscilation_sub()
When SEL_SUB = B_NOT_USE (sub-clock is not used as the system clock)
and SEL_RTC = B_NOT_USE (sub clock is not used as the RTC count source) (1)
Sub-clock stop setting
CGC_disable_subclk()
When SEL_MAIN = B_USE (main clock is used) or SEL_PLL = B_USE (PLL clock is used) (1)
Main clock oscillation setting
CGC_oscillation_main()
When SEL_PLL = B_USE (PLL clock is used) (1)
PLL clock oscillation setting
CGC_oscillation_PLL()
SEL_HOCO = B_USE (HOCO clock is used) (1)
HOCO clock oscillation setting
CGC_oscillation_HOCO()
A

Note:

1. Change the constant value according to the user system.

Figure 3.8 Clock Initialization (1/2)


A
When REG_OPCCR = OPCM_HIGH (high-speed operating mode) (1)
Set the operating power control mode
OPCCR register ← 00h
OPCM[2:0] bits = 000b: High-speed operating mode
Wait for the transition between operating power control modes to be completed
Read the OPCCR register
OPCMTSF bit: 0: Transition completed, 1: During transition
Set the division ratio for the internal clock
SCKCR register ← 2182 1211h (2)
PCKD[3:0] bits = 0001b: Divide-by-2 is selected for the peripheral module clock D (PCLKD)
PCKB[3:0] bits = 0010b: Divide-by-4 is selected for the peripheral module clock B (PCLKB)
BCK[3:0] bits = 0010b: Divide-by-4 is selected for the external bus clock (BCLK)
PSTOP1 bit = 1: BCLK pin output is disabled. (Fixed high)
ICK[3:0] bits = 0001b: Divide-by-2 is selected for the system clock (ICLK)
FCK[3:0] bits = 0010b: Divide-by-4 is selected for the FlashIF clock (FCLK)
Set the BCLK pin output
BCKCR register ← 01h
BCLKDIV bit = 1: BCLK divided by 2
Switch the system clock
SCKCR3 register ← 0400h (1)
CKSEL[2:0] bits = 100b: PLL circuit
When REG_OPCCR > OPCM_MID_1A (middle-speed operating mode 1A) (1)
Set the operating power control mode
OPCCR register ← REG_OPCCR
OPCM[2:0] bits = REG_OPCCR
Wait for the transition between operating power control modes to be completed
Read the OPCCR register
OPCMTSF bit: 0 Transition completed, 1: During transition
Enable write protection
PRCR register ← A500h
PRC0 bit = 0: Disable writing to the registers related to the clock generation circuit.
PRC1 bit = 0: Disable writing to the registers related to the low power consumption.
PRC2 bit = 0: Disable writing to the VRCR register.
return

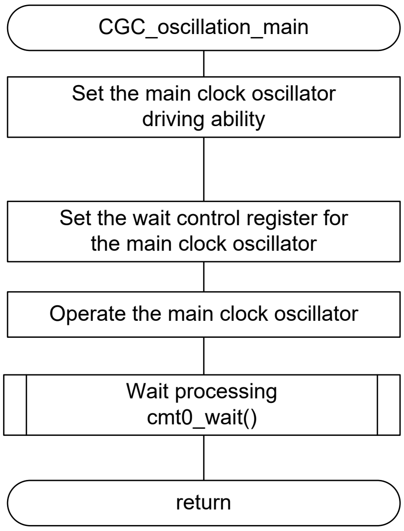
Notes:

1. Change the constant value according to the user system.
2. The setting value varies depending on the system clock selected by the constant.

Figure 3.9 Clock Initialization (2/2)

3.9.5 Main Clock Oscillation Setting

Figure 3.10 shows the Main Clock Oscillation Setting.


CGC_oscillation_main
Set the main clock oscillator driving ability
Set the wait control register for the main clock oscillator
Operate the main clock oscillator
Wait processing cmt0_wait()
return

MOFCR register ← 30h [1]
MODRV[2:0] bits = 000b: 16-MHz to 20-MHz non-lead type ceramic resonator
MODRV2[1:0] bits = 11b: 16 MHz to 20 MHz
MOSEL bit = 0: Resonator

MOSCWTCR register ← 0Dh [1]
MSTS[4:0] bits = 01101b: Wait time = 131072 cycles (approx. 6.55 ms)

MOSCCR register ← 00h
MOSTP bit = 0: Main clock oscillator is operating.

Wait for the main clock oscillation stabilization wait time [1] (approx. 13.1 ms).

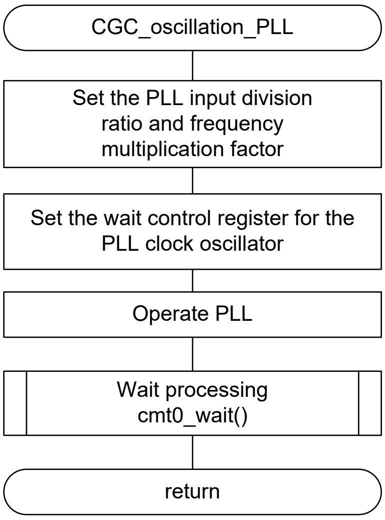
Note:

1. Change the constant value according to the user system.

Figure 3.10 Main Clock Oscillation Setting

3.9.6 PLL Clock Oscillation Setting

Figure 3.11 shows the PLL Clock Oscillation Setting.


CGC_oscillation_PLL
Set the PLL input division ratio and frequency multiplication factor
Set the wait control register for the PLL clock oscillator
Operate PLL
Wait processing cmt0_wait()
return

PLLCR register ← 0901h [1]
PLIDIV[1:0] bits = 01b: Divided by 2
STC[4:0] bits = 01001b: Multiplied by 10

PLLWTCR register ← 09h [1]
PSTS[4:0] bits = 01001b: Waiting time = 65536 cycles (approx. 655.36 μs)

PLLCR2 register ← 00h
PLLEN bit = 0: PLL is operating.

Wait for the PLL clock oscillation stabilization wait time [1] (approx. 1.50 ms).

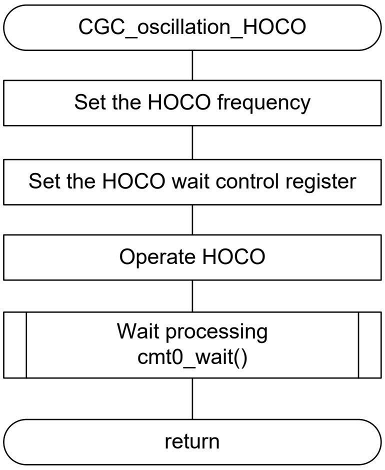
Note:

1. Change the constant value according to the user system.

Figure 3.11 PLL Clock Oscillation Setting

3.9.7 HOCO Clock Oscillation Setting

Figure 3.12 shows the HOCO Clock Oscillation Setting.



HOCOCR2 register ← 03h [1]
HCFRQ[1:0] bits = 11b: 50 MHz

HOCOWTCR2 register ← 03h [2]
HSTS2[3:0] bits = 0011b: 9216 cycles (approx. 184.32 μs)

HOCOCR register ← 00h
HCSTP bit = 0: HOCO is operating.

Wait for the HOCO clock oscillation stabilization wait time [1] (approx. 350 μs).

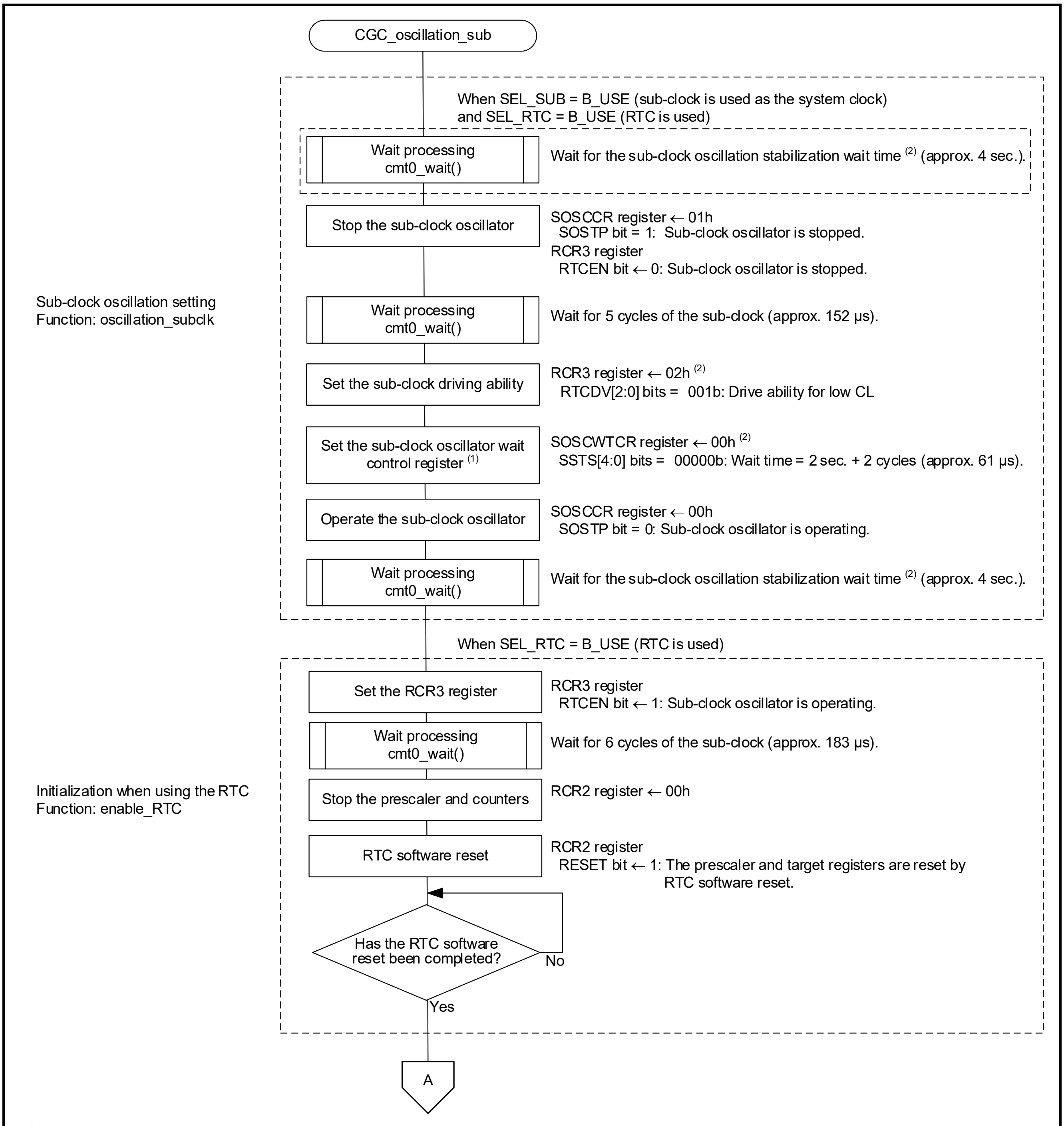
Notes:

1. Change the constant value according to the user system.
2. The setting value varies depending on the HOCO frequency selected by the constant.

Figure 3.12 HOCO Clock Oscillation Setting

3.9.8 Sub-Clock Oscillation Setting

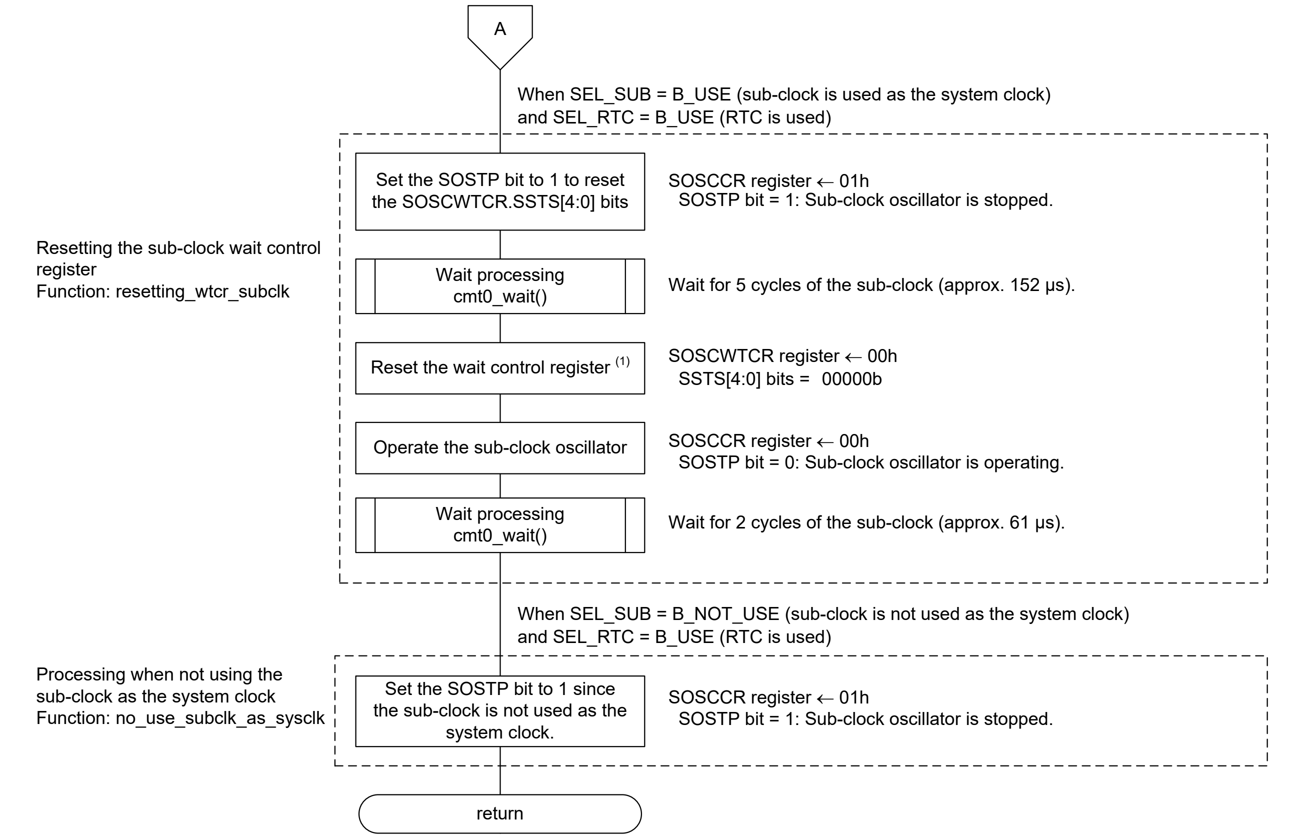
Figure 3.13 and Figure 3.14 show the sub clock oscillation setting.


CGC_oscillation_sub
When SEL_SUB = B_USE (sub-clock is used as the system clock) and SEL_RTC = B_USE (RTC is used)
Wait processing cmt0_wait()
Wait for the sub-clock oscillation stabilization wait time (2) (approx. 4 sec.).
Stop the sub-clock oscillator
SOSCCR register ← 01h
SOSTP bit = 1: Sub-clock oscillator is stopped.
RCR3 register
RTCEN bit ← 0: Sub-clock oscillator is stopped.
Sub-clock oscillation setting
Function: oscillation_subclk
Wait processing cmt0_wait()
Wait for 5 cycles of the sub-clock (approx. 152 μs).
Set the sub-clock driving ability
RCR3 register ← 02h (2)
RTCDV[2:0] bits = 001b: Drive ability for low CL
Set the sub-clock oscillator wait control register (1)
SOSCWTCR register ← 00h (2)
SSTS[4:0] bits = 00000b: Wait time = 2 sec. + 2 cycles (approx. 61 μs).
Operate the sub-clock oscillator
SOSCCR register ← 00h
SOSTP bit = 0: Sub-clock oscillator is operating.
Wait processing cmt0_wait()
Wait for the sub-clock oscillation stabilization wait time (2) (approx. 4 sec.).
When SEL_RTC = B_USE (RTC is used)
Set the RCR3 register
RCR3 register
RTCEN bit ← 1: Sub-clock oscillator is operating.
Wait processing cmt0_wait()
Wait for 6 cycles of the sub-clock (approx. 183 μs).
Initialization when using the RTC
Function: enable_RTC
Stop the prescaler and counters
RCR2 register ← 00h
RTC software reset
RCR2 register
RESET bit ← 1: The prescaler and target registers are reset by RTC software reset.
Has the RTC software reset been completed?
No
Yes
A

Notes:

1. Set the sub-clock oscillator wait control register to a value greater than or equal to the sub-clock oscillation stabilization time recommended by the crystal/ceramic resonator manufacturer.
2. Change the constant value according to the user system.
3. When setting the START bit to 0, set b7 to 0 at the same time.

Figure 3.13 Sub-Clock Oscillation Setting (1/2)


A
When SEL_SUB = B_USE (sub-clock is used as the system clock)
and SEL_RTC = B_USE (RTC is used)
Resetting the sub-clock wait control register
Function: resetting_wtcr_subclk
Set the SOSTP bit to 1 to reset the SOSCWTCR.SSTS[4:0] bits
SOSCCR register ← 01h
SOSTP bit = 1: Sub-clock oscillator is stopped.
Wait processing
cmt0_wait()
Wait for 5 cycles of the sub-clock (approx. 152 µs).
Reset the wait control register (1)
SOSCWTCR register ← 00h
SSTS[4:0] bits = 00000b
Operate the sub-clock oscillator
SOSCCR register ← 00h
SOSTP bit = 0: Sub-clock oscillator is operating.
Wait processing
cmt0_wait()
Wait for 2 cycles of the sub-clock (approx. 61 µs).
When SEL_SUB = B_NOT_USE (sub-clock is not used as the system clock)
and SEL_RTC = B_USE (RTC is used)
Processing when not using the sub-clock as the system clock
Function: no_use_subclk_as_sysclk
Set the SOSTP bit to 1 since the sub-clock is not used as the system clock.
SOSCCR register ← 01h
SOSTP bit = 1: Sub-clock oscillator is stopped.
return

Note:

1. When exiting from software standby mode while the sub-clock is selected as the system clock, the clock provision into the LSI is started after the specified time set in the SOSCWTCR.SSTS[4:0] bits has elapsed since the condition to exit has occurred. However when the RCR3.RTCEN bit is 1 (operating), the sub-clock does not stop even during software standby mode. Then the sub-clock oscillation stabilization wait time is not required when exiting. Therefore 00000b is set to the SOSCWTCR.SSTS[4:0] bits to make the sub-clock oscillation stabilization wait time become minimum.

Figure 3.14 Sub-Clock Oscillation Setting (2/2)

3.9.9 Sub-Clock Stop Setting

Figure 3.15 shows the Sub-Clock Stop Setting.

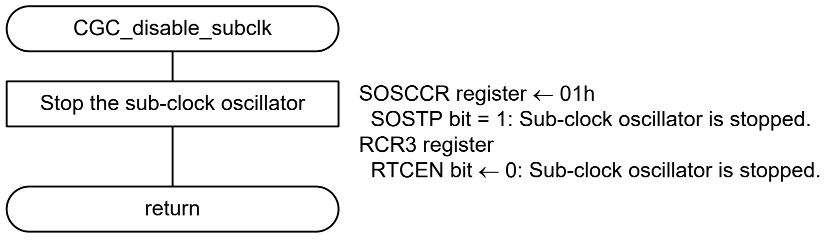
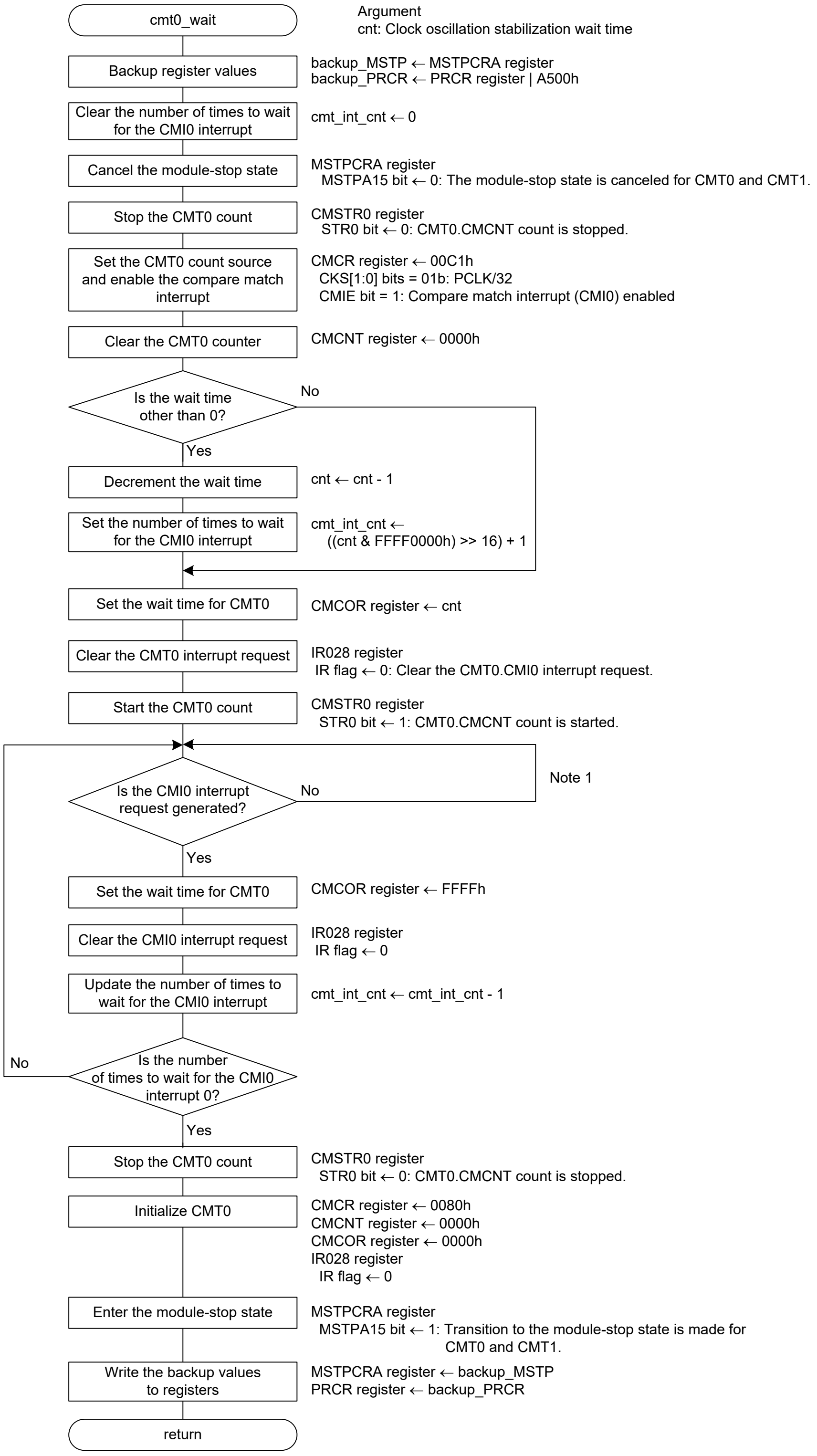

CGC_disable_subclk
Stop the sub-clock oscillator
SOSCCR register ← 01h
SOSTP bit = 1: Sub-clock oscillator is stopped.
RCR3 register
RTCEN bit ← 0: Sub-clock oscillator is stopped.
return

Figure 3.15 Sub-Clock Stop Setting

3.9.10 Wait Processing

Figure 3.16 shows the Wait Processing.


cmt0_wait
Argument
cnt: Clock oscillation stabilization wait time
Backup register values
backup_MSTP ← MSTPCRA register
backup_PRCR ← PRCR register | A500h
Clear the number of times to wait for the CMI0 interrupt
cmt_int_cnt ← 0
Cancel the module-stop state
MSTPCRA register
MSTPA15 bit ← 0: The module-stop state is canceled for CMT0 and CMT1.
Stop the CMT0 count
CMSTR0 register
STR0 bit ← 0: CMT0.CMCNT count is stopped.
Set the CMT0 count source and enable the compare match interrupt
CMCR register ← 00C1h
CKS[1:0] bits = 01b: PCLK/32
CMIE bit = 1: Compare match interrupt (CMI0) enabled
Clear the CMT0 counter
CMCNT register ← 0000h
Is the wait time other than 0?
No
Yes
Decrement the wait time
cnt ← cnt - 1
Set the number of times to wait for the CMI0 interrupt
cmt_int_cnt ←
((cnt & FFFF0000h) >> 16) + 1
Set the wait time for CMT0
CMCOR register ← cnt
Clear the CMT0 interrupt request
IR028 register
IR flag ← 0: Clear the CMT0.CMI0 interrupt request.
Start the CMT0 count
CMSTR0 register
STR0 bit ← 1: CMT0.CMCNT count is started.
Is the CMI0 interrupt request generated?
No
Note 1
Yes
Set the wait time for CMT0
CMCOR register ← FFFFh
Clear the CMI0 interrupt request
IR028 register
IR flag ← 0
Update the number of times to wait for the CMI0 interrupt
cmt_int_cnt ← cmt_int_cnt - 1
No
Is the number of times to wait for the CMI0 interrupt 0?
Yes
Stop the CMT0 count
CMSTR0 register
STR0 bit ← 0: CMT0.CMCNT count is stopped.
Initialize CMT0
CMCR register ← 0080h
CMCNT register ← 0000h
CMCOR register ← 0000h
IR028 register
IR flag ← 0
Enter the module-stop state
MSTPCRA register
MSTPA15 bit ← 1: Transition to the module-stop state is made for CMT0 and CMT1.
Write the backup values to registers
MSTPCRA register ← backup_MSTP
PRCR register ← backup_PRCR
return

Note:

1. When the counter of the watchdog timer (WDT) or independent watchdog timer (IWDT) is operating, refresh the counter in this loop as required.

Figure 3.16 Wait Processing

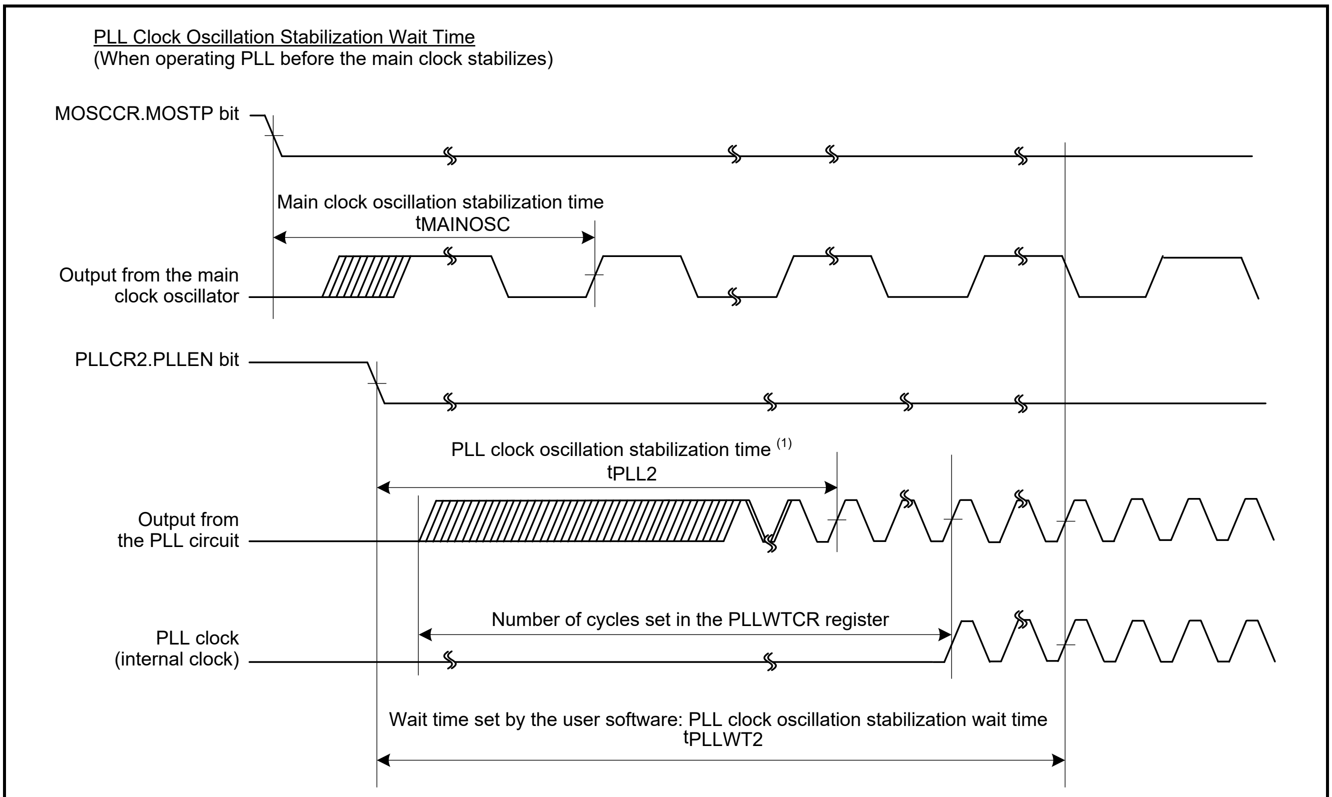
4. Appendix

4.1 Clock Oscillation Stabilization Wait Time

4.1.1 When Operating PLL before Main Clock Oscillation Stabilizes

When oscillating the main clock and PLL clock, their oscillation stabilization wait times can be combined into a single wait time.

Figure 4.1 shows the PLL Oscillation Stabilization Wait Time (when Operating PLL before the Main Clock Stabilizes) and Table 4.1 lists the Setting Value for the PLL Clock Wait Control Register and Oscillation Stabilization Wait Time (when Operating PLL before Main Clock Oscillation Stabilizes).


PLL Clock Oscillation Stabilization Wait Time
(When operating PLL before the main clock stabilizes)
MOSCCR.MOSTP bit
Main clock oscillation stabilization time
tMAINOSC
Output from the main clock oscillator
PLLCR2.PLLEN bit
PLL clock oscillation stabilization time (1)
tPLL2
Output from the PLL circuit
Number of cycles set in the PLLWTCR register
PLL clock (internal clock)
Wait time set by the user software: PLL clock oscillation stabilization wait time
tPLLWT2

Note:
1. Refer to the Electrical Characteristics chapter in the User's Manual: Hardware.

Figure 4.1 PLL Oscillation Stabilization Wait Time (when Operating PLL before the Main Clock Stabilizes)

Table 4.1 Setting Value for the PLL Clock Wait Control Register and Oscillation Stabilization Wait Time (when Operating PLL before Main Clock Oscillation Stabilizes)

Setting Item	Condition of Setting Value
PLL wait control register (PLLWTCR.PSTS[4:0] bits)	Value greater than or equal to the main clock oscillation stabilization time recommended by the crystal/ceramic resonator manufacturer plus tPLL1 (max. 500 μs)
Oscillation stabilization wait time (tPLLWT2)	Value greater than or equal to two times the number of cycles set in the PLLWTCR register.

5. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

6. Reference Documents

User’s Manual: Hardware

RX210 Group User’s Manual: Hardware Rev.1.50 (R01UH0037EJ)
The latest version can be downloaded from the Renesas Electronics website.

Technical Update/Technical News

The latest information can be downloaded from the Renesas Electronics website.

User’s Manual: Development Tools

RX Family C/C++ Compiler Package V.1.01 User’s Manual Rev.1.00 (R20UT0570EJ)
The latest version can be downloaded from the Renesas Electronics website.

Website and Support

Renesas Electronics website

http://www.renesas.com

Inquiries

http://www.renesas.com/contact/

REVISION HISTORY	RX210 Group Application Note Initial Setting

Rev.	Date	Description	
		Page	Summary
1.00	Jul. 2, 12	—	First edition issued
2.00	Mar. 1, 13	—	Revised in accordance with the overhaul of specifications. - Added stop processing for active peripheral functions after a reset. - Changed the specifications of the nonexistent port initialization and clock initialization. - Covered chip versions A, B and C.
2.10	Sep. 2, 13	—	Revised in accordance with the following: - Changed the board used. - Covered 144-pin and 145-pin packages. - Modified the wait processing with CMT0 to support 32-bit argument.
		1	Abstract: Rewritten the explanation.
		1	Products: Added 145-pin and 144-pin packages.
		1	Changes from Rev. 1.00: Deleted.
		3	1.1 Stopping Peripheral Functions Operating after a Reset: Added the description of RAM1.
		3	1.2 Configuring Nonexistent Ports: Changed the descriptions to cover less than 144 pins.
		6	Table 2.1 Operation Confirmation Conditions: - MCU used: Changed to R5F5210BBDFP. - iodefine.h version: Updated the version to 1.40 - Board used: Changed to R0K505210C002BE.
		7	Table 3.1 Peripheral Modules whose Module-Stop States are Canceled after a Reset: Added the RAM1 row.
		8, 9	Table 3.2 Nonexistent Ports (1/2) and Table 3.3 Nonexistent Ports (2/2) : - Added nonexistent port list for 100-pin package. - Added the following ports which exist in 144-pin and 145-pin packages in the lists: PORT6 to 9, F, K, and L. - Changed pin numbers for some ports in each list.
		17	Table 3.12 Constants Used in the Sample Code (2/2): - Deleted the ROM_SIZE constant and note 9. - Added the MSTP_STATE_RAM1 constant.
		18	Table 3.13 Constants Used in the Sample Code: Deleted the UPTO_512KB and OVER_512KB constants.
		19	Table 3.14 Constants when a 145-Pin or 144-Pin Package is Used (PIN_SIZE=145 or PIN_SIZE=144): Added.
		19 to 21	Table 3.15 Constants when a 100-Pin Package is Used, Table 3.16 Constants when a 80-Pin Package is Used, Table 3.17 Constants when a 64-Pin Package is Used, and Table 3.18 Constants when a 48-Pin Package is Used: Added the following ports which exist in 144-pin and 145-pin packages in the lists: PORT6 to 9, F, K, and L.
		23	3.8 Function Specifications: Deleted the Change from Rev. 1.00 from the R_INIT_NonExistentPort and R_INIT_Clock functions.

Rev.	Date	Description	
		Page	Summary
2.10	Sep. 2, 13	26	Figure 3.6 Stop Processing for Active Peripheral Functions after a Reset: Added the description of RAM1.
		27	Figure 3.7 Nonexistent Port Initialization: Added the following ports which exist in 144-pin and 145-pin packages in the lists: PORT6 to 9, F, K, and L.
		34	Figure 3.16 Wait Processing: Revised.
2.20	Apr. 1, 14	—	Revised to support for 69-pin package.
		1	Products: - Added 69-pin package. - Modified the maximum ROM size for the 100-pin package.
		6	Table 2.1 Operation Confirmation Conditions: - Sample code version: Updated the version to 2.20.
		9	Table 3.3 Nonexistent Ports (2/2): Added the description of 69-pin package.
		9	3.2.2 Selecting the Number of Pins: Added the description of 69-pin package.
		20	Table 3.17 Constants when a 69-Pin or 64-Pin Package is Used (PIN_SIZE=69 or PIN_SIZE=64): Added the description of 69-pin package.
		32	Figure 3.13 Sub-Clock Oscillation Setting (1/2): Added note 3.
		34	Figure 3.16 Wait Processing: Revised the procedure after the second branch.
2.21	Feb. 1, 21	9	Table 2.1 Integrated development environment, C compiler, iodefine.h and Sample code version, changed.
		32	Figure 3.13 Subclock Oscillation Enable (2/2), changed.
		37, 38	Fixed the format of the date of Revision History.
		program	Technical update TN-RX*-A239B/E, supported.

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/.