RX63N Group, RX631 Group

Initial Setting

R01AN1245EJ0110
Rev. 1.10
Jan. 6, 2014

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

- RX63N Group 177-pin and 176-pin packages with a ROM size between 768 KB and 2 MB
- RX63N Group 145-pin and 144-pin packages with a ROM size between 768 KB and 2 MB
- RX63N Group 100-pin package with a ROM size between 768 KB and 2 MB
- RX631 Group 177-pin and 176-pin packages with a ROM size between 256 KB and 2 MB
- RX631 Group 145-pin and 144-pin packages with a ROM size between 256 KB and 2 MB
- RX631 Group 100-pin package with a ROM size between 256 KB and 2 MB
- RX631 Group 64-pin and 48-pin packages with a ROM size between 256 KB and 512 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, EXDMAC, RAM0, RAM1. Although the sample code includes processing for stopping these peripheral functions, it is not executed in the sample code. Change the oscillation parameters as required to execute processing. Also the RTC needs to be stopped when it is not used. The setting to stop the RTC is performed in the clock setting part of the sample code.

1.2 Configuring Nonexistent Ports

Ports which are not connected to pins must be set as output for products with less than 176 pins. In the sample code, initial values are set for 176-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. Main clock setting
2. Sub-clock setting
3. PLL clock setting
4. System clock switching

Processing of the sub-clock setting differs depending on sub-clock usage including processing when the sub-clock is not used. Five setting patterns are introduced in the application note (see section 3.3.2. for more information).

In the sample code, the PLL clock is used as the system clock without using the sub-clock.

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 on 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 Start-Up Time	Remarks
Crystal/ceramic resonator for the main clock	12 MHz	4.2 ms (2)	
Crystal for the sub-clock	32.768 kHz (1)	1.3 sec. (2)	For standard clock loads
PLL clock	192 MHz (main clock divided by 1 and multiplied by 16)	Maximum of 500 µs (3)	

Notes:
1. Sub-clock oscillation is disabled in the sample code.
2. The start-up 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 start-up 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.

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	R5F563NBDDFC (RX63N Group)
Operating frequencies	- Main clock: 12 MHz - Sub-clock: 32.768 kHz (stopped in patterns A and B, and a 48-pin package) - PLL: 192 MHz (main clock divided by 1 and multiplied by 16) - HOCO: Stopped - System clock (ICLK): 96 MHz (PLL divided by 2) - Peripheral module clock A (PCLKA): 96 MHz (PLL divided by 2) - Peripheral module clock B (PCLKB): 48 MHz (PLL divided by 4) - External bus clock (BCLK): 48 MHz (PLL divided by 4) - FlashIF clock (FCLK): 48 MHz (PLL divided by 4) - IEBUS clock (IECLK): 48 MHz (PLL divided by 4)
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics Corporation High-performance Embedded Workshop Version 4.09.01
C compiler	Renesas Electronics Corporation C/C++ Compiler Package for RX Family V.1.02 Release 01
	Compile options -cpu=rx600 -output=obj="$(CONFIGDIR)\$(FILELEAF).obj" -debug -nologo (The default setting is used in the integrated development environment.)
iodefine.h version	Version 1.6A
Endian	Little endian
Operating mode	Single-chip mode
Processor mode	Supervisor mode
Sample code version	Version 1.10
Board used	Renesas Starter Kit+ for RX63N (product part no.: R0K50563NC000BE)

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 the 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)
EXDMAC	MSTPCRA.MSTPA29 bit		
RAM0	MSTPCRC.MSTPC0 bit		
RAM1	MSTPCRC.MSTPC1 bit		

3.2 Nonexistent Port Initialization

3.2.1 Overview

When using a product with less than 176 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	144-Pin and 145-Pin Packages	No. of Pins	100-Pin Package	No. of Pins
PORT0	—	—	P00 to P03	4
PORT1	P10 and P11	2	P10 and P11	2
PORT2	—	—	—	—
PORT3	—	—	—	—
PORT4	—	—	—	—
PORT5	P57	1	P56 and P57	2
PORT6	—	—	P60 to P67	8
PORT7	—	—	P70 to P77	8
PORT8	P84 and P85	2	P80 to P87	8
PORT9	P94 to P97	4	P90 to P97	8
PORTA	—	—	—	—
PORTB	—	—	—	—
PORTC	—	—	—	—
PORTD	—	—	—	—
PORTE	—	—	—	—
PORTF	PF0 to PF4	5	PF0 to PF5	6
PORTG	PG0 to PG7	8	PG0 to PG7	8
PORTJ	—	—	PJ5	1

Table 3.3 Nonexistent Ports (2/2)

Port Symbol	64-Pin Package	No. of Pins	48-Pin Package	No. of Pins
PORT0	P00 to P03, P07	5	P00 to P03, P05, P07	6
PORT1	P10 to P13	4	P10 to P13	4
PORT2	P20 to P25	6	P20 to P25	6
PORT3	P32 to P34	3	P32 to P34	3
PORT4	P45 and P47	2	P43 to P45, P47	4
PORT5	P50 to P53, P56, P57	6	P50 to P57	8
PORT6	P60 to P67	8	P60 to P67	8
PORT7	P70 to P77	8	P70 to P77	8
PORT8	P80 to P87	8	P80 to P87	8
PORT9	P90 to P97	8	P90 to P97	8
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	PF0 to PF5	6	PF0 to PF5	6
PORTG	PG0 to PG7	8	PG0 to PG7	8
PORTJ	PJ3, PJ5	2	PJ3, PJ5	2

3.2.2 Selecting the Number of Pins

The number of pins in the sample code is set for the 176-pin package (PIN_SIZE=176). This application note covers 177-pin, 176-pin, 145-pin, 144-pin, 100-pin, 64-pin, and 48-pin packages. When using products with less than 176 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 procedure 2. Sub-clock and RTC configuration, select a pattern from Table 3.5 according to the user system.

Steps 1 to 6 are all performed in the sample code. In processing, operate the main clock, PLL, and HOCO. Set the main clock as the RTC count source, switch the system clock to HOCO, and stop the sub-clock in the sub-clock setting (pattern A). Then switch the system clock to PLL and stop HOCO.

Table 3.4 Clock Setting Procedure

Step	Processing	Details	Setting in the Sample Code
1	Main clock oscillation setting	Set the MOSCWTCR register with a wait time until the main clock output is provided to the internal clock, and enable main clock oscillation. Then wait for the main clock oscillation stabilization wait time (1) by software.	Main clock oscillator: Operating
2	Sub-clock and RTC configuration	Select the sub-clock setting pattern from five patterns listed in Table 3.5 according to the user system and configure the settings accordingly. - Patterns A and B, and the pattern for a 48-pin package Operate HOCO, set the RTC count source, switch the system clock to HOCO, and then configure the sub-clock and RTC. - Patterns C, D and E Set the RTC count source, and configure the sub-clock and RTC.	Sub-clock oscillator: Stopped (pattern A)
3	PLL oscillation setting (2)	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 PLL clock oscillation stabilization wait time (1) by software.	PLL: Operating
4	Clock division ratio setting (3)	Set the clock division ratio.	- ICLK,PCLKA: Divided by 2 - PCLKB, BCLK, FCLK.IECLK: Divided by 4 - SDCLK, BCLK: Stopped - UCLK: Not used
5	Switching the system clock	Switch the system clock according to the user system.	Clock switched to: PLL
6	HOCO setting	Set HOCO to be operating or stopped. When not using the HOCO clock, power consumption can be reduced by turning off the HOCO power supply.	- HOCO: Stopped - HOCO power supply: OFF

Notes:
1. Refer to 3.3.3 Oscillation Stabilization Wait Time for Each Clock for details on the oscillation stabilization wait time.
2. When not using PLL, the PLL clock setting is not necessary.
3. Do not set divide-by-1 and divide-by-2 when selecting the main clock as the system clock.

3.3.2 Sub-Clock Setting Patterns

The following five patterns are provided for step 2 in Table 3.4.

A. Neither the sub-clock nor the RTC is used. Or the package used is a 48-pin package.
B. The sub-clock is not used and the RTC uses the main clock as the count source.
C. The sub-clock is used as the system clock, and the RTC is not used.
D. The sub-clock is used as the RTC count source.
E. The sub-clock is used as both the system clock and RTC count source.

Table 3.5 lists the Sub-Clock Setting Patterns. Select a pattern from patterns A to E according to the user system. For a 48-pin package, select pattern A.

Table 3.5 Sub-Clock Setting Patterns

Pattern	Sub-Clock		RTC		PCLKB (1) when Setting the Sub-Clock
	Crystal Usage	System Clock	Usage	Count Source	
A	Not used	—	Not used	Main clock	HOCO divided by 1
B	Not used	—	Used	Main clock	HOCO divided by 1
C	Used	Used	Not used	Sub-clock	LOCO divided by 1
D	Used	Not used	Used	Sub-clock	LOCO divided by 1
E	Used	Used	Used	Sub-clock	LOCO divided by 1

Note:
1. The set value for PCLKB must satisfy the following:
PCLKB ≥ RTC count source.

3.3.3 Oscillation Stabilization Wait Time for Each Clock

This section describes the wait control registers and oscillation stabilization wait times for the main clock, PLL and sub-clock. This section also describes values for the oscillation stabilization wait times specified in the sample code.

3.3.3.1 Main Clock Oscillation Stabilization Wait Time

Figure 3.1 shows the Main Clock Oscillation Stabilization Wait Time and Table 3.6 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 oscillator start-up time (tMAINOSC) recommended by the crystal/ceramic resonator manufacturer. Set the main clock oscillation stabilization wait time (tMAINOSCWT) to a value greater than 'tMAINOSC recommended by the crystal/ceramic resonator manufacturer + (wait time set in the MOSCWTCR register + 16384 cycles)'.

tMAINOSC used in the sample code is 4.2 ms, thus the setting value in the MOSCWTCR register is 0Ch (approximately 5.46 ms), and the setting value for tMAINOSCWT is approximately 11.026 ms.

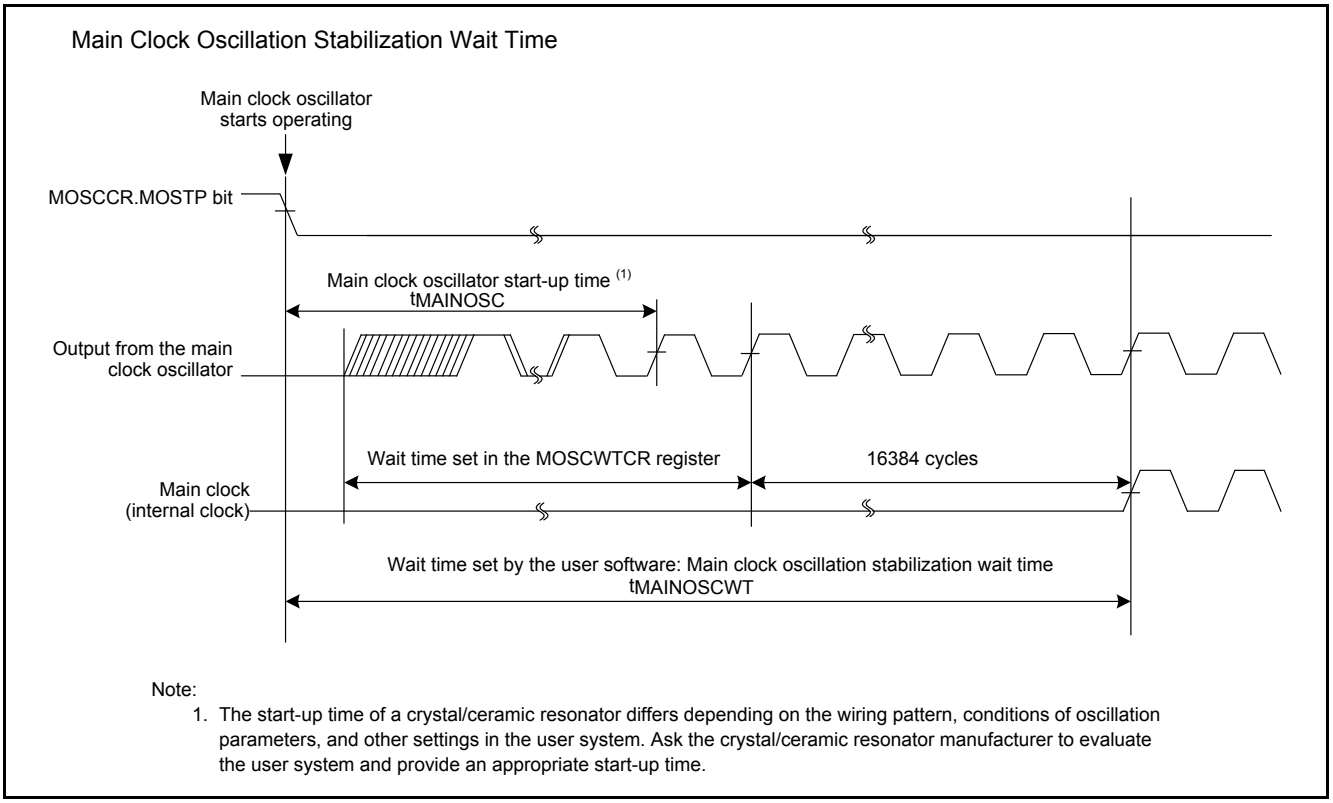


Figure 3.1 Main Clock Oscillation Stabilization Wait Time

Table 3.6 Setting Value for the MOSCWTCR Register and Oscillation Stabilization Wait Time

	Condition of Setting Value and Calculation Method	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	0Ch (approx. 5.46 ms)
Oscillation stabilization wait time (tMAINOSCWT)	When n is the wait time selected by the MOSCWTCR.MSTS[4:0] bits: $tMAINOSC + \frac{n + 16384}{fMAIN}$	Approx. 11.026 ms

3.3.3.2 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 3.7 lists the Setting Value of the PLLWTCR Register and Oscillation Stabilization Wait Time.

Set a value greater than or equal to PLL lock time (tPLL1 (max. 500 µs)) to the PLL wait control register (PLLWTCR). Set the PLL clock oscillation stabilization wait time (tPLLWT1) to a value greater than 'tPLL1 (500 µs) + (wait time set in the PLLWTCR register + 131072 cycles)'.

tPLL1 is a maximum of 500 µs, thus the setting value in the PLLWTCR register is 0Ah (approximately 681.6 µs), and the setting value for the tPLLWT1 is approximately 1.865 ms.

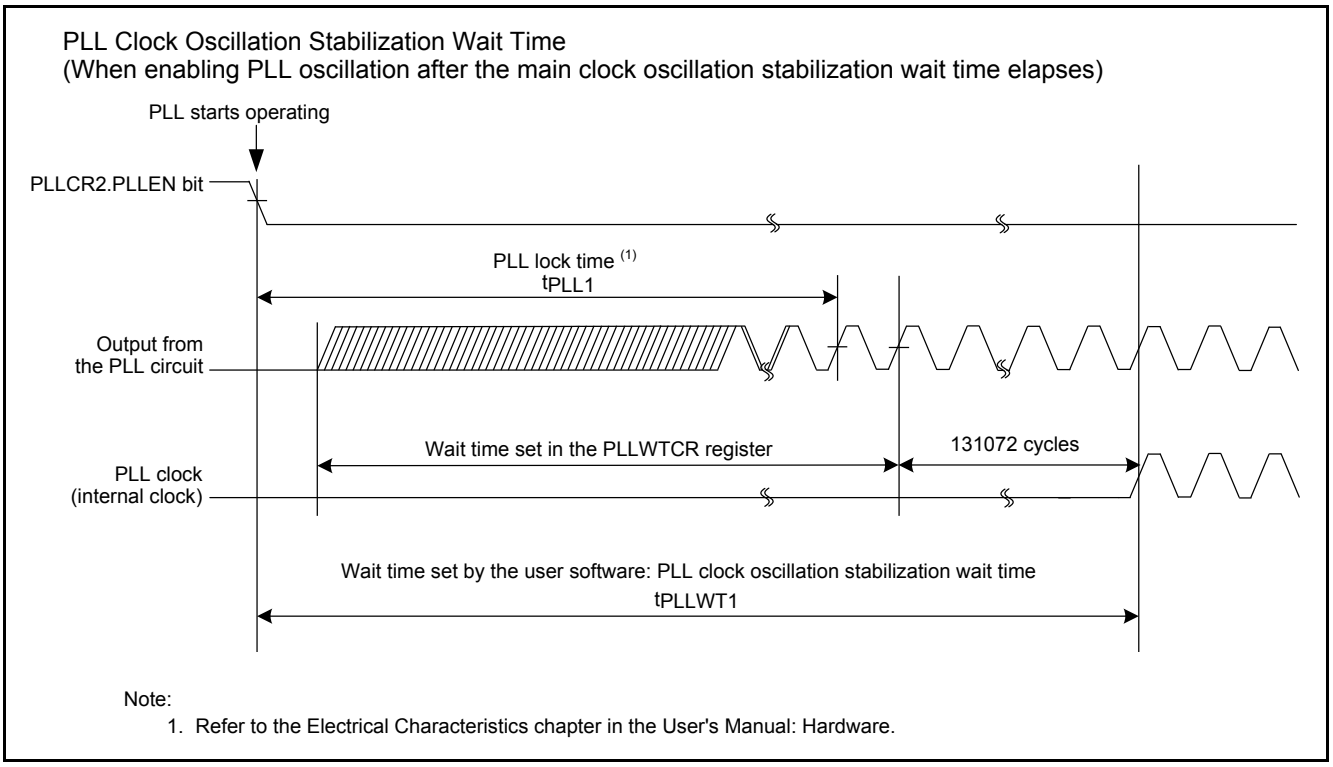


Figure 3.2 PLL Clock Oscillation Stabilization Wait Time

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

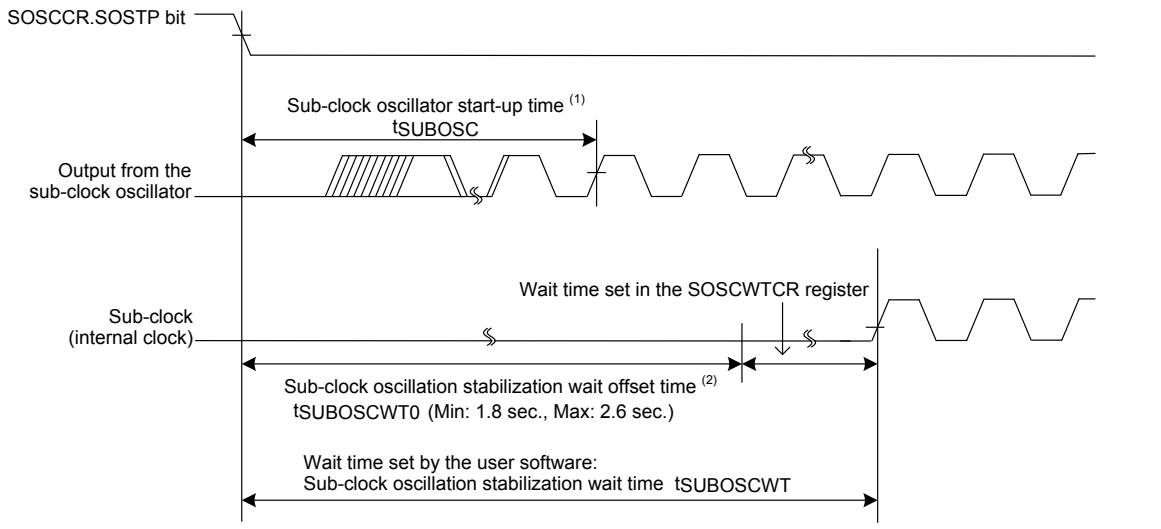
	Condition of Setting Value and Calculation Method	Setting Value in the Sample Code
PLLWTCR.PSTS[4:0] bits	Value greater than or equal to tPLL1 (max. 500 µs)	0Ah (approx. 681.6 µs)
Oscillation stabilization wait time (tPLLWT1)	When n is the wait time selected by the PLLWTCR.PSTS[4:0] bits: $tPLL1 + \frac{n + 131072}{fPLL}$	Approx. 1.865 ms

3.3.3.3 Sub-Clock Oscillation Stabilization Wait Time (Except 64-Pin Packages)

Figure 3.3 shows the Sub-Clock Oscillation Stabilization Wait Time and Table 3.8 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 oscillator start-up time (tSUBOSC) recommended by the crystal/ceramic resonator manufacturer minus the minimum value of the sub-clock oscillation stabilization wait offset time (tSUBOSCWT0 (1.8 sec.)). Set the sub-clock oscillation stabilization wait time (tSUBOSCWT) to a value greater than 'tSUBOSC recommended by the crystal/ceramic resonator manufacturer' or 'maximum value of tSUBOSCWT0 (2.6 sec.)', whichever is greater plus the wait time 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 (approximately 61 µs), and the setting value for tSUBOSCWT is approximately 2.6 seconds.



Notes:
1. Contact the crystal/ceramic resonator manufacturer to determine the start-up time of a crystal/ceramic resonator for the user system. The start-up 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 3.8 Setting Value of the SOSCWTCR Register and Oscillation Stabilization Wait Time

	Condition of Setting Value and Calculation Method	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 minus the minimum value of tSUBOSCWT0 (1.8 sec.)	00h (approx. 61 µs)
Oscillation stabilization wait time (tSUBOSCWT)	When n is the wait time selected by the SOSCWTCR.SSTS[4:0] bits: Maximum value of $tSUBOSCWT0$ (2.6 sec.) + $\frac{n}{fSUB}$	Approx. 2.6 sec.

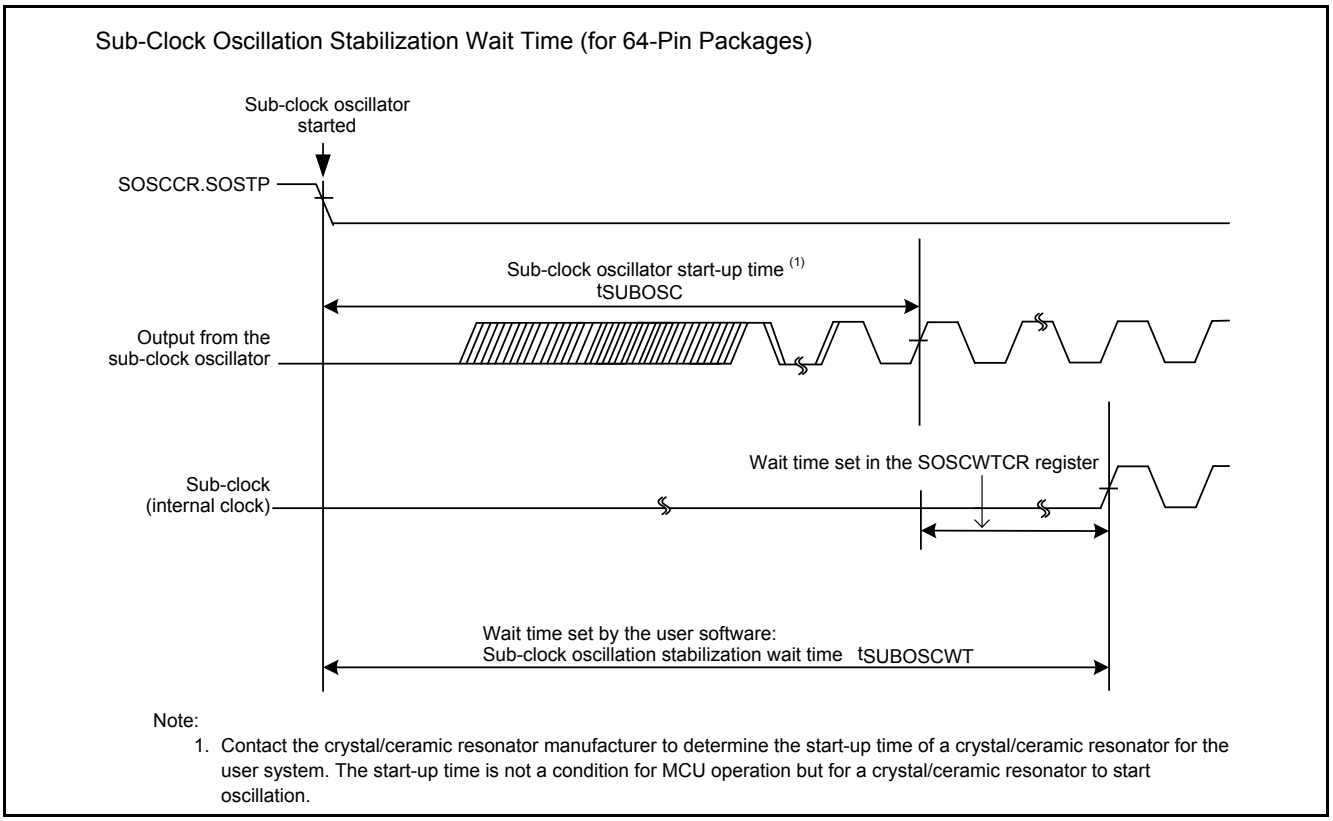
3.3.3.4 Sub-Clock Oscillation Stabilization Wait Time (for 64-Pin Packages)

The sub-clock oscillation stabilization wait offset time (tSUBOSCWT0) is not included in 64-pin packages.

Figure 3.4 shows the Sub-Clock Oscillation Stabilization Wait Time (for 64-Pin Packages) and Table 3.9 lists the Setting Value of the SOSCWTCR Register and Oscillation Stabilization Wait Time (for 64-Pin Packages).

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

tSUBOSC used in the sample code is 1.3 seconds, thus the setting value in the SOSCWTCR register is 0Ch (approximately 2 sec.), and the setting value for tSUBOSCWT is approximately 3.3 seconds.



Note:
1. Contact the crystal/ceramic resonator manufacturer to determine the start-up time of a crystal/ceramic resonator for the user system. The start-up time is not a condition for MCU operation but for a crystal/ceramic resonator to start oscillation.

Figure 3.4 Sub-Clock Oscillation Stabilization Wait Time (for 64-Pin Packages)

Table 3.9 Setting Value of the SOSCWTCR Register and Oscillation Stabilization Wait Time (for 64-Pin Packages)

	Condition of Setting Value and Calculation Method	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	0Ch (approx. 2 sec.)
Oscillation stabilization wait time (tSUBOSCWT)	When n is the wait time selected by the SOSCWTCR.SSTS[4:0] bits: $tSUBOSC + \frac{n}{fSUB}$	Approx. 3.3 sec.

3.4 File Composition

Table 3.10 lists the Files Used in the Sample Code. Files generated by the integrated development environment are not included in this table.

Table 3.10 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 3.11 lists the Option-Setting Memory Configured in the Sample Code. When necessary, set a value suited to the user system.

Table 3.11 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 3.12 and Table 3.13 list the Constants Used in the Sample Code, Table 3.14 lists the Constants when a 177-Pin or 176-Pin Package is Used (PIN_SIZE=177 or PIN_SIZE=176), Table 3.15 lists the Constants when a 145-Pin or 144-Pin Package is Used (PIN_SIZE=145 or PIN_SIZE=144), Table 3.16 lists the Constants when a 100-Pin Package is Used (PIN_SIZE=100), Table 3.17 lists the Constants when a 64-Pin Package is Used (PIN_SIZE=64), and Table 3.18 lists the Constants when a 48-Pin Package is Used (PIN_SIZE=48).

Table 3.12 Constants Used in the Sample Code (1/2)

Constant Name	Setting Value	Contents
MAIN_CLOCK_Hz (1)	12,000,000 L	Oscillation frequency of a crystal/ceramic resonator for the main clock (Hz)
SUB_CLOCK_Hz (1)	32,768 L	Oscillation frequency of a crystal for the sub-clock (Hz)
WAIT_TIME_FOR_MAIN_ OSCILLATION (1)	11,026,000 L	Main clock oscillation stabilization wait time (ns)
WAIT_TIME_FOR_SUB_ OSCILLATION (1)	2,600,000,000 L	Sub-clock oscillation stabilization wait time (ns) (except 64-pin packages)
WAIT_TIME_FOR_SUB_ OSCILLATION (1)	3,300,000,000 L	Sub-clock oscillation stabilization wait time (ns) (for 64-pin packages)
WAIT_TIME_FOR_PLL_ OSCILLATION (1)	1,865,000 L	PLL clock oscillation stabilization wait time (ns)
WAIT_TIME_FOR_HOCO_ OSCILLATION	2,000,000 L	HOCO clock oscillation stabilization wait time (ns)
PATTERN_A	1	Sub-clock setting pattern A
PATTERN_B	2	Sub-clock setting pattern B
PATTERN_C	3	Sub-clock setting pattern C
PATTERN_D	4	Sub-clock setting pattern D
PATTERN_E	5	Sub-clock setting pattern E
PATTERN_48	(PATTERN_A)	Sub-clock setting pattern for a 48-pin package
SELECT_SUB (1)	PATTERN_A	Selection of the sub-clock setting pattern
LOW_CL (1)	—	Low driving ability is used.
MSTP_STATE_DMACDTC (2)	MODULE_STOP_DISABLE	Cancel the module-stop state for DMAC and DTC
MSTP_STATE_EXDMAC (2)	MODULE_STOP_DISABLE	Cancel the module-stop state for EXDAMC
MSTP_STATE_RAM0 (2)	MODULE_STOP_DISABLE	Cancel the module-stop state for RAM0
MSTP_STATE_RAM1 (2)	MODULE_STOP_DISABLE	Cancel the module-stop state for RAM1
PIN_SIZE (3)	176	Number of pins of the product used
MAIN_CLOCK_CYCLE	(1/MAIN_CLOCK_Hz)*10^9	Main clock cycle (ns)
SUB_CLOCK_CYCLE	(1/SUB_CLOCK_Hz)*10^9	Sub-clock cycle (ns)

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.

Table 3.13 Constants Used in the Sample Code (2/2)

Constant Name	Setting Value	Contents
FOR_CMT0_TIME (when the RTC count source is the sub-clock)	1/LOCO*32	Count period (ns) of the CMT0 timer for the oscillation stabilization wait time (LOCO = 143.75 kHz (max.), PCLKB divided by 32)
FOR_CMT0_TIME (when the RTC count source is the main clock)	1/HOCO*32	Count period (ns) of the CMT0 timer for the oscillation stabilization wait time (HOCO = 55 MHz (max.), PCLKB divided by 32)
FOR_CMT0_LOCO (when the RTC count source is the main clock)	1/LOCO*32	Count period (ns) of the CMT0 timer for the main clock and PLL oscillation stabilization wait times when the RTC count source is the main clock. (LOCO = 143.75 kHz (max.), PCLKB divided by 32)
MODULE_STOP_ENABLE	1	Module stop-state
MODULE_STOP_DISABLE	0	Module stop-state is canceled

Table 3.14 Constants when a 177-Pin or 176-Pin Package is Used (PIN_SIZE=177 or PIN_SIZE=176)

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_PGPDR	0x00	Setting value for the port PG direction register
DEF_PJPDR	0x00	Setting value for the port PJ direction register

Table 3.15 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	0x03	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	0x80	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	0x30	Setting value for the port P8 direction register
DEF_P9PDR	0xF0	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	0x1F	Setting value for the port PF direction register
DEF_PGPDR	0xFF	Setting value for the port PG direction register
DEF_PJPDR	0x00	Setting value for the port PJ direction register

Table 3.16 Constants when a 100-Pin Package is Used (PIN_SIZE=100)

Constant Name	Setting Value	Contents
DEF_P0PDR	0x0F	Setting value for the port P0 direction register
DEF_P1PDR	0x03	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	0xC0	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	0xFF	Setting value for the port P8 direction register
DEF_P9PDR	0xFF	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	0x3F	Setting value for the port PF direction register
DEF_PGPDR	0xFF	Setting value for the port PG direction register
DEF_PJPDR	0x20	Setting value for the port PJ direction register

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

Constant Name	Setting Value	Contents
DEF_P0PDR	0x8F	Setting value for the port P0 direction register
DEF_P1PDR	0x0F	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	0xA0	Setting value for the port P4 direction register
DEF_P5PDR	0xCF	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	0xFF	Setting value for the port P8 direction register
DEF_P9PDR	0xFF	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	0x3F	Setting value for the port PF direction register
DEF_PGPDR	0xFF	Setting value for the port PG direction register
DEF_PJPDR	0x28	Setting value for the port PJ direction register

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

Constant Name	Setting Value	Contents
DEF_P0PDR	0xAF	Setting value for the port P0 direction register
DEF_P1PDR	0x0F	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	0xFF	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	0xFF	Setting value for the port P8 direction register
DEF_P9PDR	0xFF	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	0x3F	Setting value for the port PF direction register
DEF_PGPDR	0xFF	Setting value for the port PG direction register
DEF_PJPDR	0x28	Setting value for the port PJ 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_no_use_subclk	Sub-clock setting pattern A (when the sub-clock is not used as the system clock or RTC count source, or when the package sued is a 48-pin package)
CGC_disable_subclk_RTC_use_mainclk	Sub-clock setting pattern B (when the sub-clock is not used and the RTC operates using the main clock)
CGC_subclk_as_sysclk	Sub-clock setting pattern C (when the sub-clock is used as the system clock and not used as the RTC count source)
CGC_subclk_as_RTC	Sub-clock setting pattern D (when the sub-clock is used as the RTC count source and not used as the system clock)
CGC_subclk_as_sysclk_RTC	Sub-clock setting pattern E (when the sub-clock is used as both the system clock and RTC count source)
disable_subclk	Disabling the sub-clock
oscillation_subclk	Sub-clock oscillation setting
no_use_subclk_as_sysclk	Processing when the sub-clock is not used as the system clock
resetting_wtcr_mainclk	Resetting the wait control register (RTC count source is the main clock)
resetting_wtcr_subclk	Resetting the wait control register (RTC count source is the sub-clock)
enable_RTC	Initialization when using the RTC
disable_RTC_mainclk	Initialization when not using the RTC (RTC count source is the main clock)
disable_RTC_subclk	Initialization when not using the RTC (RTC count source is the sub-clock)
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 176 pins.
Arguments	None
Return Value	None
Remarks	The number of pins in the sample code is set for the 176-pin package (PIN_SIZE=176). 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 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
Remarks	This processing is not necessary if PLL is not used as the system clock.

CGC_oscillation_HOCO	
Outline	HOCO clock oscillation setting
Header	r_init_clock.h
Declaration	void CGC_oscillation_HOCO(void)
Description	Oscillate HOCO and then wait for HOCO oscillation stabilization wait time by software.
Arguments	None
Return Value	None

CGC_no_use_subclk	
Outline	Sub-clock setting pattern A
Header	r_init_clock.h
Declaration	void CGC_no_use_subclk(void)
Description	Configure the setting when the sub-clock is not used as the system clock or RTC count source, or configure the setting for a 48-pin package.
Arguments	None
Return Value	None

CGC_disable_subclk_RTC_use_mainclk	
Outline	Sub-clock setting pattern B
Header	r_init_clock.h
Declaration	void CGC_disable_subclk_RTC_use_mainclk(void)
Description	Configure the setting when the sub-clock is not used and the RTC operates using the main clock.
Arguments	None
Return Value	None

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

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

CGC_subclk_as_sysclk_RTC	
Outline	Sub-clock setting pattern E
Header	r_init_clock.h
Declaration	void CGC_subclk_as_sysclk_RTC(void)
Description	Configure the setting when the sub-clock is used as both the system clock and RTC count source.
Arguments	None
Return Value	None

disable_subclk	
Outline	Disabling the sub-clock
Header	None
Declaration	static void disable_subclk(void)
Description	Configure the setting when the sub-clock is not used as the system clock or RTC count source.
Arguments	None
Return Value	None

oscillation_subclk	
Outline	Sub-clock oscillation setting
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 for processing when the sub-clock is used only as the RTC count source.
Arguments	None
Return Value	None

resetting_wtcr_mainclk	
Outline	Resetting the wait control register (RTC count source is the main clock)
Header	None
Declaration	static void resetting_wtcr_mainclk(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

resetting_wtcr_ subclk	
Outline	Resetting the wait control register (RTC count source is the sub-clock)
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

disable_RTC_mainclk	
Outline	Initialization when not using the RTC (RTC count source is the main clock)
Header	None
Declaration	static void disable_RTC_mainclk(void)
Description	Initialize the settings when not using the RTC.
Arguments	None
Return Value	None

disable_RTC_ subclk	
Outline	Initialization when not using the RTC (RTC count source is the sub-clock)
Header	None
Declaration	static void disable_RTC_subclk(void)
Description	Initialize the settings when not using the RTC.
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 3.3.3 Oscillation Stabilization Wait Time for Each Clock. 2. The value of FOR_CMT0_TIME varies depending on the RTC count source. FOR_CMT0_TIME is calculated using 55 MHz (max.) as a HOCO value when the RTC count source is the main clock, and 143.75 kHz (max.) as a LOCO value when the RTC count source is the sub-clock. The actual wait time may differ according to the HOCO or LOCO frequency.	

3.9 Flowcharts

3.9.1 Main Processing

Figure 3.5 shows the Main Processing.

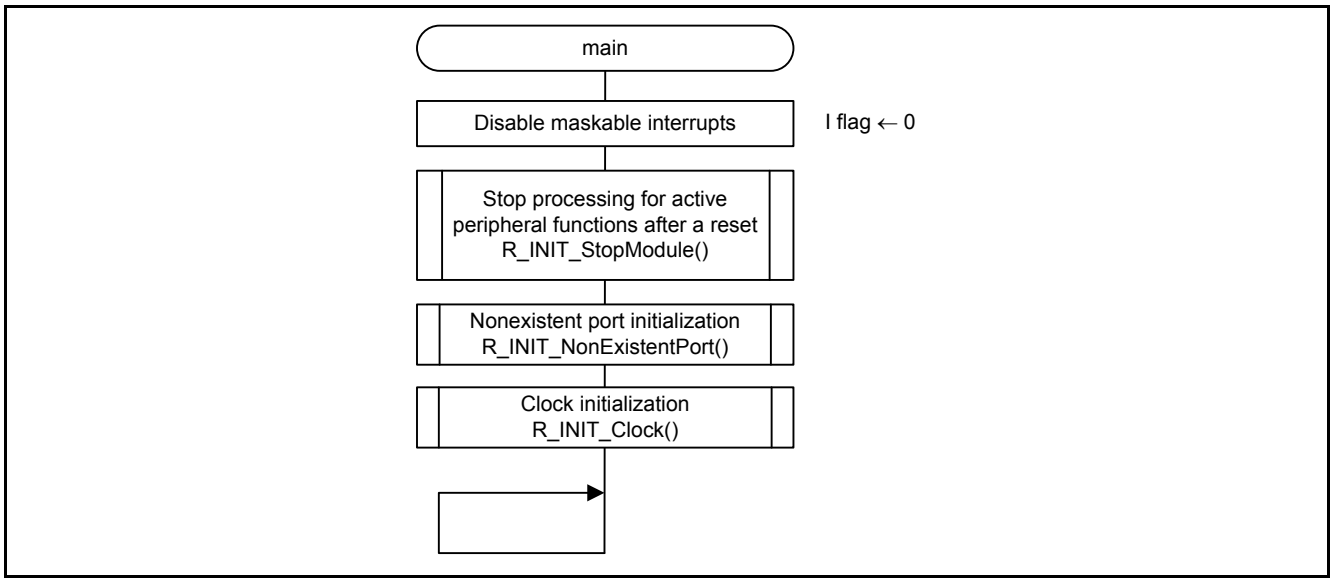
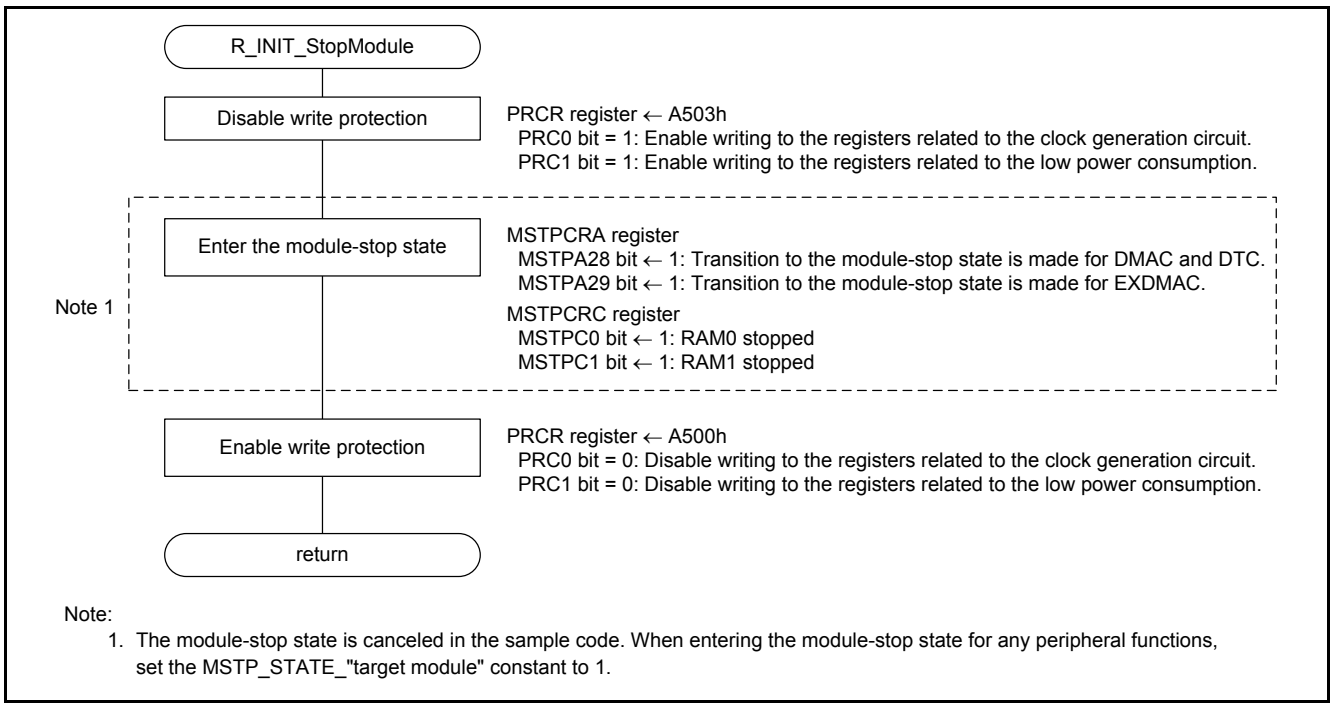


Figure 3.5 Main Processing

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.

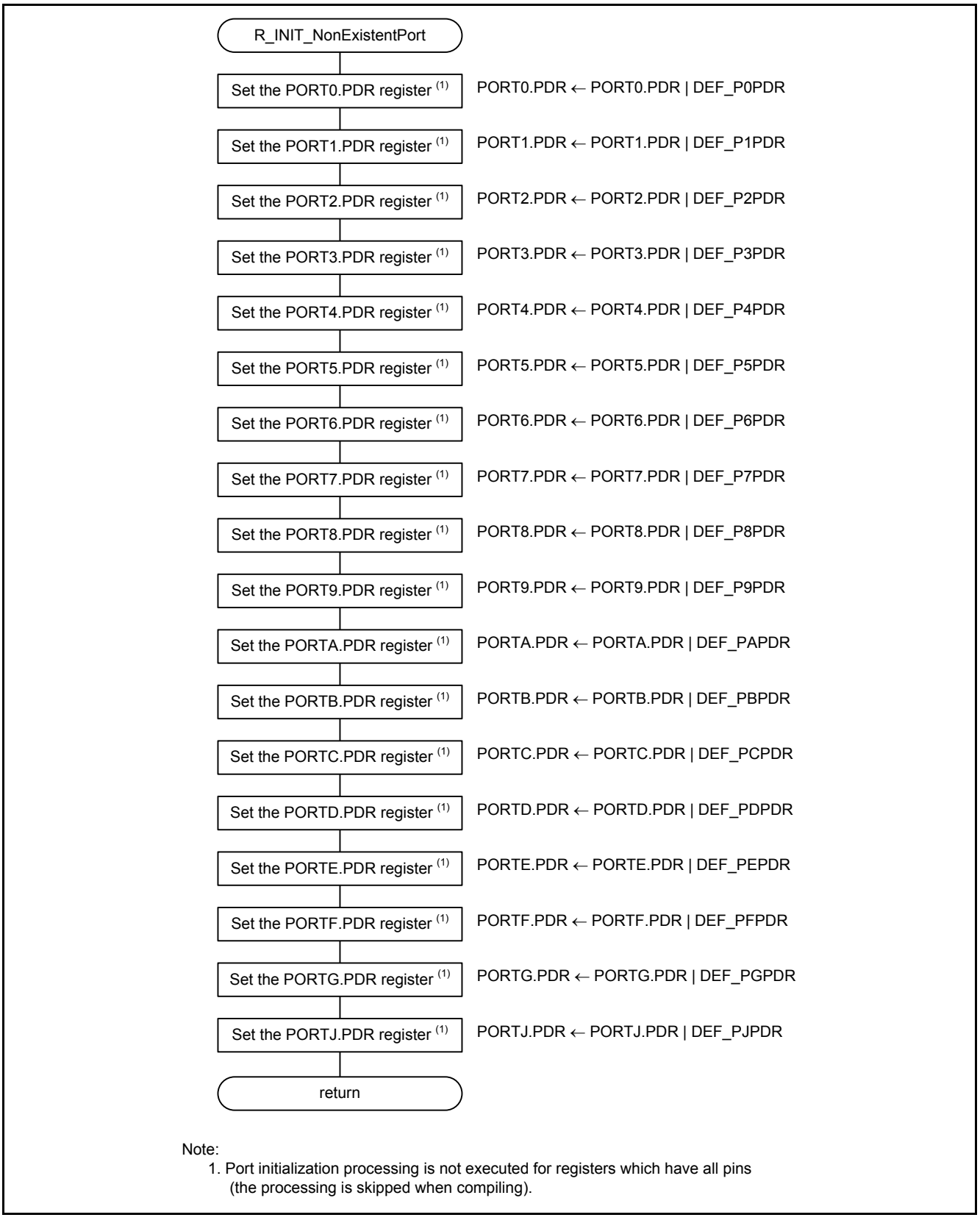


Note:
1. The module-stop state is canceled in the sample code. When entering the module-stop state for any peripheral functions, set the MSTP_STATE_"target module" constant to 1.

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.

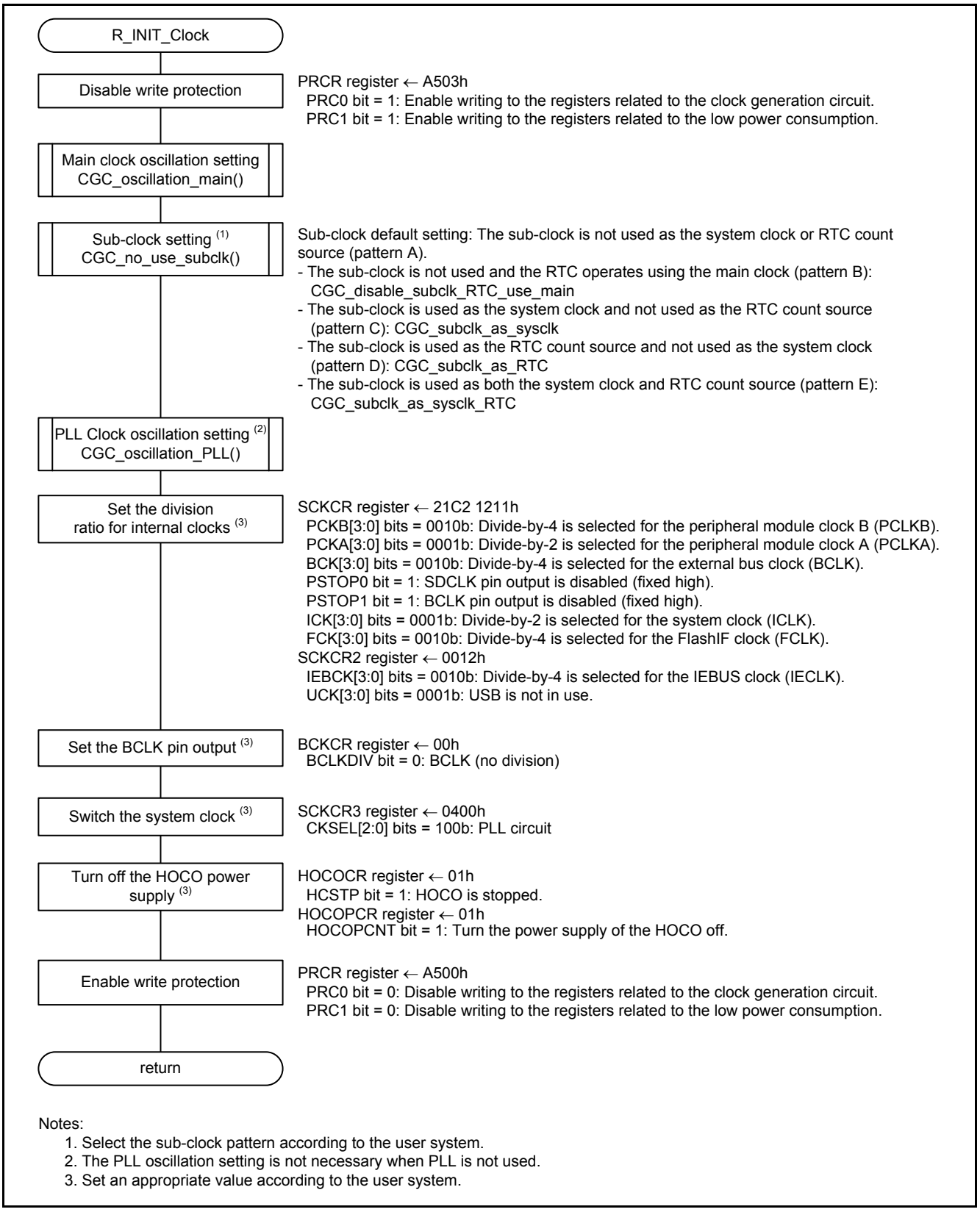


Note:
1. Port initialization processing is not executed for registers which have all pins (the processing is skipped when compiling).

Figure 3.7 Nonexistent Port Initialization

3.9.4 Clock Initialization

Figure 3.8 shows the Clock Initialization.



Notes:
1. Select the sub-clock pattern according to the user system.
2. The PLL oscillation setting is not necessary when PLL is not used.
3. Set an appropriate value according to the user system.

Figure 3.8 Clock Initialization

3.9.5 Main Clock Oscillation Setting

Figure 3.9 shows the Main Clock Oscillation Setting.

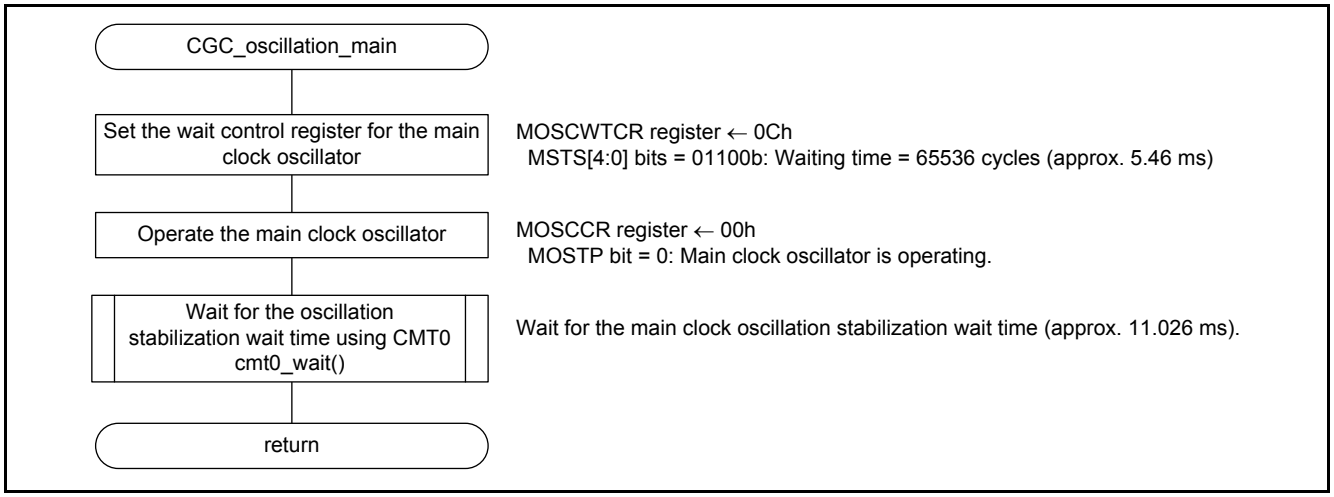


Figure 3.9 Main Clock Oscillation Setting

3.9.6 PLL Clock Oscillation Setting

Figure 3.10 shows the PLL Clock Oscillation Setting.

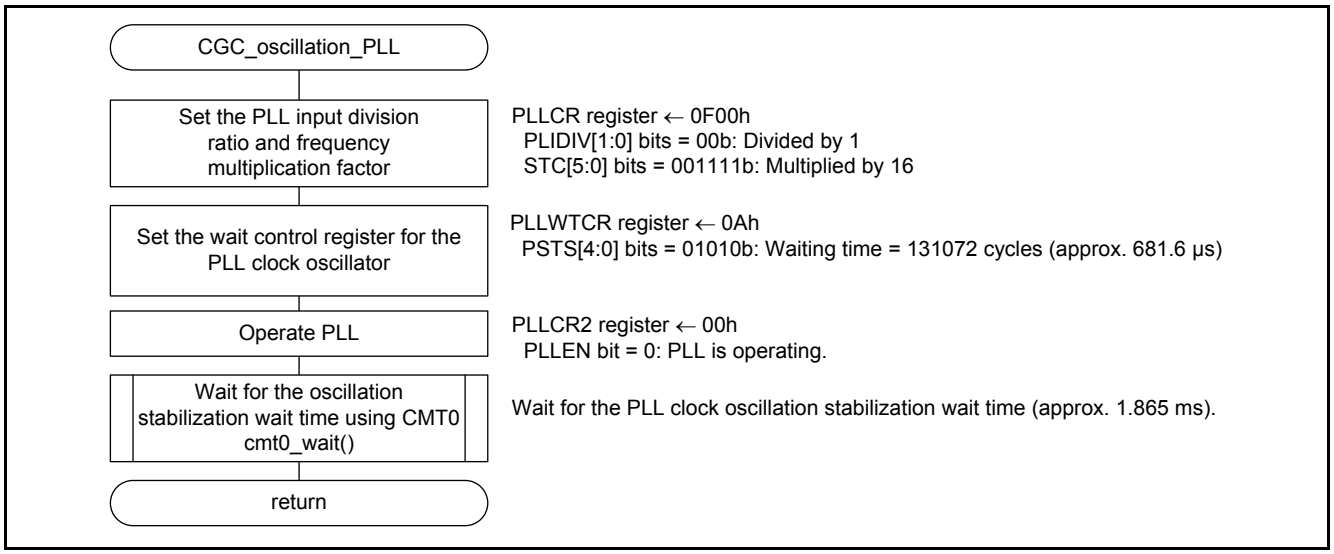


Figure 3.10 PLL Clock Oscillation Setting

3.9.7 HOCO Clock Oscillation Setting

Figure 3.11 shows the HOCO Clock Oscillation Setting.

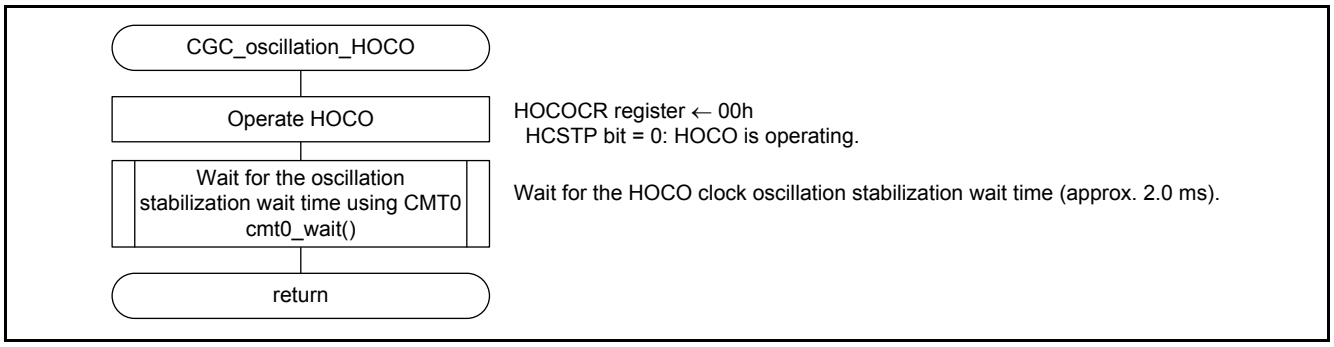


Figure 3.11 HOCO Clock Oscillation Setting

3.9.8 Sub-Clock Setting

Figure 3.12 to Figure 3.16 show the sub-clock setting patterns A to E.

Figure 3.12 shows Sub-Clock Setting Pattern A (when the Sub-Clock is not Used as the System Clock or RTC Count Source).

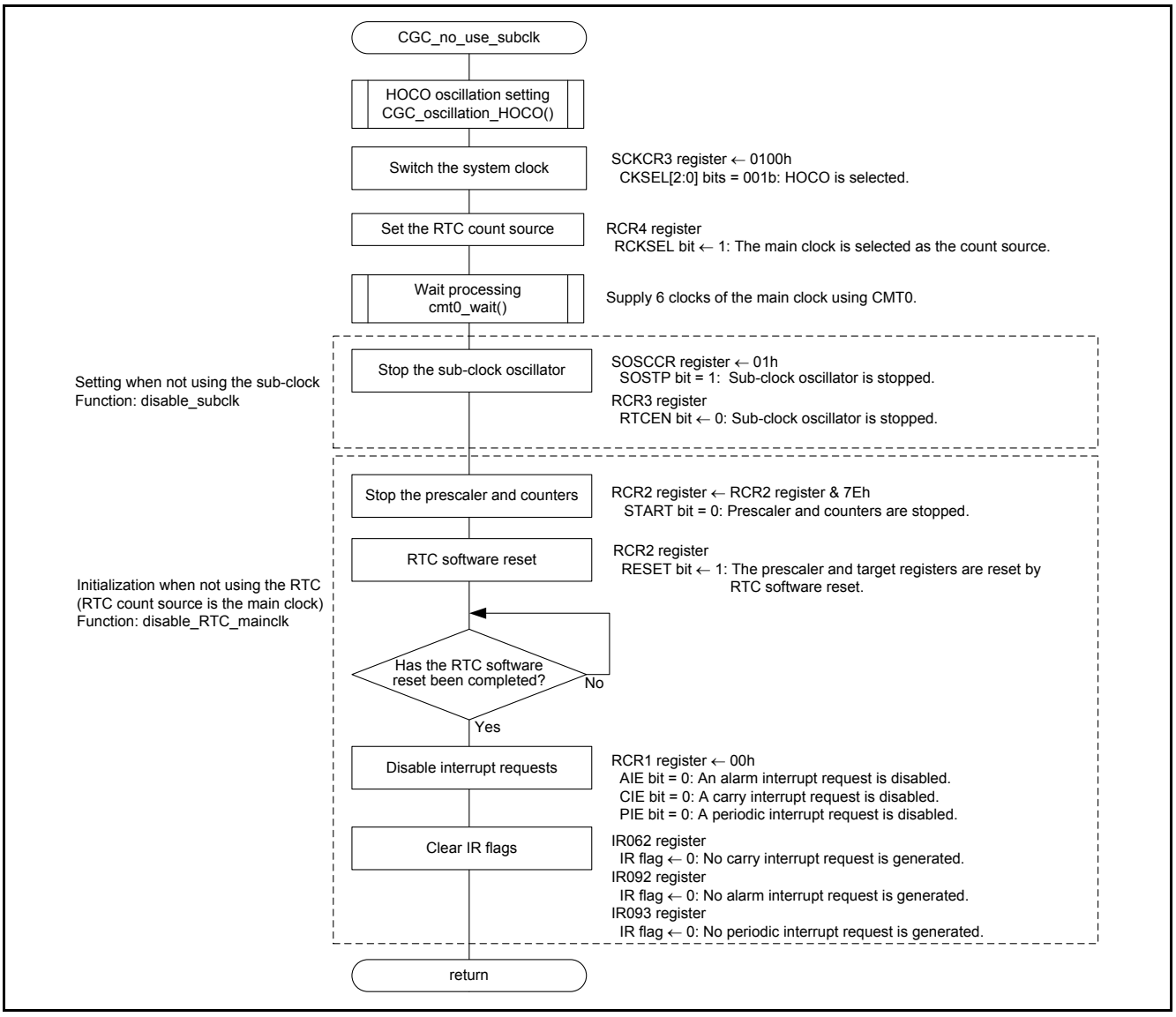
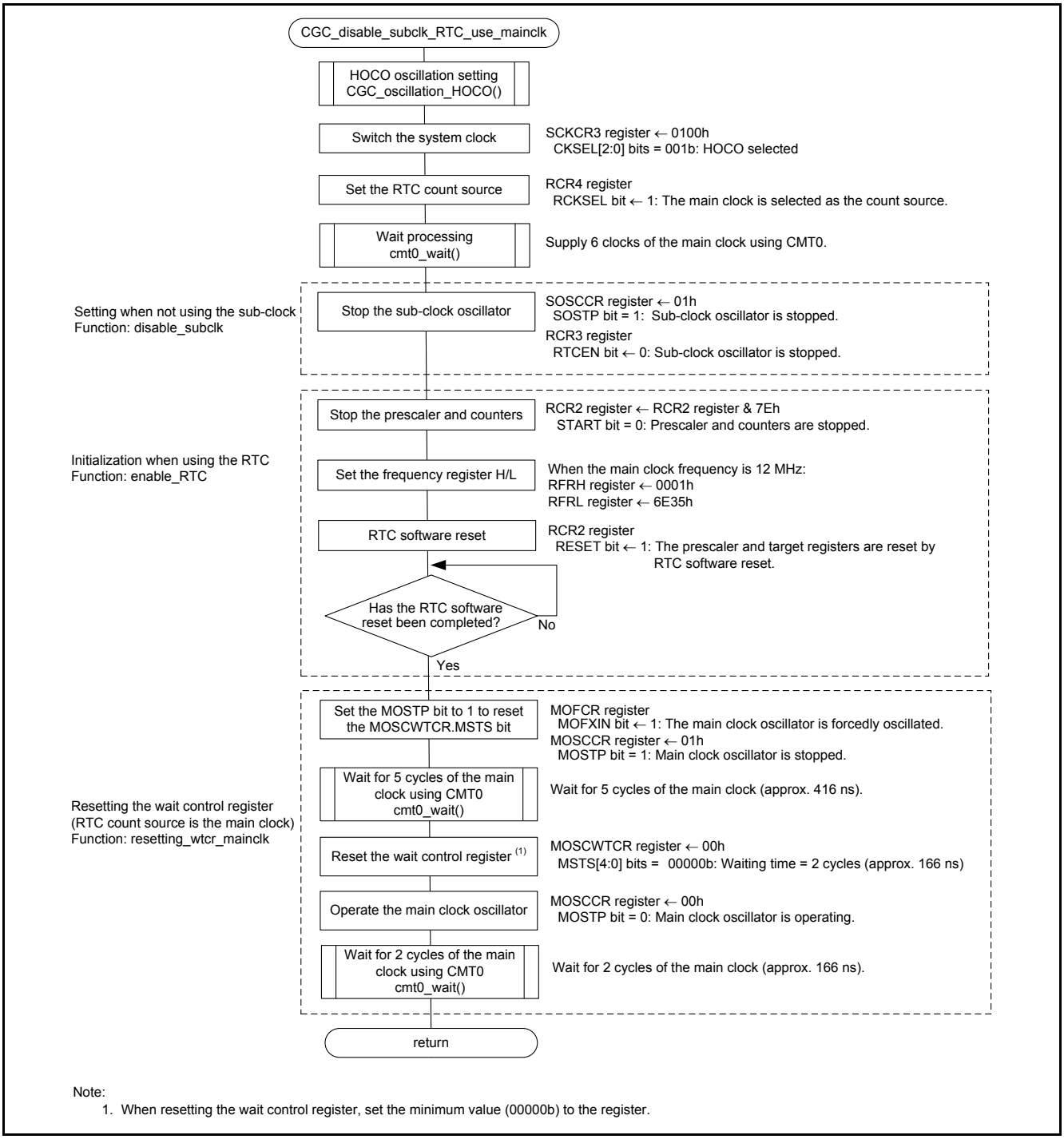


Figure 3.12 Sub-Clock Setting Pattern A (when the Sub-Clock is not Used as the System Clock or RTC Count Source)

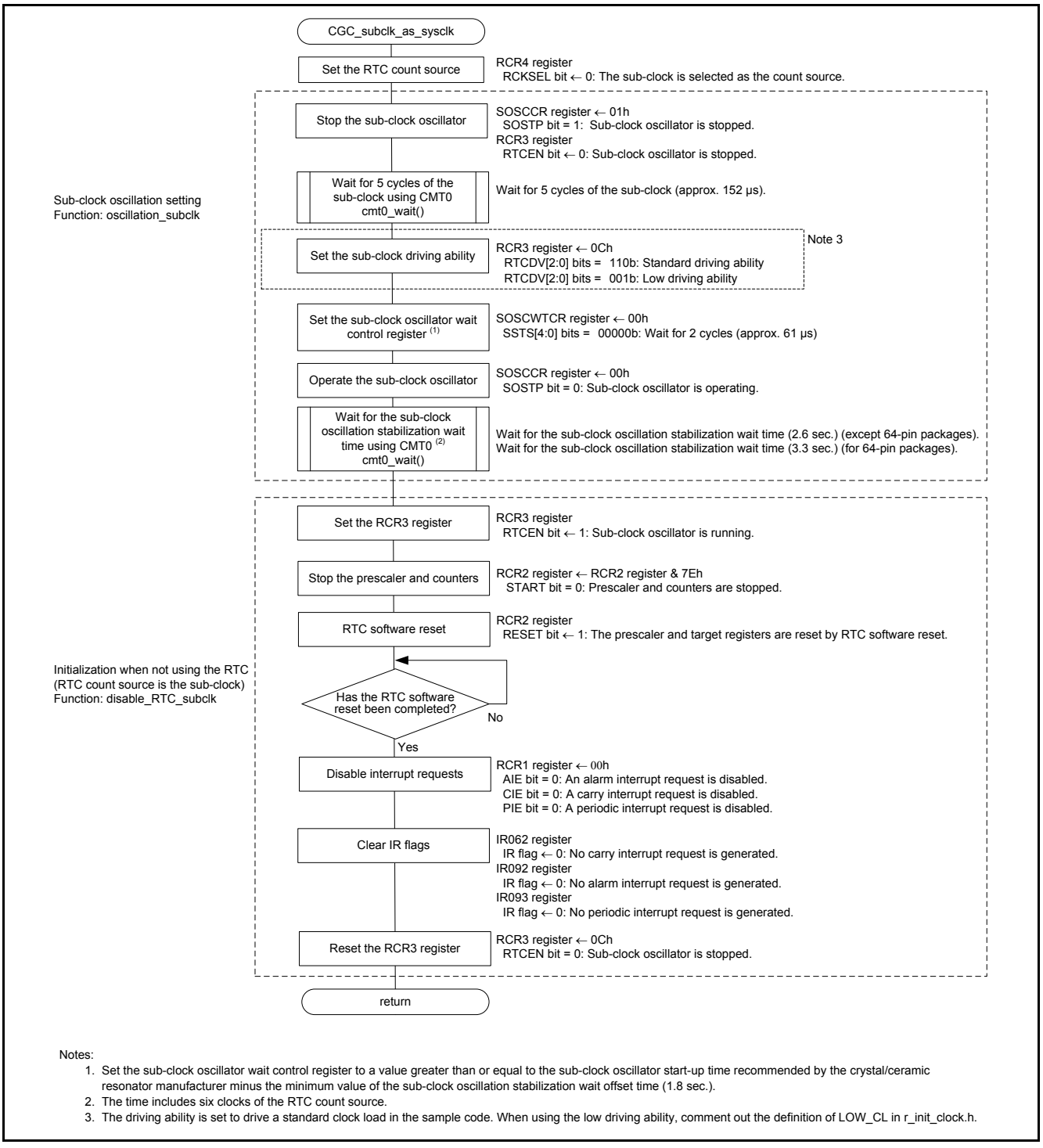
Figure 3.13 shows the Sub-Clock Setting Pattern B (when the Sub-Clock is Stopped and the RTC Operates Using the Main Clock).



Note:
1. When resetting the wait control register, set the minimum value (00000b) to the register.

Figure 3.13 Sub-Clock Setting Pattern B (when the Sub-Clock is Stopped and the RTC Operates Using the Main Clock)

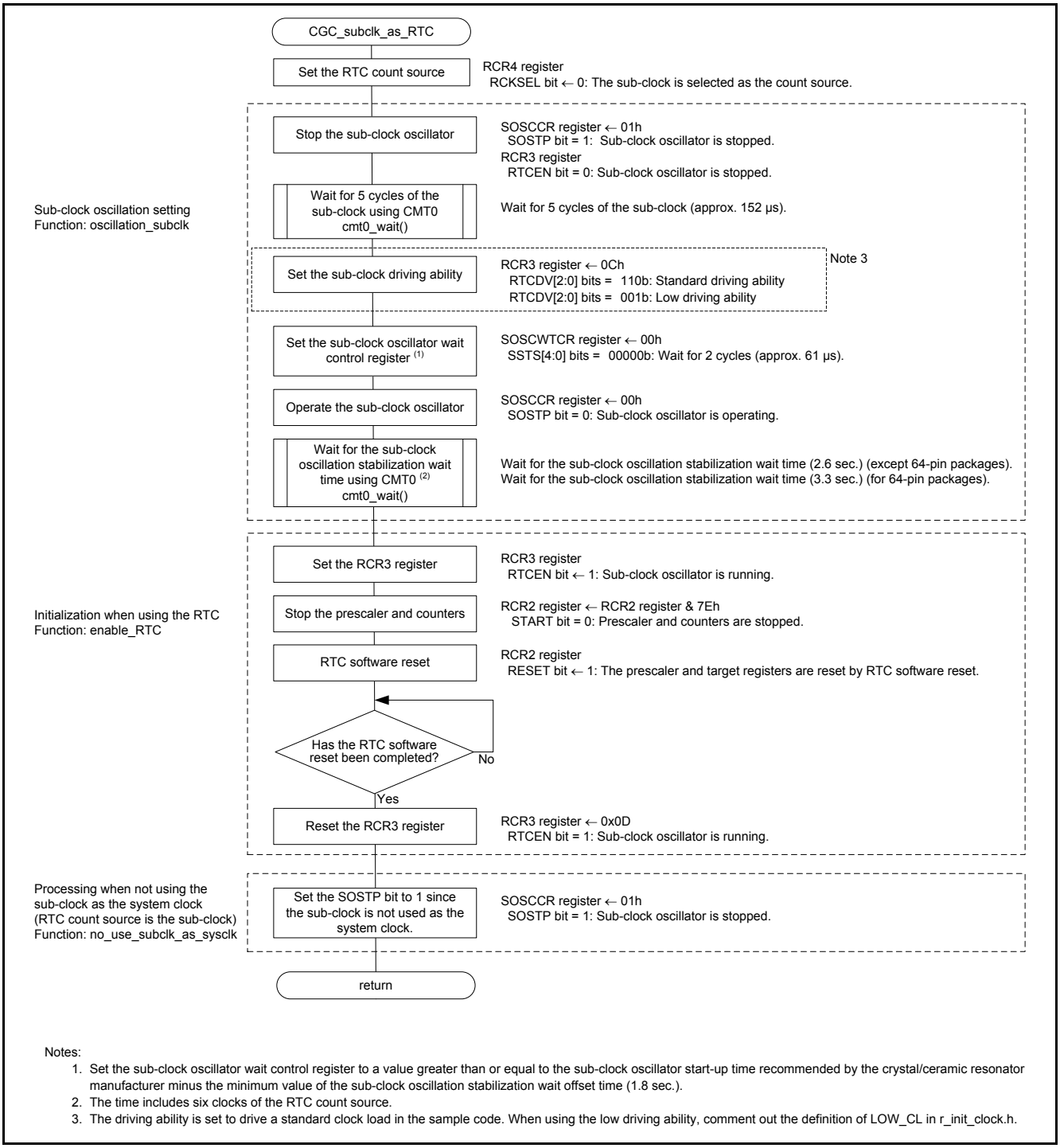
Figure 3.14 shows the Sub-Clock Setting Pattern C (when the Sub-Clock is Used as the System Clock and not Used as the RTC Count Source).



Notes:
1. Set the sub-clock oscillator wait control register to a value greater than or equal to the sub-clock oscillator start-up time recommended by the crystal/ceramic resonator manufacturer minus the minimum value of the sub-clock oscillation stabilization wait offset time (1.8 sec.).
2. The time includes six clocks of the RTC count source.
3. The driving ability is set to drive a standard clock load in the sample code. When using the low driving ability, comment out the definition of LOW_CL in r_init_clock.h.

Figure 3.14 Sub-Clock Setting Pattern C (when the Sub-Clock is Used as the System Clock and not Used as the RTC Count Source)

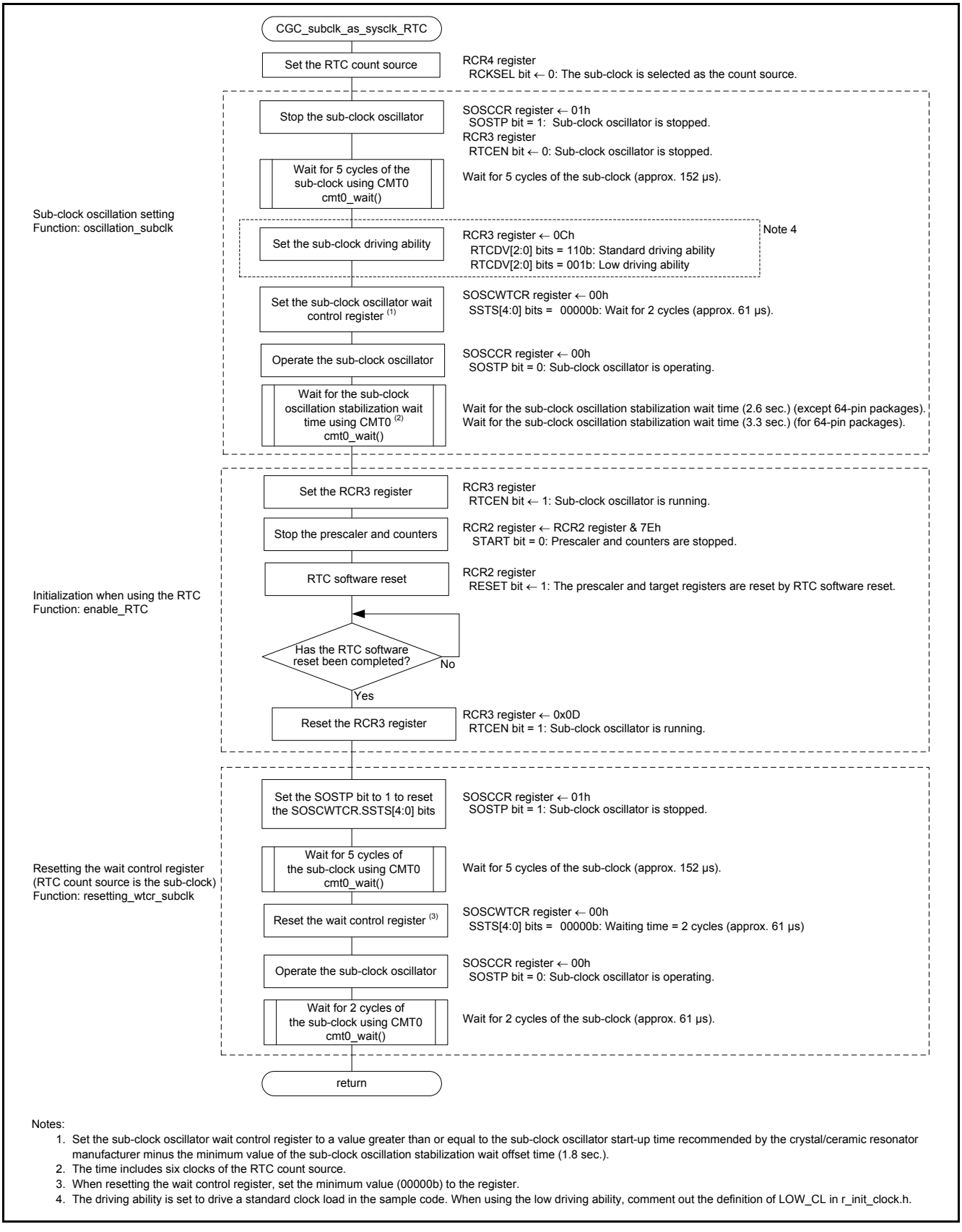
Figure 3.15 shows the Sub-Clock Setting Pattern D (when the Sub-Clock is Used as the RTC Count Source and not Used as the System Clock).



Notes:
1. Set the sub-clock oscillator wait control register to a value greater than or equal to the sub-clock oscillator start-up time recommended by the crystal/ceramic resonator manufacturer minus the minimum value of the sub-clock oscillation stabilization wait offset time (1.8 sec.).
2. The time includes six clocks of the RTC count source.
3. The driving ability is set to drive a standard clock load in the sample code. When using the low driving ability, comment out the definition of LOW_CL in r_init_clock.h.

Figure 3.15 Sub-Clock Setting Pattern D (when the Sub-Clock is Used as the RTC Count Source and not Used as the System Clock)

Figure 3.16 shows the Sub-Clock Setting Pattern E (when the Sub-Clock is Used as Both the System Clock and RTC count source).

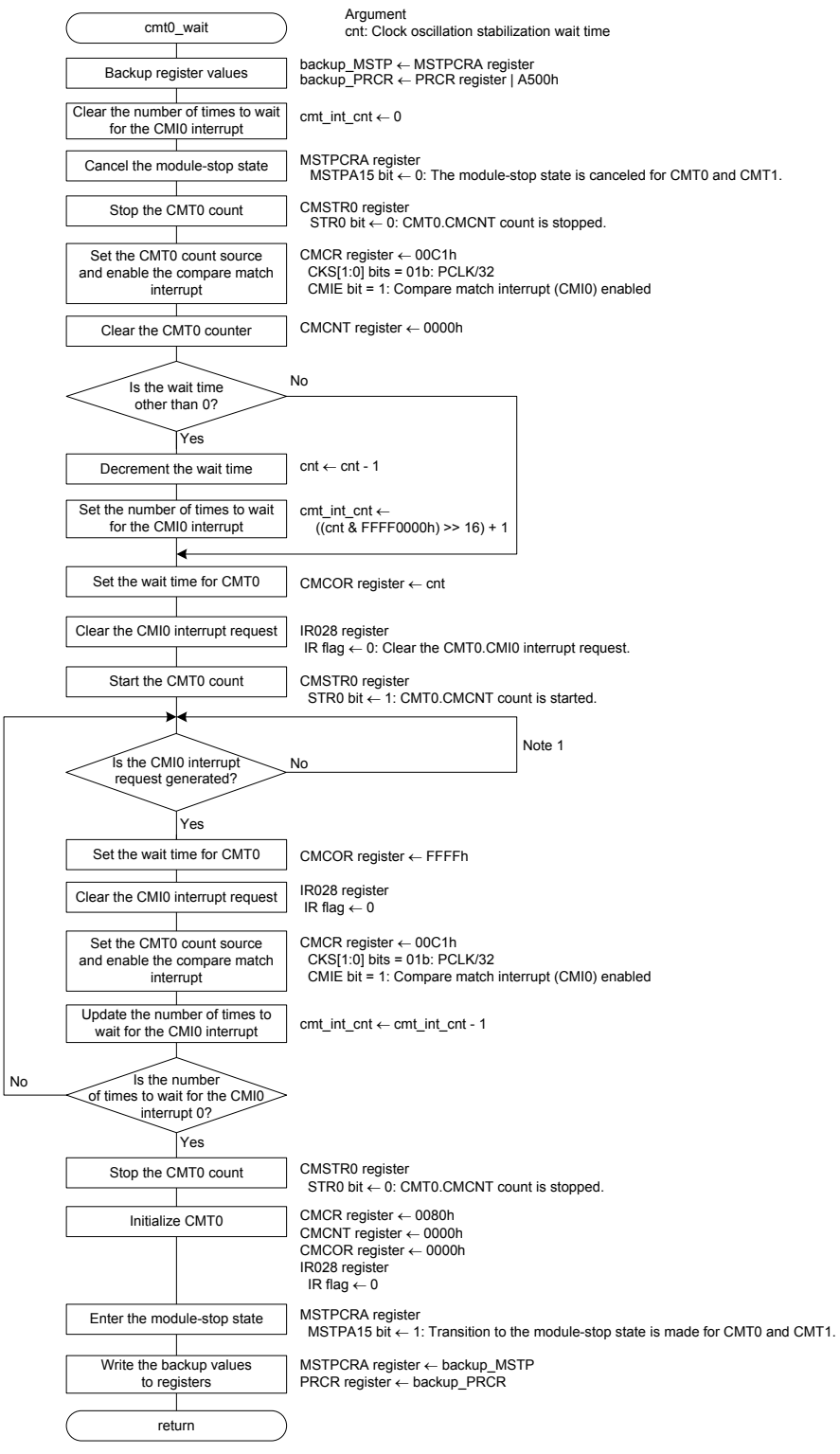


Notes:
1. Set the sub-clock oscillator wait control register to a value greater than or equal to the sub-clock oscillator start-up time recommended by the crystal/ceramic resonator manufacturer minus the minimum value of the sub-clock oscillation stabilization wait offset time (1.8 sec.).
2. The time includes six clocks of the RTC count source.
3. When resetting the wait control register, set the minimum value (00000b) to the register.
4. The driving ability is set to drive a standard clock load in the sample code. When using the low driving ability, comment out the definition of LOW_CL in r_init_clock.h.

Figure 3.16 Sub-Clock Setting Pattern E (when the Sub-Clock is Used as Both the System Clock and RTC count source)

3.9.9 Wait Processing

Figure 3.17 shows the Wait Processing.



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 3.17 Wait Processing

4. Appendices

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).

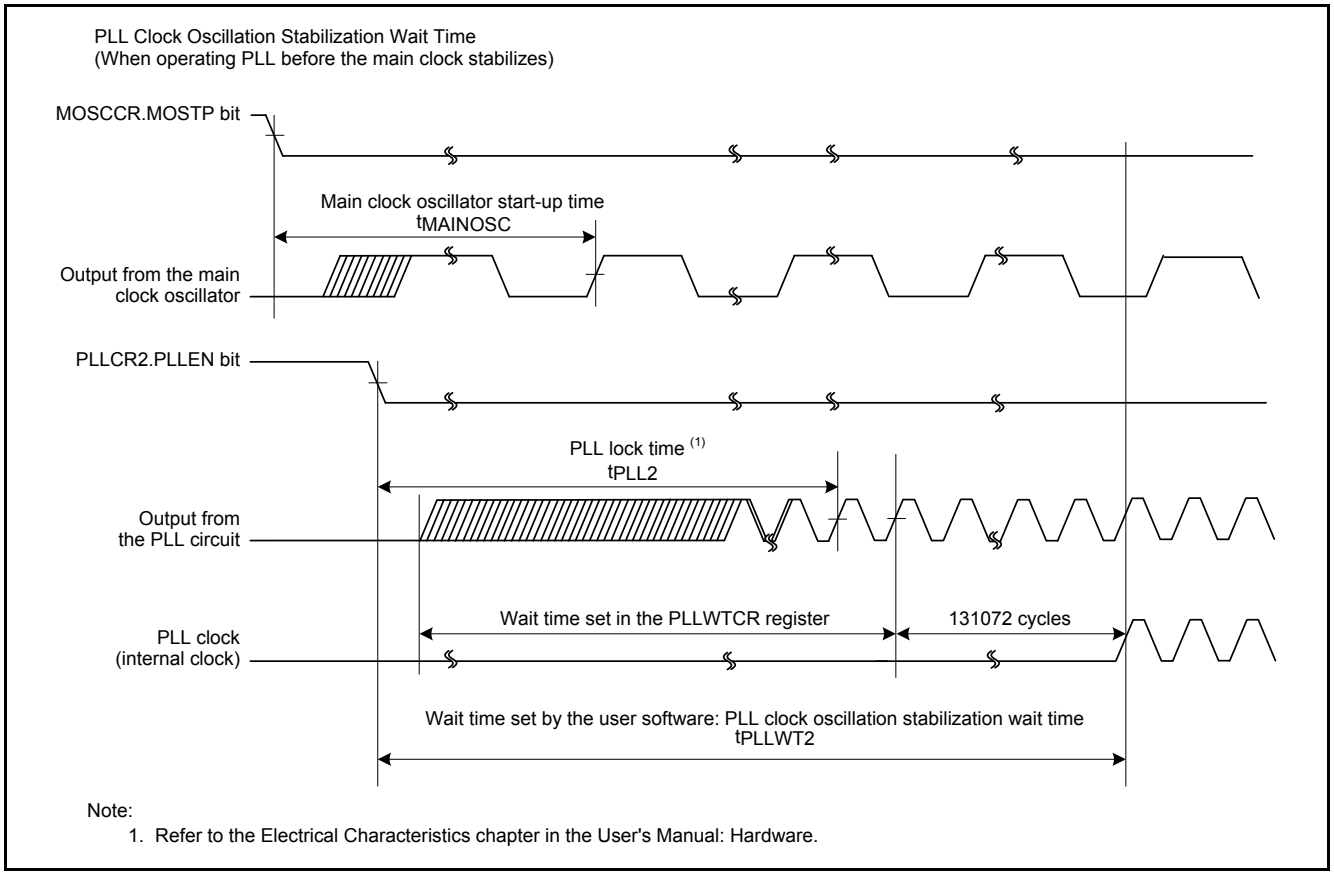


Figure 4.1 PLL Oscillation Stabilization Wait Time (when Operating PLL before the Main Clock Stabilizes)

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)

	Condition of Setting Value and Calculation Method
PLL wait control register (PLLWTCR.PSTS[4:0] bits)	Value greater than or equal to the main clock oscillator start-up time recommended by the crystal/ceramic resonator manufacturer plus tPLL1 (max. 500 µs)
Oscillation stabilization wait time (tPLLWT2)	When n is the wait time selected by the PLLWTCR.PSTS[4:0] bits: $$tMAINOSC + tPLL1 + \frac{n + 131072}{fPLL}$$

4.1.2 When the Sub-Clock Oscillator Start-Up Time is Greater than the Sub-Clock Oscillation Stabilization Wait Offset Time

The sample code assumes that the sub-clock oscillator start-up time (tSUBOSC) is less than the sub-clock oscillation stabilization wait offset time (tSUBOSCWT0).

Figure 4.2 shows the Sub-Clock Oscillation Stabilization Wait Time (when tSUBOSC is Greater than tSUBOSCWT0) and Table 4.2 lists the Setting Value for the Sub-Clock Oscillator Wait Control Register and Oscillation Stabilization Wait Time (when tSUBOSC is Greater than tSUBOSCWT0).

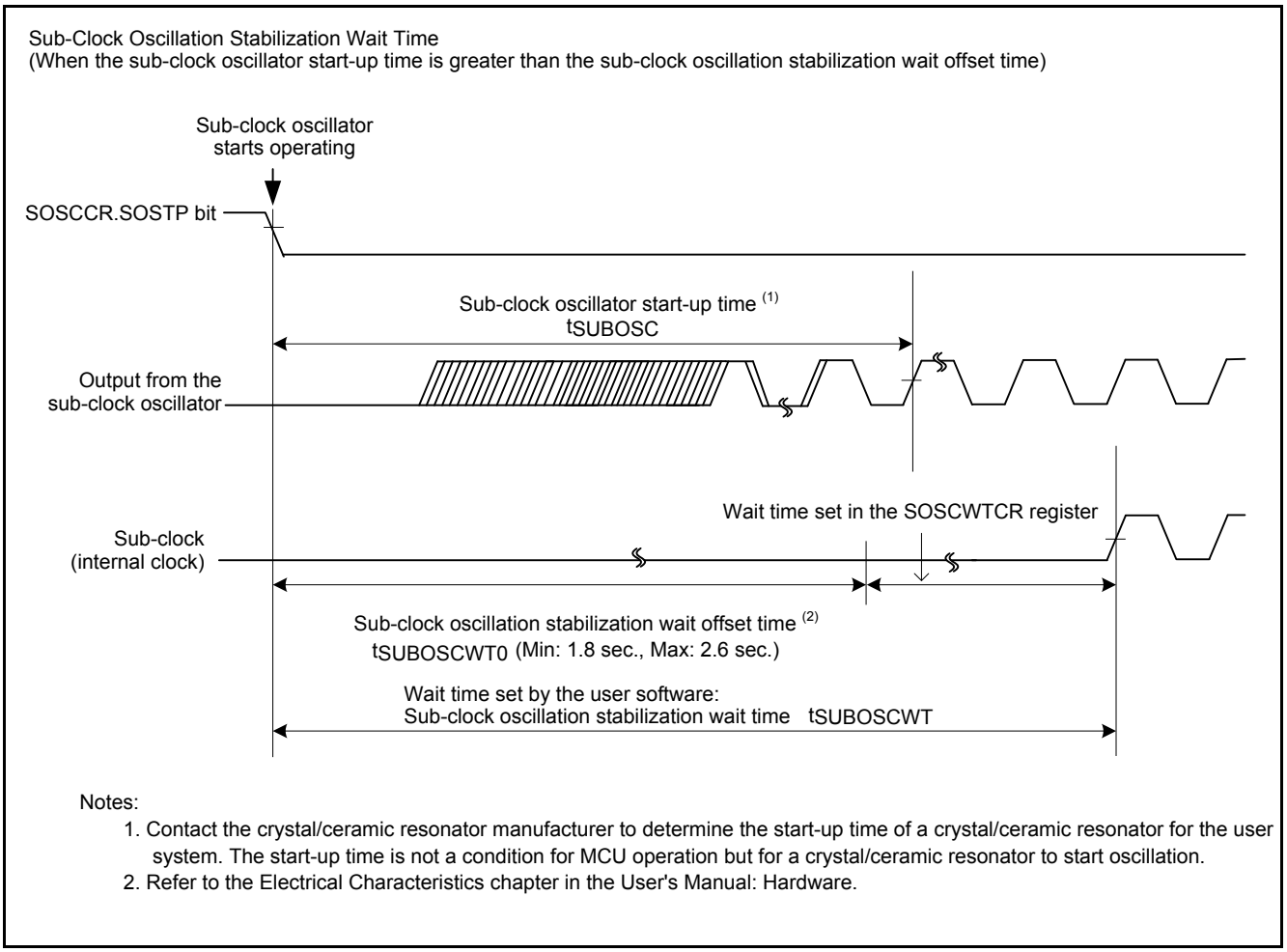


Figure 4.2 Sub-Clock Oscillation Stabilization Wait Time (when tSUBOSC is Greater than tSUBOSCWT0)

Table 4.2 Setting Value for the Sub-Clock Oscillator Wait Control Register and Oscillation Stabilization Wait Time (when tSUBOSC is Greater than tSUBOSCWT0)

	Condition of Setting Value and Calculation Method
Sub-clock oscillator wait control register (SOSCWTCR.SSTS[4:0] bits)	Value greater than or equal to tSUBOSC recommended by the crystal/ceramic resonator manufacturer minus the minimum value of tSUBOSCWT0 (1.8 sec.)
Oscillation stabilization wait time (tSUBOSCWT)	When n is the wait time selected in the SOSCWTCR.SSTS[4:0] bits: $tSUBOSC + \frac{n}{fSUB}$

5. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

6. Reference Documents

User's Manual: Hardware

RX63N Group, RX631 Group User's Manual: Hardware Rev.1.70 (R01UH0041EJ)
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	RX63N Group, RX631 Group Application Note Initial Setting

Rev.	Date	Description	
		Page	Summary
1.00	Mar. 1, 2013	—	First edition issued
1.10	Jan. 6, 2014	—	- Covered 64-pin and 48-pin packages. - Modified the wait processing with CMT0 to support 32-bit argument.
		5	Table 2.1 Operation Confirmation Conditions: - Operating frequencies: Added the description regarding 48-pin version. - iodefine.h version: Updated the version to 1.6A
		8	Table 3.3 Nonexistent Ports (2/2): Added the nonexistent port list for 48-pin and 64-pin packages.
		9	Table 3.4 Clock Setting Procedure: Added the description of a 48-pin package in step 2.
		13	3.3.3.3 Sub-Clock Oscillation Stabilization Wait Time (Except 64-Pin Packages): Changed the specifications to support packages except 64-pin packages.
		14	3.3.3.4 Sub-Clock Oscillation Stabilization Wait Time (for 64-Pin Packages): Added.
		16	Table 3.12 Constants Used in the Sample Code: - Added the WAIT_TIME_FOR_SUB_OSCILLATION constant for 64-pin package. - Added the PATTERN_48 constant. - Added the LOW_CL constant.
		19	Table 3.17 Constants when a 60-Pin Package is Used (PIN_SIZE=60): Added.
		19	Table 3.18 Constants when a 48-Pin Package is Used (PIN_SIZE=48): Added.
		20	3.7 Functions: Added the description of a 48-pin package to the CGC_no_use_subclk function.
		33 to 35	Figure 3.14 Sub-Clock Setting Pattern C, Figure 3.15 Sub-Clock Setting Pattern D, and Figure 3.16 Sub-Clock Setting Pattern E: Added descriptions regarding the low driving ability and 64-pin packages.
		36	3.9.9 Wait Processing: Revised.

General Precautions in the Handling of MPU/MCU Products

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

1. Handling of Unused Pins

   Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

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

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 LanGao Rd., Putuo District, Shanghai, China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 3.0