RX23E-A Group

Initial Settings Example

Summary

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

Target Devices

RX23E-A Group 48-pin version, ROM capacity: 128 KB to 256 KB

RX23E-A Group 40-pin version, ROM capacity: 128 KB to 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 others. These include the DMAC, DTC and RAM. 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

Port direction registers which have nonexistent ports need to be specified with determined values. In the sample code, initial values are set for port direction registers in 48-pin products. Change the values according to the product used.

1.3 Setting Clocks

1.3.1 Overview

Clocks are configured in the following steps:

1. Main clock setting
2. PLL clock setting
3. HOCO clock setting
4. 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. 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 time are calculated using values listed in Table 1.1.

Table 1.1 Clock Specifications Used in Sample Code

Clock	Oscillation Frequency	Oscillation Stabilization Time	Remarks
Crystal/ceramic resonator for the main clock	8 MHz	4.2 ms*2	Crystal used
PLL clock	32 MHz	74.4 μs*3	
HOCO clock	32 MHz*1	41.3 μs*3	

Notes: 1. The clock 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. Contact the crystal/ceramic resonator manufacturer to evaluate the user system and provide an appropriate oscillation stabilization time.
3. Refer to the Electrical Characteristics in the User's Manual: Hardware.

1.3.3 Selecting Clocks

In the sample code, users can select the system clock source, whether clocks are oscillating or stopped, and other settings by changing the constants defined in r_init_clock.h. Refer to Table 3.6 for constants that can be changed.

Table 1.2 lists examples of clock selections.

Table 1.2 Examples of Clock Selections

No.		**1**	**2**	**3**
System clock		PLL	Main clock	HOCO
PLL clock		Oscillating	Stopped	Stopped
Main clock		Oscillating	Oscillating	Stopped
HOCO clock		Stopped	Stopped	Oscillating
Operating power control mode		High-speed operating mode	High-speed operating mode	High-speed operating mode
Constants	SEL_SYSCLK	CLK_PLL	CLK_MAIN	CLK_HOCO
	SEL_PLL	B_USE	B_NOT_USE	B_NOT_USE
	SEL_MAIN	B_USE	B_USE	B_NOT_USE
	SEL_HOCO	B_NOT_USE	B_NOT_USE	B_USE
	SEL_OPCM	OPCM_HIGH	OPCM_HIGH	OPCM_HIGH

2. Operation Confirmation Conditions

The sample code has been run and confirmed under the conditions below.

Table 2.1 Operation Confirmation Conditions

Item		Contents
MCU used		R5F523E6ADFL (RX23E-A Group)
Operating frequencies	When the PLL clock is selected as the system clock	• Main clock: 8 MHz • PLL: 32 MHz (main clock × 1/2 × 8) • LOCO: 4 MHz • System clock (ICLK): 32 MHz (PLL × 1) • Peripheral module clock A (PCLKA): 32 MHz (PLL × 1) • Peripheral module clock B (PCLKB): 32 MHz (PLL × 1) • Peripheral module clock D (PCLKD): 32 MHz (PLL × 1) • FlashIF clock (FCLK): 32 MHz (PLL × 1)
	When the main clock is selected as the system clock	• Main clock: 8 MHz • LOCO: 4 MHz • System clock (ICLK): 8 MHz (main clock × 1) • Peripheral module clock A (PCLKA): 8 MHz (main clock × 1) • Peripheral module clock B (PCLKB): 8 MHz (main clock × 1) • Peripheral module clock D (PCLKD): 8 MHz (main clock × 1) • FlashIF clock (FCLK): 8 MHz (main clock × 1)
	When the HOCO clock is selected as the system clock	• LOCO: 4 MHz • HOCO: 32 MHz • System clock (ICLK): 32 MHz (HOCO × 1) • Peripheral module clock A (PCLKA): 32 MHz (HOCO × 1) • Peripheral module clock B (PCLKB): 32 MHz (HOCO × 1) • Peripheral module clock D (PCLKD): 32 MHz (HOCO × 1) • FlashIF clock (FCLK): 32 MHz (HOCO × 1)
Operating voltage		3.3 V
Integrated development environment		Renesas Electronics Corporation e^2 studio version 7.6.0
C compiler		Renesas Electronics Corporation C/C++ Compiler Package for RX Family V.3.01.00
		Compile options The default setting is used in the integrated development environment.
iodefine.h version		V1.00
Endian order		Little endian
Operating mode		Single-chip mode
Processor mode		Supervisor mode
Sample code version		Version 1.00
Board used		Renesas Solution Starter Kit for RX23E-A (product No.: RTK0ESXB10C00001BE)

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, the MSTP_STATE_“target module” constant is set 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)
RAM	MSTPCRC.MSTPC0 bit		

3.2 Nonexistent Port Initialization

3.2.1 Overview

Bits corresponding to the nonexistent ports in the PRD register are set to 1. 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 I/O select bits for nonexistent ports to 1, and set the output data store bits for nonexistent ports to 0.

3.2.2 Selecting the Number of Pins

The number of pins in the sample code is set for the 48-pin package (PIN_SIZE = 48). This application note covers 48-pin and 40-pin packages. When using products other than the 48 pin-package, change PIN_SIZE in r_init_port_initialize.h to the number of pins on the package used.

3.3 Clock Settings

3.3.1 Clock Setting Procedure

Table 3.2 lists the clock setting procedure with each processing and setting in the sample code. In the sample code, the main clock and PLL are operating, and the HOCO is stopped.

Table 3.2 Clock Setting Procedure

Step	Processing	Details		Setting in the Sample Code
1	Main clock setting*[1]	Not used	No setting is required.	Main clock is used.
		Used	The main clock drive capability is set, the MOSCWTCR register is set with a wait time until the main clock output is provided to the internal clock, and then the main clock oscillation is enabled. Then wait for the oscillation stabilization time is processed.	
2	PLL clock setting*[1]	Not used	No setting is required.	PLL clock is used.
		Used	The PLL input frequency division ratio and frequency multiplication factor are set, and PLL clock oscillation is enabled. Then wait for the oscillation stabilization time is processed.	
3	HOCO clock setting *[1]	Not used	No setting is required.	HOCO clock is not used.
		Used	The HOCO clock oscillation is enabled. Then wait for the oscillation stabilization time is processed.	
4	Operating power control mode setting	The operating power control mode is set according to the operating frequency and operating voltage in the user system.		High-speed operating mode is set.
5	Clock division ratio setting	The clock division ratio is changed.		FCLK, ICLK, PCLKA, PCLKB, PCLKD: $\times$ 1
6	System clock switching	The system clock is switched according to the user system.		Switched to the PLL.

Note: 1. When selecting each clock usage, change the appropriate constant in r_init_clock.h as required.

3.4 Section Configuration

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

For the method of adding, changing, and deleting sections, refer to the latest version of RX Family: CC-RX Compiler User's Manual.

Table 3.3 Section Information Changed in Sample Code

Section Name	Change	Address	Description
End_of_RAM	Added	0000 7FFCh*1	On-chip RAM end address

Note: 1. The on-chip RAM capacity differs depending on the product configuration. Change this address to match the on-chip RAM capacity of your device.

3.5 File Composition

Table 3.4 lists the files used in the sample code. Files generated by the integrated development environment should not be listed in this table.

Table 3.4 Files Used in 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_port_initialize.c	Nonexistent port initialization	
r_init_port_initialize.h	Header file for r_init_port_initialize.c	
r_init_clock.c	Clock initialization	
r_init_clock.h	Header file for r_init_clock.c	

3.6 Option-Setting Memory

Table 3.5 lists the option-setting memory configured in the sample code. When necessary, set a value suited to the user system.

Table 3.5 Option-Setting Memory Configured in Sample Code

Symbol	Address	Setting Value	Contents
OFS0	FFFF FF8Fh to FFFF FF8Ch	FFFF FFFFh	The IWDT is stopped after a reset.
OFS1	FFFF FF8Bh to FFFF FF88h	FFFF FFFFh	Fast startup time at power-on is disabled. The voltage monitor 0 reset is disabled after a reset. HOCO oscillation is disabled after a reset.
MDE	FFFF FF83h to FFFF FF80h	FFFF FFFFh	Little endian

3.7 Constants

Table 3.6 lists the constants used in the sample code, which can be changed by users. Table 3.7 lists the constants used in the sample code, which cannot be changed by users. Table 3.8 lists the constants for Initialization of the Port Direction Register.

Table 3.6 Constants Used in Sample Code (Users can change the constants listed in this table.)

Constant Name	Setting Value	Contents
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	8,000,000L	Oscillation frequency of crystal/ceramic resonator for the main clock (Hz)
REG_MOFCR*1	00h	Setting for the drive capability of the main clock oscillator (setting value in the MOFCR register)
REG_MOSCWTCR*1	06h	Setting value in the main clock wait control register
SEL_HOCO	B_NOT_USE	Selection of the HOCO clock operation: • B_USE: Used (HOCO clock oscillating) • B_NOT_USE: Not used (HOCO clock stopped)
SEL_PLL	B_ USE	Selection of the PLL clock operation: • B_USE: Used (PLL clock oscillating) • B_NOT_USE: Not used (PLL clock stopped)
REG_PLLCR	0F01h	PLL input frequency division ratio and frequency multiplication factor settings (setting value in the PLLCR register)
SEL_SYSCLK*1	CLK_PLL	Clock source selection for the system clock • CLK_HOCO: HOCO clock • CLK_MAIN: Main clock • CLK_PLL: PLL clock
REG_SCKCR	0000 0000h	Setting for the internal clock division ratio (setting value in the SCKCR register)
SEL_OPCM*1	OPCM_HIGH	Selection of the operating power control mode*4 • OPCM_HIGH: High-speed operating mode • OPCM_MID: Middle-speed operating mode
MSTP_STATE_DMACDTC*2	MODULE_STOP _DISABLE	Selection of the module-stop state for DMAC/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 RAM • MODULE_STOP_DISABLE: Operating • MODULE_STOP_ENABLE: Stopped
PIN_SIZE*3	48	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_ port_initialize.h according to the user system.
4. The ranges of the operating frequency and operating voltage differ depending on operating modes. Refer to the User's Manual: Hardware for details.

Table 3.7 Constants Used in Sample Code (Users cannot change the constants listed in this table.)

Constant Name	Setting Value	Contents
B_NOT_USE	0	Not used
B_USE	1	Used
CLK_MAIN	0200h	Clock source: Main clock
CLK_PLL	0400h	Clock source: PLL
CLK_HOCO	0100h	Clock source: HOCO
OPCM_MID	02h	Operating power control mode: Middle-speed operating mode
OPCM_HIGH	00h	Operating power control mode: High-speed operating mode
OPCM_DEFAULT	OPCM_MID	Operating mode after reset cancellation
MODULE_STOP_ENABLE	1	Transition to the module stop-state is made
MODULE_STOP_DISABLE	0	Module stop-state is canceled

Table 3.8 Constants for Initialization of the Port Direction Register

Constant Name	Setting Value		Contents
	48-Pin Package	40-Pin Package	
DEF_P1PDR	0Fh	0Fh	Setting value for the port P1 direction register
DEF_P2PDR	3Fh	3Fh	Setting value for the port P2 direction register
DEF_P3PDR	1Ch	1Ch	Setting value for the port P3 direction register
DEF_PBPDR	FCh	FCh	Setting value for the port PB direction register
DEF_PCPDR	0Fh	CFh	Setting value for the port PC direction register
DEF_PHPDR	F0h	FCh	Setting value for the port PH direction register

3.8 Functions

Table 3.9 lists the functions used in the sample code.

Table 3.9 Functions Used in Sample Code

Function Name	Outline
Main	Main processing
R_INIT_StopModule	Stop processing for active peripheral functions after a reset
R_INIT_Port_Initialize	Nonexistent port initialization
R_INIT_Clock	Clock initialization
cgc_oscillation_main	Main clock oscillation setting
cgc_oscillation_hoco	HOCO clock oscillation setting
cgc_oscillation_pll	PLL clock oscillation setting

3.9 Function Specifications

The following tables list the sample code function specifications.

Main	
Outline	Main processing
Header	None
Declaration	void main(void)
Description	Calls 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	Configures 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_Port_Initialize	
Outline	Nonexistent port initialization
Header	r_init_port_initialize.h
Declaration	void R_INIT_Port_Initialize(void)
Description	Initializes port direction registers for ports that do not exist in products.
Arguments	None
Return Value	None
Remarks	The pin count setting in the sample code is for a 48-pin package (PIN_SIZE = 48). After this function is called, when writing in byte units to PDR registers or PODR registers which have nonexistent ports, set the I/O select bits for the nonexistent ports in the PDR registers to 1 and the output data store bits for the nonexistent ports in the PODR registers to 0.

R_INIT_Clock	
Outline	Clock initialization
Header	r_init_clock.h
Declaration	void R_INIT_Clock(void)
Description	Initializes the clock.
Arguments	None
Return Value	None
Remarks	The sample code selects processing which uses the PLL as the system clock.

cgc_oscillation_main	
Outline	Main clock oscillation setting
Header	r_init_clock.h
Declaration	void cgc_oscillation_main(void)
Description	Sets the main clock drive capability, sets the MOSCWTCR register, and enables main clock oscillation. Then waits for the main clock oscillation stabilization time.
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	Enables the HOCO oscillation. Then waits for the HOCO clock oscillation stabilization time.
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	Sets the PLL input frequency division ratio and frequency multiplication factor, and enables PLL clock oscillation. Then waits for the PLL oscillation stabilization time.
Arguments	None
Return Value	None

3.10 Flowcharts

3.10.1 Main Processing

Figure 3.1 shows the main processing.

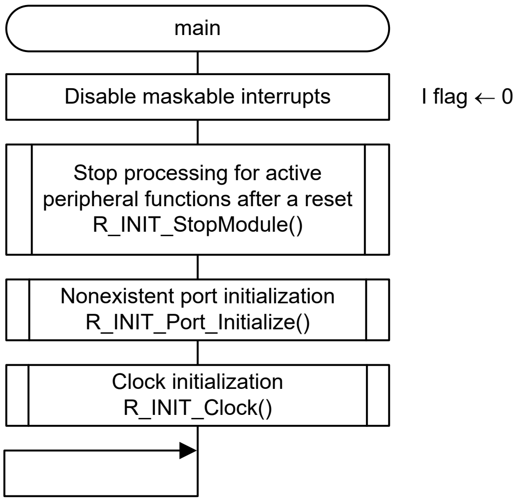
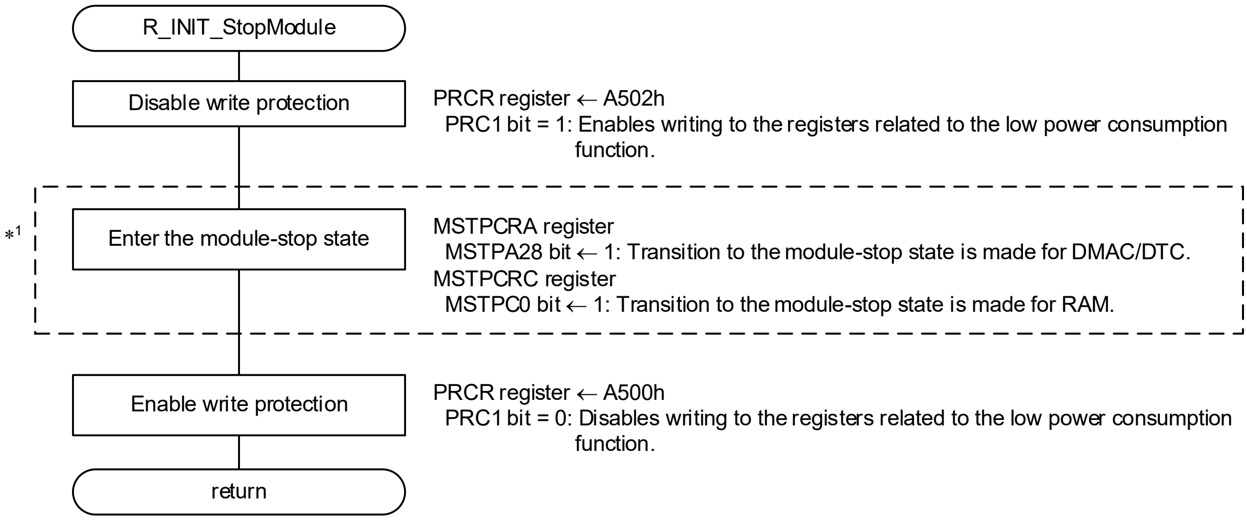


Figure 3.1 Main Processing

3.10.2 Stop Processing for Active Peripheral Functions after a Reset

Figure 3.2 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 #define MSTP_STATE_"target module" constant to 1.

Figure 3.2 Stop Processing for Active Peripheral Functions after a Reset

3.10.3 Nonexistent Port Initialization

Figure 3.3 shows the nonexistent port initialization.

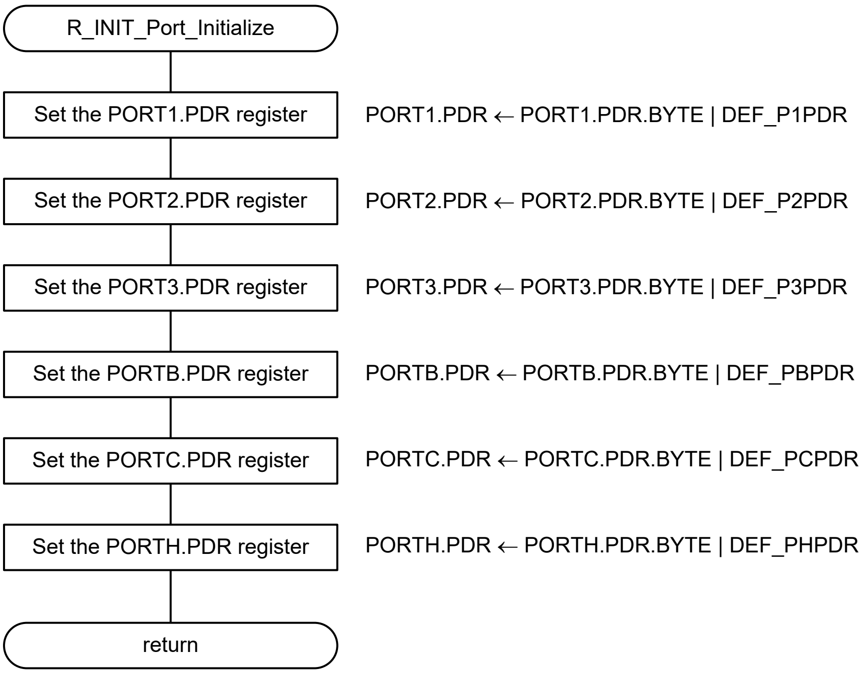
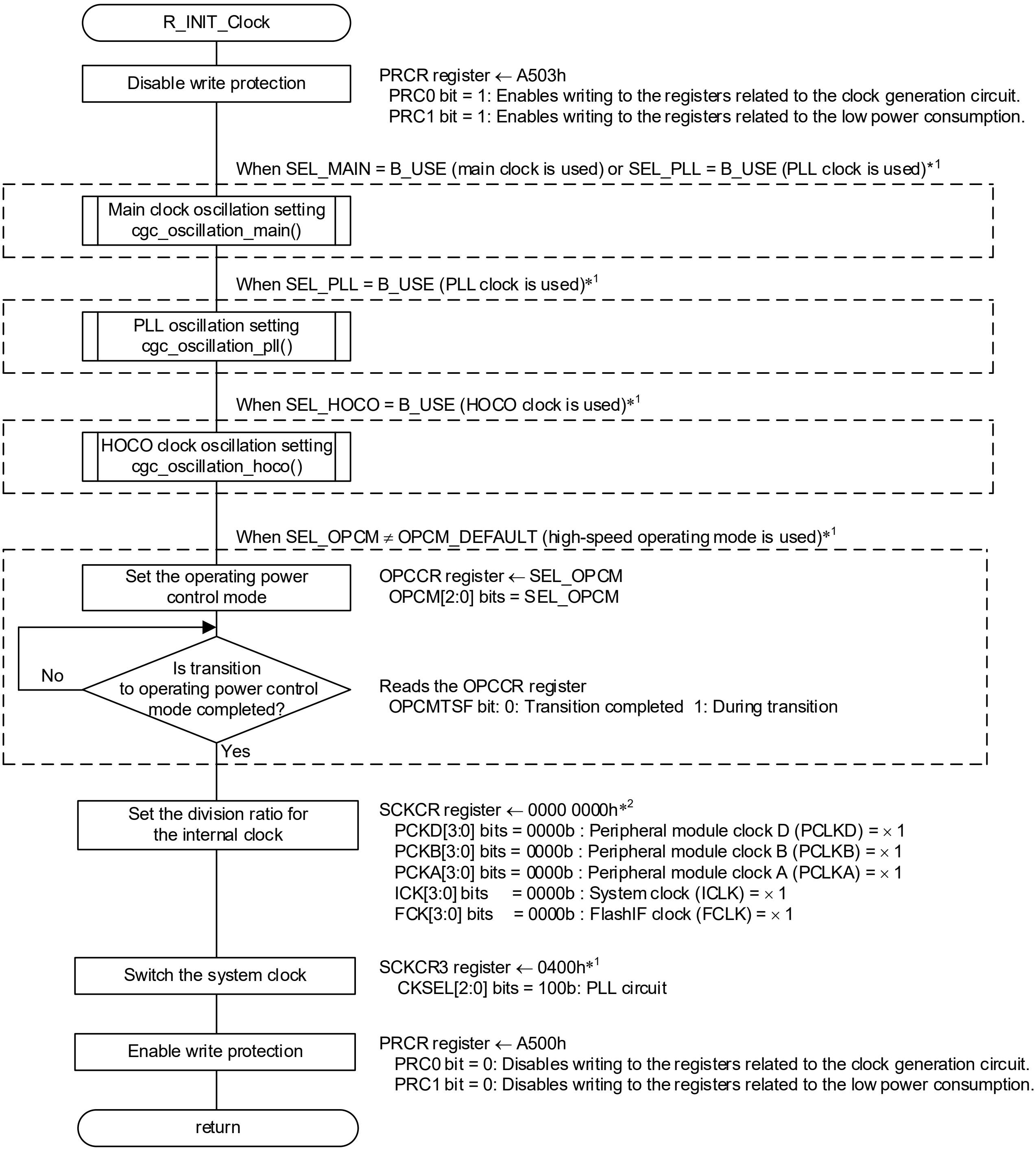


Figure 3.3 Nonexistent Port Initialization

3.10.4 Clock Initialization

Figure 3.4 shows the clock initialization.

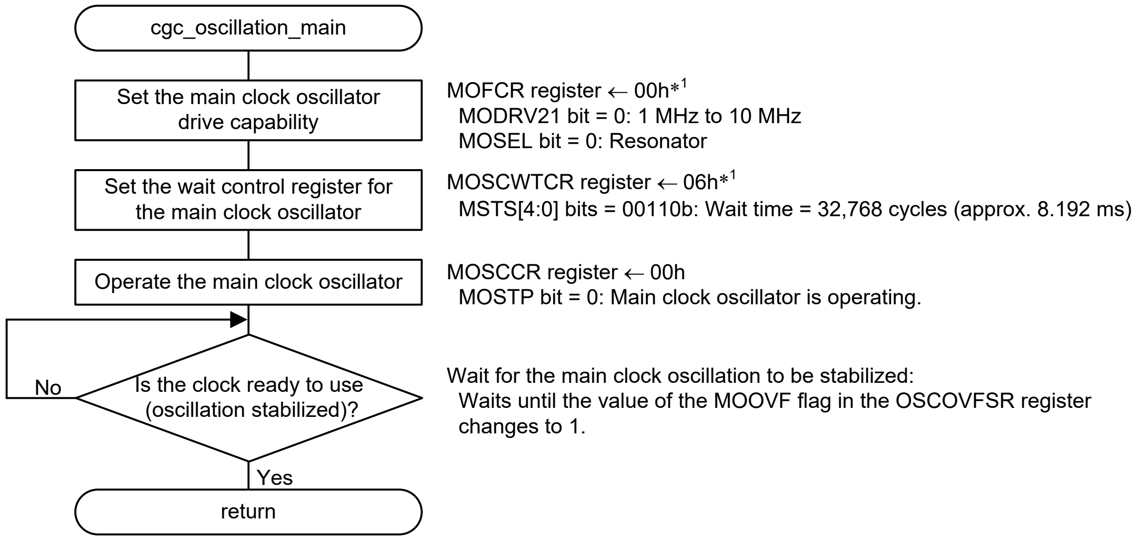


Notes: 1. Change the constant value according to the user system.
2. The setting value varies depending on the system clock selected by the constant.

Figure 3.4 Clock Initialization

3.10.5 Main Clock Oscillation Setting

Figure 3.5 shows the main clock oscillation setting.



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

Figure 3.5 Main Clock Oscillation Setting

3.10.6 HOCO Clock Oscillation Setting

Figure 3.6 shows the HOCO clock oscillation setting.

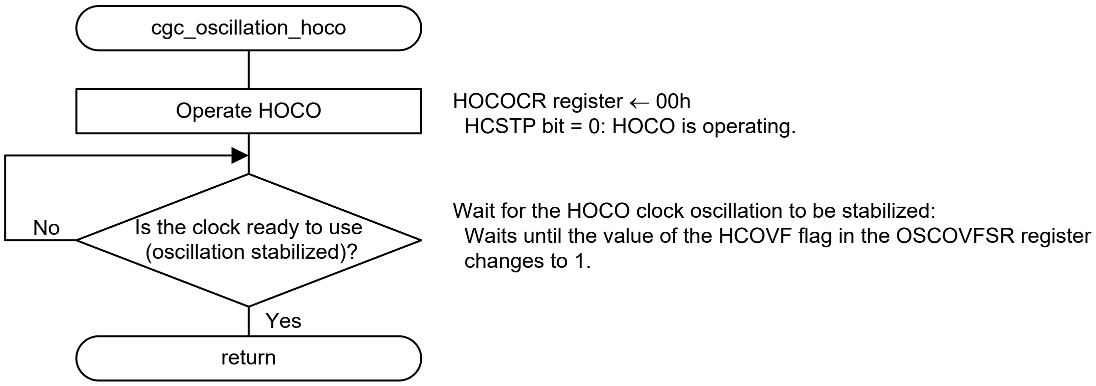
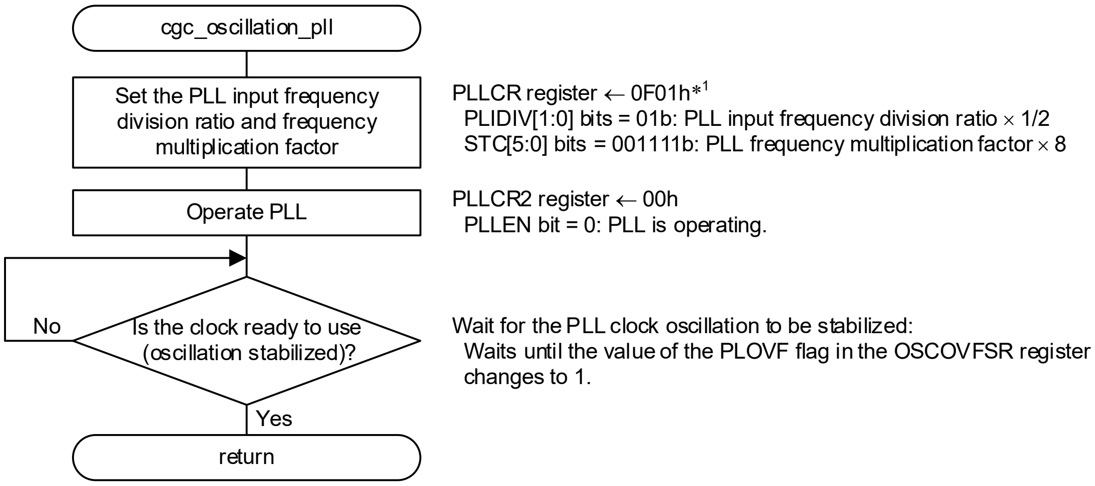


Figure 3.6 HOCO Clock Oscillation Setting

3.10.7 PLL Clock Oscillation Setting

Figure 3.7 shows the PLL clock oscillation setting.



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

Figure 3.7 PLL Clock Oscillation Setting

4. Importing a Project

After importing the sample project, make sure to confirm build and debugger setting.

4.1 Importing a Project into e2 studio

Follow the steps below to import your project into e² studio. Pictures may be different depending on the version of e² studio to be used.

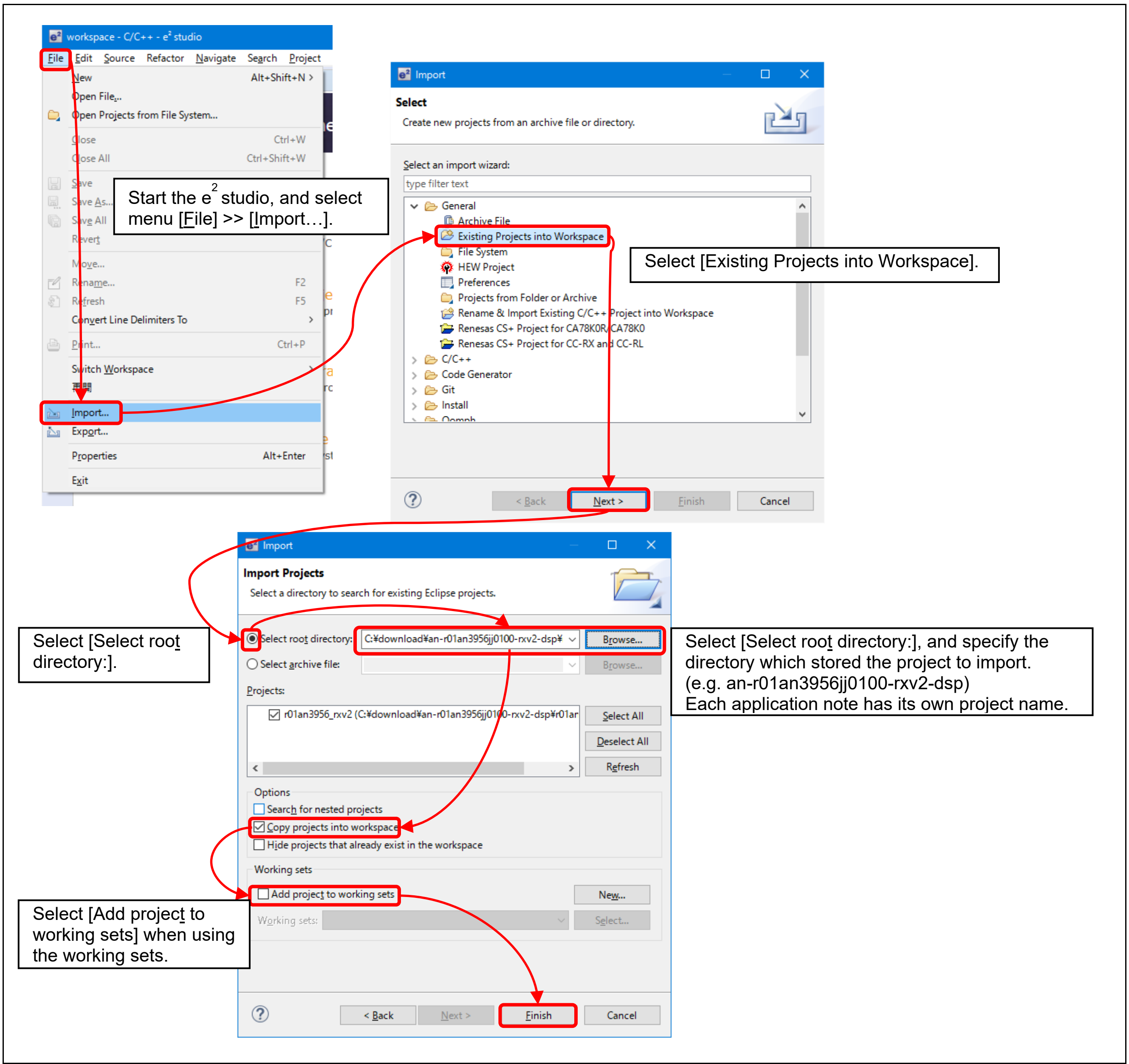


Figure 4.1 Importing a Project into e² studio

4.2 Importing a Project into CS+

Follow the steps below to import your project into CS+. Pictures may be different depending on the version of CS+ to be used.

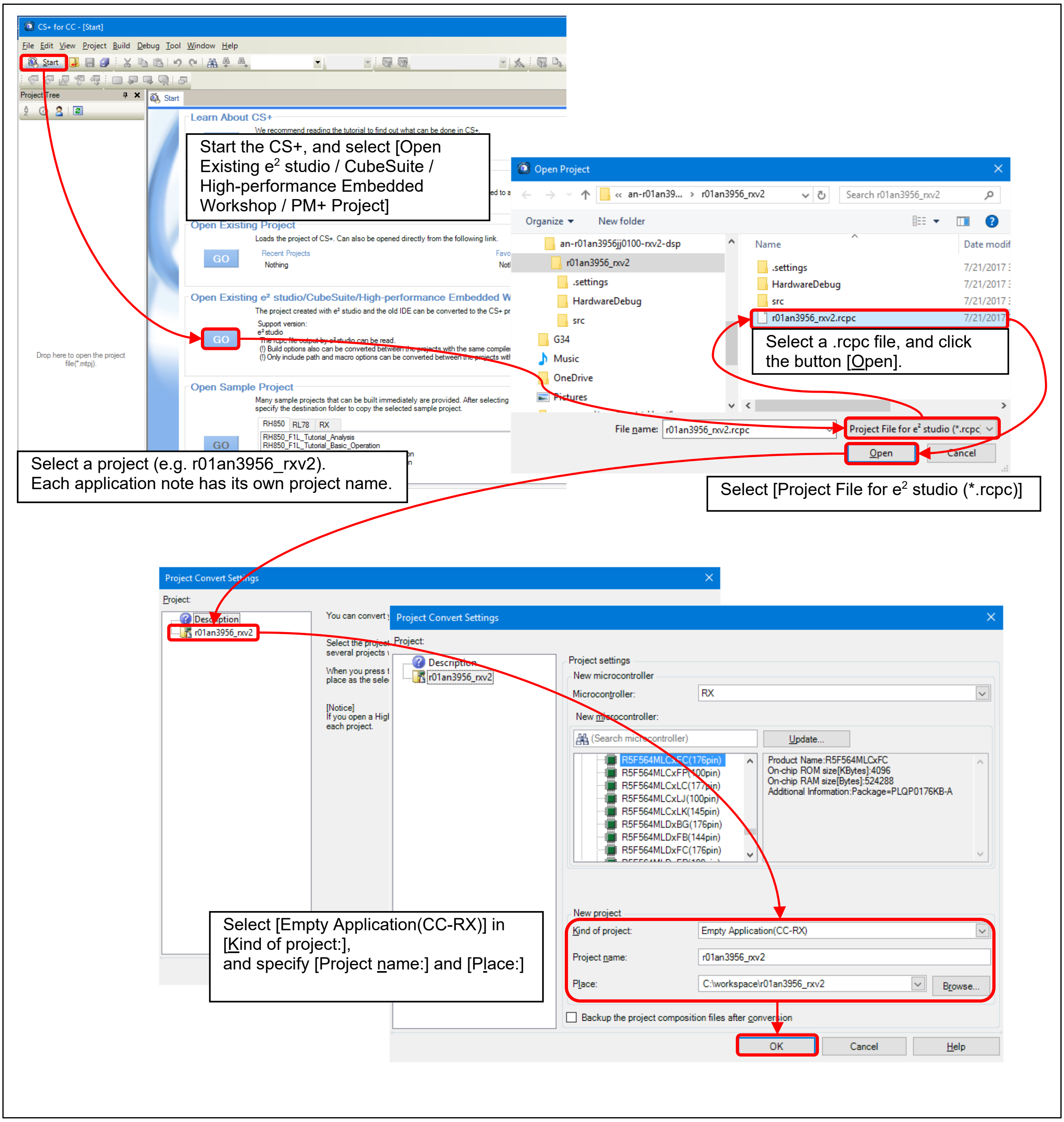


Figure 4.2 Importing a Project into CS+

5. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

6. Reference Documents

User's Manual: Hardware

RX23E-A Group User's Manual: Hardware (R01UH0801)
The latest version can be downloaded from the Renesas Electronics website.

User's Manual: Development Tools

RX Family Compiler CC-RX User's Manual (R20UT3248)
The latest version can be downloaded from the Renesas Electronics website.

Revision History

Rev.	Date	Description	
		Page	**Summary**
1.00	Nov. 01, 2019	—	First edition issued

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

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

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

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

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

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

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

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

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

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/.