R8C/M12A Group

Power Control in Wait Mode

R01AN0370EJ0100
Rev. 1.00
Sep. 16, 2011

Abstract

This document describes the setting method and an application example for power control using wait mode in the R8C/M12A Group.

Product

MCU: R8C/M12A Group

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

After reset and setting the operating mode, the MCU enters wait mode. The MCU returns from wait mode by the falling edge input of the $\overline{\text{INT0}}$ pin or a timer interrupt every second. This document describes the two sample programs described below.

- Sample program 1
  Reset → Low-speed on-chip oscillator mode (no division) → High-speed on-chip oscillator mode (no division) → Wait mode → High-speed on-chip oscillator mode (no division)
- Sample program 2
  Reset → Low-speed on-chip oscillator mode (no division) → Low-speed on-chip oscillator mode (divided-by-8) → Wait mode → Low-speed on-chip oscillator mode (no division) → High-speed clock mode (no division)

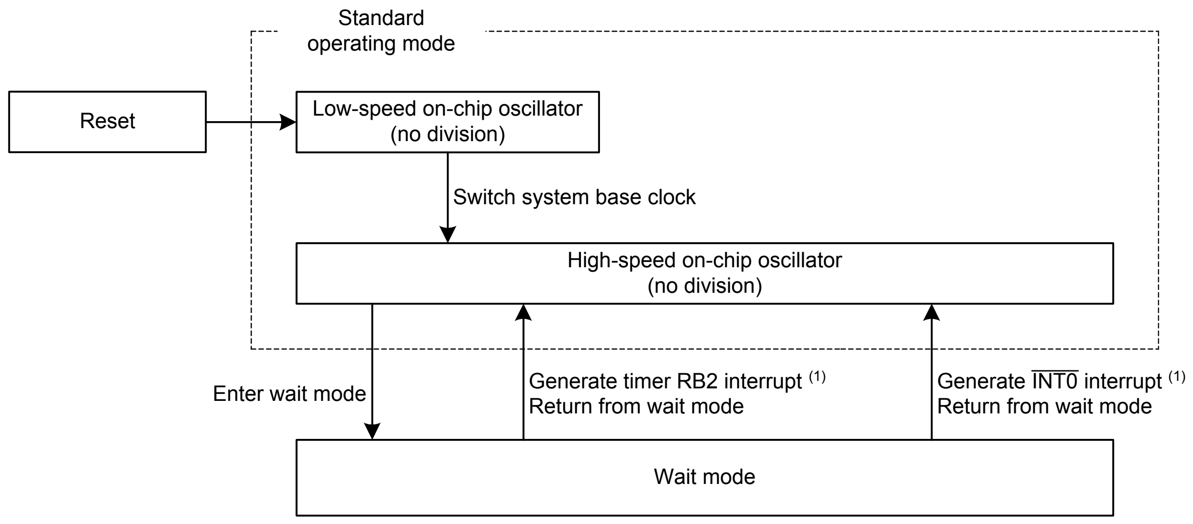
Tables 1.1 and 1.2 list the Peripheral Functions and Their Applications (Sample Programs 1 and 2). Figures 1.1 to 1.2 show Usage Examples.

Table 1.1 Peripheral Functions and Their Applications (Sample Program 1)

Peripheral Function	Application
$\overline{\text{INT0}}$ interrupt	Return from wait mode
Timer RJ2	Count source of timer RB2
Timer RB2	Return from wait mode

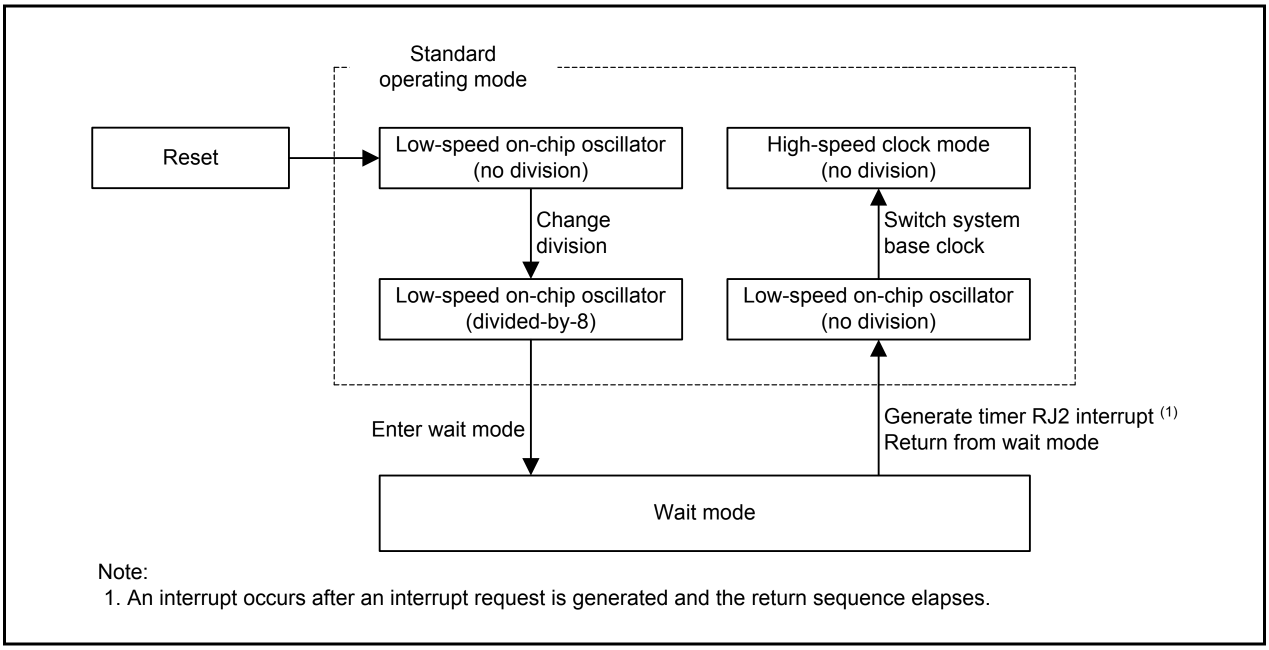
Table 1.2 Peripheral Function and Its Application (Sample Program 2)

Peripheral Function	Application
Timer RJ2	Return from wait mode



Note:
1. An interrupt occurs after an interrupt request is generated and then return sequence elapses.

Figure 1.1 Usage Example for Sample Program 1


Standard
operating mode
Reset
Low-speed on-chip oscillator (no division)
Change division
Low-speed on-chip oscillator (divided-by-8)
High-speed clock mode (no division)
Switch system base clock
Low-speed on-chip oscillator (no division)
Enter wait mode
Generate timer RJ2 interrupt (1)
Return from wait mode
Wait mode

Note:
1. An interrupt occurs after an interrupt request is generated and the return sequence elapses.

Figure 1.2 Usage Example for Sample Program 2

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	R8C/M12A Group
Operating frequencies	Sample program 1 • High-speed on-chip oscillator clock: 20 MHz (typical) • System clock (f) (before entering wait mode): 20 MHz (typical) • System clock (f) (after returning from wait mode): 20 MHz (typical) • CPU clock (fs) (before entering wait mode): 20 MHz (typical) • CPU clock (fs) (after returning from wait mode): 20 MHz (typical) Sample program 2 • Low-speed on-chip oscillator clock: 125 kHz (typical) • XIN clock: 20 MHz • System clock (f) (before entering wait mode): 125 kHz (typical) • System clock (f) (after returning from wait mode): 125 kHz (typical) • System clock (f) (after switching system base clock): 20 MHz • CPU clock (fs) (before entering wait mode): 15.625 kHz (typical) • CPU clock (fs) (after returning from wait mode): 125 kHz (typical) • CPU clock (fs) (after switching system base clock): 20 MHz
Operating voltage	5.0 V (2.7 to 5.5 V)
Integrated development environment	Renesas Electronics Corporation High-performance Embedded Workshop Version 4.07
C compiler	Renesas Electronics Corporation M16C Series, R8C Family C Compiler V.5.45 Release 01
	Compile options -D__UART0__ -c -finfo -dir "$(CONFIGDIR)" -R8C (Default setting is used in the integrated development environment.)

3. Hardware

3.1 Pin Used

Table 3.1 lists the Pin Used and Its Function.

Table 3.1 Pin Used and Its Function

Pin Name	I/O	Function
P1_4/$\overline{\text{INT0}}$	Input	$\overline{\text{INT0}}$ interrupt input

4. Software for Sample Program 1

4.1 Operation Overview of Sample Program 1

After reset, the MCU enters high-speed on-chip oscillator mode (no division) from low-speed on-chip oscillator mode (no division) by a program. Disable maskable interrupts, enable the $\overline{\text{INT0}}$ interrupt input used to return from wait mode, start timer RB2 and timer RJ2 counts, disable CPU rewrite mode, disable the voltage detection circuit, and set a clock after returning from wait mode. Then, enable maskable interrupts and execute the WAIT instruction to enter wait mode.

The system clock stops in wait mode. However, since fHOCO is selected as the count source of timer RJ2, timer RB2 and timer RJ2 continue the count operation. The MCU returns from wait mode using an interrupt by the falling edge input of the $\overline{\text{INT0}}$ pin or the timer RB2 interrupt every second. A clock immediately before entering wait mode is automatically selected for the CPU clock when returning from wait mode. When the MCU returns from wait mode using the timer RB2 interrupt, it enters wait mode by the WAIT instruction again. When the MCU returns from wait mode using the $\overline{\text{INT0}}$ interrupt, disable the $\overline{\text{INT0}}$ interrupt input, and stop timer RB2 and timer RJ2.

(1) Oscillate the high-speed on-chip oscillator by a program after reset.

(2) After waiting until oscillation of the high-speed on-chip oscillator stabilizes, switch the operating mode from low-speed on-chip oscillator mode (no division) to high-speed on-chip oscillator mode (no division) and stop the low-speed on-chip oscillator.

(3) Disable maskable interrupts, and perform the settings described below.

Settings
- Enable the $\overline{\text{INT0}}$ interrupt input.
- Start the timer RB2 and timer RJ2 count.
- Disable CPU rewrite mode.
- Disable voltage detection.
- Set the WCKSTP bit in the CKSTPR register to 1 (the system clock stops in wait mode).
- Set the PHISRS bit in the CKRSCR register to 0 (setting values of bits PHISSEL0 to PHISSEL2 in the SCKCR register are enabled).
- Set the WAITRS bit in the CKRSCR register to 0 (return from wait mode using the system base clock immediately before entering wait mode).
- Set 1 to the variable (mode).

(4) Enable maskable interrupts, and execute the WAIT instruction to enter wait mode.

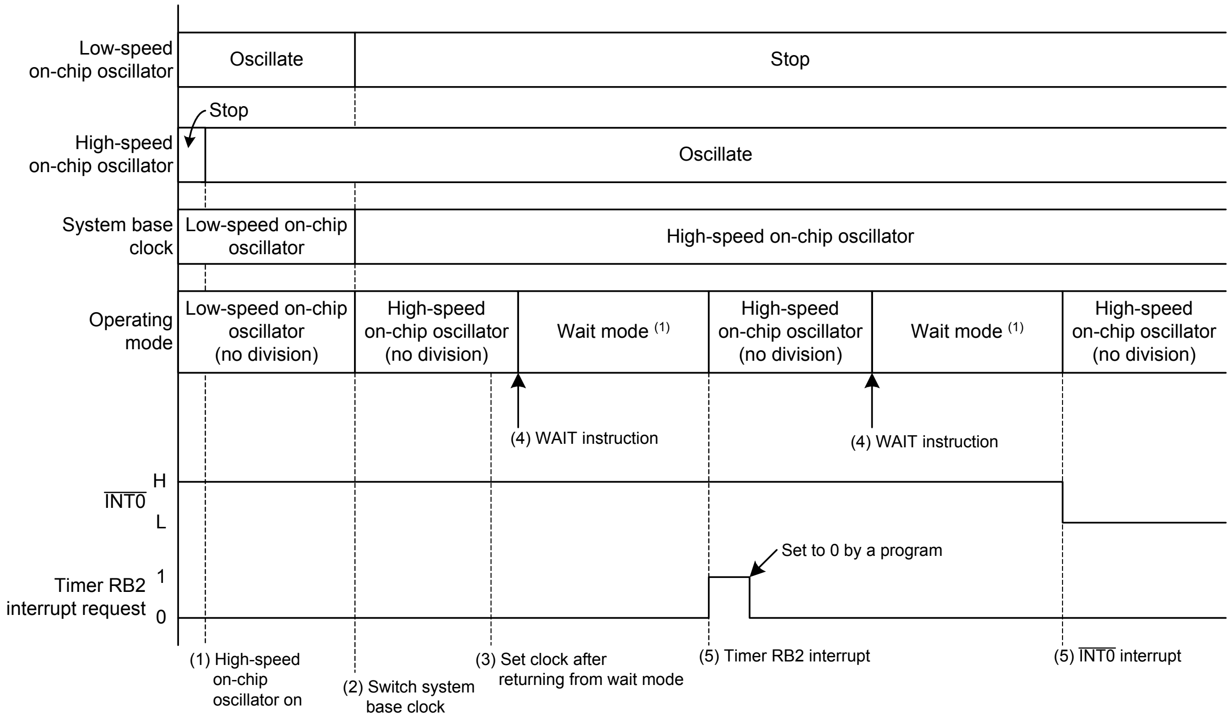
(5) Return from wait mode using the timer RB2 interrupt (every second) or $\overline{\text{INT0}}$ interrupt (falling edge signal). Clocks set in (3) are selected for the CPU clock when returning from wait mode.

When the MCU returns from wait mode using the timer RB2 interrupt:
After returning to main processing, the MCU enters wait mode again.

When the MCU returns from wait mode using the $\overline{\text{INT0}}$ interrupt:
Set 0 to the variable (mode) to return to main processing, disable the $\overline{\text{INT0}}$ interrupt input, and stop timer RB2 and timer RJ2.

(6) Repeat steps (3) to (5).

Figure 4.1 shows the Operating Example in Wait Mode.


Low-speed on-chip oscillator
Oscillate
Stop
Stop
High-speed on-chip oscillator
Oscillate
System base clock
Low-speed on-chip oscillator
High-speed on-chip oscillator
Operating mode
Low-speed on-chip oscillator (no division)
High-speed on-chip oscillator (no division)
Wait mode (1)
High-speed on-chip oscillator (no division)
Wait mode (1)
High-speed on-chip oscillator (no division)
(4) WAIT instruction
(4) WAIT instruction
INT0
H
L
Set to 0 by a program
Timer RB2 interrupt request
1
0
(1) High-speed on-chip oscillator on
(2) Switch system base clock
(3) Set clock after returning from wait mode
(5) Timer RB2 interrupt
(5) INT0 interrupt

Note:

1. The peripheral functions that use the CPU clock and system clock and a divided system clock generated by the prescaler are stopped.

Figure 4.1 Operating Example in Wait Mode

4.2 Required Memory Size

Table 4.1 lists the Required Memory Size.

Table 4.1 Required Memory Size for Sample Program 1

Memory Used	Size	Remarks
ROM	443 bytes	In the r01an0370_src_sample1.c module
RAM	1 byte	In the r01an0370_src_sample1.c module
Maximum user stack usage	10 bytes	
Maximum interrupt stack usage	18 bytes	

The required memory size varies depending on the C compiler version and compile options.

4.3 Variables

Table 4.2 lists the Global Variable.

Table 4.2 Global Variable

Type	Variable Name	Contents	Function Used
unsigned char	mode	Select to enter wait mode	power_control, _int0

4.4 Functions

Table 4.3 lists the Functions.

Table 4.3 Functions

Function Name	Outline
mcu_init	System clock setting
int0_init	Initial setting of $\overline{\text{INT0}}$ interrupt
timer_rj2_init	Initial setting of timer RJ2
timer_rb2_init	Initial setting of timer RB2
power_control	Wait mode processing
_int0	$\overline{\text{INT0}}$ interrupt handling
_timer_rb	Timer RB2 interrupt handling

4.5 Function Specifications

The following tables list the sample code function specifications.

mcu_init	
Outline	System clock setting
Header	None
Declaration	void mcu_init(void)
Description	Set the system clock.
Argument	None
Returned value	None
Remark	—

int0_init	
Outline	Initial setting of $\overline{\text{INT0}}$ interrupt
Header	None
Declaration	void int0_init(void)
Description	Perform initial setting to use the $\overline{\text{INT0}}$ interrupt.
Argument	None
Returned value	None
Remark	—

timer_rj2_init	
Outline	Initial setting of timer RJ2
Header	None
Declaration	void timer_rj2_init(void)
Description	Perform initial setting to use timer RJ2 in timer mode.
Argument	None
Returned value	None
Remark	—

timer_rb2_init	
Outline	Initial setting of timer RB2
Header	None
Declaration	void timer_rb2_init(void)
Description	• Perform initial setting to use timer RB2 in timer mode. • Use timer RJ2 underflow as the count source.
Argument	None
Returned value	None
Remark	—

power_control	
Outline	Wait mode processing
Header	None
Declaration	void power_control(void)
Description	Enter wait mode.
Argument	None
Returned value	None
Remark	—

_int0	
Outline	$\overline{\text{INT0}}$ interrupt handling
Header	None
Declaration	void _int0(void)
Description	Set 0 to the variable (mode).
Argument	None
Returned value	None
Remark	—

_timer_rb	
Outline	Timer RB2 interrupt handling
Header	None
Declaration	void _timer_rb(void)
Description	Perform timer RB2 interrupt handling.
Argument	None
Returned value	None
Remark	—

4.6 Flowcharts

4.6.1 Main Processing

Figure 4.2 shows the Main Processing.

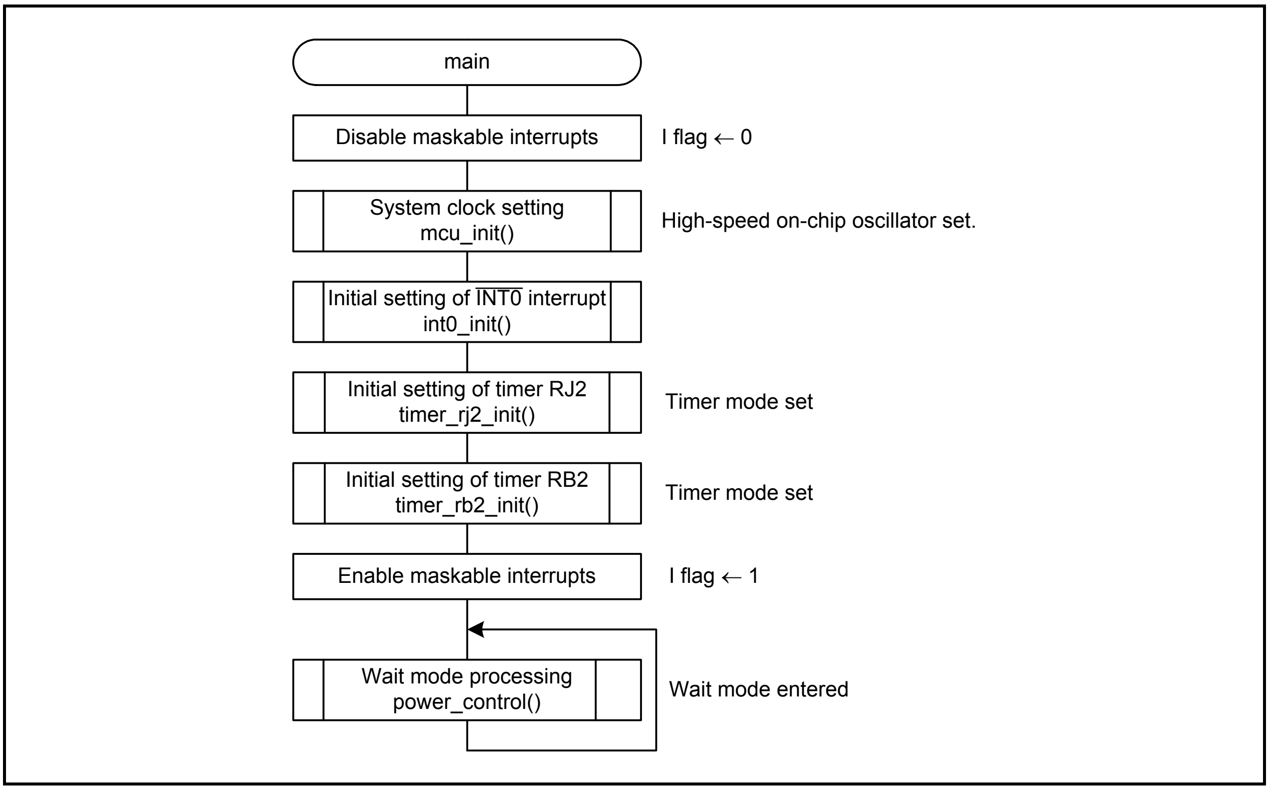

main
Disable maskable interrupts
I flag ← 0
System clock setting
mcu_init()
High-speed on-chip oscillator set.
Initial setting of $\overline{\text{INT0}}$ interrupt
int0_init()
Initial setting of timer RJ2
timer_rj2_init()
Timer mode set
Initial setting of timer RB2
timer_rb2_init()
Timer mode set
Enable maskable interrupts
I flag ← 1
Wait mode processing
power_control()
Wait mode entered

Figure 4.2 Main Processing

4.6.2 System Clock Setting

Figure 4.3 shows the System Clock Setting.

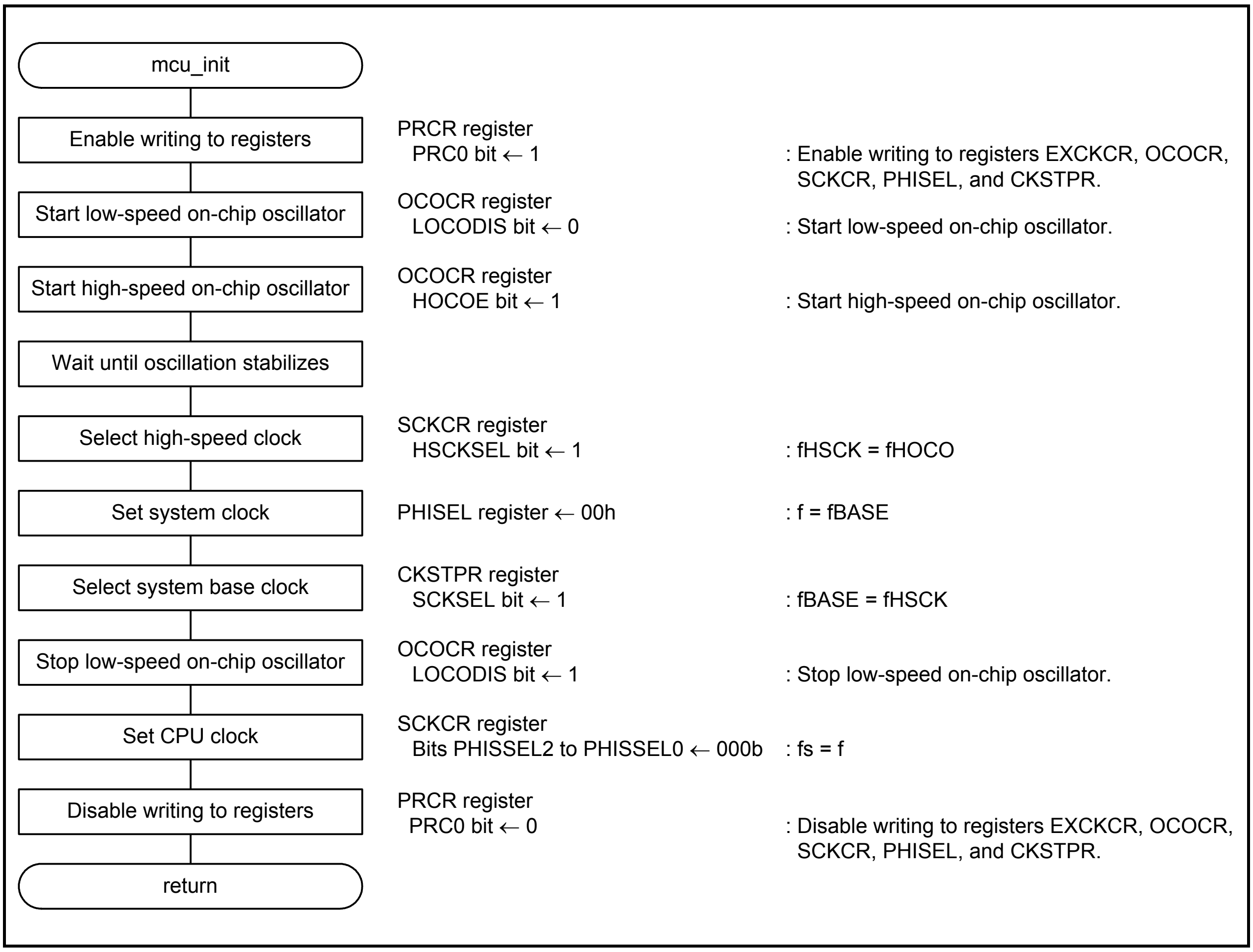
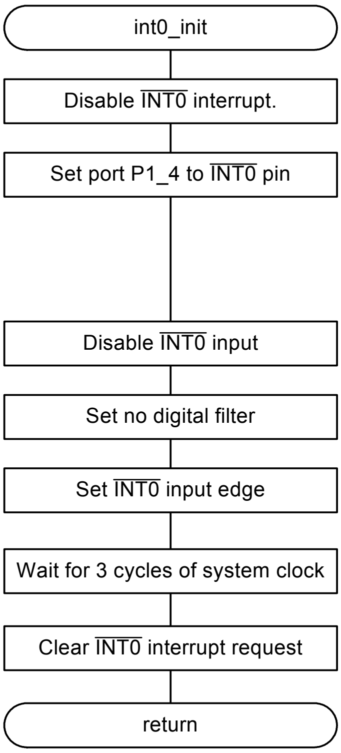

mcu_init
Enable writing to registers
PRCR register
PRC0 bit ← 1
: Enable writing to registers EXCKCR, OCOCR, SCKCR, PHISEL, and CKSTPR.
Start low-speed on-chip oscillator
OCOCR register
LOCODIS bit ← 0
: Start low-speed on-chip oscillator.
Start high-speed on-chip oscillator
OCOCR register
HOCOE bit ← 1
: Start high-speed on-chip oscillator.
Wait until oscillation stabilizes
Select high-speed clock
SCKCR register
HSCKSEL bit ← 1
: fHSCK = fHOCO
Set system clock
PHISEL register ← 00h
: f = fBASE
Select system base clock
CKSTPR register
SCKSEL bit ← 1
: fBASE = fHSCK
Stop low-speed on-chip oscillator
OCOCR register
LOCODIS bit ← 1
: Stop low-speed on-chip oscillator.
Set CPU clock
SCKCR register
Bits PHISSEL2 to PHISSEL0 ← 000b
: fs = f
Disable writing to registers
PRCR register
PRC0 bit ← 0
: Disable writing to registers EXCKCR, OCOCR, SCKCR, PHISEL, and CKSTPR.
return

Figure 4.3 System Clock Setting

4.6.3 Initial Setting of the $\overline{\text{INT0}}$ Interrupt

Figure 4.4 shows the $\overline{\text{INT0}}$ Interrupt Setting.


int0_init
Disable $\overline{\text{INT0}}$ interrupt.
Set port P1_4 to $\overline{\text{INT0}}$ pin
Disable $\overline{\text{INT0}}$ input
Set no digital filter
Set $\overline{\text{INT0}}$ input edge
Wait for 3 cycles of system clock
Clear $\overline{\text{INT0}}$ interrupt request
return

ILVLE register
Bits ILVLE5 and ILVLE4 = 00b : Level 0 (interrupts disabled)

PMH1E register
P14SEL2 bit = 0 : Use the P14SEL2 bit with bits P14SEL0 and P14SEL1 in the PMH1 register to select the function of the P1_4 pin.

PMH1 register
Bits P14SEL1 and P14SEL0 = 11b : $\overline{\text{INT0}}$

INTEN register
INT0EN bit ← 0

INTF0 register ← 00h
Bits INT0F1 and INT0F0 = 00b : No $\overline{\text{INT0}}$ input filter

ISCR0 register ← 00h
Bits INT0SB and INT0SA = 00b : Generate the interrupt request at the falling edge of $\overline{\text{INT0}}$ input.

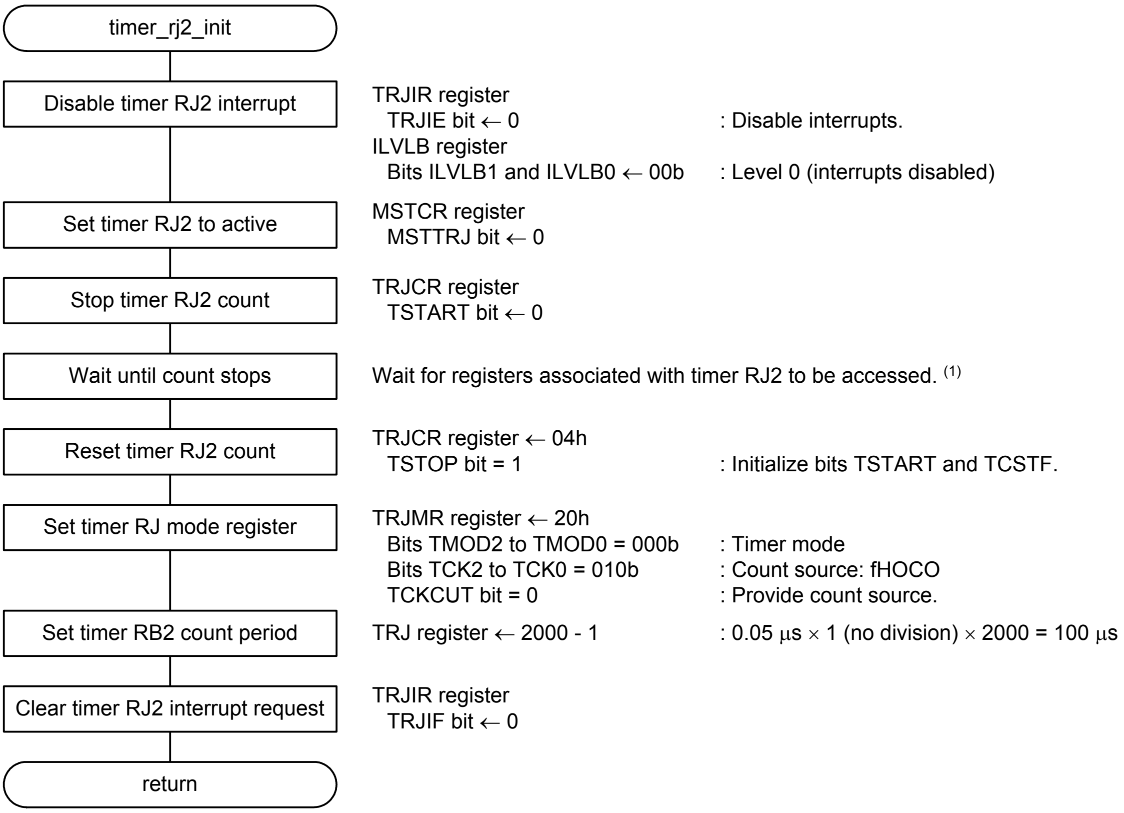
Wait for registers associated with the $\overline{\text{INT0}}$ interrupt to be accessed.

IRR3 register
IRI0 bit ← 0

Figure 4.4 $\overline{\text{INT0}}$ Interrupt Setting

4.6.4 Initial Setting of Timer RJ2

Figure 4.5 shows the Initial Setting of Timer RJ2.


timer_rj2_init
Disable timer RJ2 interrupt
Set timer RJ2 to active
Stop timer RJ2 count
Wait until count stops
Reset timer RJ2 count
Set timer RJ mode register
Set timer RB2 count period
Clear timer RJ2 interrupt request
return

TRJIR register
  TRJIE bit ← 0 : Disable interrupts.
ILVLB register
  Bits ILVLB1 and ILVLB0 ← 00b : Level 0 (interrupts disabled)

MSTCR register
  MSTTRJ bit ← 0

TRJCR register
  TSTART bit ← 0

Wait for registers associated with timer RJ2 to be accessed. (1)

TRJCR register ← 04h
  TSTOP bit = 1 : Initialize bits TSTART and TCSTF.

TRJMR register ← 20h
  Bits TMOD2 to TMOD0 = 000b : Timer mode
  Bits TCK2 to TCK0 = 010b : Count source: fHOCO
  TCKCUT bit = 0 : Provide count source.

TRJ register ← 2000 - 1 : 0.05 µs × 1 (no division) × 2000 = 100 µs

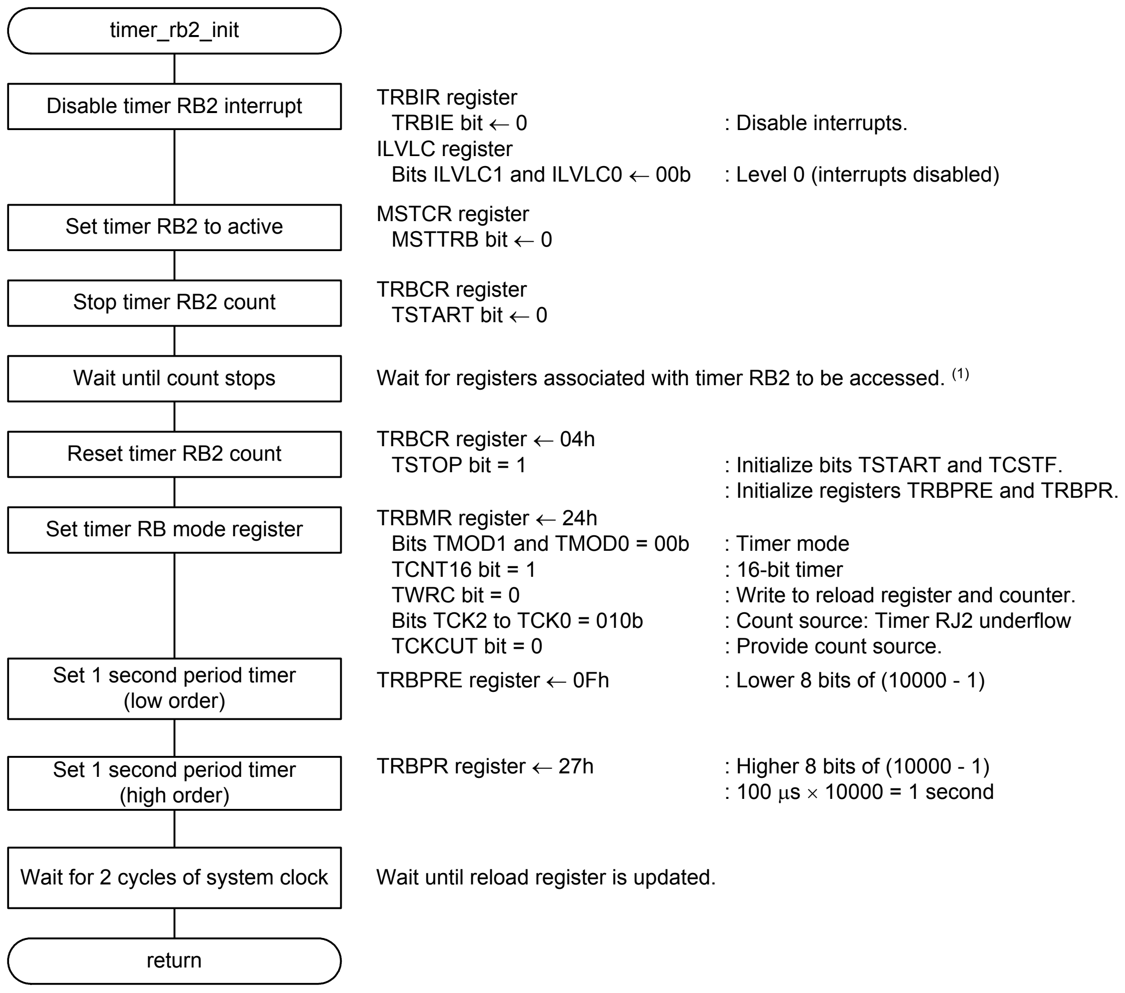
TRJIR register
  TRJIF bit ← 0

Note:
1. Registers associated with timer RJ2: TRJ, TRJCR, and TRJMR

Figure 4.5 Initial Setting of Timer RJ2

4.6.5 Initial Setting of Timer RB2

Figure 4.6 shows the Initial Setting of Timer RB2.


timer_rb2_init
Disable timer RB2 interrupt
Set timer RB2 to active
Stop timer RB2 count
Wait until count stops
Reset timer RB2 count
Set timer RB mode register
Set 1 second period timer (low order)
Set 1 second period timer (high order)
Wait for 2 cycles of system clock
return

```
TRBIR register
  TRBIE bit ← 0                        : Disable interrupts.
ILVLC register
  Bits ILVLC1 and ILVLC0 ← 00b         : Level 0 (interrupts disabled)

MSTCR register
  MSTTRB bit ← 0

TRBCR register
  TSTART bit ← 0

Wait for registers associated with timer RB2 to be accessed. (1)

TRBCR register ← 04h
  TSTOP bit = 1                        : Initialize bits TSTART and TCSTF.
                                       : Initialize registers TRBPRE and TRBPR.
TRBMR register ← 24h
  Bits TMOD1 and TMOD0 = 00b           : Timer mode
  TCNT16 bit = 1                       : 16-bit timer
  TWRC bit = 0                         : Write to reload register and counter.
  Bits TCK2 to TCK0 = 010b             : Count source: Timer RJ2 underflow
  TCKCUT bit = 0                       : Provide count source.
TRBPRE register ← 0Fh                  : Lower 8 bits of (10000 - 1)

TRBPR register ← 27h                   : Higher 8 bits of (10000 - 1)
                                       : 100 µs × 10000 = 1 second

Wait until reload register is updated.
```

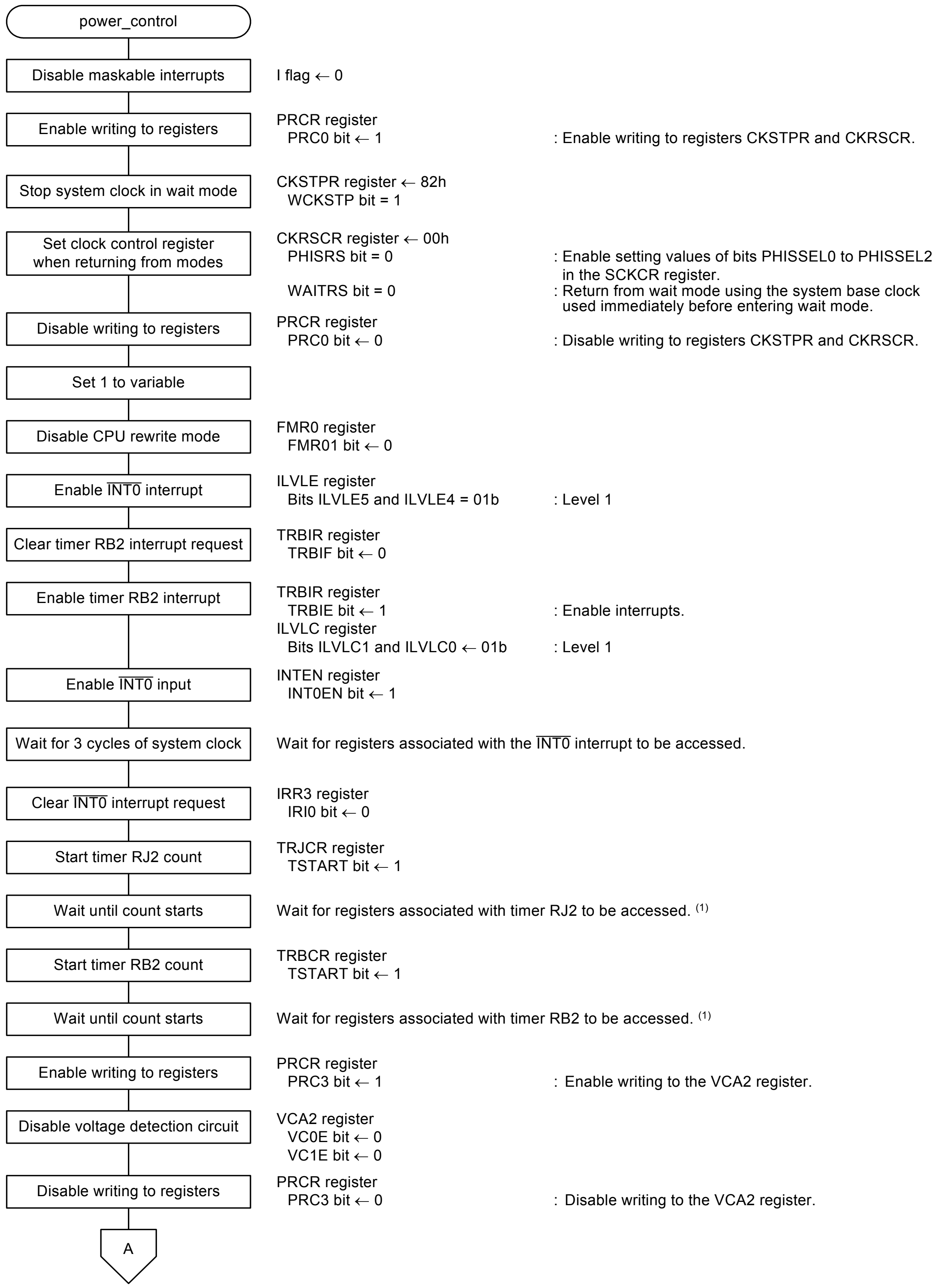
Note:

1. Registers associated with timer RB2: TRBCR, TRBMR, TRBPRE, and TRBPR

Figure 4.6 Initial Setting of Timer RB2

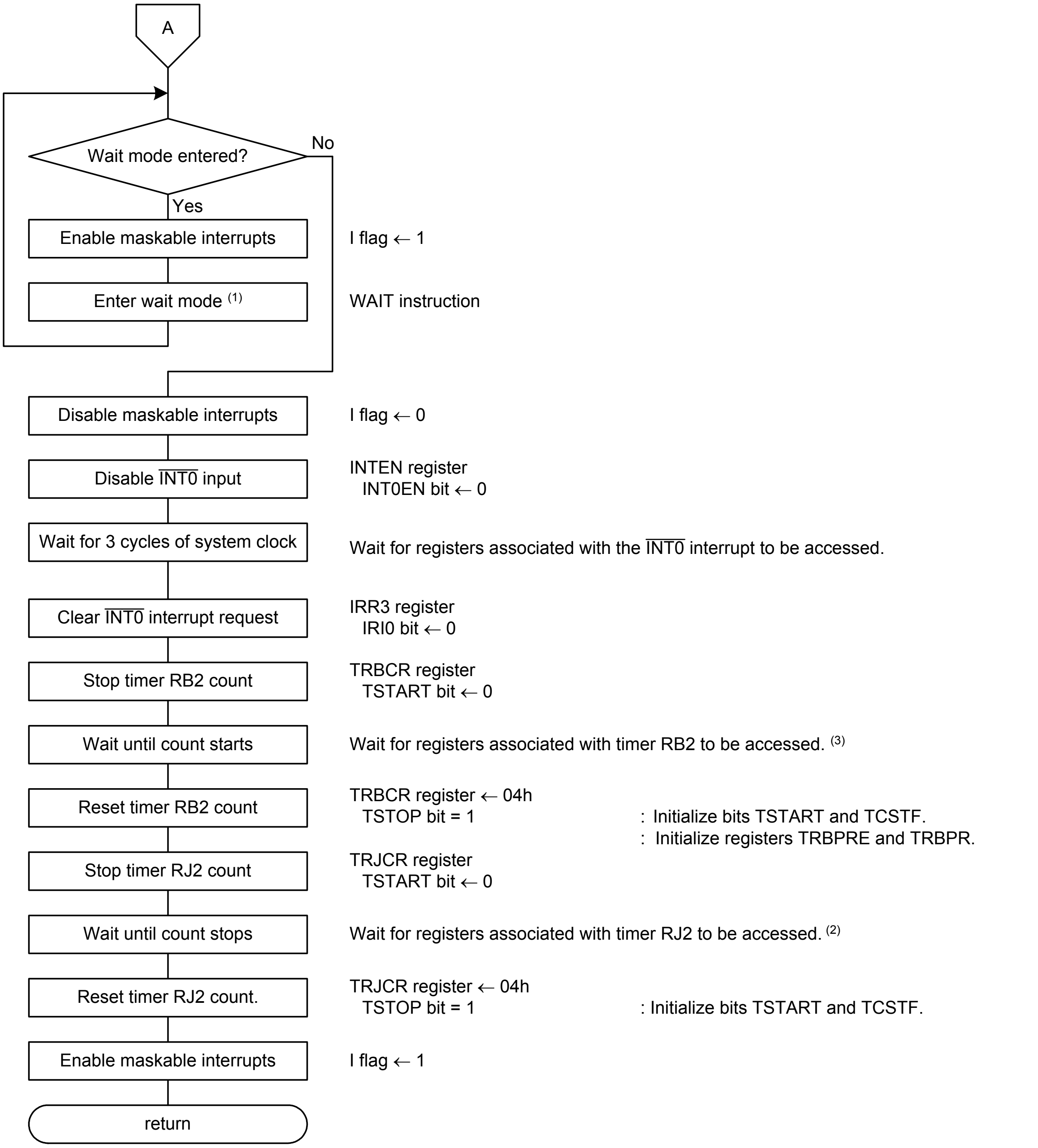
4.6.6 Wait Mode Processing

Figures 4.7 and 4.8 show Wait Mode Processing.


power_control
Disable maskable interrupts
I flag ← 0
Enable writing to registers
PRCR register
PRC0 bit ← 1
: Enable writing to registers CKSTPR and CKRSCR.
Stop system clock in wait mode
CKSTPR register ← 82h
WCKSTP bit = 1
Set clock control register when returning from modes
CKRSCR register ← 00h
PHISRS bit = 0
: Enable setting values of bits PHISSEL0 to PHISSEL2 in the SCKCR register.
WAITRS bit = 0
: Return from wait mode using the system base clock used immediately before entering wait mode.
Disable writing to registers
PRCR register
PRC0 bit ← 0
: Disable writing to registers CKSTPR and CKRSCR.
Set 1 to variable
Disable CPU rewrite mode
FMR0 register
FMR01 bit ← 0
Enable INT0 interrupt
ILVLE register
Bits ILVLE5 and ILVLE4 = 01b
: Level 1
Clear timer RB2 interrupt request
TRBIR register
TRBIF bit ← 0
Enable timer RB2 interrupt
TRBIR register
TRBIE bit ← 1
: Enable interrupts.
ILVLC register
Bits ILVLC1 and ILVLC0 ← 01b
: Level 1
Enable INT0 input
INTEN register
INT0EN bit ← 1
Wait for 3 cycles of system clock
Wait for registers associated with the INT0 interrupt to be accessed.
Clear INT0 interrupt request
IRR3 register
IRI0 bit ← 0
Start timer RJ2 count
TRJCR register
TSTART bit ← 1
Wait until count starts
Wait for registers associated with timer RJ2 to be accessed. (1)
Start timer RB2 count
TRBCR register
TSTART bit ← 1
Wait until count starts
Wait for registers associated with timer RB2 to be accessed. (1)
Enable writing to registers
PRCR register
PRC3 bit ← 1
: Enable writing to the VCA2 register.
Disable voltage detection circuit
VCA2 register
VC0E bit ← 0
VC1E bit ← 0
Disable writing to registers
PRCR register
PRC3 bit ← 0
: Disable writing to the VCA2 register.
A

Notes:
1. Registers associated with timer RJ2: TRJ, TRJCR, and TRJMR
2. Registers associated with timer RB2: TRBCR, TRBMR, TRBPRE, and TRBPR

Figure 4.7 Wait Mode Processing (1/2)


A
Wait mode entered?
No
Yes
Enable maskable interrupts
I flag ← 1
Enter wait mode (1)
WAIT instruction
Disable maskable interrupts
I flag ← 0
Disable $\overline{\text{INT0}}$ input
INTEN register
INT0EN bit ← 0
Wait for 3 cycles of system clock
Wait for registers associated with the $\overline{\text{INT0}}$ interrupt to be accessed.
Clear $\overline{\text{INT0}}$ interrupt request
IRR3 register
IRI0 bit ← 0
Stop timer RB2 count
TRBCR register
TSTART bit ← 0
Wait until count starts
Wait for registers associated with timer RB2 to be accessed. (3)
Reset timer RB2 count
TRBCR register ← 04h
TSTOP bit = 1
: Initialize bits TSTART and TCSTF.
: Initialize registers TRBPRE and TRBPR.
Stop timer RJ2 count
TRJCR register
TSTART bit ← 0
Wait until count stops
Wait for registers associated with timer RJ2 to be accessed. (2)
Reset timer RJ2 count.
TRJCR register ← 04h
TSTOP bit = 1
: Initialize bits TSTART and TCSTF.
Enable maskable interrupts
I flag ← 1
return

Notes:

1. Since the 4 bytes are prefetched from the WAIT instruction and then the program stops, insert at least four NOP instructions as a program restriction when entering wait mode.
2. Registers associated with timer RJ2: TRJ, TRJCR, and TRJMR
3. Registers associated with timer RB2: TRBCR, TRBMR, TRBPRE, and TRBPR

Figure 4.8 Wait Mode Processing (2/2)

4.6.7 $\overline{\mathrm{INT0}}$ Interrupt Handling

Figure 4.9 shows $\overline{\mathrm{INT0}}$ Interrupt Handling.

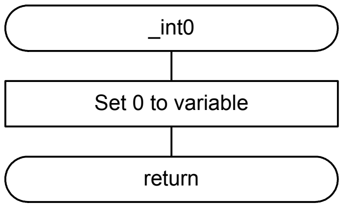


Figure 4.9 $\overline{\mathrm{INT0}}$ Interrupt Handling

4.6.8 Timer RB2 Interrupt Handling

Figure 4.10 shows Timer RB2 Interrupt Handling.

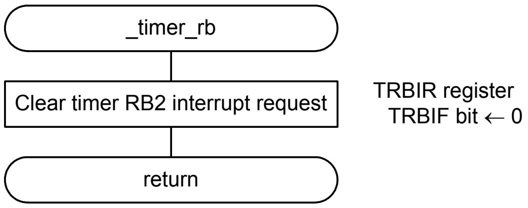


Figure 4.10 Timer RB2 Interrupt Handling

5. Software for Sample Program 2

5.1 Operation Overview of Sample Program 2

After reset, the MCU enters low-speed on-chip oscillator mode (divided-by-8) from low-speed on-chip oscillator mode (no division) by a program. Set the interval of refresh operations in low-current-consumption read mode to the FREFR register, and set the FMR27 bit in the FMR2 register to 1 (low-current-consumption read mode enabled) to use low-current-consumption read mode. Disable maskable interrupts, start the timer RJ2 count used to return from wait mode, disable CPU rewrite mode, disable voltage detection circuit, set a clock after returning from wait mode, disable low-current-consumption read mode, stop flash memory in wait mode, and enable low-power-consumption wait mode. After enabling maskable interrupts, execute the WAIT instruction to enter wait mode.

The MCU returns from wait mode using the timer RJ2 interrupt every second. The low-speed on-chip oscillator mode (no division) is automatically selected for the operating mode when returning from wait mode. Oscillate the XIN clock by a program and enter high-speed clock mode (no division) after waiting until XIN clock oscillation stabilizes.

(1) Set low-speed on-chip oscillator mode (divided-by-8) by a program after reset.

(2) After setting the interval of refresh operations in low-current-consumption read mode, set the FMR27 bit in the FMR2 register to 1 (low-current-consumption read mode enabled).

(3) Disable maskable interrupts, and perform the settings described below.

Settings

- Start the timer RJ2 count.
- Disable CPU rewrite mode.
- Disable voltage detection.
- Set the PHISRS bit in the CKRSCR register to 1 (no division).
- Set the WAITRS bit in the CKRSCR register to 0 (return from wait mode using the system base clock immediately before entering wait mode).
- Set the FMR27 bit to 0 (low-current-consumption read mode disabled).
- Set the WTFMSTP bit to 1 (flash memory stops in wait mode).
- Set the LPE bit in the VCA2 register to 1 (low-power-consumption wait mode enabled).

(4) Enable maskable interrupts and execute the WAIT instruction to enter wait mode.

(5) Return from wait mode using the time RJ2 interrupt (1 second period). Clocks set in (3) are selected for the CPU clock when returning from wait mode.

(6) Oscillate the XIN clock by a program.

(7) After waiting until XIN clock oscillation stabilizes, switch the system base clock from the low-speed on-chip oscillator to the XIN clock, and enter high-speed clock mode (no division).

Figure 5.1 shows an Operating Example in Wait Mode.

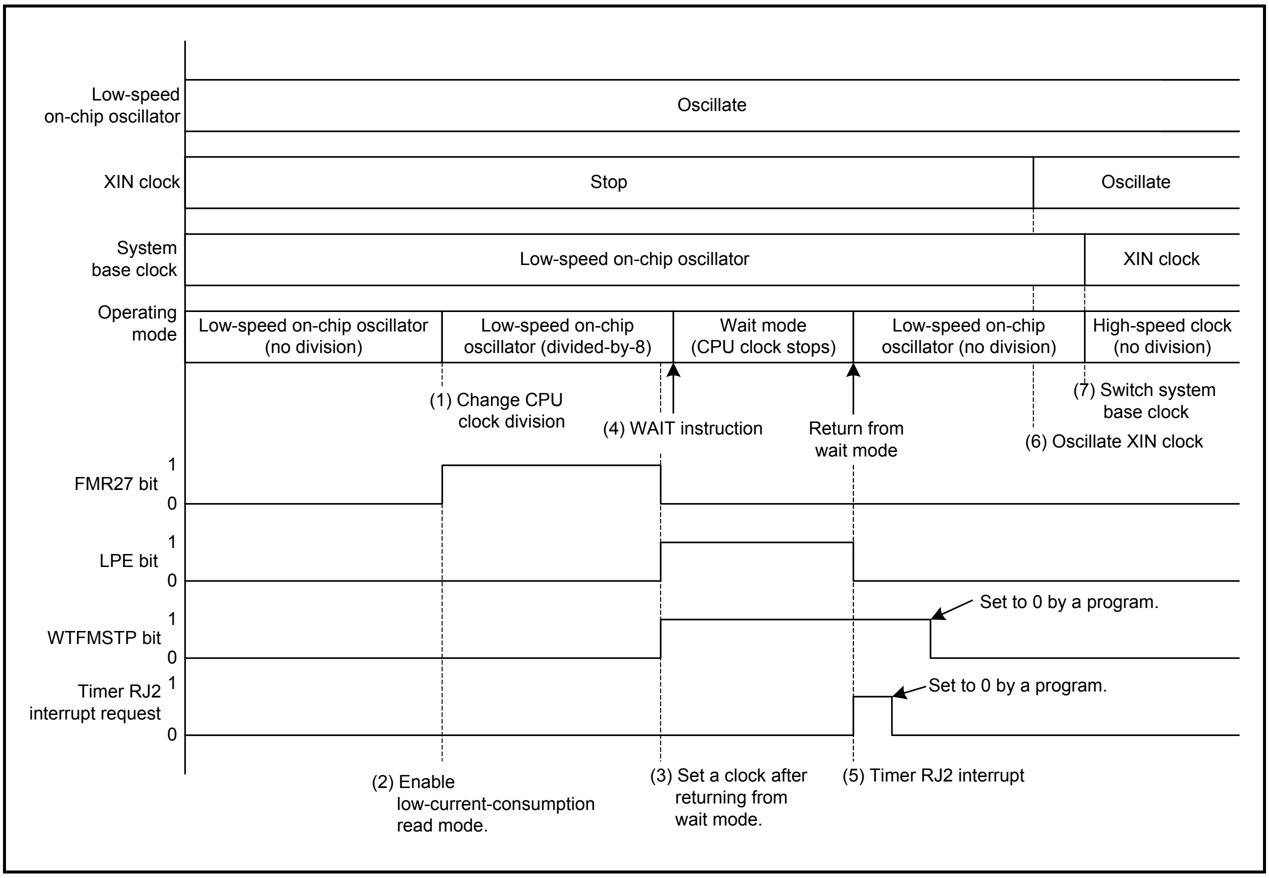

Low-speed on-chip oscillator
Oscillate
XIN clock
Stop
Oscillate
System base clock
Low-speed on-chip oscillator
XIN clock
Operating mode
Low-speed on-chip oscillator (no division)
Low-speed on-chip oscillator (divided-by-8)
Wait mode (CPU clock stops)
Low-speed on-chip oscillator (no division)
High-speed clock (no division)
(1) Change CPU clock division
(4) WAIT instruction
Return from wait mode
(7) Switch system base clock
(6) Oscillate XIN clock
FMR27 bit
1
0
LPE bit
1
0
WTFMSTP bit
1
0
Set to 0 by a program.
Timer RJ2 interrupt request
1
0
Set to 0 by a program.
(2) Enable low-current-consumption read mode.
(3) Set a clock after returning from wait mode.
(5) Timer RJ2 interrupt

Figure 5.1 Operating Example in Wait Mode

5.2 Required Memory Size

Table 5.1 lists the Required Memory Size.

Table 5.1 Required Memory Size for Sample Program 2

Memory Used	Size	Remarks
ROM	311 bytes	In the r01an0370_src_sample2.c module
RAM	0 bytes	In the r01an0370_src_sample2.c module
Maximum user stack usage	10 bytes	
Maximum interrupt stack usage	18 bytes	

The required memory size varies depending on the C compiler version and compile options.

5.3 Functions

Table 5.2 lists the Functions.

Table 5.2 Functions

Function Name	Outline
mcu_init	System clock setting
timer_rj2_init	Initial setting of timer RJ2
power_control	Wait mode processing
_timer_rj	Timer RJ2 interrupt handling

5.4 Function Specifications

The following tables list the sample code function specifications.

mcu_init	
Outline	System clock setting
Header	None
Declaration	void mcu_init(void)
Description	Set the system clock.
Argument	None
Returned value	None
Remark	—

timer_rj2_init	
Outline	Initial setting of timer RJ2
Header	None
Declaration	void timer_rj2_init(void)
Description	Perform initial setting to use timer RJ2 in timer mode.
Argument	None
Returned value	None
Remark	—

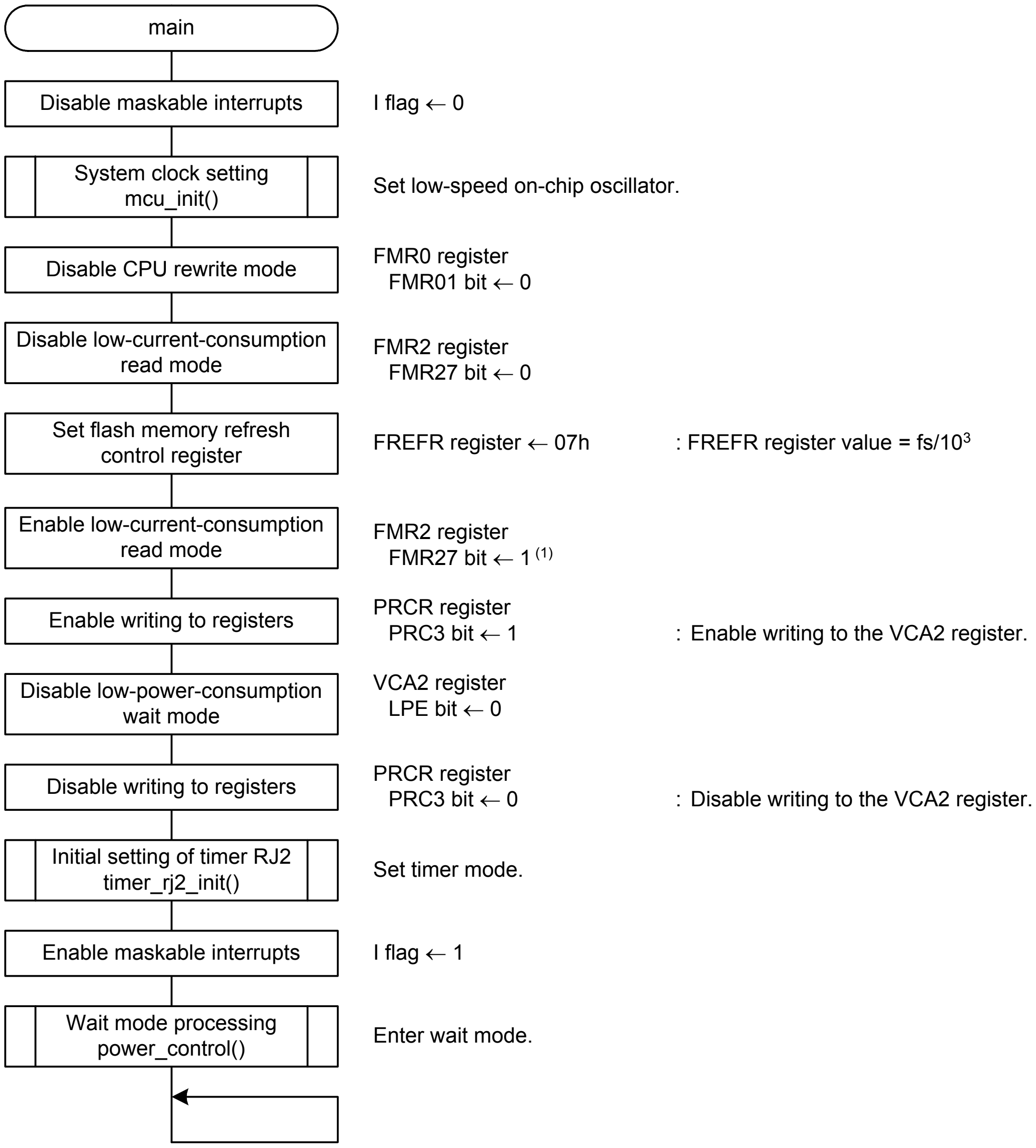
wait_control	
Outline	Wait mode processing
Header	None
Declaration	void power_control(void)
Description	Enter wait mode.
Argument	None
Returned value	None
Remark	—

_timer_rj2	
Outline	Timer RJ2 interrupt handling
Header	None
Declaration	void _timer_rj(void)
Description	Perform timer RJ2 interrupt handling.
Argument	None
Returned value	None
Remark	—

5.5 Flowcharts

5.5.1 Main Processing

Figure 5.2 shows the Main Processing.

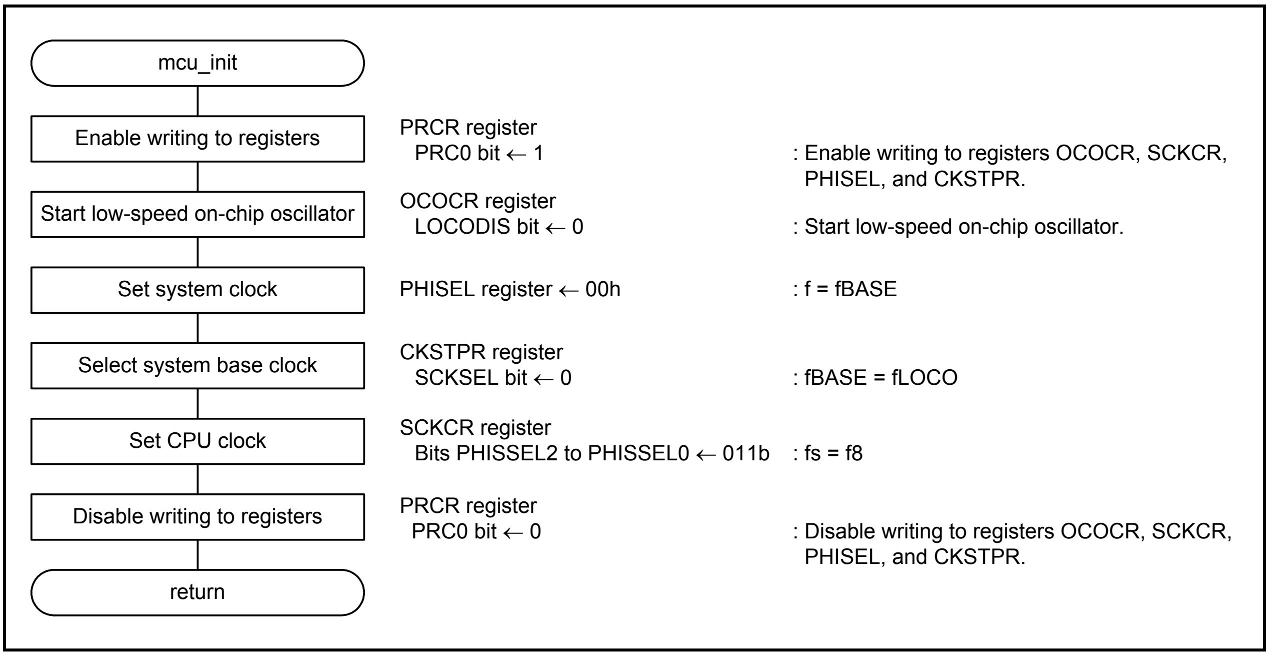

main
Disable maskable interrupts
I flag ← 0
System clock setting mcu_init()
Set low-speed on-chip oscillator.
Disable CPU rewrite mode
FMR0 register FMR01 bit ← 0
Disable low-current-consumption read mode
FMR2 register FMR27 bit ← 0
Set flash memory refresh control register
FREFR register ← 07h
: FREFR register value = fs/10^3
Enable low-current-consumption read mode
FMR2 register FMR27 bit ← 1 (1)
Enable writing to registers
PRCR register PRC3 bit ← 1
: Enable writing to the VCA2 register.
Disable low-power-consumption wait mode
VCA2 register LPE bit ← 0
Disable writing to registers
PRCR register PRC3 bit ← 0
: Disable writing to the VCA2 register.
Initial setting of timer RJ2 timer_rj2_init()
Set timer mode.
Enable maskable interrupts
I flag ← 1
Wait mode processing power_control()
Enter wait mode.

Note:
1. To set the FMR27 bit to 1, first write 0 and then write 1 immediately.

Figure 5.2 Main Processing

5.5.2 System Clock Setting

Figure 5.3 shows the System Clock Setting.


mcu_init
Enable writing to registers
Start low-speed on-chip oscillator
Set system clock
Select system base clock
Set CPU clock
Disable writing to registers
return

PRCR register
PRC0 bit ← 1 : Enable writing to registers OCOCR, SCKCR, PHISEL, and CKSTPR.

OCOCR register
LOCODIS bit ← 0 : Start low-speed on-chip oscillator.

PHISEL register ← 00h : f = fBASE

CKSTPR register
SCKSEL bit ← 0 : fBASE = fLOCO

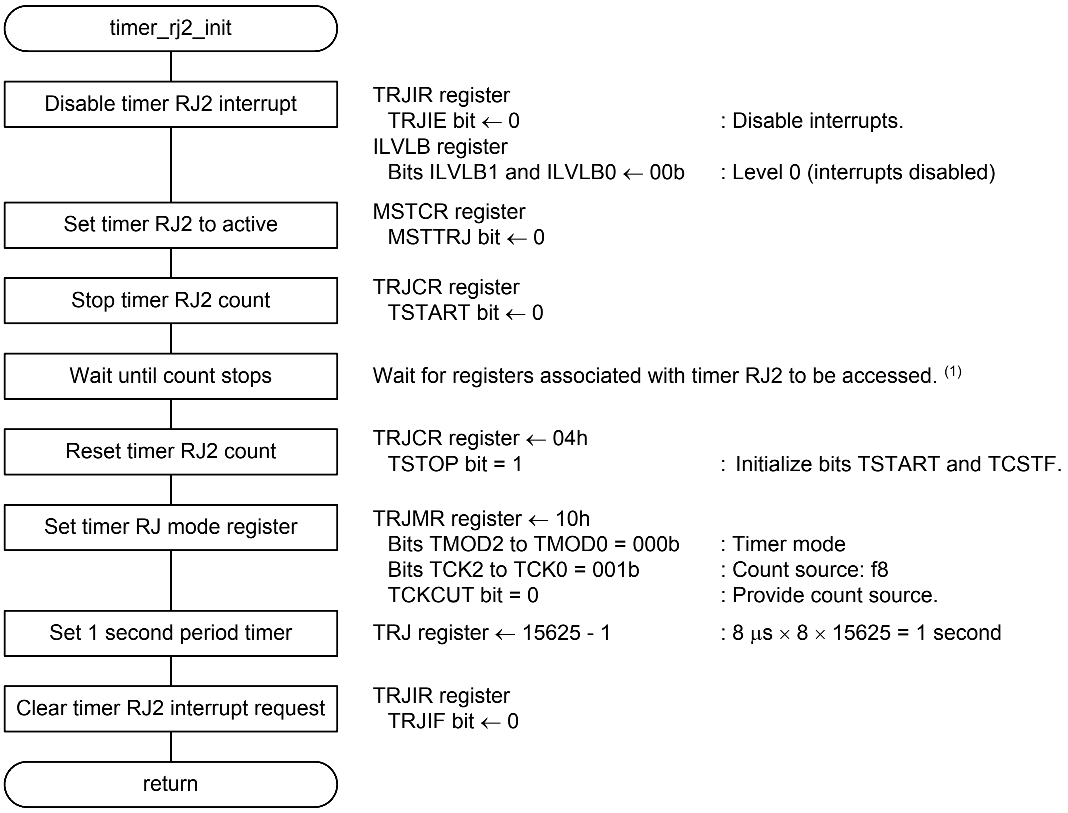
SCKCR register
Bits PHISSEL2 to PHISSEL0 ← 011b : fs = f8

PRCR register
PRC0 bit ← 0 : Disable writing to registers OCOCR, SCKCR, PHISEL, and CKSTPR.

Figure 5.3 System Clock Setting

5.5.3 Initial Setting of Timer RJ2

Figure 5.4 shows the Initial Setting of Timer RJ2.

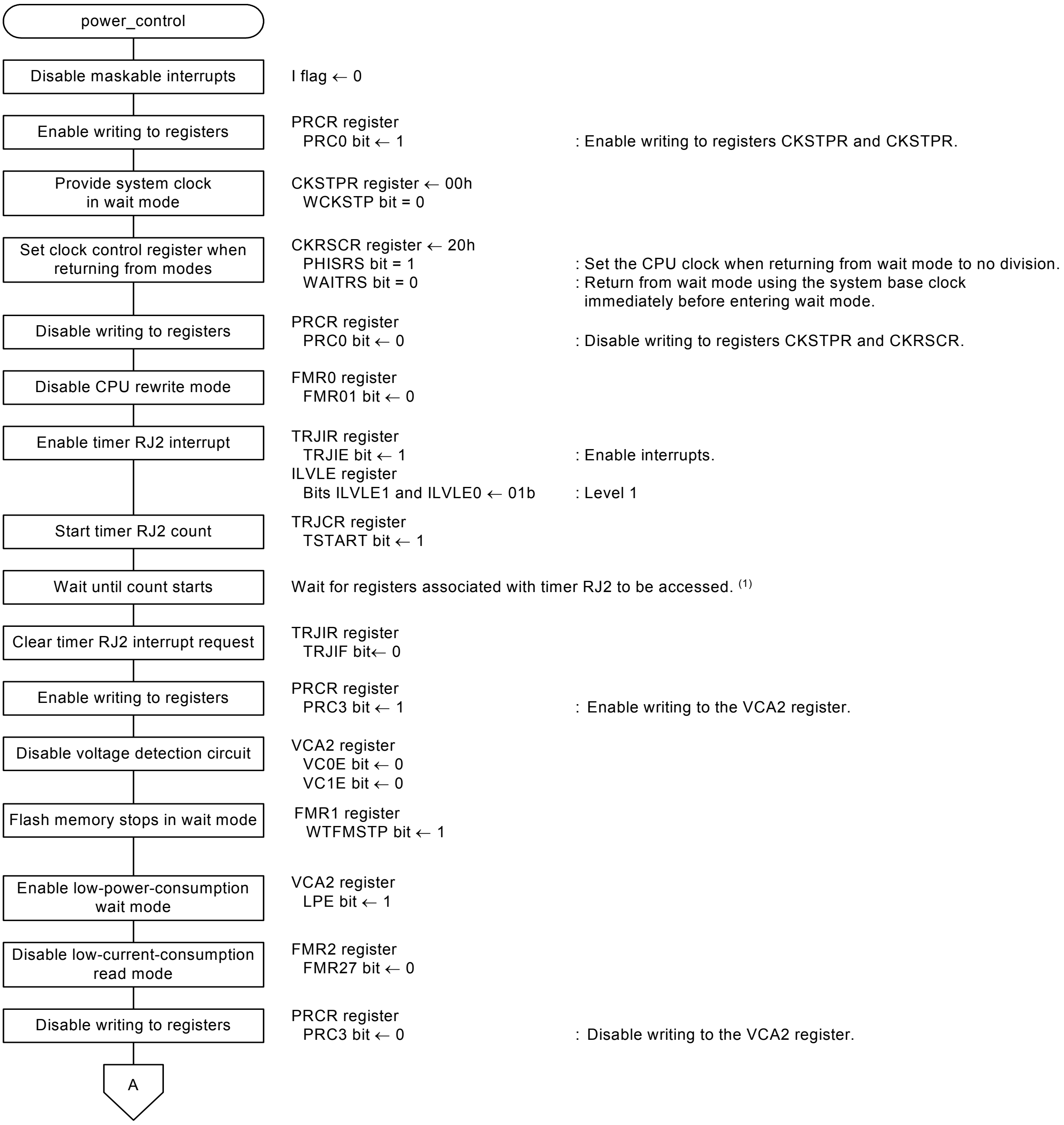

timer_rj2_init
Disable timer RJ2 interrupt
TRJIR register
TRJIE bit ← 0 : Disable interrupts.
ILVLB register
Bits ILVLB1 and ILVLB0 ← 00b : Level 0 (interrupts disabled)
Set timer RJ2 to active
MSTCR register
MSTTRJ bit ← 0
Stop timer RJ2 count
TRJCR register
TSTART bit ← 0
Wait until count stops
Wait for registers associated with timer RJ2 to be accessed. (1)
Reset timer RJ2 count
TRJCR register ← 04h
TSTOP bit = 1 : Initialize bits TSTART and TCSTF.
Set timer RJ mode register
TRJMR register ← 10h
Bits TMOD2 to TMOD0 = 000b : Timer mode
Bits TCK2 to TCK0 = 001b : Count source: f8
TCKCUT bit = 0 : Provide count source.
Set 1 second period timer
TRJ register ← 15625 - 1 : 8 µs × 8 × 15625 = 1 second
Clear timer RJ2 interrupt request
TRJIR register
TRJIF bit ← 0
return

Note:
1. Registers associated with timer RJ2: TRJ, TRJCR, and TRJMR

Figure 5.4 Initial Setting of Timer RJ2

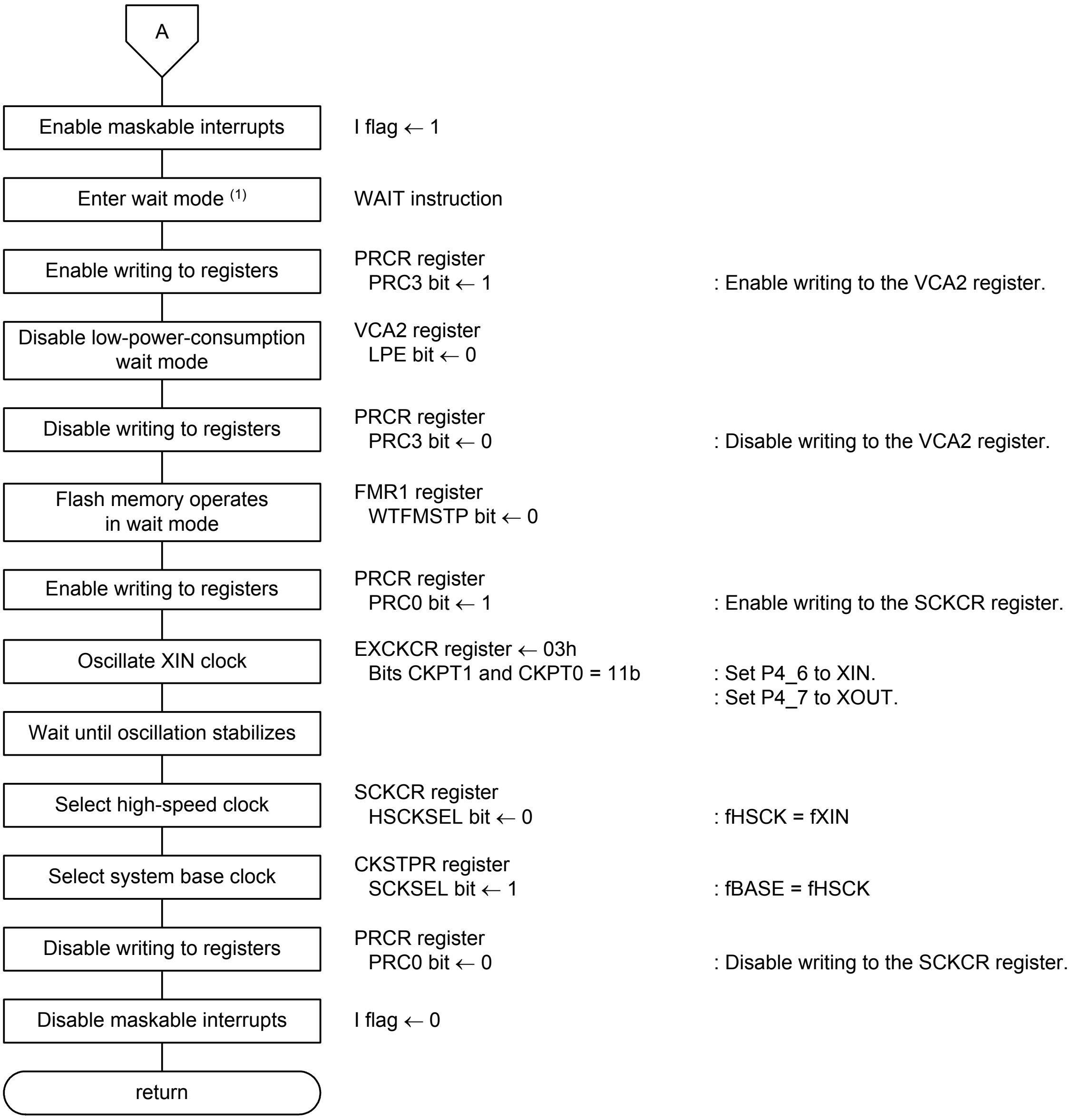
5.5.4 Wait Mode Processing

Figure 5.5 and 5.6 show Wait Mode Processing.


power_control
Disable maskable interrupts
I flag ← 0
Enable writing to registers
PRCR register
PRC0 bit ← 1
: Enable writing to registers CKSTPR and CKSTPR.
Provide system clock in wait mode
CKSTPR register ← 00h
WCKSTP bit = 0
Set clock control register when returning from modes
CKRSCR register ← 20h
PHISRS bit = 1
WAITRS bit = 0
: Set the CPU clock when returning from wait mode to no division.
: Return from wait mode using the system base clock immediately before entering wait mode.
Disable writing to registers
PRCR register
PRC0 bit ← 0
: Disable writing to registers CKSTPR and CKRSCR.
Disable CPU rewrite mode
FMR0 register
FMR01 bit ← 0
Enable timer RJ2 interrupt
TRJIR register
TRJIE bit ← 1
: Enable interrupts.
ILVLE register
Bits ILVLE1 and ILVLE0 ← 01b
: Level 1
Start timer RJ2 count
TRJCR register
TSTART bit ← 1
Wait until count starts
Wait for registers associated with timer RJ2 to be accessed. (1)
Clear timer RJ2 interrupt request
TRJIR register
TRJIF bit← 0
Enable writing to registers
PRCR register
PRC3 bit ← 1
: Enable writing to the VCA2 register.
Disable voltage detection circuit
VCA2 register
VC0E bit ← 0
VC1E bit ← 0
Flash memory stops in wait mode
FMR1 register
WTFMSTP bit ← 1
Enable low-power-consumption wait mode
VCA2 register
LPE bit ← 1
Disable low-current-consumption read mode
FMR2 register
FMR27 bit ← 0
Disable writing to registers
PRCR register
PRC3 bit ← 0
: Disable writing to the VCA2 register.
A

Note:
1. Registers associated with timer RJ2: TRJ, TRJCR, and TRJMR

Figure 5.5 Wait Mode Processing (1/2)


A
Enable maskable interrupts
I flag ← 1
Enter wait mode (1)
WAIT instruction
Enable writing to registers
PRCR register
PRC3 bit ← 1
: Enable writing to the VCA2 register.
Disable low-power-consumption wait mode
VCA2 register
LPE bit ← 0
Disable writing to registers
PRCR register
PRC3 bit ← 0
: Disable writing to the VCA2 register.
Flash memory operates in wait mode
FMR1 register
WTFMSTP bit ← 0
Enable writing to registers
PRCR register
PRC0 bit ← 1
: Enable writing to the SCKCR register.
Oscillate XIN clock
EXCKCR register ← 03h
Bits CKPT1 and CKPT0 = 11b
: Set P4_6 to XIN.
: Set P4_7 to XOUT.
Wait until oscillation stabilizes
Select high-speed clock
SCKCR register
HSCKSEL bit ← 0
: fHSCK = fXIN
Select system base clock
CKSTPR register
SCKSEL bit ← 1
: fBASE = fHSCK
Disable writing to registers
PRCR register
PRC0 bit ← 0
: Disable writing to the SCKCR register.
Disable maskable interrupts
I flag ← 0
return

Note:

1. Since the 4 bytes are prefetched from the WAIT instruction and then the program stops, insert at least four NOP instructions as a program restriction when entering wait mode.

Figure 5.6 Wait Mode Processing (2/2)

5.5.5 Timer RJ2 Interrupt Handling

Figure 5.7 shows Timer RJ2 Interrupt Handling.

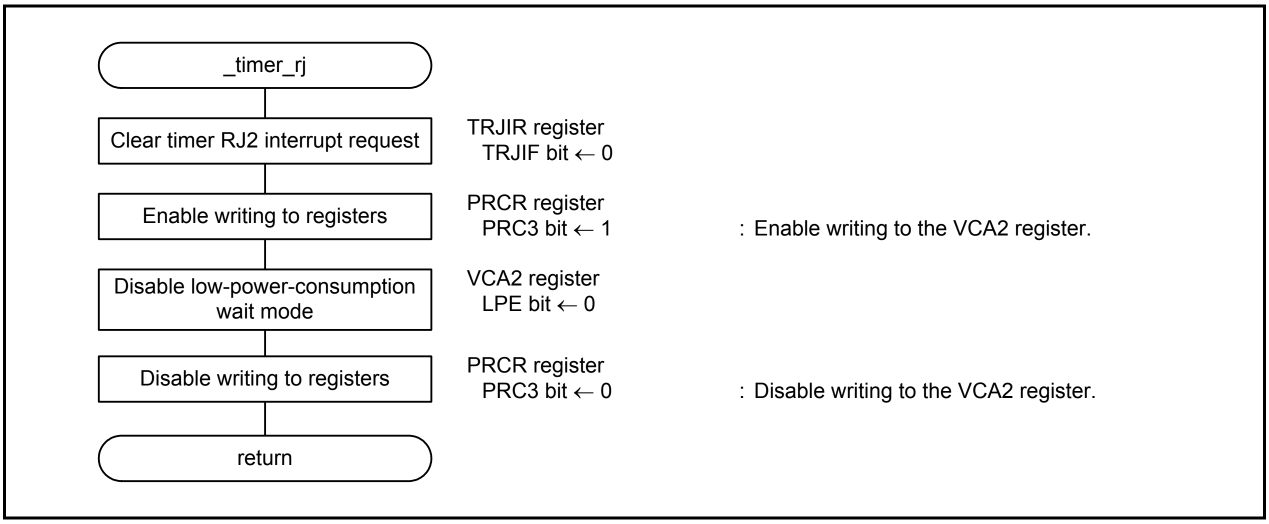

_timer_rj
Clear timer RJ2 interrupt request
TRJIR register
TRJIF bit ← 0
Enable writing to registers
PRCR register
PRC3 bit ← 1
: Enable writing to the VCA2 register.
Disable low-power-consumption wait mode
VCA2 register
LPE bit ← 0
Disable writing to registers
PRCR register
PRC3 bit ← 0
: Disable writing to the VCA2 register.
return

Figure 5.7 Timer RJ2 Interrupt Handling

6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

7. Reference Documents

R8C/M12A Group User's Manual: Hardware Rev.1.00
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.

C Compiler Manual
M16C Series, R8C Family C Compiler Package V.5.45
C Compiler User's Manual Rev.2.00
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/inquiry

Revision History	R8C/M12A Group Power Control in Wait Mode

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 16, 2011	—	First edition issued

All trademarks and registered trademarks are the property of their respective owners.

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 manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins

   Handle unused pins in accord 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 one with a different part number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. 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.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.
6. 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.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. 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 categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
   "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.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. 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.
9. 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 system 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. 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.
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: +44-1628-585-100, Fax: +44-1628-585-900

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 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

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.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

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.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2011 Renesas Electronics Corporation. All rights reserved.

Colophon 1.1