RL78/G13

R01AN3128EJ0111
Rev.1.11
May. 20, 2019

CPU Clock Changing and Standby Settings (C Language) CC-RL

Introduction

This application note describes how to change the RL78/G13's CPU clock and set it to standby (changing operation modes).

This application uses switch input to change the CPU clock and the operation mode, while controlling 4 LEDs to indicate the CPU clock status and the operation mode.

Target Device

RL78/G13

When applying the sample program covered in this application note to another microcomputer, modify the program according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.

Contents

1. Specifications

This application describes how to switch the CPU clock and operation mode using switch input, as shown in Figure 1.1 Operating Mode Status Transition Diagram.

In addition, the application controls four LEDs to indicate the status of the CPU clock and the operation mode.

The Peripheral Functions and Applications used in this application note, Operating Mode Status Transition Diagram, and Operation Modes and Corresponding LED Status are show in Table 1.1, Figure 1.1, and Table 1.2, correspondingly.

Table 1.1 Peripheral Functions and Applications

Peripheral Function	Application
Port output	Controls LEDs (LED0-LED3) connected to pins P52, P53, P62, and P63.
External interrupt	Interrupt (INTP1) that detects a pin input edge according to switch input (SW1).
12-bit interval timer	Interrupt (INTIT) that detects an interval signal from the 12-bit interval timer
A/D converter	Converts analog signal input level of the P26/ANI6 pin.

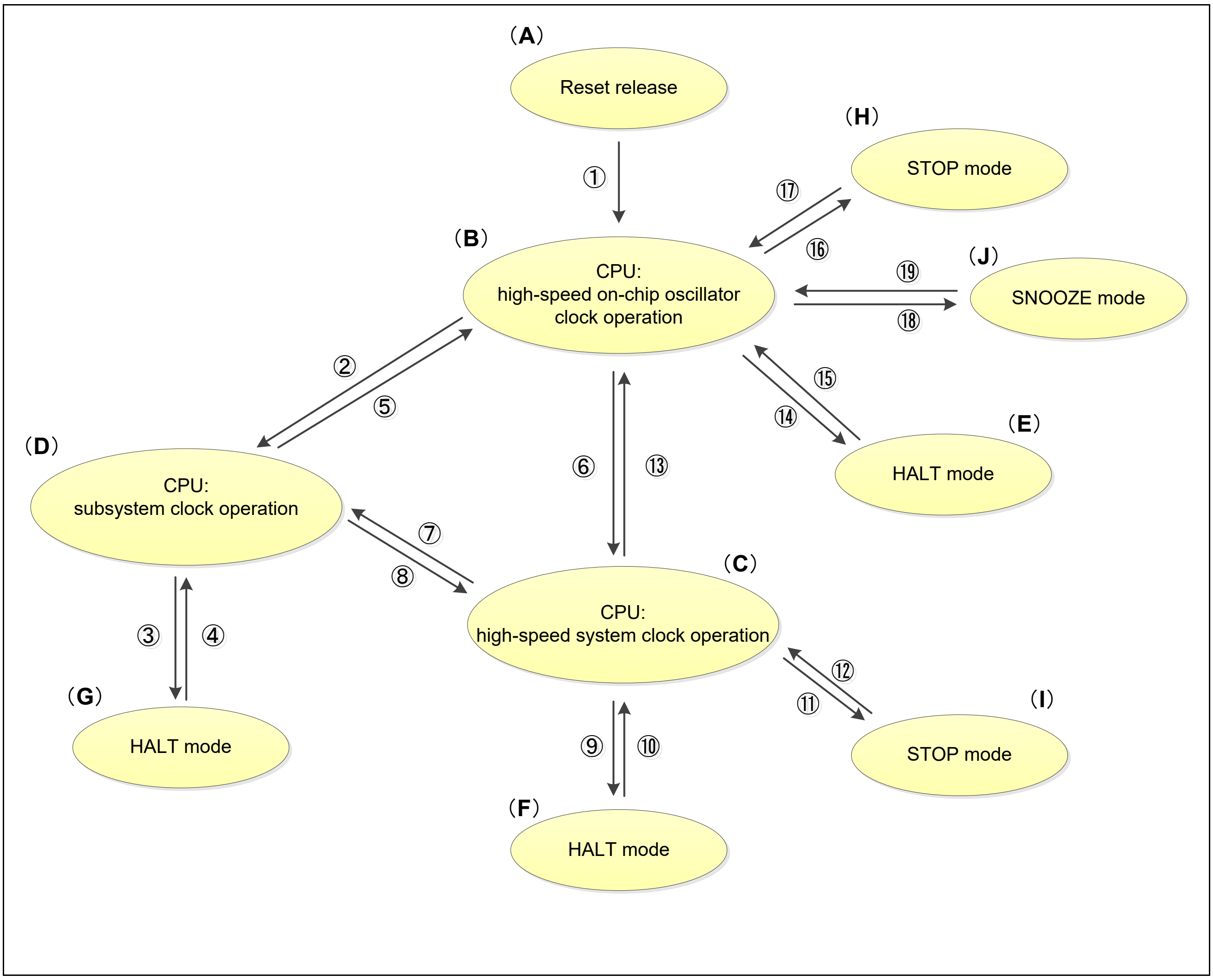


Figure 1.1 Operating Mode Status Transition Diagram

Table 1.2 Operation Modes and Corresponding LED Status

CPU/Peripheral Hardware Clock (f_{CLK})	Operation mode	LED Status			
		LED0	LED1	LED2	LED3
High-speed on-chip oscillator clock (f_{IH})	Normal operation mode	ON	ON	ON	ON
	HALT mode	ON	ON	ON	OFF
	SNOOZE mode	ON	ON	OFF	ON
	STOP mode	ON	ON	OFF	OFF
High-speed system clock (f_{MX})	Normal operation mode	ON	OFF	ON	ON
	HALT mode	ON	OFF	ON	OFF
	STOP mode	ON	OFF	OFF	OFF
Subsystem clock (f_{SUB})	Normal operation mode	OFF	ON	ON	ON
	HALT mode	OFF	ON	ON	OFF

1.1 CPU Clock Changes

This section describes the special function register (SFR) settings required for changing the CPU clock.

- Changing from high-speed on-chip oscillator clock to high-speed system clock
- Changing from high-speed on-chip oscillator clock to subsystem clock
- Changing from high-speed system clock to high-speed on-chip oscillator clock
- Changing from high-speed system clock to subsystem clock
- Changing from subsystem clock to high-speed on-chip oscillator clock
- Changing from subsystem clock to high-speed system clock

1.1.1 Changing from high-speed on-chip oscillator clock to high-speed system clock

When changing the CPU clock from the high-speed on-chip oscillator clock to the high-speed system clock, set the oscillator and start oscillation using the following registers: clock operation mode control register (CMC), oscillation stabilization time select register (OSTS), clock operation status control register (CSC). Next, wait for the oscillation to stabilize using the oscillation stabilization time counter status register (OSTC).

After the oscillation stabilizes, set the high-speed system clock to fCLK using the system clock control register (CKC).

Confirm that the status of the main system clock has changed to the high-speed system clock, and then stop the high-speed on-chip oscillator.

① Set (1) the OSCSEL bit of the CMC register (when fx > 10MHz, set (1) the AMPH bit) to operate the X1 oscillator. Set (1) the EXCLK bit and OSCSEL bit when using the external clock.

	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS	OSCSELS	0	AMPHS1	AMPHS0	AMPH
	0	**1**	x	x	0	x	x	**0/1**

AMPH bit: clear to 0 when the X1 oscillation clock is 10 MHZ or lower.

② Using the OSTS register, select the oscillation stabilization time of the X1 oscillation circuit. This setting does not have to exist at the time the external clock is used.

Example: Set the following values for a wait of at least 102 μs based on a 10 MHz resonator.

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0
	0	0	0	0	0	**0**	**1**	**0**

③ Clear (0) the MSTOP bit of the CSC register to start oscillating the X1 oscillator. After the external clock is input to the external clock signal, to clear (0) the MSTOP bit.

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP
	0	x	0	0	0	0	0	0

④ Use the OSTC register to wait for oscillation of the X1 oscillator to stabilize. External clock is not required oscillation stabilization wait.

Example: Wait until the bits reach the following values for a wait of at least 102 μs based on a 10 MHz resonator.

	7	6	5	4	3	2	1	0
OSTC	MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18
	1	**1**	**1**	**0**	**0**	**0**	**0**	**0**

⑤ Set the MCM0 bit of the CKC register to specify the X1 oscillation clock as the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	0	0
	0	0	0	**1**	0	0	0	0

Register setting values:
x: unused bit; blank space; unchanged bit; -: reserved bits or unassigned bit

⑥ Wait for the MCS bit of the CKC register to switch, then set (1) HIOSTOP and stop the high-speed on-chip oscillator clock.

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP
	0	x	0	0	0	0	0	**1**

Register setting values:

x: unused bit; blank space; unchanged bit; -: reserved bits or unassigned bit

1.1.2 Changing from high-speed on-chip oscillator clock to subsystem clock

When changing the CPU clock from the high-speed on-chip oscillator clock to the subsystem clock, set the oscillator and start oscillation using the following registers: subsystem clock supply mode control register (OSMC), clock operation mode control register (CMC), clock operation status control register (CSC). Next, wait for the oscillation to stabilize using the timer or similar function. After the oscillation stabilizes, set the subsystem clock to f_{CLK} using the system clock control register (CKC). Confirm that the status of the CPU/peripheral hardware clock has switched to the subsystem clock, and then stop the high-speed on-chip oscillator.

① In this application, a 12-bit interval timer is used to count the oscillation stabilization time for the subsystem clock resonator. Set the WUTMMCK0 to 1 to use the low-speed on-chip oscillator clock as the count clock for the 12-bit interval timer. To run only the real-time clock and 12-bit interval timer on the subsystem clock in the STOP mode or HALT mode (during CPU operation with the subsystem clock), set the RTCLPC bit to 1.

	7	6	5	4	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	0	0	0	0
	x	0	0	**1**	0	0	0	0

② Select XT1 oscillation mode using the CMC register. Set (1) the EXCLKS bit and OSCSELS bit when using the external clock.

	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS	OSCSELS	0	AMPHS1	AMPHS0	AMPH
	x	x	**0/1**	**1**	0	**0/1**	**0/1**	x

③ Clear (0) the XTSTOP bit of the CSC register to start oscillating the XT1 oscillator. After the external clock is input to the external clock signal, to clear (0) the XTSTOP bit.

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP
	x	**0**	0	0	0	0	0	0

④ Use the timer function or another function to wait for oscillation of the subsystem clock oscillator to stabilize using software. This application note is counted using a 12-bit interval timer. External clock is not required oscillation stabilization wait.

⑤ Set the CSS bit of the CKC register to specify the subsystem clock as the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	0	0
	0	**1**	0	0	0	0	0	0

⑥ Confirm that the CLS bit of the CKC register has changed to 1, then set (1) HIOSTOP and stop the high-speed on-chip oscillator clock.

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP
	x	0	0	0	0	0	0	**1**

Register setting values:

x: unused bit; blank space; unchanged bit; -: reserved bits or unassigned bit

1.1.3 Changing from high-speed system clock to high-speed on-chip oscillator clock

When changing the CPU clock from the high-speed system clock to the high-speed on-chip oscillator clock, start the oscillation using the clock operation status control register (CSC), then wait for the oscillation to stabilize using the timer function or another function. After the oscillation stabilization time has elapsed, set the high-speed on-chip oscillator clock to f_{CLK} using the system clock control register (CKC). Confirm that the status of the main system clock status has switched to the high-speed on-chip oscillator clock, and then stop the X1 oscillator.

① Clear (0) the HIOSTOP bit of the CSC register, and then start oscillating the high-speed on-chip oscillator.

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP
	0	x	0	0	0	0	0	**0**

② Use a software wait to wait for the oscillation of the high-speed on-chip oscillator to stabilize. Count the wait time (oscillation stabilization time: 18μs to 65μs) using the timer function or another function. This application note is counted using a 12-bit interval timer.

③ Set the MCM0 bit of the CKC register to specify the high-speed on-chip oscillator clock as the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	0	0
	0	0	1	**0**	0	0	0	0

④ After confirming that the MCS bit of the CKC register has switched, set (1) the MSTOP bit and stop the oscillating the X1 oscillator.

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP
	1	x	0	0	0	0	0	0

Register setting values:
x: unused bit; blank space; unchanged bit; -: reserved bits or unassigned bit

1.1.4 Changing from high-speed system clock to subsystem clock

When changing the CPU clock from the high-speed system clock to the subsystem clock, start the oscillation using the subsystem clock supply mode control register (OSMC) and the clock operation status control register (CSC), then wait for the oscillation to stabilize using the timer function or another function. After the oscillation stabilization time has elapsed, set the subsystem clock to f_{CLK} using the system clock control register (CKC). Confirm that the status of the CPU/peripheral hardware clock has switched to the subsystem clock, and then stop the X1 oscillator.

① To run only the real-time clock and 12-bit interval timer on the subsystem clock in the STOP mode or HALT mode (during CPU operation with the subsystem clock), set the RTCLPC bit to 1.

	7	6	5	4	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	0	0	0	0
	x	0	0	**1**	0	0	0	0

② Clear (0) the XSTOP bit of the CSC register and start oscillating the XT1 oscillator. After the external clock is input to the external clock signal, to clear (0) the XTSTOP bit.

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP
	0	**0**	0	0	0	0	0	1

③ Use the timer function or another function to wait for oscillation of the subsystem clock oscillator to stabilize using software. This application note is counted using a 12-bit interval timer. External clock is not required oscillation stabilization wait.

④ Set the CSS bit of the CKC register to specify the subsystem clock as the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	0	0
	0	**1**	1	1	0	0	0	0

⑤ After confirming that the CLS bit of the CKC register has switched, set (1) the MSTOP bit and stop the oscillation of the X1 oscillator.

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP
	1	0	0	0	0	0	0	1

Register setting values:
x: unused bit; blank space; unchanged bit; -: reserved bits or unassigned bit

1.1.5 Changing from subsystem clock to high-speed on-chip oscillator clock

When changing the CPU clock from the subsystem clock to the high-speed on-chip oscillator clock, start the oscillation using the clock operation status control register (CSC), then wait for the oscillation to stabilize using the timer function or another function. After the oscillation stabilization time has elapsed, set the high-speed on-chip oscillator clock to f_{CLK} using the system clock control register (CKC). Confirm that the status of the CPU/peripheral hardware clock has switched to the high-speed on-chip oscillator clock, and then stop the XT1 oscillator.

① Clear (0) the HIOSTOP bit of the CSC register, and start oscillating the high-speed on-chip oscillator.

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP
	x	0	0	0	0	0	0	**0**

② Use a software wait to wait for the oscillation of the high-speed on-chip oscillator to stabilize. Count the wait time (oscillation stabilization time: 18μs to 65μs) using the timer function or another function. This application note is counted using a 12-bit interval timer.

③ Set the CSS bit of the CKC register to specify the high-speed on-chip oscillator clock as the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0[Note]	0	0	0	0
	1	**0**	0	0	0	0	0	0

④ After confirming that the CLS bit of the CKC register has switched, clear (0) the XTSTOP bit and stop the oscillation of the XT1 oscillator.

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP
	x	**1**	0	0	0	0	0	0

Note: Changing the value of the MCM0 bit is prohibited while the CPU/peripheral hardware clock is operating with the subsystem clock

Register setting values:
x: unused bit; blank space; unchanged bit; -: reserved bits or unassigned bit

1.1.6 Changing from subsystem clock to high-speed system clock

When changing the CPU clock from the subsystem clock to the high-speed system clock, start the oscillation using the clock operation mode control register (CMC) and the clock operation status control register (CSC), then use the oscillation stabilization timer count register (OSTC) to wait for the oscillation to stabilize. After the oscillation stabilization time has elapsed, set the high-speed system clock to f_{CLK} using the system clock control register (CKC). Confirm that the status of the main system clock has changed to the high-speed system clock, and then stop the XT1 oscillator.

① Using the OSTC register, select the oscillation stabilization time for the X1 oscillator. This setting does not have to exist at the time the external clock is used.

Example: Set the following values when a wait of at least 102 μs is set based on a 10 MHz resonator.

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0
	0	0	0	0	0	**0**	**1**	**0**

② Clear (0) the MSTOP bit of the CSC register and start oscillating theX1 oscillator. After the external clock is input to the external clock signal, to clear (0) the MSTOP bit.

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP
	0	0	0	0	0	0	0	x

③ Use the OSTC register to wait for oscillation of the X1 oscillator to stabilize. External clock is not required oscillation stabilization wait.

Example: Wait until the bits reach the following values for a wait of at least 102 μs based on a 10 MHz resonator.

	7	6	5	4	3	2	1	0
OSTC	MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18
	1	**1**	**1**	**0**	**0**	**0**	**0**	**0**

④ Set the CSS bit of the CKC register to specify the high-speed system clock as the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0[Note]	0	0	0	0
	1	**0**	1	1	0	0	0	0

⑤ After confirming that the CLS bit of the CKC register has switched, set (1) the XTSTOP bit and stop the oscillation of the XT1 oscillator.

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP
	0	**1**	0	0	0	0	0	x

Note: Changing the value of the MCM0 bit is prohibited while the CPU/peripheral hardware clock is operating with the subsystem clock

Register setting values:
x: unused bit; blank space; unchanged bit; -: reserved bits or unassigned bit

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	RL78/G13 (R5F100LEA)
Operating frequencies	● High-speed on-chip oscillator clock: 32MHz ● High-speed system clock: 20MHz ● Subsystem clock: 32.768kHz ● CPU/peripheral hardware clock: 32MHz/20MHz/32.768kHz^Note
Operating voltage	5.0V (operating range 2.9V to 5.5V) LVD operations (V_{LVD}): reset mode 2.81V (2.76V to 2.87V)
Integrated development environment (CS+)	CS+ for CC V4.00.00 from Renesas Electronics Corp.
C compiler (CS+)	CC-RL V1.02.00 from Renesas Electronics Corp.
Integrated development environment (e^2 studio)	e^2 studio V4.3.0.008 from Renesas Electronics Corp.
C compiler (e^2 studio)	CC-RL V1.02.00 from Renesas Electronics Corp.
Board used	Renesas Electronics Corp. RL78/G13 Starter Kit (R0K50100LS000BE/900BE)

Note: CPU/peripheral hardware clock settings are changed in the application.

3. Related Application Notes

Application notes related to this document are shown below. Please refer to these as needed.

RL78/G13 Initialization (R01AN2575J) Application Note

4. Hardware Explanation

4.1 Hardware Configuration Example

Figure 4.1 shows an example of the hardware configuration used in this application note.

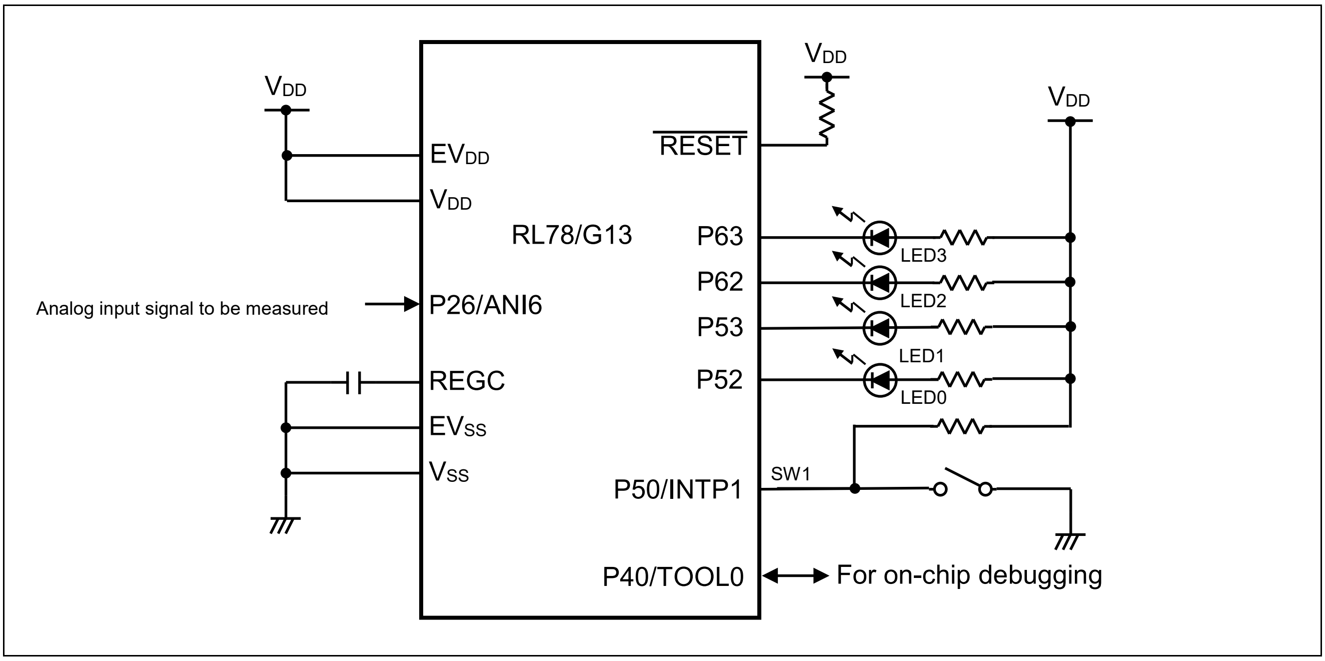


Figure 4.1 Hardware Configuration

Note: 1.This simplified circuit diagram was created to show an overview of connections only. When actually designing your circuit, make sure the design includes sufficient pin processing and meets electrical characteristic requirements.
(Connect each input-only port to V_{DD} or V_{SS} through a resistor.)

2. If a pin name starts with EV_{SS}, connect the pin to V_{SS}, if it starts with EV_{DD}, connect it to V_{DD}.
3. Make V_{DD} higher than the RESET release voltage (V_{LVD}) set in LVD.

4.2 Used Pin List

Table 4.1 provides List of Pins and Functions

Table 4.1 List of Pins and Functions

Pin Name	Input/Output	Function
P50	Input	Switch (SW1) input port
P26/AIN6	Input	A/D converter analog input port
P52	Output	LED (LED0) control port
P53	Output	LED (LED1) control port
P62	Output	LED (LED2) control port
P63	Output	LED (LED3) control port

5. Software Explanation

5.1 Operation Outline

This application enables the user to change the CPU clock and the operation mode using switch input. Steps 1 through 19 below describe how to change the clock and mode.

Sequence for changing CPU clock and operation mode:

1. High-speed on-chip oscillator clock: CPU operation (B)
2. High-speed on-chip oscillator clock: CPU operation (B) → subsystem clock: CPU operation (D)
3. Subsystem clock: CPU operation (D) → subsystem clock: HALT mode (G)
4. Subsystem clock: HALT mode (G) → subsystem clock: CPU operation (D)
5. Subsystem clock: CPU operation (D) → high-speed on-chip oscillator clock: CPU operation (B)
6. High-speed on-chip oscillator clock: CPU operation (B) → high-speed system clock: CPU operation (C)
7. High-speed system clock: CPU operation (C) → subsystem clock: CPU operation (D)
8. Subsystem clock: CPU operation (D) → high-speed system clock: CPU operation (C)
9. High-speed system clock: CPU operation (C) → High-speed system clock: HALT mode (F)
10. High-speed system clock: HALT mode (F) → high-speed system clock: CPU operation (C)
11. High-speed system clock: CPU operation (C) → STOP mode (I)
12. STOP mode (I) → High-speed system clock (C)
13. High-speed system clock: CPU operation (C) → high-speed on-chip oscillator clock: CPU operation (B)
14. High-speed on-chip oscillator clock: CPU operation (B) → high-speed on-chip oscillator clock: HALT mode (E)
15. High-speed on-chip oscillator clock: HALT mode (E) → high-speed on-chip oscillator clock: CPU operation (B)
16. High-speed on-chip oscillator clock: CPU operation (B) → STOP mode (H)
17. STOP mode (H) → high-speed on-chip oscillator clock: CPU operation (B)
18. High-speed on-chip oscillator clock: CPU operation (B) → SNOOZE mode (transition from STOP mode to SNOOZE mode) (J)
19. SNOOZE mode (J) → high-speed on-chip oscillator clock: CPU operation (B)

(1) Input/output port initialization

- P52-P53 and P62-P63 pins: set as output ports (use to control LEDs)
- P50/INTP1 pin: set as input port (use for switch input)
- P26/ANI6 pin: set as analog input port (use as A/D conversion analog input channel)

(2) Clock generator initialization

<Setting conditions>

- Set the flash operation mode to HS (high-speed main) mode using user option byte (000C2H/010C2H).)
- High-speed on-chip oscillator clock frequency: set to 32 MHz
- Set the operation mode of the subsystem clock pin to XT1 oscillation, and connect a crystal resonator to the XT1/123 and XT2/EXCLKS/P124 pins.
- Set the oscillation mode of the XT1 oscillator to ultra-low power consumption oscillation. (Select the optimal oscillation mode for the oscillator connected to the board.)
- Set the operation mode of the high-speed system clock pin to X1 oscillation, and connect a crystal resonator to the X1/P121 and X2/EXCLK/P122 pins.
- Select the main system clock (f_{MAIN}) as the CPU/peripheral hardware clock (f_{CLK}).

(3) Interrupt processing initialization

- Set the INTP1 pin valid edge to falling edge and enable switch input.
- Use the 12-bit interval timer to confirm switch input. The voltage level of the pin is checked approximately every 5 ms. If the voltage level matches twice consecutively, the switch input is recognized as valid (prevents chattering).

(4) The CPU clock and operation mode change as follows each time the falling edge of a signal (switch) input to the P50/INTP1 pin is detected.

① Wait for switch input after LED lighting control is complete (LED0: ON, LED1: ON, LED2: ON, LED3: ON)

② Change CPU clock from high-speed on-chip oscillator clock to subsystem clock, then implement LED lighting control (LED0: OFF, LED1: ON, LED2: ON, LED3: ON), and wait for switch input.

③ After LED lighting control is complete (LED0: OFF, LED1: ON, LED2: ON, LED3: OFF), transition to HALT mode, then wait for switch input.

④ Implement LED lighting control (LED0: OFF, LED1: ON, LED2: ON, LED3: ON), then wait for switch input.

⑤ Change CPU clock from subsystem clock to high-speed on-chip oscillator clock, implement LED lighting control (LED0: ON, LED1: ON, LED2: ON, LED3: ON), then wait for switch input.

⑥ Change CPU clock from high-speed on-chip oscillator clock to high-speed system clock, implement LED lighting control (LED0: ON, LED1: OFF, LED2: ON, LED3: ON), then wait for switch input.

⑦ Change CPU clock from high-speed system clock to subsystem clock, implement LED lighting control (LED0: OFF, LED1: ON, LED2: ON, LED3: ON), then wait for switch input.

⑧ Change CPU clock from subsystem clock to high-speed system clock, implement LED lighting control (LED0: ON, LED1: OFF, LED2: ON, LED3: ON), then wait for switch input.

⑨ Implement LED lighting control (LED0: ON, LED1: OFF, LED2: ON, LED3: OFF), transition to HALT mode, then wait for switch input.

⑩ Implement LED lighting control (LED0: ON, LED1: OFF, LED2: ON, LED3: ON), then wait for switch input.

⑪ Implement LED lighting control (LED0: ON, LED1: OFF, LED2: OFF, LED3: OFF), transition to STOP mode, then wait for switch input.

⑫ Implement LED lighting control (LED0: ON, LED1: OFF, LED2: ON, LED3: ON), then wait for switch input.

⑬ Change CPU clock from high-speed system clock to high-speed on-chip oscillator clock, implement LED lighting control (LED0: ON, LED1: ON, LED2: ON, LED3: ON), then wait for switch input.

⑭ Implement LED lighting control (LED0: ON, LED1: OFF, LED2: ON, LED3: OFF), transition to HALT mode, then wait for switch input.

⑮ Implement LED lighting control (LED0: ON, LED1: ON, LED2: ON, LED3: ON), then wait for switch input.

⑯ Implement LED lighting control (LED0: ON, LED1: OFF, LED2: OFF, LED3: OFF), transition to STOP mode, then wait for switch input.

⑰ Implement LED lighting control (LED0: ON, LED1: ON, LED2: ON, LED3: ON), then wait for switch input.

⑱ S Implement LED lighting control (LED0: ON, LED1: ON, LED2: OFF, LED3: ON), transition to SNOOZE mode, then wait for generation of A/D conversion interrupt.

⑲ Implement LED lighting control (LED0: ON, LED1: ON, LED2: ON, LED3: ON), then wait for switch input.

After the CPU clock and operation mode have been changed according to steps 1 to 19 above, the falling edge of a signal (switch) input to the P50/INTP1 pin is detected, all LEDs are turned OFF, and the CPU goes to HALT mode (only RESET input in standby recovery).

In addition, if the CPU clock can't be status transition to a certain period of time such as by oscillation failure of the crystal oscillator is all LEDs are turned OFF and end the status transition in error processing.

Note: Refer to the RL78/G13 User's Manual for usage notes concerning this device.

5.2 Option Byte Settings

Table5.1 lists the option byte settings.

Table 5.1 Option Byte Settings

Address	Setting Value	Contents
000C0H/010C0H	01101110B	Watchdog timer operation is stopped (count is stopped after reset)
000C1H/010C1H	01111111B	LVD operation (V_{LVD}): reset mode Detection voltage: Rising edge 2.81 V/falling edge 2.75 V
000C2H/010C2H	11101000B	HS mode, HOCO: 32 MHz
000C3H/010C3H	10000100B	On-chip debugging enabled

5.3 Variables

Table5.2 lists the global variables.

Table 5.2 Variables

Type	Variable Name	Contents	Function Used
8-bit	g_int_cnt	Number of interval signal detection interrupts for12-bit interval timer	R_MAIN_NOP_Loop, r_intit_interrupt
8-bit	g_int_flg	Confirm the external interrupt generation detection flag	R_MAIN_BtoD, R_MAIN_DtoB, R_MAIN_BtoC, R_MAIN_CtoD, R_MAIN_DtoC, R_MAIN_CtoB, r_intc1_interrupt

5.4 Functions (subroutines)

Table 5.3 lists the functions (subroutines).

Table 5.3 Functions

Function Name	Outline	Numbers of operating mode status transition diagram
R_MAIN_AtoB	Status transition processing from (A) to (B)	①
R_MAIN_BtoD	Status transition processing from (B) to (D)	②
R_MAIN_DtoG	Status transition processing from (D) to (G)	③
R_MAIN_GtoD	Status transition processing from (G) to (D)	④
R_MAIN_DtoB	Status transition processing from (D) to (B)	⑤
R_MAIN_BtoC	Status transition processing from (B) to (C)	⑥
R_MAIN_CtoD	Status transition processing from (C) to (D)	⑦
R_MAIN_DtoC	Status transition processing from (D) to (C)	⑧
R_MAIN_CtoF	Status transition processing from (C) to (F)	⑨
R_MAIN_FtoC	Status transition processing from (F) to (C)	⑩
R_MAIN_CtoI	Status transition processing from (C) to (I)	⑪
R_MAIN_ItoC	Status transition processing from (I) to (C)	⑫
R_MAIN_CtoB	Status transition processing from (C) to (B)	⑬
R_MAIN_BtoE	Status transition processing from (B) to (E)	⑭
R_MAIN_EtoB	Status transition processing from (E) to (B)	⑮
R_MAIN_BtoH	Status transition processing from (B) to (H)	⑯
R_MAIN_HtoB	Status transition processing from (H) to (B)	⑰
R_MAIN_BtoJ	Status transition processing from (B) to (J)	⑱
R_MAIN_JtoB	Status transition processing from (J) to (B)	⑲
R_MAIN_NOP_Loop	Continuous NOP instruction execution processing	-
R_MAIN_END	End processing of status transition	-
R_MAIN_ERROR	Error processing of status transition	-
R_MAIN_Set_SnoozeOn	A/D converter setting	-
r_intc1_interrupt	Confirm the external interrupt generation detection flag update processing	-
r_intit_interrupt	12-bit interval timer interval signal detection interrupt count processing	-
r_adc_interrupt	SNOOZE mode release processing	-

5.5 Function (subroutine) Specifications

The following are the sample code functions (subroutines) used in this application note.

[Function Name] R_MAIN_AtoB

Outline	Status transition processing from (A) to (B)
Declaration	void R_MAIN_AtoB(void)
Description	Control LED lighting. (CPU clock: high-speed on-chip oscillator clock)
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_BtoD

Outline	Status transition processing from (B) to (D)
Declaration	void R_MAIN_BtoD(void)
Description	Change the CPU clock from high-speed on-chip oscillator clock to subsystem clock. After the clock is switched, control LED lighting.
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_DtoG

Outline	Status transition processing from (D) to (G)
Declaration	void R_MAIN_DtoG(void)
Description	Control LED lighting, transition to HALT mode. (CPU clock stopped (when using subsystem clock))
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_GtoD

Outline	Status transition processing from (G) to (D)
Declaration	void R_MAIN_GtoD(void)
Description	Control LED lighting (CPU clock: subsystem clock)
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_DtoB

Outline	Status transition processing from (D) to (B)
Declaration	void R_MAIN_DtoB(void)
Description	Change the CPU clock from subsystem to high-speed on-chip oscillator clock. After the clock is switched, control LED lighting.
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_BtoC

Outline	Status transition processing from (B) to (C)
Declaration	void R_MAIN_BtoC(void)
Description	Change the CPU clock from high-speed on-chip oscillator clock to high-speed system clock. After the clock is switched, control LED lighting.
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_CtoD

Outline	Status transition processing from (C) to (D)
Declaration	void R_MAIN_CtoD(void)
Description	Change the CPU clock from high-speed system clock to subsystem clock. After the clock is switched, control LED lighting.
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_DtoC

Outline	Status transition processing from (D) to (C)
Declaration	void R_MAIN_DtoC(void)
Description	Change the CPU clock from subsystem clock to high-speed system clock. After the clock is switched, control LED lighting.
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_CtoF

Outline	Status transition processing from (C) to (F)
Declaration	void R_MAIN_CtoF(void)
Description	Control LED lighting, then transition to HALT mode. (CPU clock stopped (when using high-speed system clock))
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_FtoC

Outline	Status transition processing from (F) to (C)
Declaration	void R_MAIN_FtoC(void)
Description	Control LED lighting. (CPU clock: high-speed system clock)
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_CtoI

Outline	Status transition processing from (C) to (I)
Declaration	void R_MAIN_CtoI(void)
Description	Control LED lighting, then transition to STOP mode. (Stop CPU clock (when using high-speed system clock))
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_ItoC

Outline	Status transition processing from (I) to (C)
Declaration	void R_MAIN_ItoC(void)
Description	Control LED lighting. (CPU clock: high-speed system clock)
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_CtoB

Outline	Status transition processing from (C) to (B)
Declaration	void R_MAIN_CtoB(void)
Description	Change the CPU clock from high-speed system clock to high-speed on-chip oscillator clock. After the clock is switched, control LED lighting.
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_BtoE

Outline	Status transition processing from (B) to (E)
Declaration	void R_MAIN_BtoE(void)
Description	Control LED lighting, then transition to HALT mode. (CPU clock stopped (when using high-speed on-chip oscillator clock))
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_EtoB

Outline	Status transition processing from (E) to (B)
Declaration	void R_MAIN_EtoB(void)
Description	Control LED lighting. (CPU clock: high-speed on-chip oscillator clock)
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_BtoH

Outline	Status transition processing from (B) to (H)
Declaration	void R_MAIN_BtoH(void)
Description	Control LED lighting, then transition to STOP mode. (CPU clock stopped (when using high-speed on-chip oscillator clock))
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_HtoB

Outline	Status transition processing from (H) to (B)
Declaration	void R_MAIN_HtoB(void)
Description	Control LED lighting. (CPU clock: high-speed on-chip oscillator clock)
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_BtoJ

Outline	Status transition processing from (B) to (J)
Declaration	void R_MAIN_BtoJ(void)
Description	Set A/D converter and control LED lighting. Then, transition to SNOOZE mode.
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_JtoB

Outline	Status transition processing from (J) to (B)
Declaration	void R_MAIN_JtoB(void)
Description	Set SNOOZE release and stop A/D converter. Then control LED lighting.
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_NOP_Loop

Outline	Continuous NOP instruction execution processing
Declaration	void R_MAIN_NOP_Loop(void)
Description	Execute NOP instruction continuously. End processing when external interrupt generation detection flag is confirmed.
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_END

Outline	End processing of status transition
Declaration	void R_MAIN_END(void)
Description	Disable interrupts. Control LED lighting (all off).
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_ERROR

Outline	Error processing of status transition
Declaration	void R_MAIN_ERROR(void)
Description	Disable interrupts. Control LED lighting (all off). Loop processing with in the function(Return is only reset input).
Argument	None
Return Value	None
Notes	None

[Function Name] R_MAIN_AD_SnoozeOn

Outline	A/D converter setting
Declaration	void R_MAIN_AD_SnoozeOn(void)
Description	Set A/D converter to hardware trigger wait mode with 12-bit interval timer interrupt signal. Enable SNOOZE mode and transition to A/D conversion wait status.
Argument	None
Return Value	None
Notes	None

[Function Name] r_intc1_interrupt

Outline	External interrupt generation detection flag confirmation processing
Declaration	static void __near r_intc1_interrupt(void)
Description	Confirm external interrupt generation detection flag with generation of external interrupt. End processing when switch input level changes to high.
Argument	None
Return Value	None
Notes	None

[Function Name] r_intit_interrupt

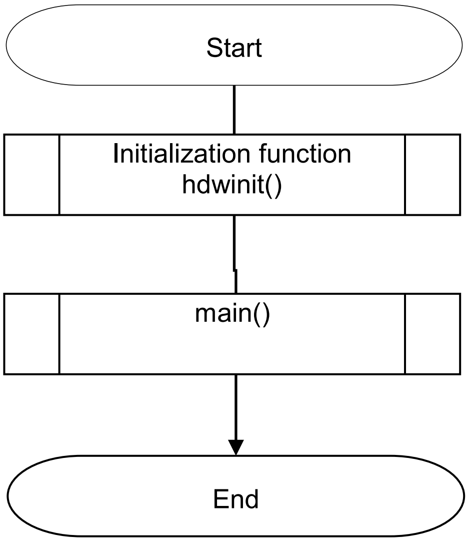
Outline	12-bit interval timer interval signal detection interrupt count processing
Declaration	static void __near r_intit_interrupt(void)
Description	Increment the RITCOUNT each time the 12-bit interval timer interrupt signal detection interrupt is generated.
Argument	None
Return Value	None
Notes	None

[Function Name] r_adc_interrupt

Outline	SNOOZE mode release processing
Declaration	static void __near r_adc_interrupt(void)
Description	Clear the AWC bit of the ADM2 register and release the SNOOZE mode.
Argument	None
Return Value	None
Notes	None

5.6 Flowcharts

Figure 5.1 shows the entire flow for this application note.



Reference the option byte before executing the CPU initialization function.

Figure 5.1 Overall Flowchart

5.6.1 Initialization Function

Figure 5.2 shows the flowchart for the initialization function.

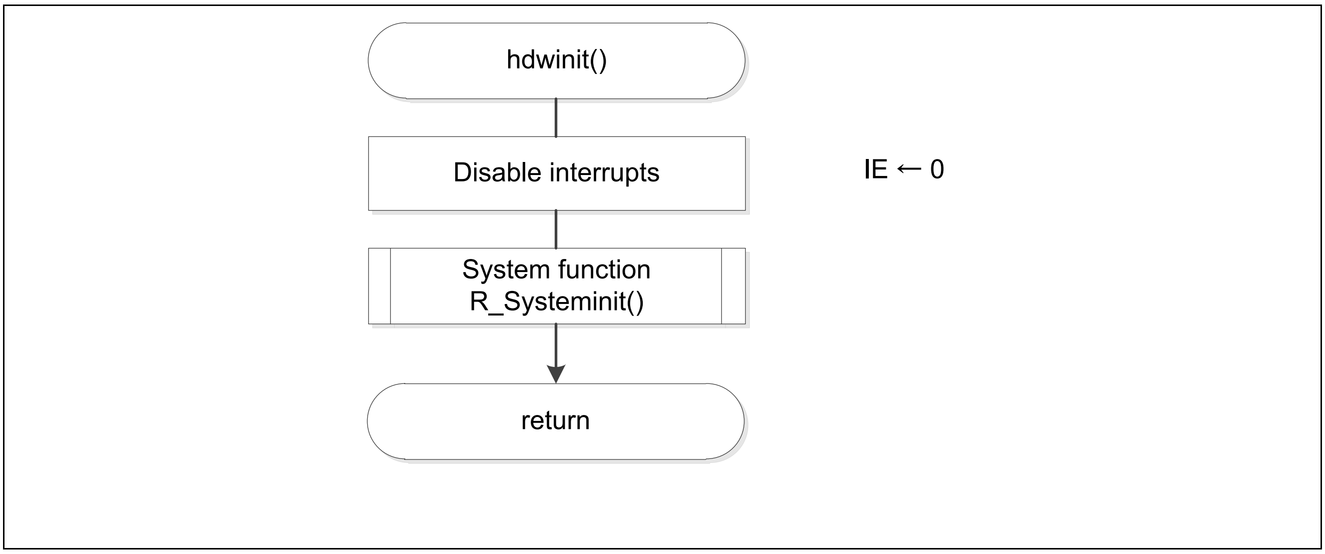


Figure 5.2 Initialization Function

5.6.2 System Function

Figure 5.3 shows the flowchart for system function.

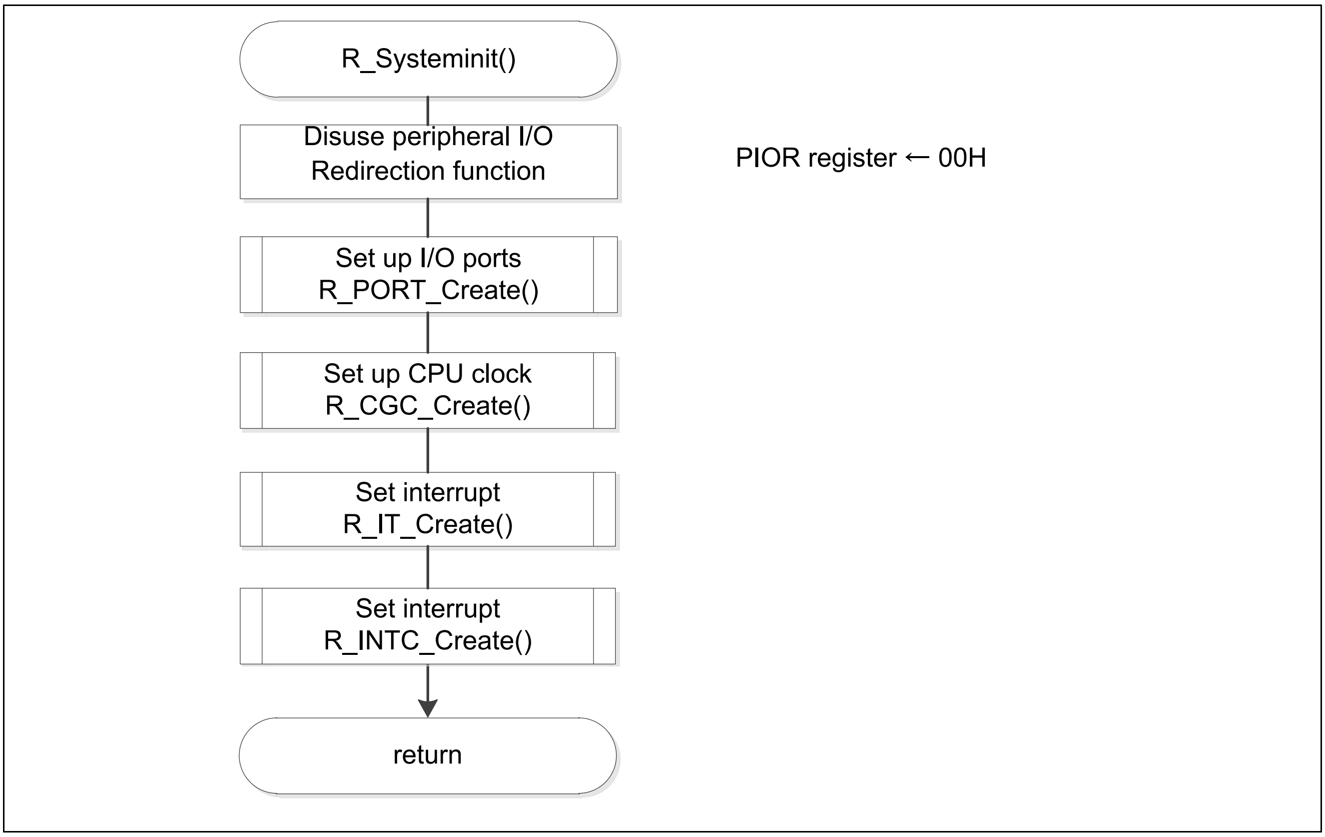
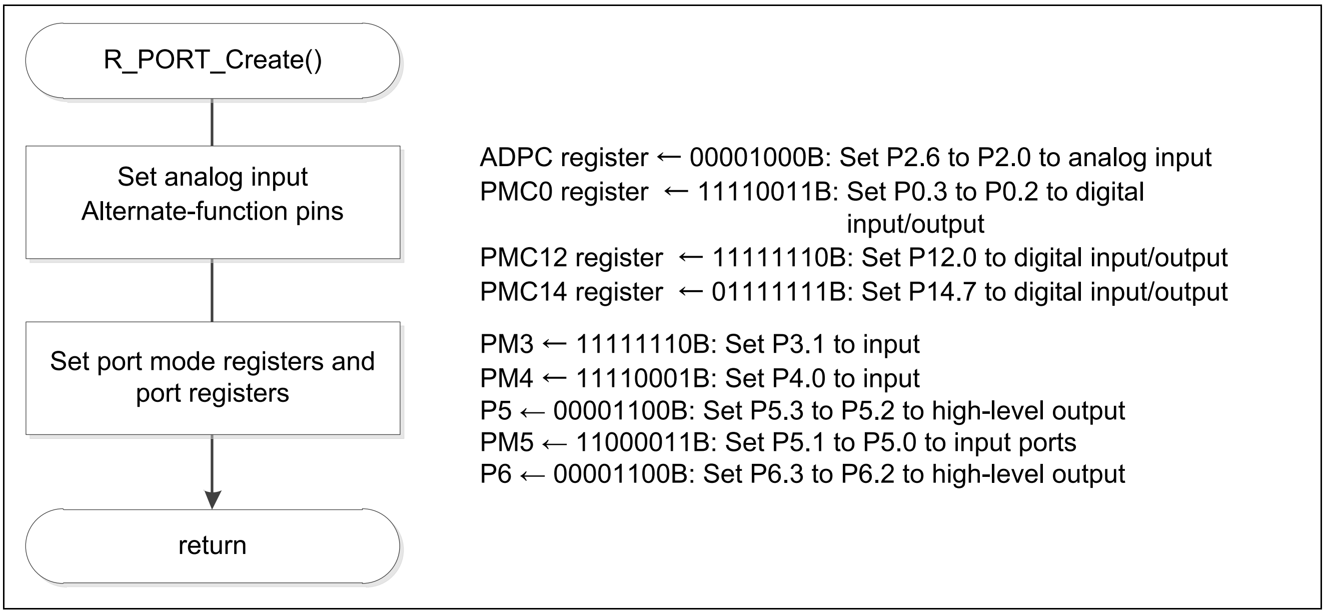

R_Systeminit()
Disuse peripheral I/O Redirection function
PIOR register ← 00H
Set up I/O ports R_PORT_Create()
Set up CPU clock R_CGC_Create()
Set interrupt R_IT_Create()
Set interrupt R_INTC_Create()
return

Figure 5.3 System Function

5.6.3 Input/Output Port Settings

Figure 5.4 shows the flowchart for the input/output port settings.



ADPC register ← 00001000B: Set P2.6 to P2.0 to analog input
PMC0 register ← 11110011B: Set P0.3 to P0.2 to digital input/output
PMC12 register ← 11111110B: Set P12.0 to digital input/output
PMC14 register ← 01111111B: Set P14.7 to digital input/output

PM3 ← 11111110B: Set P3.1 to input
PM4 ← 11110001B: Set P4.0 to input
P5 ← 00001100B: Set P5.3 to P5.2 to high-level output
PM5 ← 11000011B: Set P5.1 to P5.0 to input ports
P6 ← 00001100B: Set P6.3 to P6.2 to high-level output

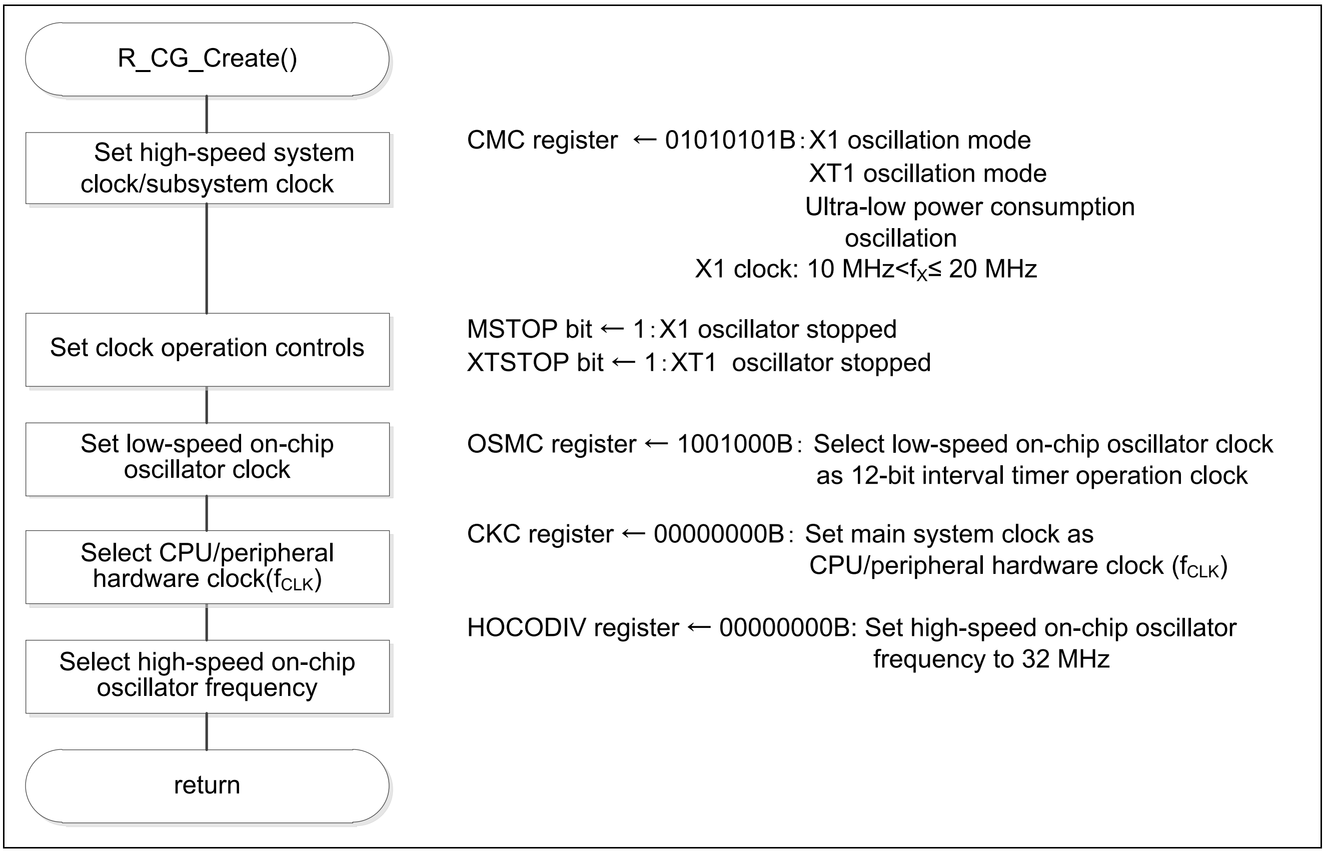
Figure 5.4 Input/Output Port Settings

Note: Refer to the initialization flowchart in the RL78/G13 Initialization (R01AN2575J) Application Note for details on how to set unused ports.

Caution: When designing circuits, always make sure unused ports are properly processed and all electrical characteristics are met. Also make sure each unused input-only port is connected to V_{DD} or V_{SS} through a resister.

5.6.4 Clock Generator Setting

Figure 5.5 shows the flowchart for setting the clock generator.


R_CG_Create()
Set high-speed system clock/subsystem clock
Set clock operation controls
Set low-speed on-chip oscillator clock
Select CPU/peripheral hardware clock(f_{CLK})
Select high-speed on-chip oscillator frequency
return

CMC register ← 01010101B : X1 oscillation mode
XT1 oscillation mode
Ultra-low power consumption oscillation
X1 clock: 10 MHz<f_X≤ 20 MHz

MSTOP bit ← 1 : X1 oscillator stopped
XTSTOP bit ← 1 : XT1 oscillator stopped

OSMC register ← 1001000B : Select low-speed on-chip oscillator clock as 12-bit interval timer operation clock

CKC register ← 00000000B : Set main system clock as CPU/peripheral hardware clock (f_{CLK})

HOCODIV register ← 00000000B: Set high-speed on-chip oscillator frequency to 32 MHz

Figure 5.5 Clock Generator Setting

Clock operation mode setting

- Clock operation mode control register (CMC)

High-speed system clock pin operation mode: input port mode
Subsystem clock pin operation mode: input port mode
XT1 oscillator oscillation mode: ultra-low power consumption oscillation
X1 clock oscillation frequency control: 1MHz ≤ fMX ≤ 10MHz

Symbol: CMC

7	6	5	4	3	2	1	0
EXCLK	OSCSEL	EXCLKS	OSCSELS	0	AMPHS1	AMPHS0	AMPH
0	**1**	**0**	**1**	0	**1**	**0**	**1**

Bits 7-6

EXCLK	OSCSEL	High-seed oscillation clock pin operation mode	X1/P121 Port	X2/EXCLK/P122 Port
0	0	Input port mode	Input port	
0	**1**	**X1 oscillation mode**	**Crystal/ceramic resonator connection**	
1	0	Input port mode	Input port	
1	1	External clock input mode	Input port	External clock input

Bits 5-4

EXCLKS	OSCSELS	Subsystem clock pin operation mode	XT1/P123 Port	XT2/EXCLKS/P124 Port
0	0	Input port mode	Input port	
0	**1**	**XT1 oscillation mode**	**Crystal resonator connection**	
1	0	Input port mode	Input port	
1	1	External clock input mode	Input port	External clock input

Bits 2-1

AMPHS1	AMPHS0	XT1 oscillator oscillation mode selection
0	0	Low-power consumption oscillation (default)
0	1	Normal oscillation
1	**0**	**Ultra-low power consumption oscillation**
1	1	Setting prohibited

Bit 0

AMPH	Control of X1 clock oscillation frequency
0	1MHz ≤ f_X ≤ 10MHz
1	**10MHz < f_X ≤ 20MHz**

Note: Refer to the RL78/G13 User’s Manual (Hardware version) for details on how to set registers.

Operation control of clocks

- Clock operation status control register (CSC)
  High-speed system clock operation control: X1 oscillator stopped
  Subsystem clock operation control: XT1 oscillator stopped
  HOCO clock operation control: HOCO operating

Symbol: CSC

7	6	5	4	3	2	1	0
MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP
1	**1**	0	0	0	0	0	**0**

Bit 7

MSTOP	High-speed system clock operation control		
	X1 oscillation mode	External clock input mode	Input port mode
0	X1 oscillator operating	External clock from EXCLK pin is valid	Input port
1	**X1 oscillator stopped**	**External clock from EXCLK pin is invalid**	

Bit 6

XTSTOP	Subsystem clock operation control		
	XT1 oscillation mode	External clock input mode	Input port mode
0	XT1 oscillator operating	External clock from EXCLKS pin is valid	Input port
1	**XT1 oscillator stopped**	**External clock from EXCLKS pin is invalid**	

Bit 0

HIOSTOP	High-speed on-chip oscillator clock operation control
0	**High-speed on-chip oscillator operating**
1	High-speed on-chip oscillator stopped

Note: Refer to the RL78/G13 User’s Manual (Hardware version) for details on how to set registers.

CPU/peripheral hardware clock (f_{CLK}) setting

- System clock control register (CKC)
  Status of f_{CLK}: main system clock
  Selection of f_{CLK}: high-speed on-chip oscillator clock （f_{IH}）

Symbol: CKC

7	6	5	4	3	2	1	0
CLS	CSS	MCS	MCM0	0	0	0	0
0	**0**	**0**	**0**	0	0	0	0

Bit 7

CLS	Status of CPU/peripheral hardware clock (f_{CLK})
0	**Main system clock (f_{MAIN})**
1	Subsystem clock (f_{SUB})

Bit 6

CSS	Selection of CPU/peripheral hardware clock (f_{CLK})
0	**Main system clock (f_{MAIN})**
1	Subsystem clock (f_{SUB})

Bit 5

MCS	Status of main system clock (f_{MAIN})
0	**High-speed on-chip oscillator clock f_{IH}）**
1	High-speed system clock (f_{MX}）

Bit 4

MCM0	Main system clock (f_{MAIN}) operation control
0	**Selects HOCO clock (f_{IH}) as main system clock (f_{MAIN})**
1	Selects high-speed system clock (f_{MX}) as main system clock (f_{MAIN}).

Note: Refer to the RL78/G13 User's Manual (Hardware version) for details on how to set registers.

Subsystem clock supply mode control

・Subsystem clock supply mode control register (OSMC)

Setting in STOP mode or in HALT mode while subsystem clock is selected as CPU clock:

enable subsystem clock supply to peripheral functions

Selection of count clock for real-time clock and 12-bit interval timer: subsystem clock

Symbol: OSMC

7	6	5	4	3	2	1	0
RTCLPC	0	0	WUTMMCK0	0	0	0	0
0	0	0	**1**	0	0	0	0

Bit 7

RTCLPC	Setting in STOP mode or in HALT mode while subsystem clock is selected as CPU clock
0	**Enables supply of subsystem clock to peripheral functions**
1	Stops supply of subsystem clock to peripheral functions other than real-time clock and 12-bit interval timer.

Bit 4

WUTMMCK0	Selection of count clock for real-time clock and 12-bit interval timer
0	Subsystem clock (f_{SUB})
1	**Low-speed internal oscillator (LOCO) clock**

Note: Refer to the RL78/G13 User's Manual (Hardware version) for details on how to set registers.

5.6.5 External Interrupt Setting

Figure 5.6 shows the flowchart for setting the external interrupt.

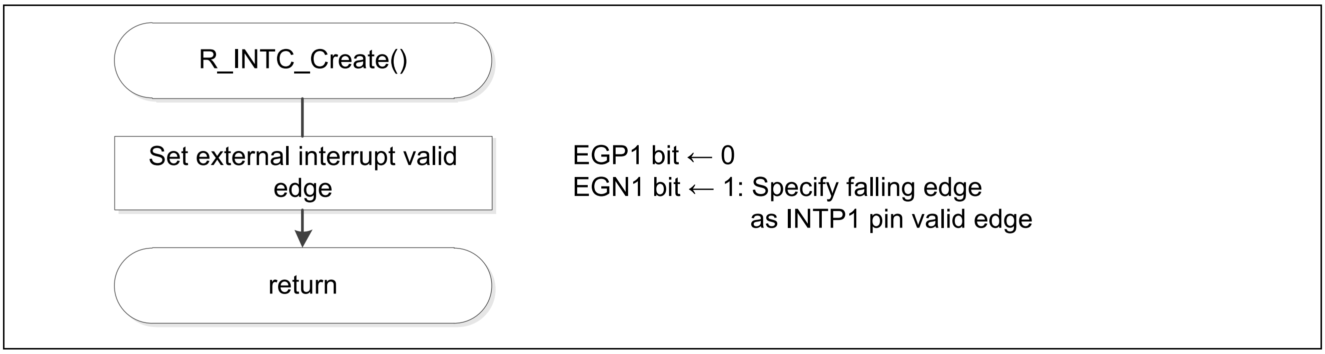

R_INTC_Create()
Set external interrupt valid edge
return
EGP1 bit ← 0
EGN1 bit ← 1: Specify falling edge
as INTP1 pin valid edge

Figure 5.6 External Interrupt Setting

Control of external interrupt valid edge

・External interrupt rising edge enable register (EGP0)

Select valid edge for INTP1 pin: falling edge

Symbol: EGP0

7	6	5	4	3	2	1	0
EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0
x	x	x	x	x	x	**0**	x

Symbol: EGN0

7	6	5	4	3	2	1	0
EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0
x	x	x	x	x	x	**1**	x

Bit 1

EGP1	EGN1	INTP1 pin valid edge selection
0	0	Edge detection disabled
0	**1**	**Falling edge**
1	0	Rising edge
1	1	Both rising and falling edges

Note: Refer to the RL78/G13 User's Manual (Hardware version) for details on how to set registers.

5.6.6 12-bit Interval Timer Setting

Figure 5.7 shows the flowchart for setting the 12-bit interval timer.

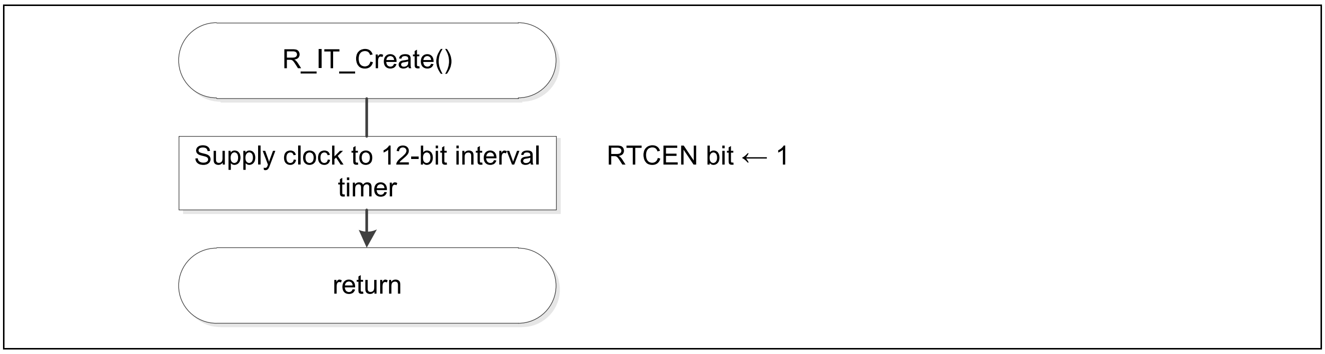


Figure 5.7 12-bit Interval Timer Setting

12-bit interval timer clock supply setting

- Peripheral enable register 0 (PER0)

  Enable clock supply to 12-bit interval timer.

Symbol: PER0

7	6	5	4	3	2	1	0
RTCEN	IICA1EN	ADCEN	IICA0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN
1	x	x	x	x	x	x	x

Bit 7

RTCEN	Control of 12-bit interval timer input clock supply
0	Stops input clock supply.
1	**Enables input clock supply.**

Note: Refer to the RL78/G13 User’s Manual (Hardware version) for details on how to set registers.

5.6.7 Main Processing

Figure 5.8 and Figure 5.9 shows the flowchart for the main processing.

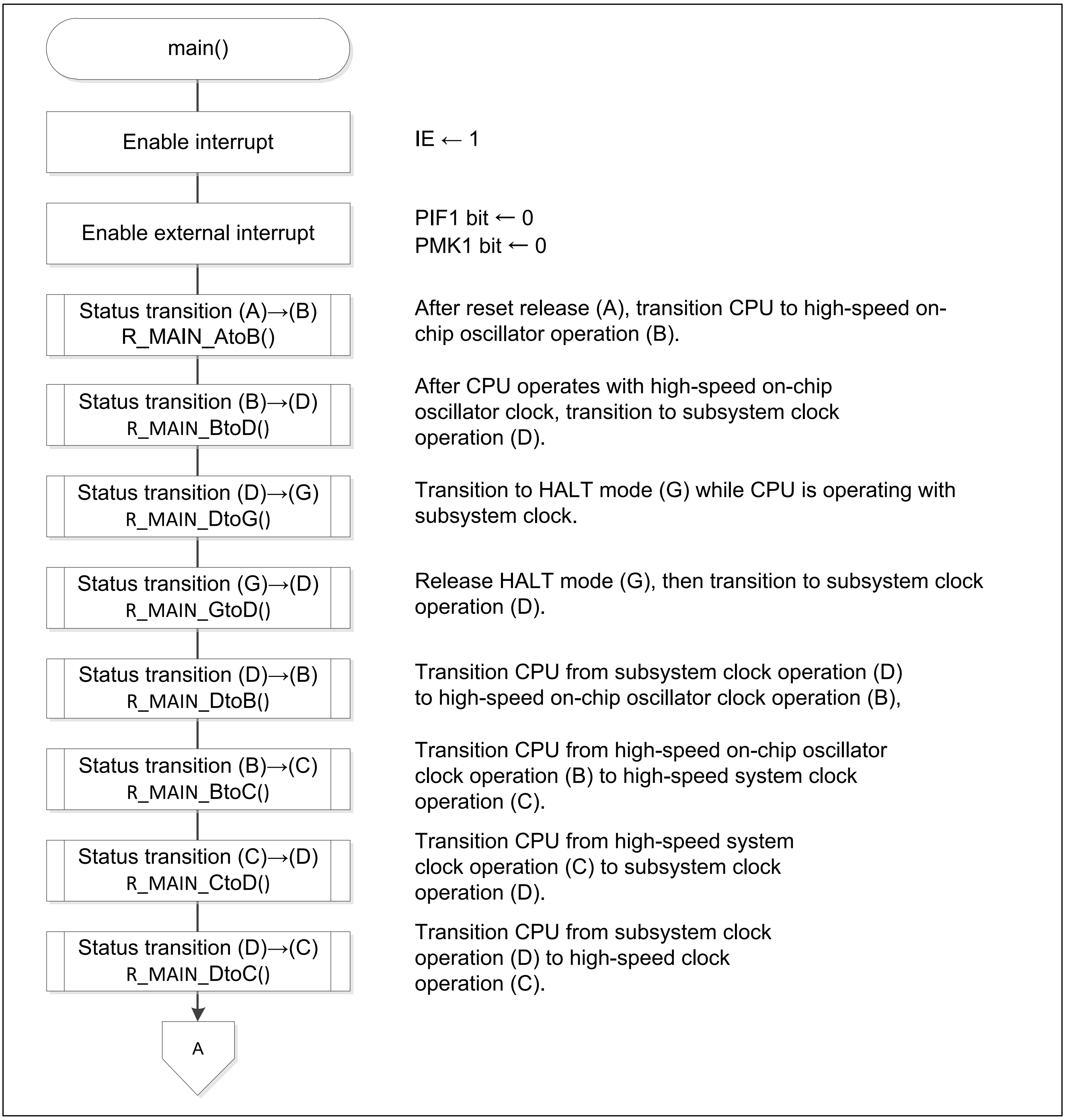

main()
Enable interrupt
IE ← 1
Enable external interrupt
PIF1 bit ← 0
PMK1 bit ← 0
Status transition (A)→(B)
R_MAIN_AtoB()
After reset release (A), transition CPU to high-speed on-chip oscillator operation (B).
Status transition (B)→(D)
R_MAIN_BtoD()
After CPU operates with high-speed on-chip oscillator clock, transition to subsystem clock operation (D).
Status transition (D)→(G)
R_MAIN_DtoG()
Transition to HALT mode (G) while CPU is operating with subsystem clock.
Status transition (G)→(D)
R_MAIN_GtoD()
Release HALT mode (G), then transition to subsystem clock operation (D).
Status transition (D)→(B)
R_MAIN_DtoB()
Transition CPU from subsystem clock operation (D) to high-speed on-chip oscillator clock operation (B),
Status transition (B)→(C)
R_MAIN_BtoC()
Transition CPU from high-speed on-chip oscillator clock operation (B) to high-speed system clock operation (C).
Status transition (C)→(D)
R_MAIN_CtoD()
Transition CPU from high-speed system clock operation (C) to subsystem clock operation (D).
Status transition (D)→(C)
R_MAIN_DtoC()
Transition CPU from subsystem clock operation (D) to high-speed clock operation (C).
A

Figure 5.8 Main Processing (1/2)

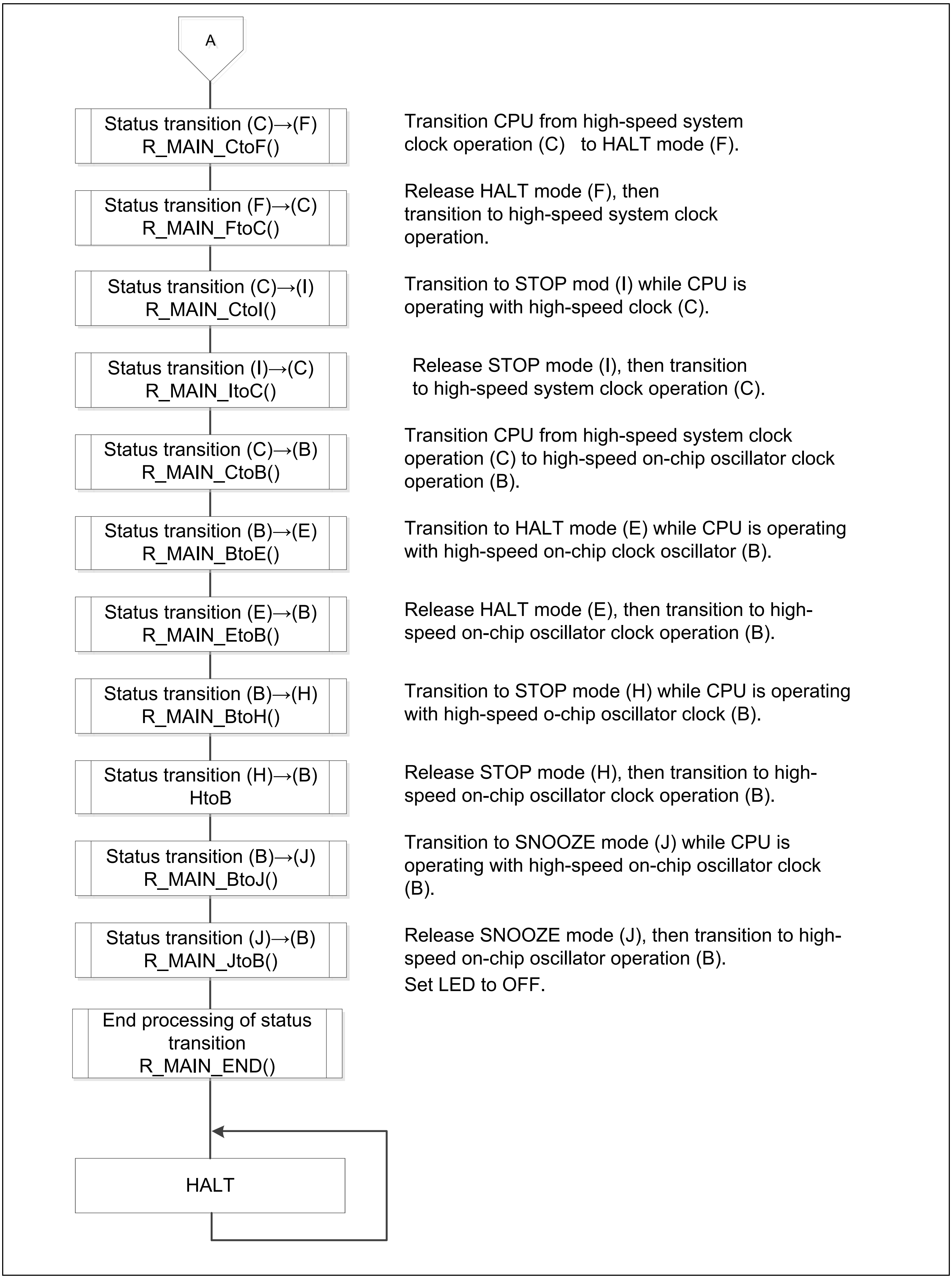

A
Status transition (C)→(F)
R_MAIN_CtoF()
Transition CPU from high-speed system clock operation (C) to HALT mode (F).
Status transition (F)→(C)
R_MAIN_FtoC()
Release HALT mode (F), then transition to high-speed system clock operation.
Status transition (C)→(I)
R_MAIN_CtoI()
Transition to STOP mod (I) while CPU is operating with high-speed clock (C).
Status transition (I)→(C)
R_MAIN_ItoC()
Release STOP mode (I), then transition to high-speed system clock operation (C).
Status transition (C)→(B)
R_MAIN_CtoB()
Transition CPU from high-speed system clock operation (C) to high-speed on-chip oscillator clock operation (B).
Status transition (B)→(E)
R_MAIN_BtoE()
Transition to HALT mode (E) while CPU is operating with high-speed on-chip clock oscillator (B).
Status transition (E)→(B)
R_MAIN_EtoB()
Release HALT mode (E), then transition to high-speed on-chip oscillator clock operation (B).
Status transition (B)→(H)
R_MAIN_BtoH()
Transition to STOP mode (H) while CPU is operating with high-speed o-chip oscillator clock (B).
Status transition (H)→(B)
HtoB
Release STOP mode (H), then transition to high-speed on-chip oscillator clock operation (B).
Status transition (B)→(J)
R_MAIN_BtoJ()
Transition to SNOOZE mode (J) while CPU is operating with high-speed on-chip oscillator clock (B).
Status transition (J)→(B)
R_MAIN_JtoB()
Release SNOOZE mode (J), then transition to high-speed on-chip oscillator operation (B).
Set LED to OFF.
End processing of status transition
R_MAIN_END()
HALT

Figure 5.9 Main Processing (2/2)

Pin input edge detection interrupt (INTP1) setting

- Interrupt request flag register (IF0L)
  Clear the PIF1 interrupt source flag.
- Interrupt mask flag register (MK0L）
  Set PMK1 interrupt mask.

Symbol: IF0L

7	6	5	4	3	2	1	0
PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF	WDTIF
x	x	x	x	**0**	x	x	x

Bit 3

PIF1	Interrupt request flag
0	**No interrupt request signal is generated**
1	Interrupt request signal is generated, interrupt request status

Symbol: MK0L

7	6	5	4	3	2	1	0
PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTMK
x	x	x	x	**0**	x	x	x

Bit 3

RTCMK	Interrupt servicing control
0	**Interrupt servicing enabled**
1	Interrupt servicing disabled

Note: Refer to the RL78/G13 User’s Manual (Hardware version) for details on how to set registers.

5.6.8 Status Transition AtoB

Figure5.10 shows the flowchart for status transition AtoB.

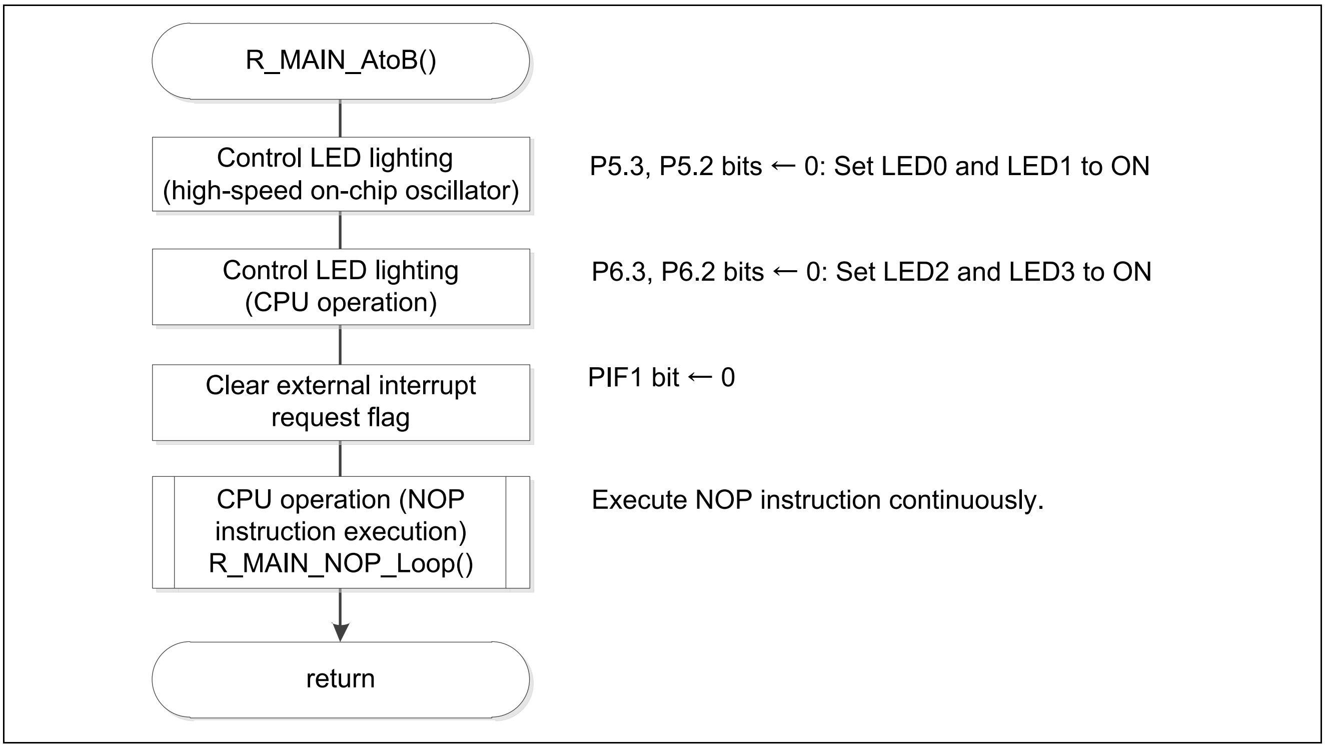

R_MAIN_AtoB()
Control LED lighting (high-speed on-chip oscillator)
P5.3, P5.2 bits ← 0: Set LED0 and LED1 to ON
Control LED lighting (CPU operation)
P6.3, P6.2 bits ← 0: Set LED2 and LED3 to ON
Clear external interrupt request flag
PIF1 bit ← 0
CPU operation (NOP instruction execution) R_MAIN_NOP_Loop()
Execute NOP instruction continuously.
return

Figure 5.10 Status Transition AtoB

5.6.9 CPU operation (NOP instruction execution)

Figure5.11 shows the flowchart for the CPU operation (NOP instruction execution)

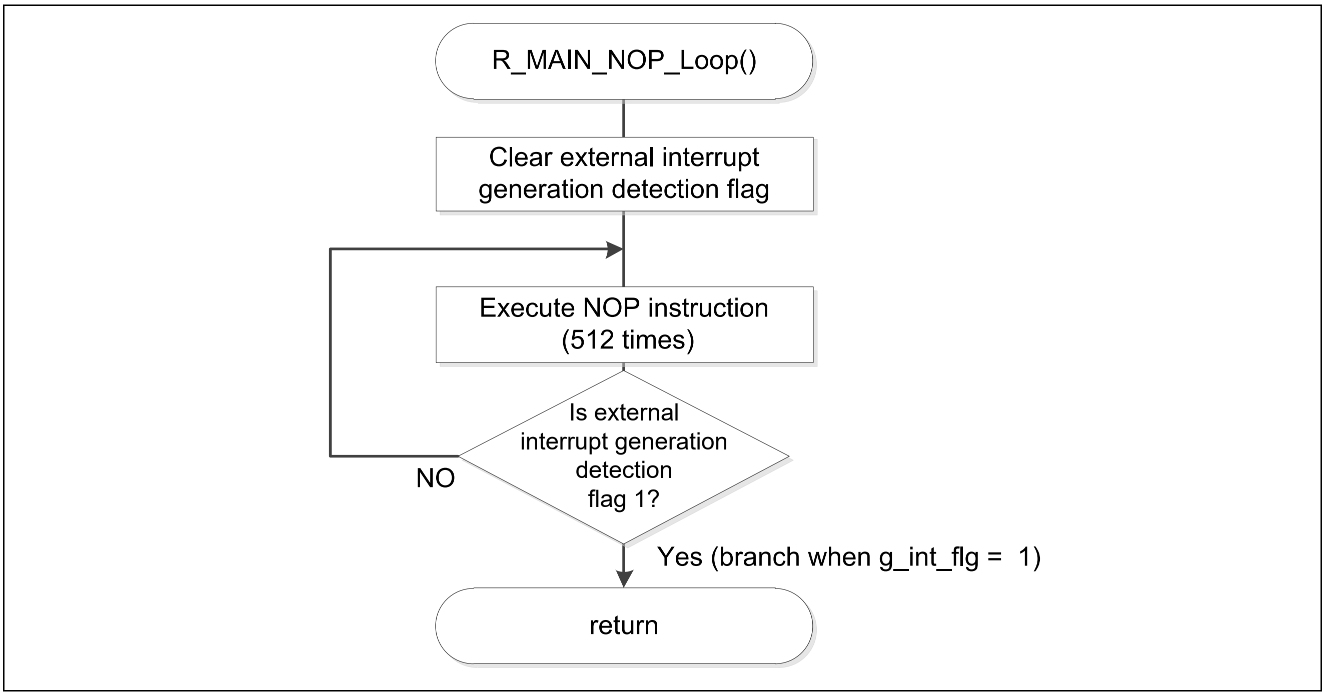

R_MAIN_NOP_Loop()
Clear external interrupt generation detection flag
Execute NOP instruction (512 times)
Is external interrupt generation detection flag 1?
NO
Yes (branch when g_int_flg = 1)
return

Figure 5.11 CPU Operation (NOP instruction execution)

5.6.10 Status Transition BtoD

Figure5.12 and Figure5.13 shows the flowchart for status transition BtoD.

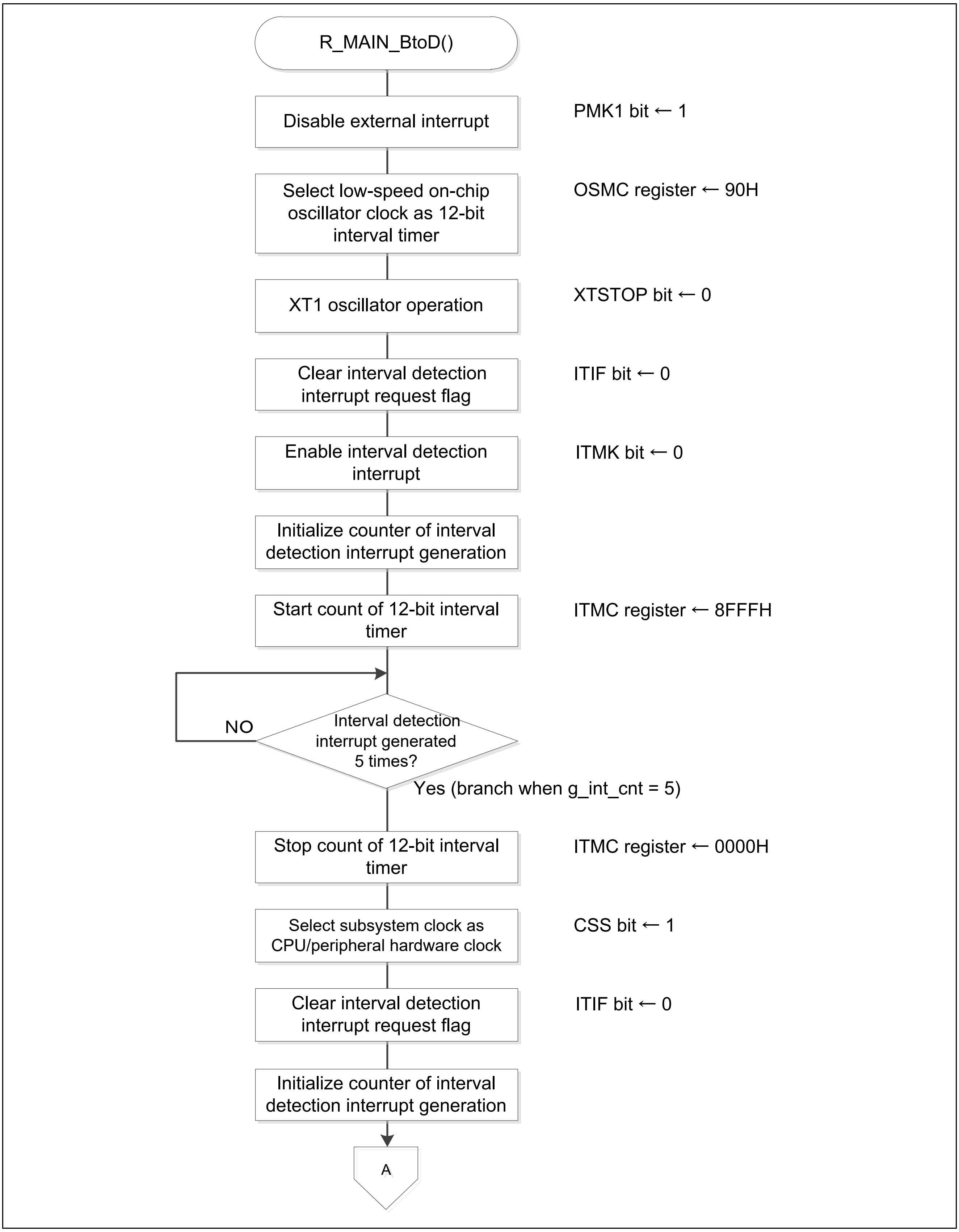

R_MAIN_BtoD()
Disable external interrupt
PMK1 bit ← 1
Select low-speed on-chip oscillator clock as 12-bit interval timer
OSMC register ← 90H
XT1 oscillator operation
XTSTOP bit ← 0
Clear interval detection interrupt request flag
ITIF bit ← 0
Enable interval detection interrupt
ITMK bit ← 0
Initialize counter of interval detection interrupt generation
Start count of 12-bit interval timer
ITMC register ← 8FFFH
NO
Interval detection interrupt generated 5 times?
Yes (branch when g_int_cnt = 5)
Stop count of 12-bit interval timer
ITMC register ← 0000H
Select subsystem clock as CPU/peripheral hardware clock
CSS bit ← 1
Clear interval detection interrupt request flag
ITIF bit ← 0
Initialize counter of interval detection interrupt generation
A

Figure 5.12 Status Transition BtoD (1/2)

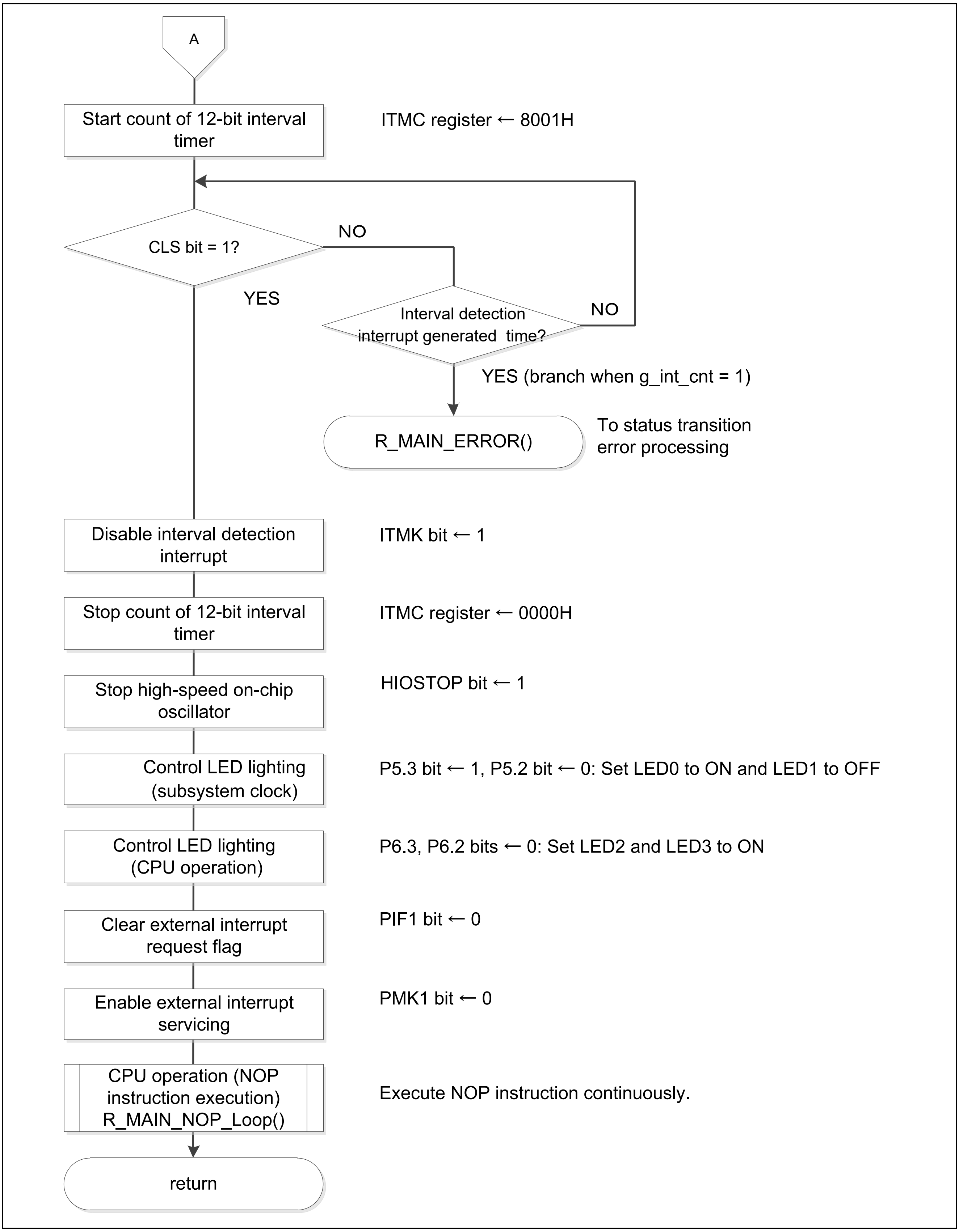

A
Start count of 12-bit interval timer
ITMC register ← 8001H
CLS bit = 1?
NO
YES
Interval detection interrupt generated time?
NO
YES (branch when g_int_cnt = 1)
R_MAIN_ERROR()
To status transition error processing
Disable interval detection interrupt
ITMK bit ← 1
Stop count of 12-bit interval timer
ITMC register ← 0000H
Stop high-speed on-chip oscillator
HIOSTOP bit ← 1
Control LED lighting (subsystem clock)
P5.3 bit ← 1, P5.2 bit ← 0: Set LED0 to ON and LED1 to OFF
Control LED lighting (CPU operation)
P6.3, P6.2 bits ← 0: Set LED2 and LED3 to ON
Clear external interrupt request flag
PIF1 bit ← 0
Enable external interrupt servicing
PMK1 bit ← 0
CPU operation (NOP instruction execution) R_MAIN_NOP_Loop()
Execute NOP instruction continuously.
return

Figure 5.13 Status Transition BtoD (2/2)

12-bit interval timer interval signal detection interrupt (INTIT) setting

- Interrupt request flag register (IF1H)

  Clear ITIF interrupt source flag.
- Interrupt mask flag register (MK1H)

  Set ITMK interrupt mask.

Symbol: IF1H

7	6	5	4	3	2	1	0
TMIF04	TMIF13	SRIF3 CSIIF31 IICIF31	STIF3 CSIIF30 IICIF30	KRIF	ITIF	RTCIF	ADIF
x	x	x	X	X	**0**	x	x

Bit 2

ITIF	Interrupt request flag
0	**No interrupt request signal is generated**
1	Interrupt request signal is generated, interrupt request status

Symbol: MK1H

7	6	5	4	3	2	1	0
TMMK04	TMMK13	SRMK3 CSIMK31 IICMK31	STMK3 CSIMK30 IICMK30	KRMK	ITMK	RTCMK	ADMK
x	x	x	x	X	**0**	x	x

Bit 2

ITMK	Interrupt servicing control
0	**Interrupt servicing enabled**
1	Interrupt servicing disabled

Note: Refer to the RL78/G13 User’s Manual (Hardware version) for details on how to set registers.

12-bit interval timer interval signal detection interrupt (INTIT) setting

- Interval timer control register (ITMC)
  Start 12-bt interval timer count operation.

Symbol: ITMC

15	14	13	2	11-0
RINTE	0	0	0	ITCMP11-ITCMP0
1	0	0	0	**FFFH**

Bit 15

RINTE	12-bit interval timer operation control
0	Count operation stopped (count clear)
1	**Count operation started**

Bits 11-0

ITCMP11-ITCMP0	Specification of 12-bit interval timer compare value
FFFH	**These bits generate an interrupt at the fixed cycle (count clock cycles x (ITCMP setting value FFFH + 1)).**
000H	Setting prohibited

Note: Refer to the RL78/G13 User’s Manual (Hardware version) for details on how to set registers.

5.6.11 Error Processing of Status Transition

Figure5.14 shows the flowchart for error processing of status transition

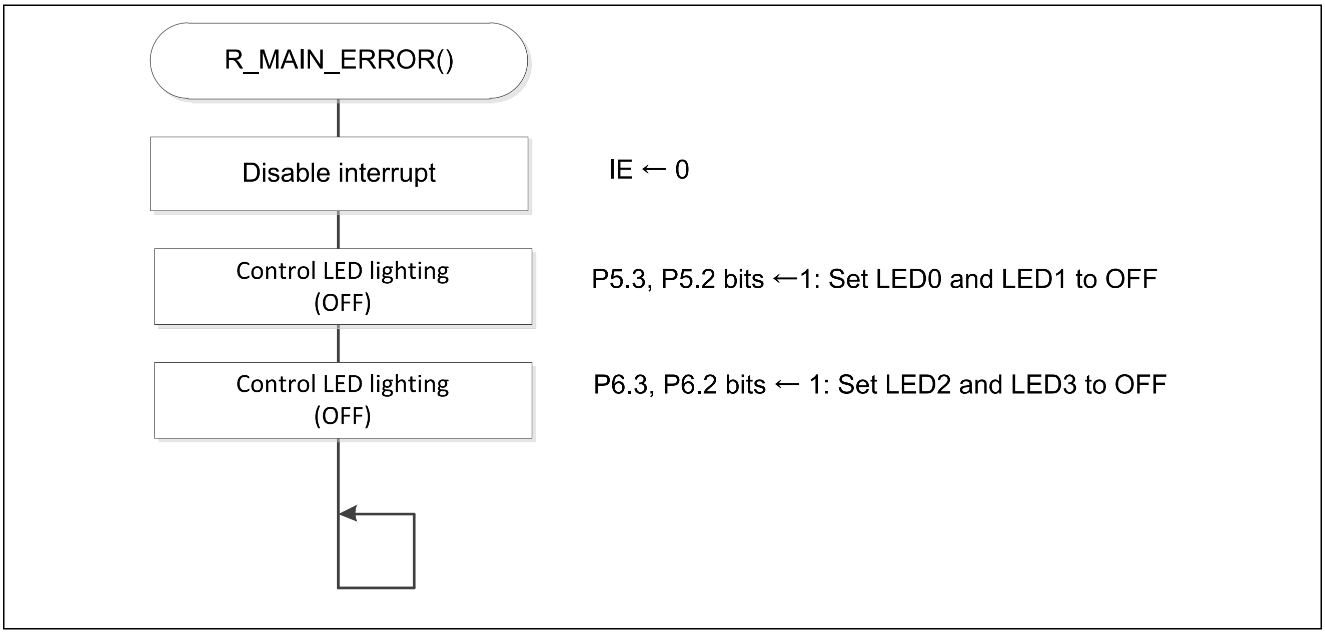

R_MAIN_ERROR()
Disable interrupt
IE ← 0
Control LED lighting (OFF)
P5.3, P5.2 bits ←1: Set LED0 and LED1 to OFF
Control LED lighting (OFF)
P6.3, P6.2 bits ← 1: Set LED2 and LED3 to OFF

Figure 5.14 Error Processing of Status Transition

5.6.12 Status Transition DtoG

Figure5.15 shows the flowchart for status transition DtoG.

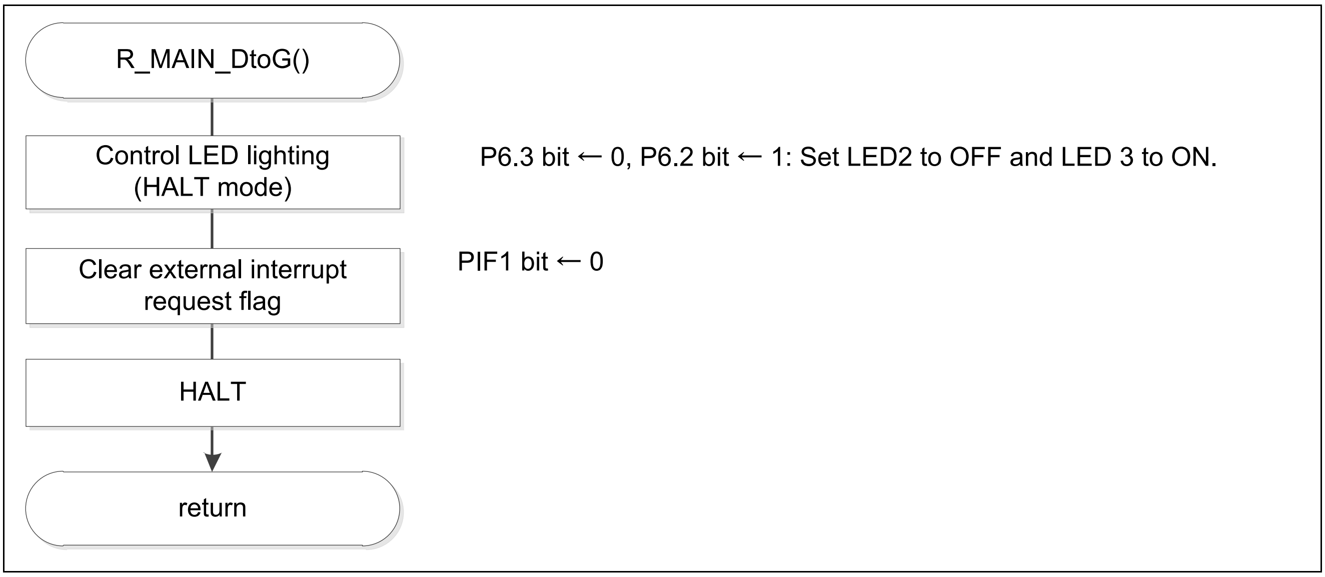

R_MAIN_DtoG()
Control LED lighting (HALT mode)
P6.3 bit ← 0, P6.2 bit ← 1: Set LED2 to OFF and LED 3 to ON.
Clear external interrupt request flag
PIF1 bit ← 0
HALT
return

Figure 5.15 Status Transition DtoG

5.6.13 Status Transition GtoD

Figure5.16 shows the flowchart for status transition GtoD.

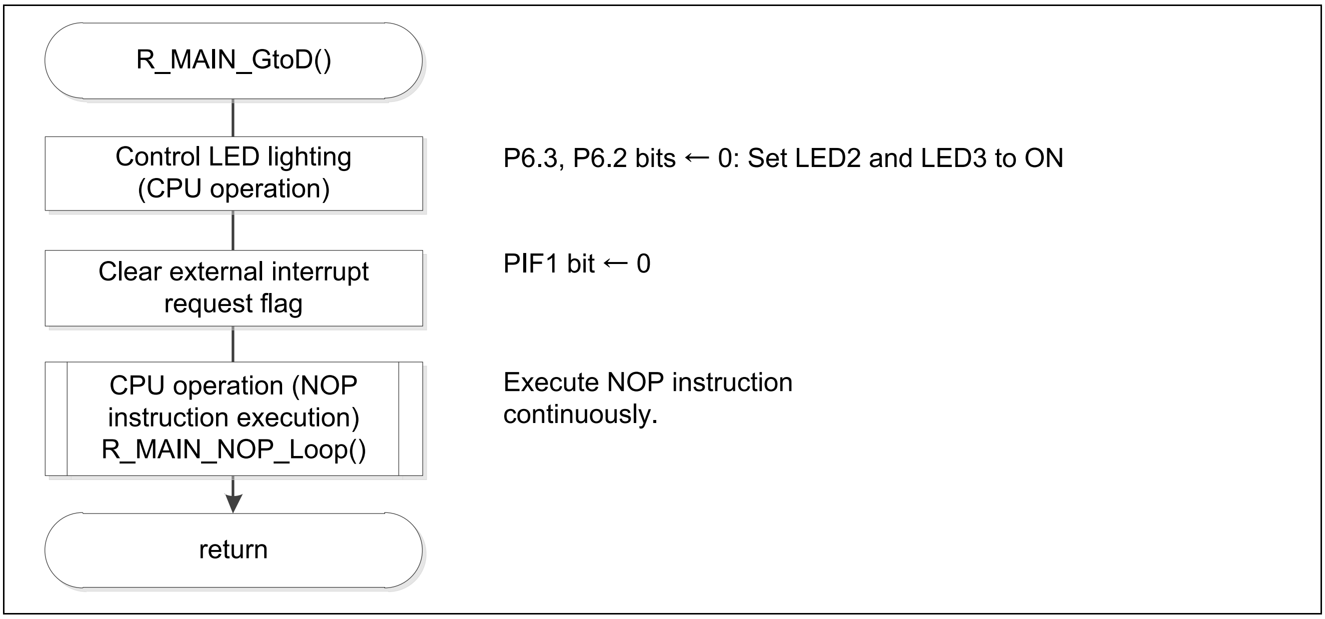

R_MAIN_GtoD()
Control LED lighting (CPU operation)
P6.3, P6.2 bits ← 0: Set LED2 and LED3 to ON
Clear external interrupt request flag
PIF1 bit ← 0
CPU operation (NOP instruction execution) R_MAIN_NOP_Loop()
Execute NOP instruction continuously.
return

Figure 5.16 Status Transition GtoD

5.6.14 Status Transition DtoB

Figure5.17 and Figure5.18 shows the flowchart for status transition DtoB.

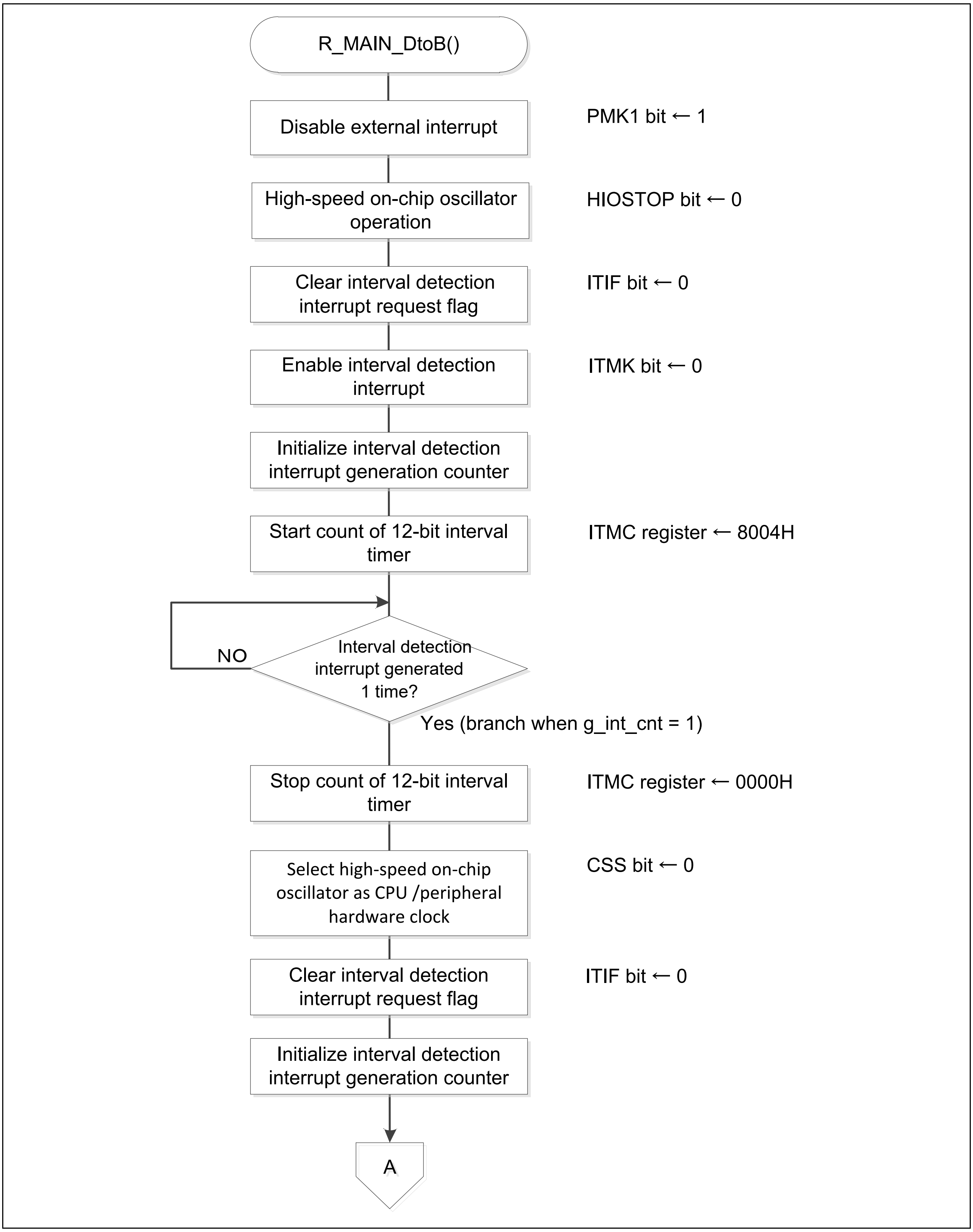

R_MAIN_DtoB()
Disable external interrupt
PMK1 bit ← 1
High-speed on-chip oscillator operation
HIOSTOP bit ← 0
Clear interval detection interrupt request flag
ITIF bit ← 0
Enable interval detection interrupt
ITMK bit ← 0
Initialize interval detection interrupt generation counter
Start count of 12-bit interval timer
ITMC register ← 8004H
NO
Interval detection interrupt generated 1 time?
Yes (branch when g_int_cnt = 1)
Stop count of 12-bit interval timer
ITMC register ← 0000H
Select high-speed on-chip oscillator as CPU /peripheral hardware clock
CSS bit ← 0
Clear interval detection interrupt request flag
ITIF bit ← 0
Initialize interval detection interrupt generation counter
A

Figure 5.17 Status Transition DtoB (1/2)

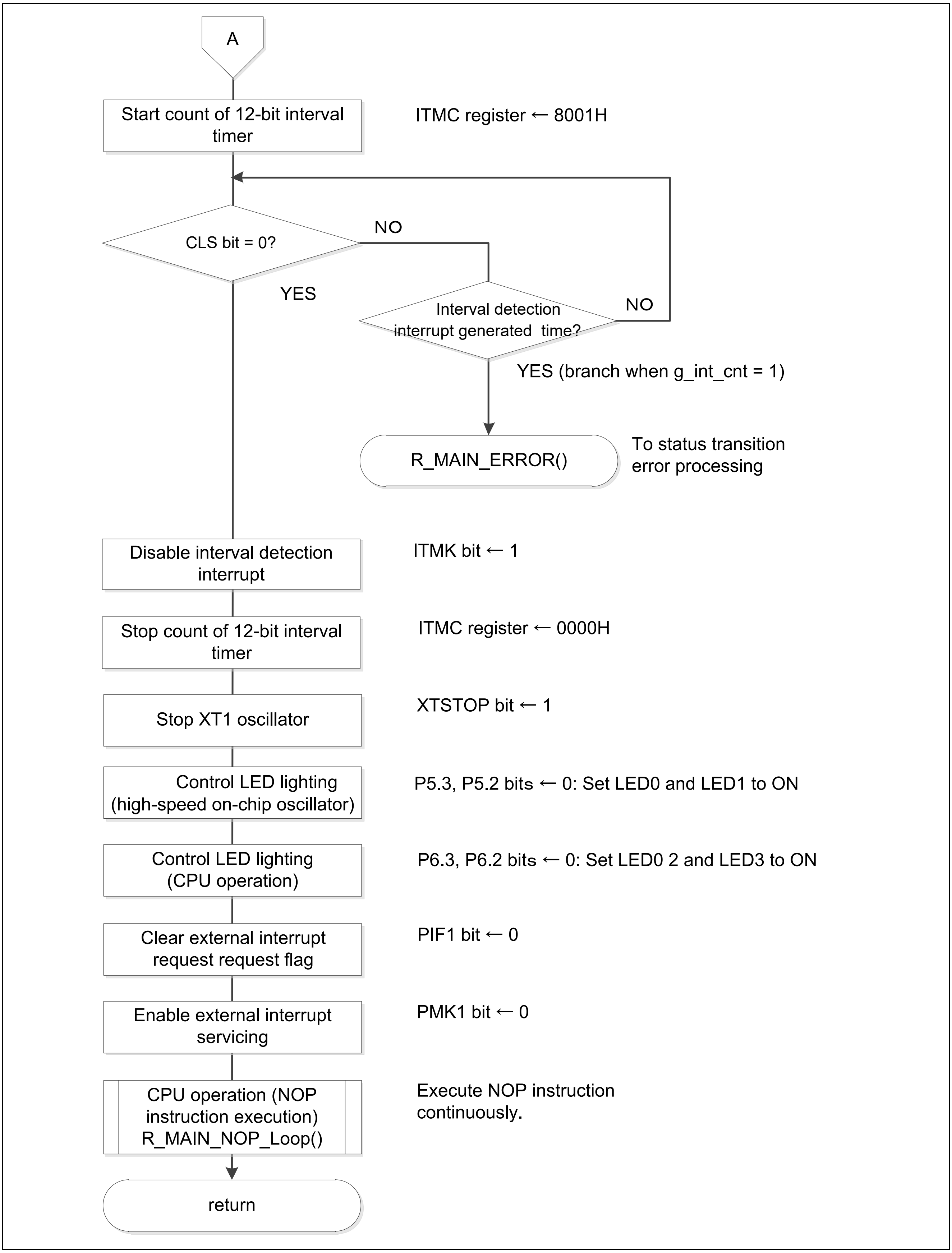

A
Start count of 12-bit interval timer
ITMC register ← 8001H
CLS bit = 0?
NO
YES
Interval detection interrupt generated time?
NO
YES (branch when g_int_cnt = 1)
R_MAIN_ERROR()
To status transition error processing
Disable interval detection interrupt
ITMK bit ← 1
Stop count of 12-bit interval timer
ITMC register ← 0000H
Stop XT1 oscillator
XTSTOP bit ← 1
Control LED lighting (high-speed on-chip oscillator)
P5.3, P5.2 bits ← 0: Set LED0 and LED1 to ON
Control LED lighting (CPU operation)
P6.3, P6.2 bits ← 0: Set LED0 2 and LED3 to ON
Clear external interrupt request request flag
PIF1 bit ← 0
Enable external interrupt servicing
PMK1 bit ← 0
CPU operation (NOP instruction execution) R_MAIN_NOP_Loop()
Execute NOP instruction continuously.
return

Figure 5.18 Status Transition DtoB (2/2)

5.6.15 Status Transition BtoC

Figure5.19 and Figure5.20 shows the flowchart for status transition BtoC.

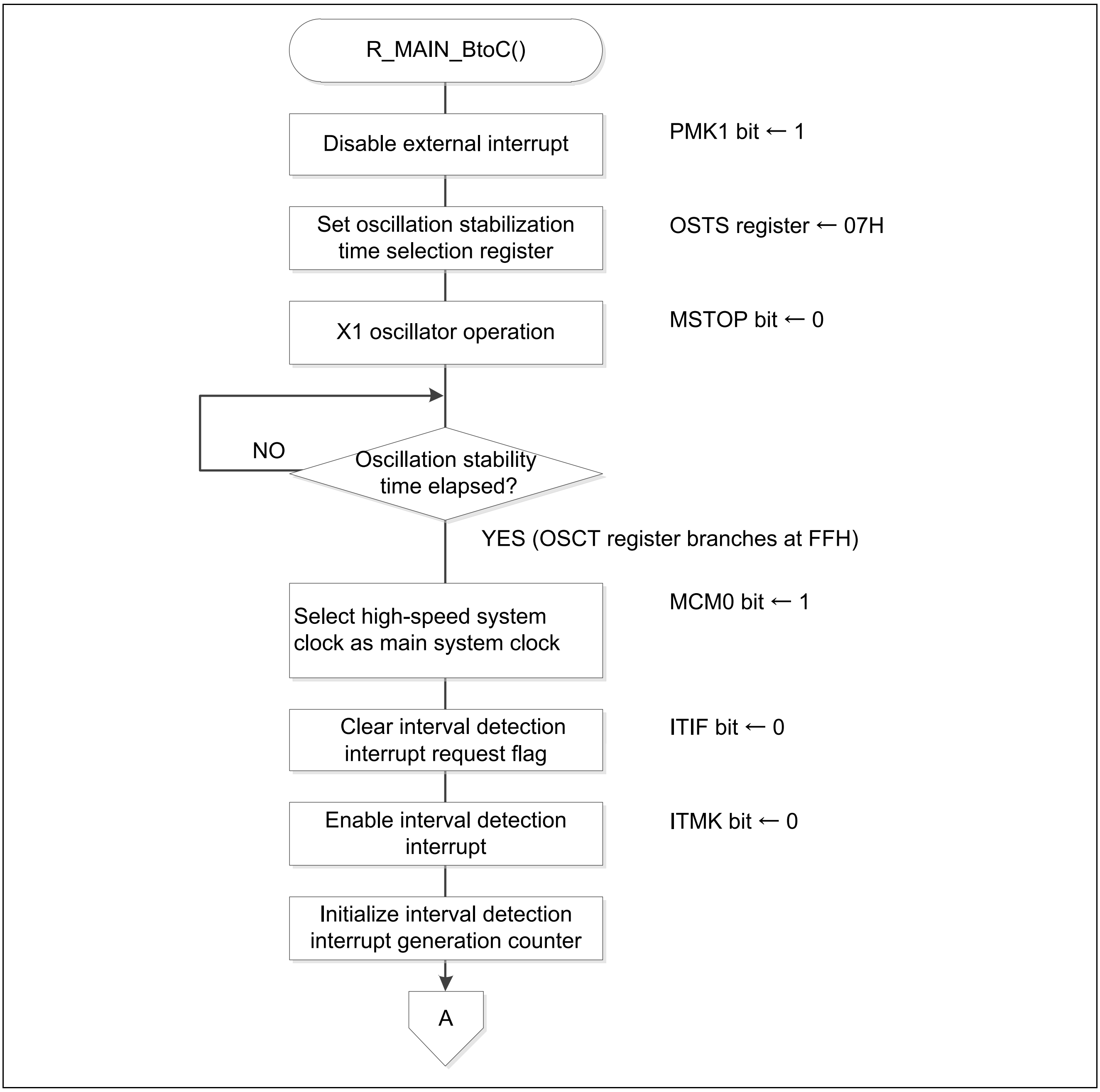

R_MAIN_BtoC()
Disable external interrupt
PMK1 bit ← 1
Set oscillation stabilization time selection register
OSTS register ← 07H
X1 oscillator operation
MSTOP bit ← 0
NO
Oscillation stability time elapsed?
YES (OSCT register branches at FFH)
Select high-speed system clock as main system clock
MCM0 bit ← 1
Clear interval detection interrupt request flag
ITIF bit ← 0
Enable interval detection interrupt
ITMK bit ← 0
Initialize interval detection interrupt generation counter
A

Figure 5.19 Status Transition BtoC (1/2)

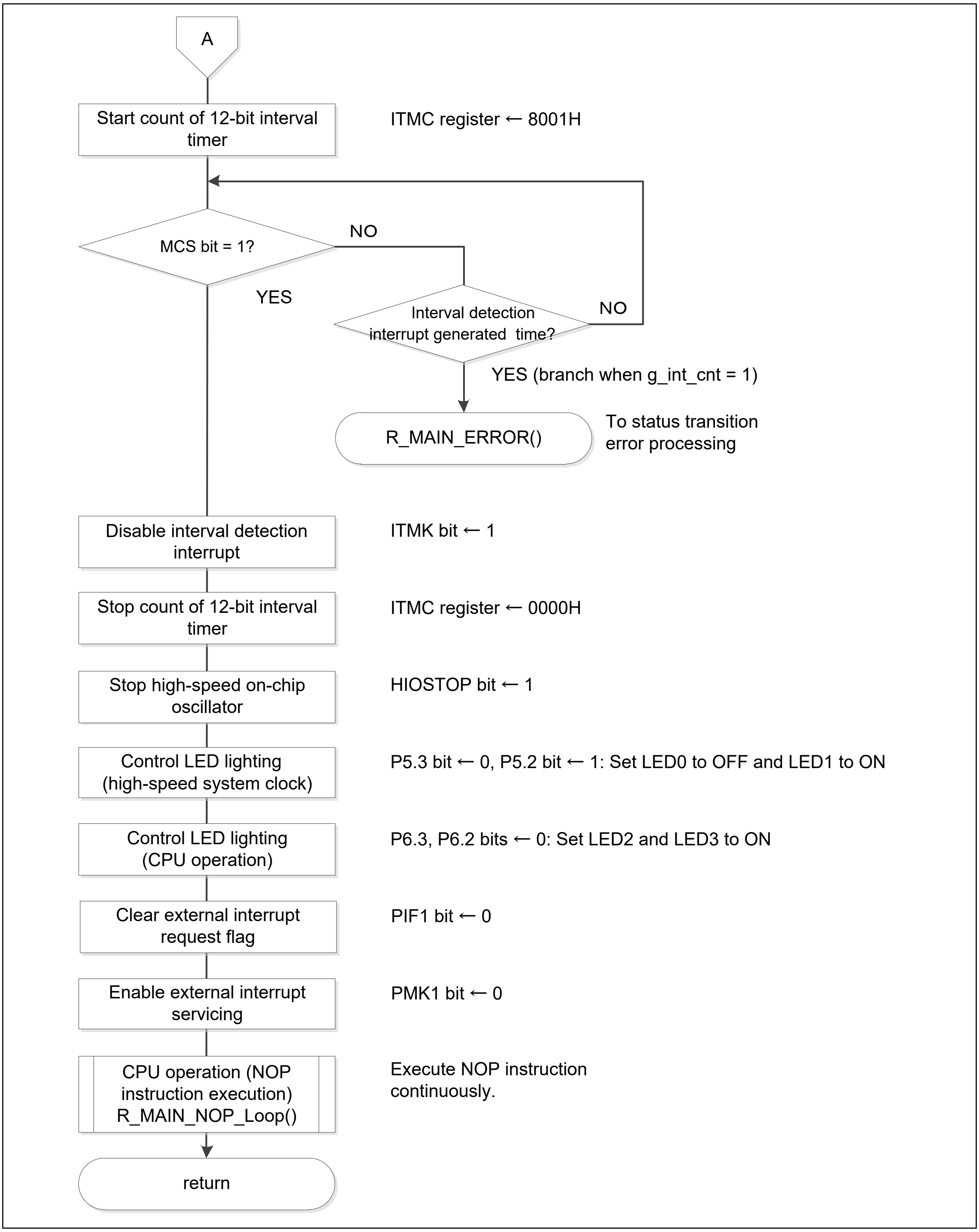

A
Start count of 12-bit interval timer
ITMC register ← 8001H
MCS bit = 1?
NO
YES
Interval detection interrupt generated time?
NO
YES (branch when g_int_cnt = 1)
R_MAIN_ERROR()
To status transition error processing
Disable interval detection interrupt
ITMK bit ← 1
Stop count of 12-bit interval timer
ITMC register ← 0000H
Stop high-speed on-chip oscillator
HIOSTOP bit ← 1
Control LED lighting (high-speed system clock)
P5.3 bit ← 0, P5.2 bit ← 1: Set LED0 to OFF and LED1 to ON
Control LED lighting (CPU operation)
P6.3, P6.2 bits ← 0: Set LED2 and LED3 to ON
Clear external interrupt request flag
PIF1 bit ← 0
Enable external interrupt servicing
PMK1 bit ← 0
CPU operation (NOP instruction execution) R_MAIN_NOP_Loop()
Execute NOP instruction continuously.
return

Figure 5.20 Status Transition BtoC (2/2)

5.6.16 Status Transition CtoD

Figure 5.21 and Figure 5.22 shows the flowchart for status transition CtoD.

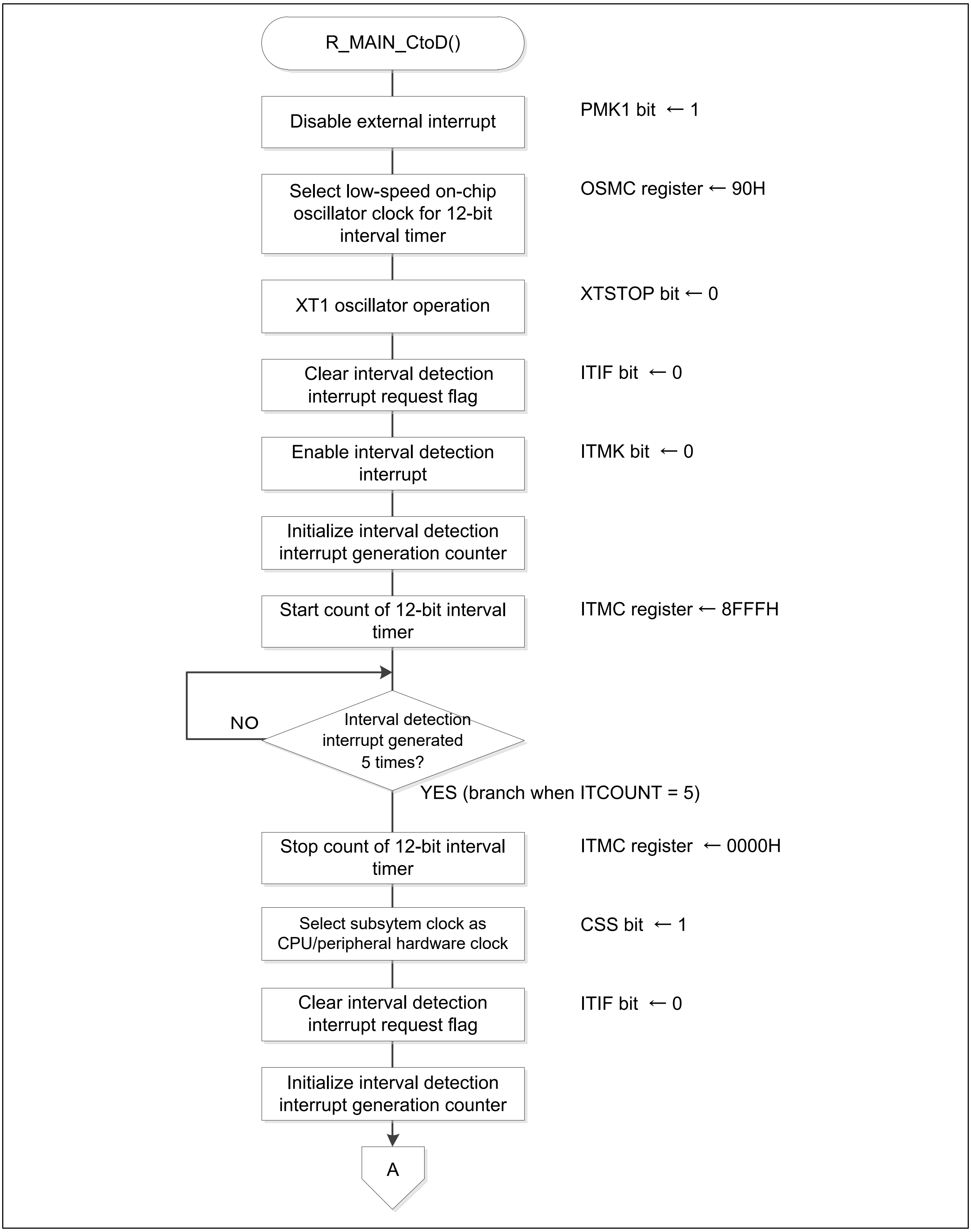

R_MAIN_CtoD()
Disable external interrupt
PMK1 bit ← 1
Select low-speed on-chip oscillator clock for 12-bit interval timer
OSMC register ← 90H
XT1 oscillator operation
XTSTOP bit ← 0
Clear interval detection interrupt request flag
ITIF bit ← 0
Enable interval detection interrupt
ITMK bit ← 0
Initialize interval detection interrupt generation counter
Start count of 12-bit interval timer
ITMC register ← 8FFFH
NO
Interval detection interrupt generated 5 times?
YES (branch when ITCOUNT = 5)
Stop count of 12-bit interval timer
ITMC register ← 0000H
Select subsytem clock as CPU/peripheral hardware clock
CSS bit ← 1
Clear interval detection interrupt request flag
ITIF bit ← 0
Initialize interval detection interrupt generation counter
A

Figure 5.21 Status Transition CtoD (1/2)

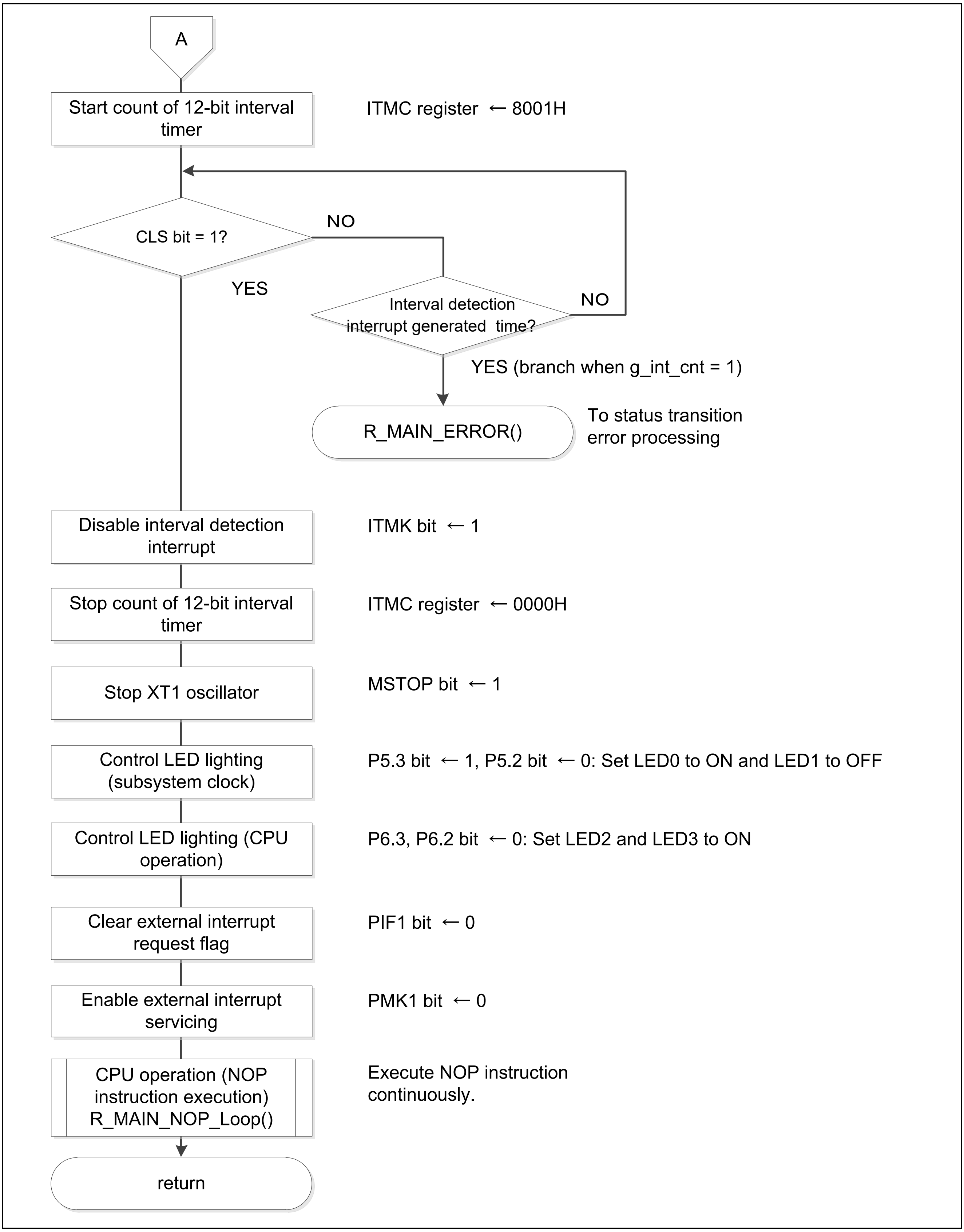

A
Start count of 12-bit interval timer
ITMC register ← 8001H
CLS bit = 1?
NO
YES
Interval detection interrupt generated time?
NO
YES (branch when g_int_cnt = 1)
R_MAIN_ERROR()
To status transition error processing
Disable interval detection interrupt
ITMK bit ← 1
Stop count of 12-bit interval timer
ITMC register ← 0000H
Stop XT1 oscillator
MSTOP bit ← 1
Control LED lighting (subsystem clock)
P5.3 bit ← 1, P5.2 bit ← 0: Set LED0 to ON and LED1 to OFF
Control LED lighting (CPU operation)
P6.3, P6.2 bit ← 0: Set LED2 and LED3 to ON
Clear external interrupt request flag
PIF1 bit ← 0
Enable external interrupt servicing
PMK1 bit ← 0
CPU operation (NOP instruction execution) R_MAIN_NOP_Loop()
Execute NOP instruction continuously.
return

Figure 5.22 Status Transition CtoD (2/2)

5.6.17 Status Transition DtoC

Figure 5.23 and Figure 5.24 shows the flowchart for status transition DtoC.

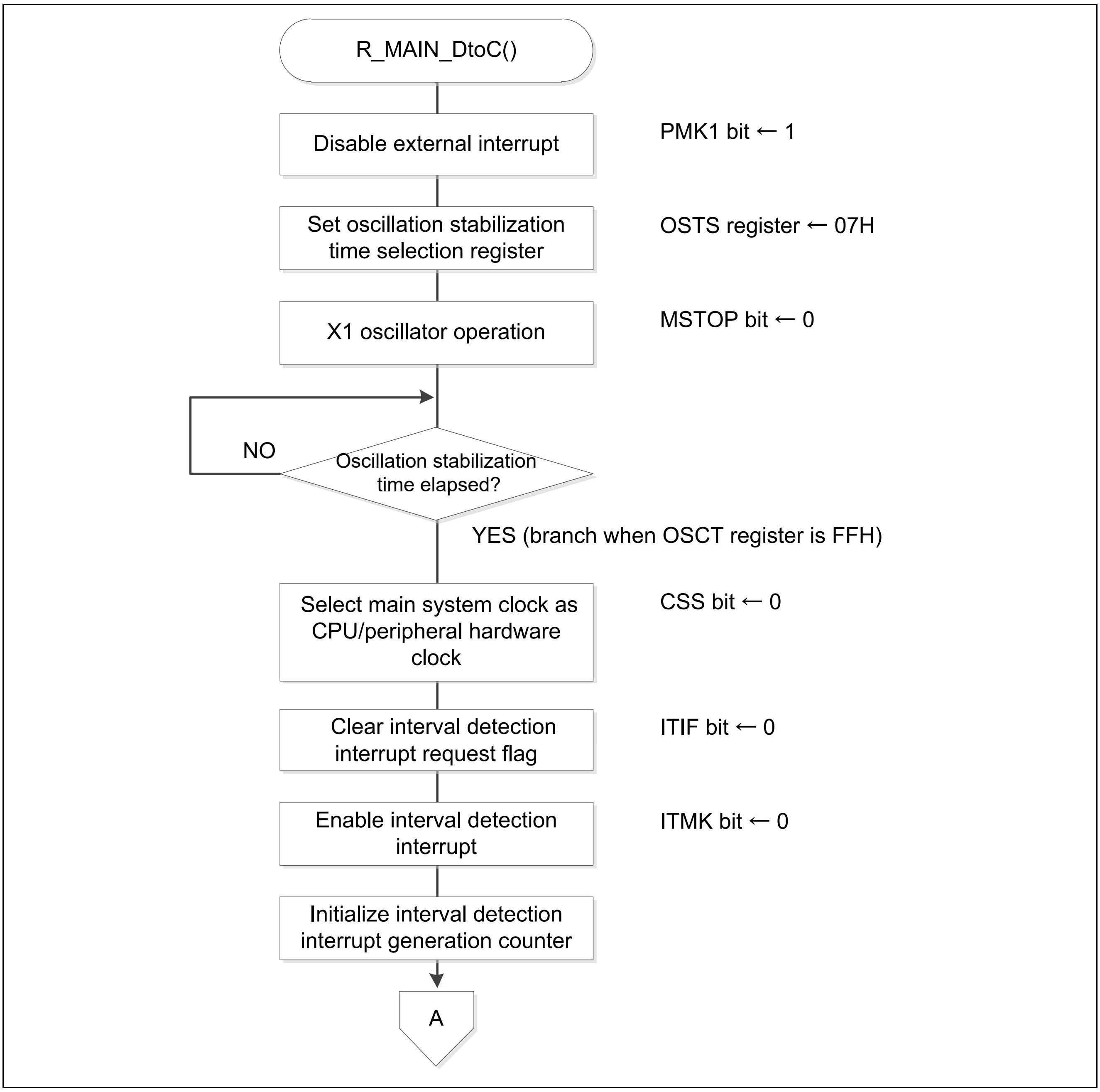

R_MAIN_DtoC()
Disable external interrupt
PMK1 bit ← 1
Set oscillation stabilization time selection register
OSTS register ← 07H
X1 oscillator operation
MSTOP bit ← 0
NO
Oscillation stabilization time elapsed?
YES (branch when OSCT register is FFH)
Select main system clock as CPU/peripheral hardware clock
CSS bit ← 0
Clear interval detection interrupt request flag
ITIF bit ← 0
Enable interval detection interrupt
ITMK bit ← 0
Initialize interval detection interrupt generation counter
A

Figure 5.23 Status Transition DtoC (1/2)

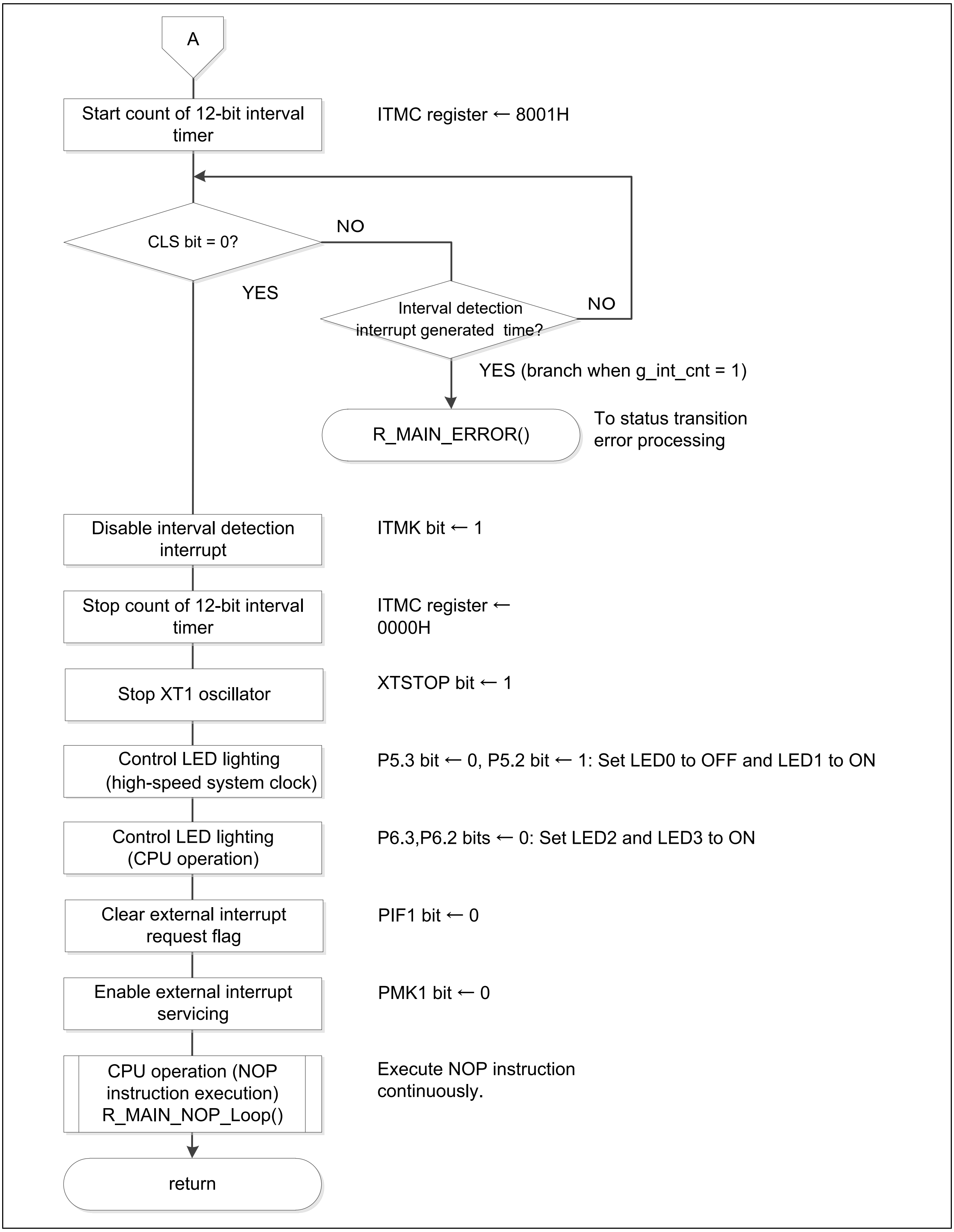

A
Start count of 12-bit interval timer
ITMC register ← 8001H
CLS bit = 0?
NO
YES
Interval detection interrupt generated time?
NO
YES (branch when g_int_cnt = 1)
R_MAIN_ERROR()
To status transition error processing
Disable interval detection interrupt
ITMK bit ← 1
Stop count of 12-bit interval timer
ITMC register ← 0000H
Stop XT1 oscillator
XTSTOP bit ← 1
Control LED lighting (high-speed system clock)
P5.3 bit ← 0, P5.2 bit ← 1: Set LED0 to OFF and LED1 to ON
Control LED lighting (CPU operation)
P6.3,P6.2 bits ← 0: Set LED2 and LED3 to ON
Clear external interrupt request flag
PIF1 bit ← 0
Enable external interrupt servicing
PMK1 bit ← 0
CPU operation (NOP instruction execution) R_MAIN_NOP_Loop()
Execute NOP instruction continuously.
return

Figure 5.24 Status Transition DtoC (2/2)

5.6.18 Status Transition CtoF

Figure 5.25 shows the flowchart for status transition CtoF.

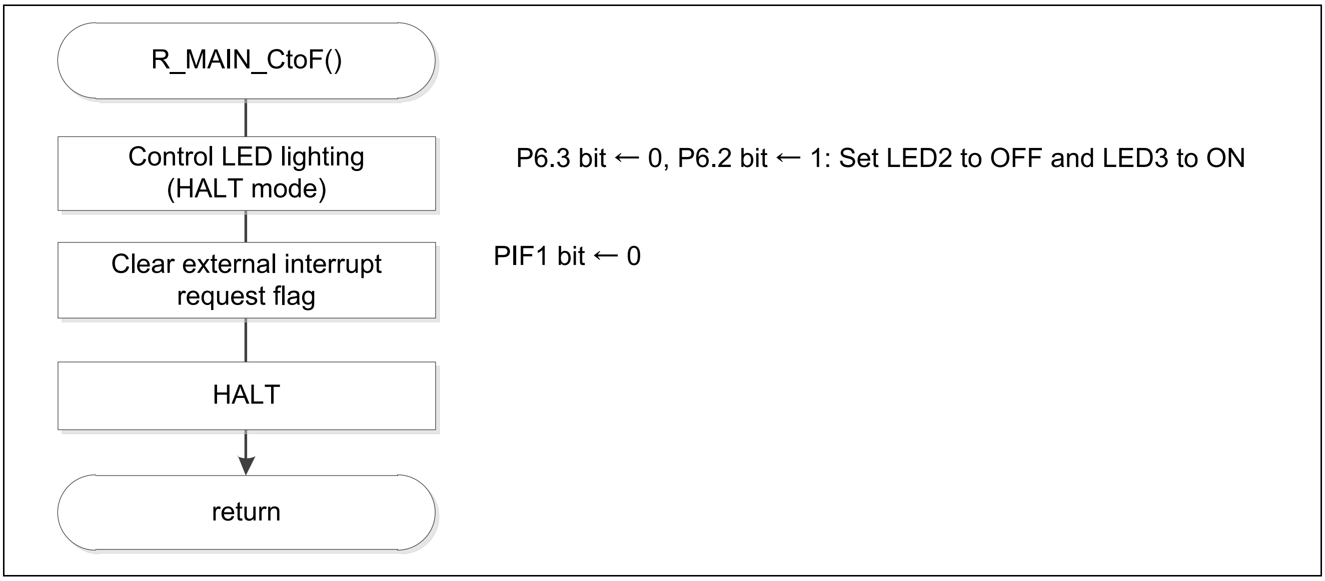

R_MAIN_CtoF()
Control LED lighting (HALT mode)
P6.3 bit ← 0, P6.2 bit ← 1: Set LED2 to OFF and LED3 to ON
Clear external interrupt request flag
PIF1 bit ← 0
HALT
return

Figure 5.25 Status Transition CtoF

5.6.19 Status Transition FtoC

Figure 5.26 shows the flowchart for status transition FtoC.

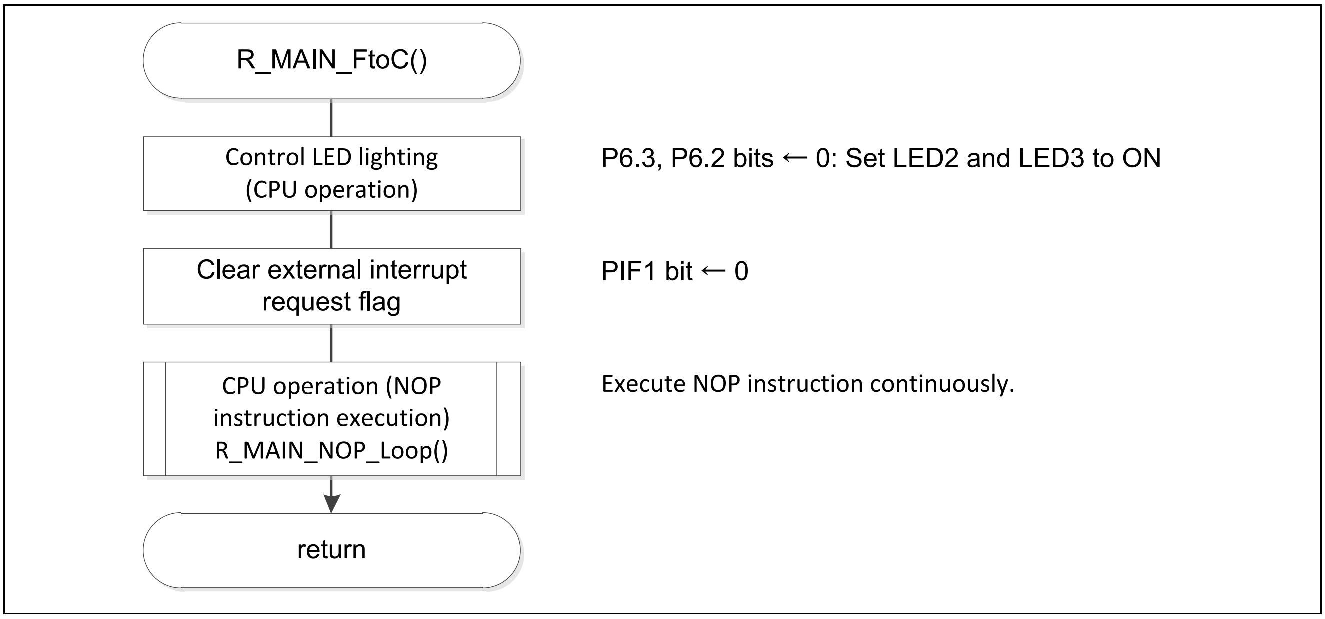

R_MAIN_FtoC()
Control LED lighting (CPU operation)
P6.3, P6.2 bits ← 0: Set LED2 and LED3 to ON
Clear external interrupt request flag
PIF1 bit ← 0
CPU operation (NOP instruction execution) R_MAIN_NOP_Loop()
Execute NOP instruction continuously.
return

Figure 5.26 Status Transition FtoC

5.6.20 Status Transition CtoI

Figure5.27 shows the flowchart for status transition CtoI.

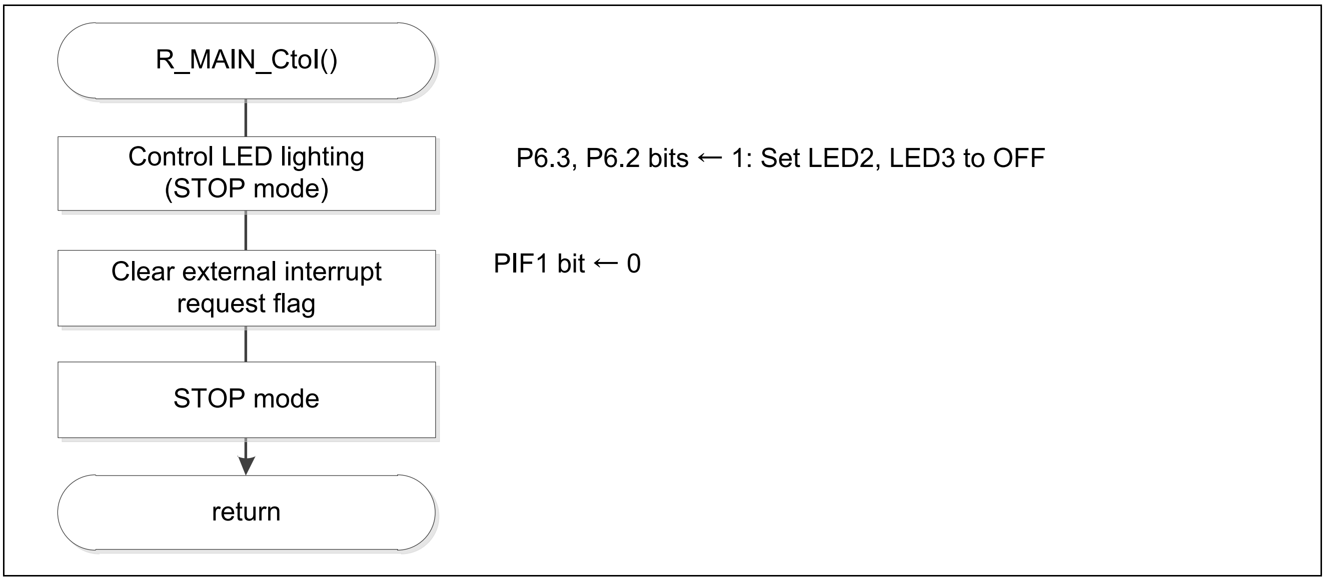

R_MAIN_CtoI()
Control LED lighting (STOP mode)
Clear external interrupt request flag
STOP mode
return
P6.3, P6.2 bits ← 1: Set LED2, LED3 to OFF
PIF1 bit ← 0

Figure 5.27 Status Transition CtoI

5.6.21 Status Transition ItoC

Figure 5.28 shows the flowchart for status transition ItoC.

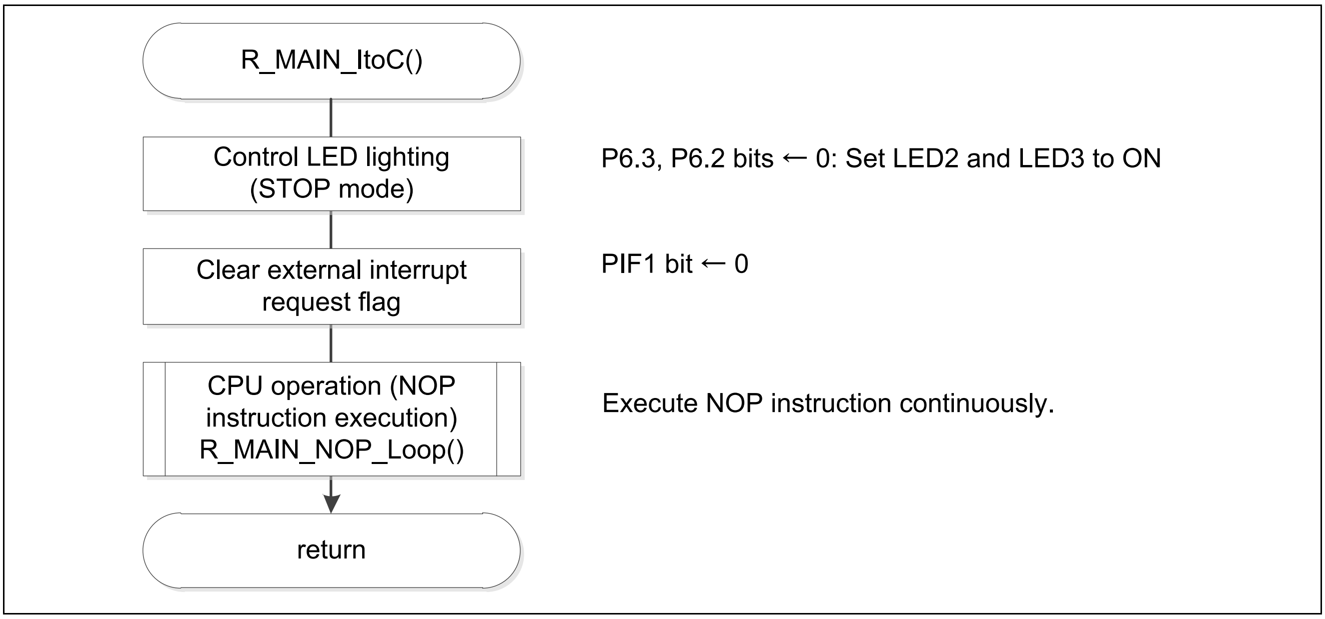

R_MAIN_ItoC()
Control LED lighting (STOP mode)
Clear external interrupt request flag
CPU operation (NOP instruction execution) R_MAIN_NOP_Loop()
return
P6.3, P6.2 bits ← 0: Set LED2 and LED3 to ON
PIF1 bit ← 0
Execute NOP instruction continuously.

Figure 5.28 Status Transition ItoC

5.6.22 Status Transition CtoB

Figure 5.29 and Figure 5.30 shows the flowchart for status transition CtoB.

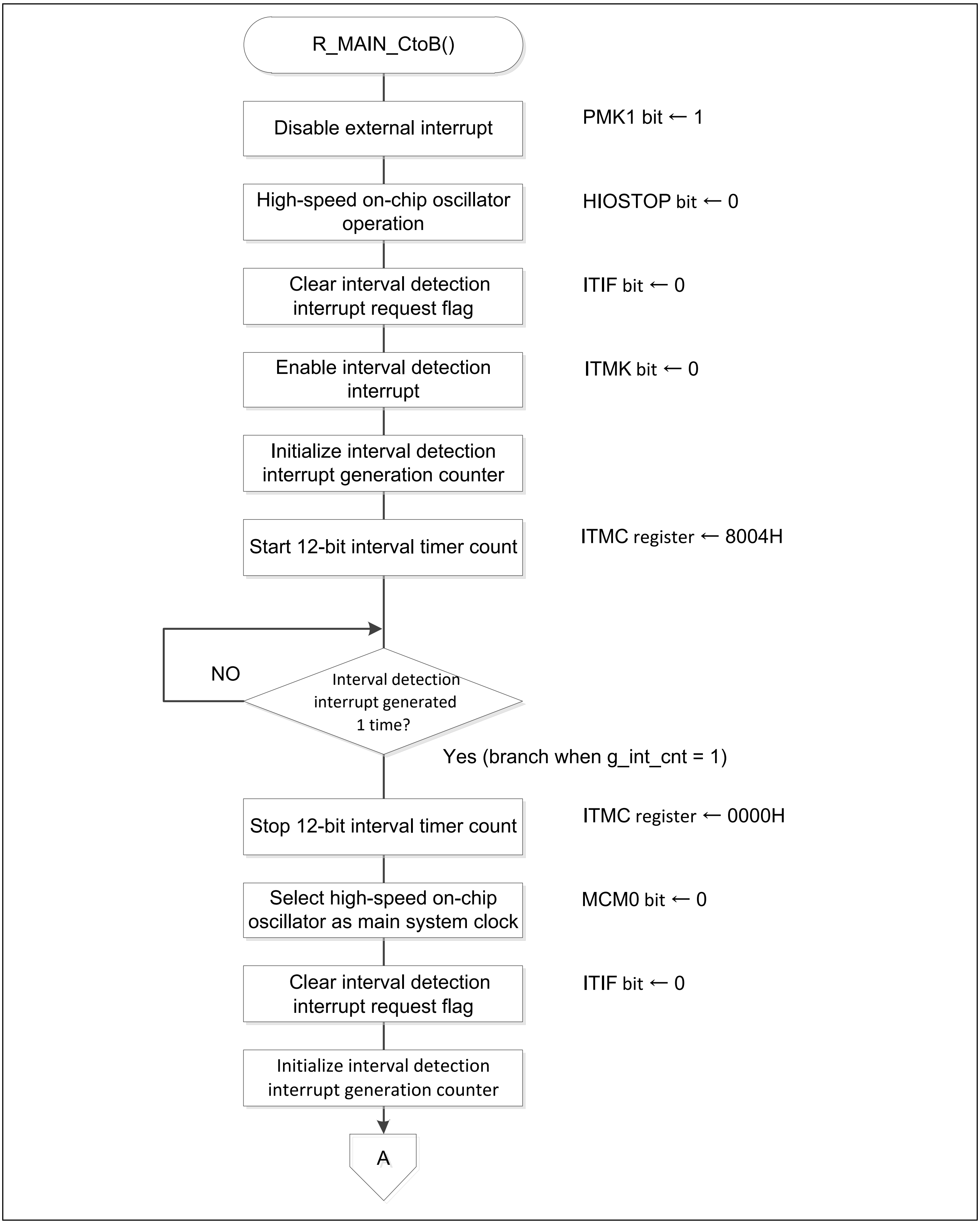

R_MAIN_CtoB()
Disable external interrupt
PMK1 bit ← 1
High-speed on-chip oscillator operation
HIOSTOP bit ← 0
Clear interval detection interrupt request flag
ITIF bit ← 0
Enable interval detection interrupt
ITMK bit ← 0
Initialize interval detection interrupt generation counter
Start 12-bit interval timer count
ITMC register ← 8004H
NO
Interval detection interrupt generated 1 time?
Yes (branch when g_int_cnt = 1)
Stop 12-bit interval timer count
ITMC register ← 0000H
Select high-speed on-chip oscillator as main system clock
MCM0 bit ← 0
Clear interval detection interrupt request flag
ITIF bit ← 0
Initialize interval detection interrupt generation counter
A

Figure 5.29 Status Transition CtoB (1/2)

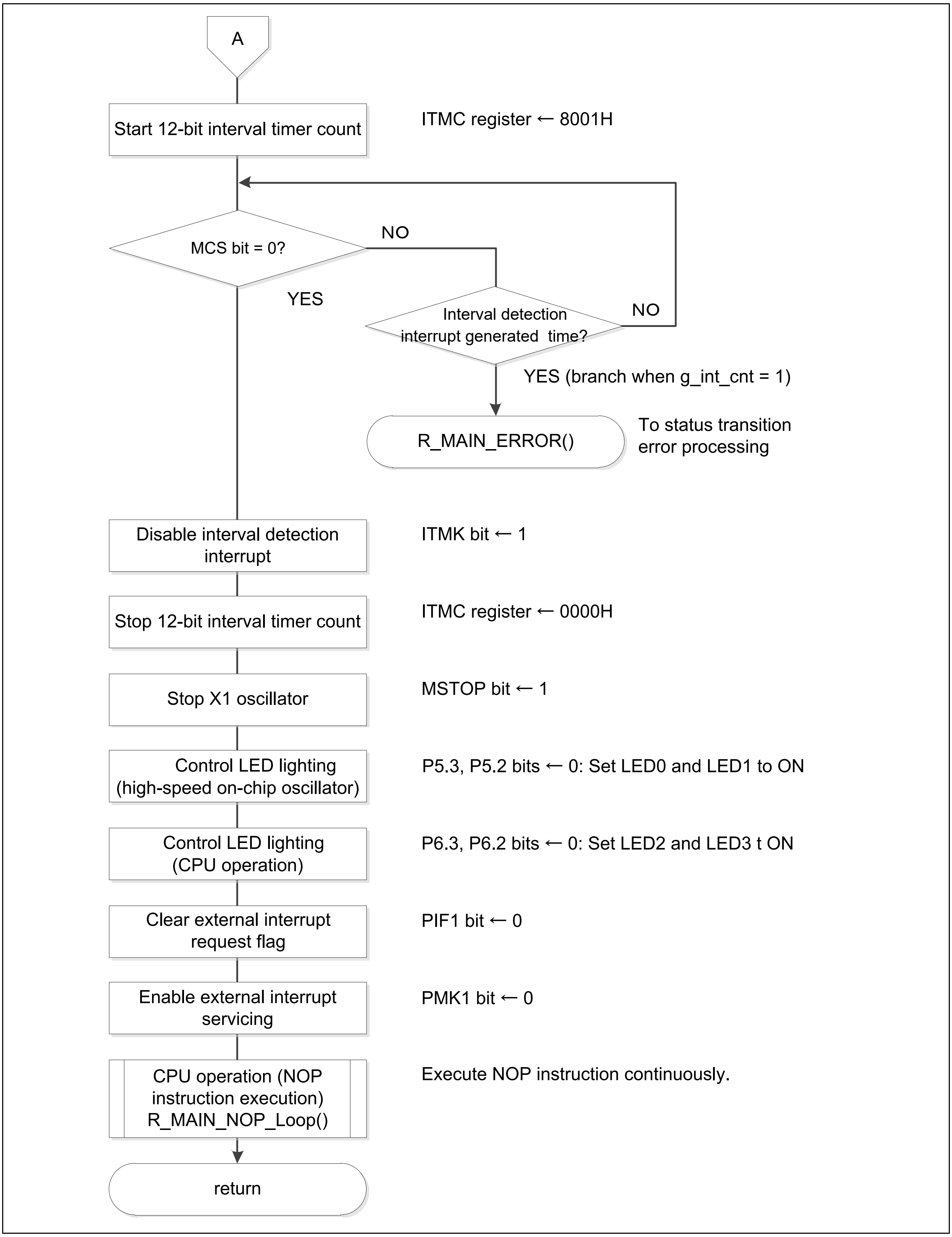

A
Start 12-bit interval timer count
ITMC register ← 8001H
MCS bit = 0?
NO
YES
Interval detection interrupt generated time?
NO
YES (branch when g_int_cnt = 1)
R_MAIN_ERROR()
To status transition error processing
Disable interval detection interrupt
ITMK bit ← 1
Stop 12-bit interval timer count
ITMC register ← 0000H
Stop X1 oscillator
MSTOP bit ← 1
Control LED lighting (high-speed on-chip oscillator)
P5.3, P5.2 bits ← 0: Set LED0 and LED1 to ON
Control LED lighting (CPU operation)
P6.3, P6.2 bits ← 0: Set LED2 and LED3 t ON
Clear external interrupt request flag
PIF1 bit ← 0
Enable external interrupt servicing
PMK1 bit ← 0
CPU operation (NOP instruction execution) R_MAIN_NOP_Loop()
Execute NOP instruction continuously.
return

Figure 5.30 Status Transition CtoB(2/2)

5.6.23 Status Transition BtoE

Figure 5.31 shows the flowchart for status transition BtoE.

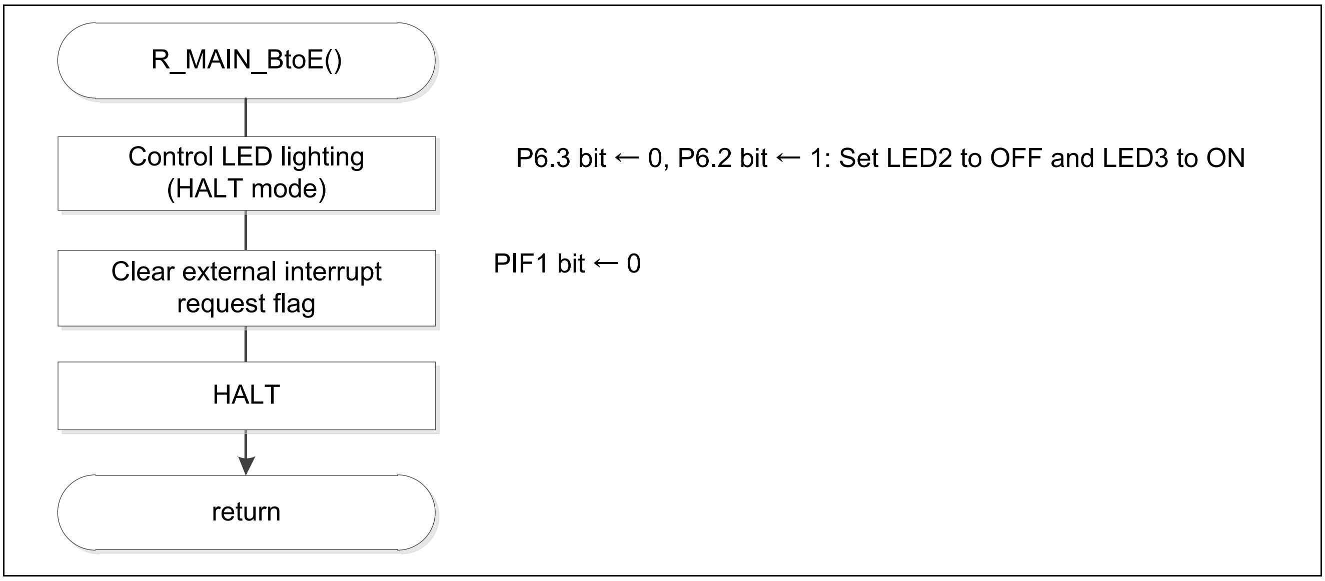

R_MAIN_BtoE()
Control LED lighting (HALT mode)
P6.3 bit ← 0, P6.2 bit ← 1: Set LED2 to OFF and LED3 to ON
Clear external interrupt request flag
PIF1 bit ← 0
HALT
return

Figure 5.31 Status Transition BtoE

5.6.24 Status Transition EtoB

Figure 5.32 shows the flowchart for status transition EtoB.

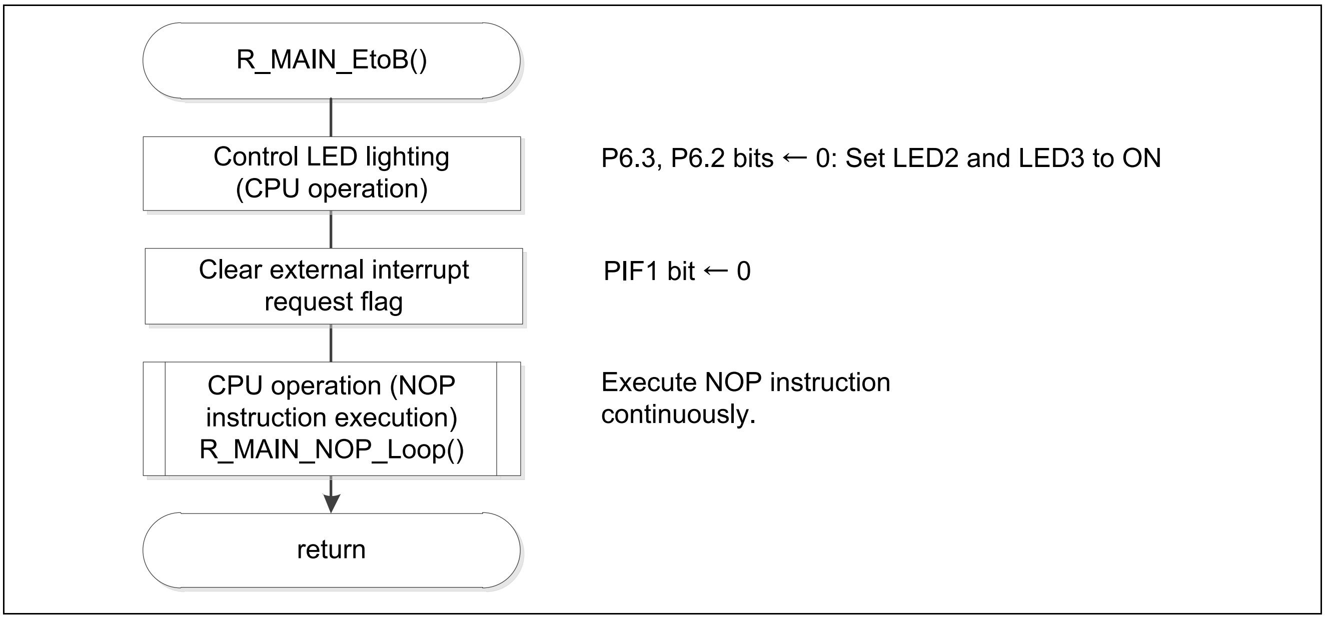

R_MAIN_EtoB()
Control LED lighting (CPU operation)
P6.3, P6.2 bits ← 0: Set LED2 and LED3 to ON
Clear external interrupt request flag
PIF1 bit ← 0
CPU operation (NOP instruction execution) R_MAIN_NOP_Loop()
Execute NOP instruction continuously.
return

Figure 5.32 Status Transition EtoB

5.6.25 Status Transition BtoH

Figure 5.33 shows the flowchart for status transition BtoH.

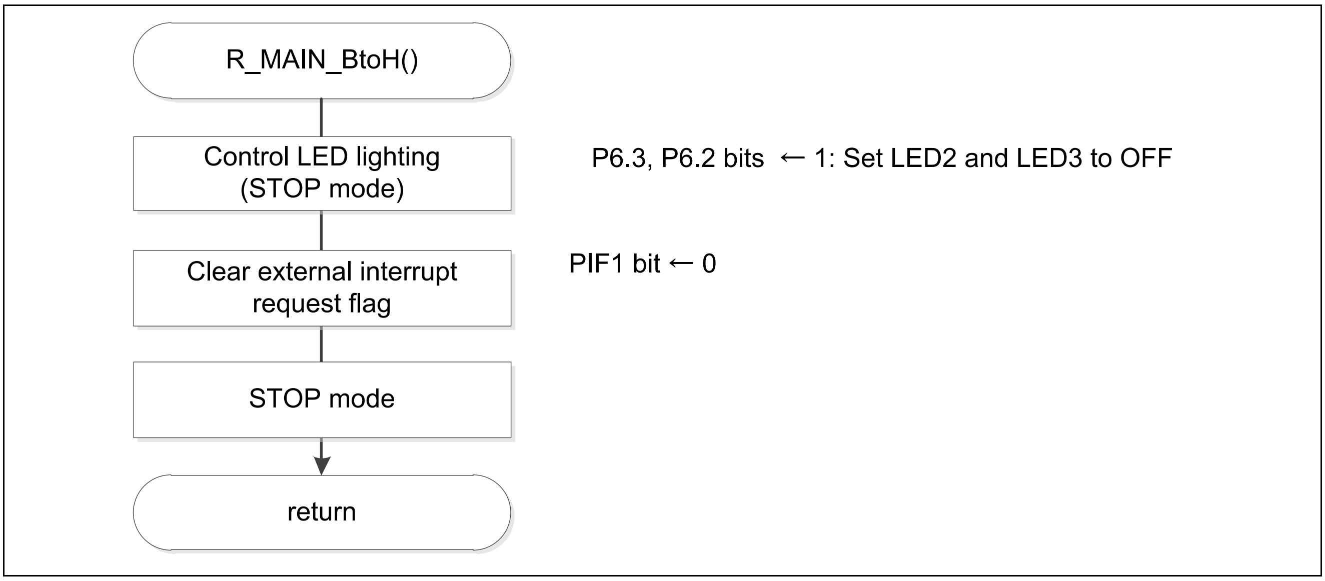

R_MAIN_BtoH()
Control LED lighting (STOP mode)
P6.3, P6.2 bits ← 1: Set LED2 and LED3 to OFF
Clear external interrupt request flag
PIF1 bit ← 0
STOP mode
return

Figure 5.33 Status Transition BtoH

5.6.26 Status Transition HtoB

Figure 5.34 shows the flowchart for status transition HtoB.

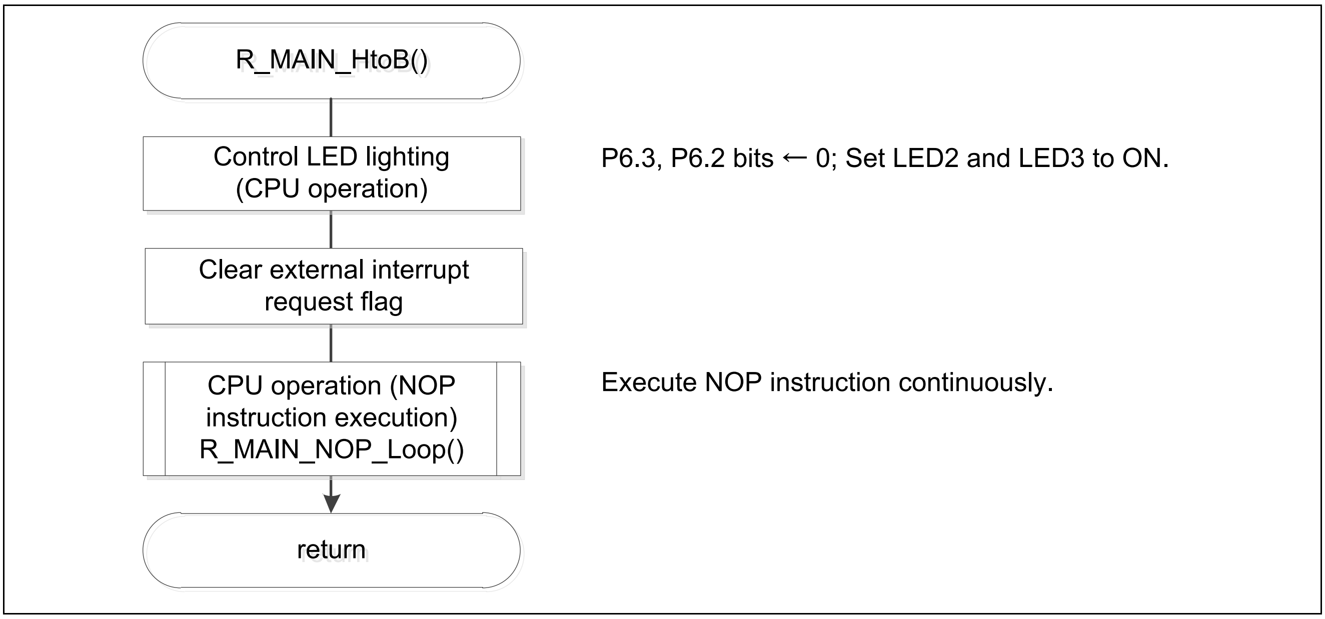

R_MAIN_HtoB()
Control LED lighting (CPU operation)
P6.3, P6.2 bits ← 0; Set LED2 and LED3 to ON.
Clear external interrupt request flag
CPU operation (NOP instruction execution) R_MAIN_NOP_Loop()
Execute NOP instruction continuously.
return

Figure 5.34 Status Transition HtoB

5.6.27 Status Transition BtoJ

Figure 5.35 shows the flowchart for status transition BtoJ.

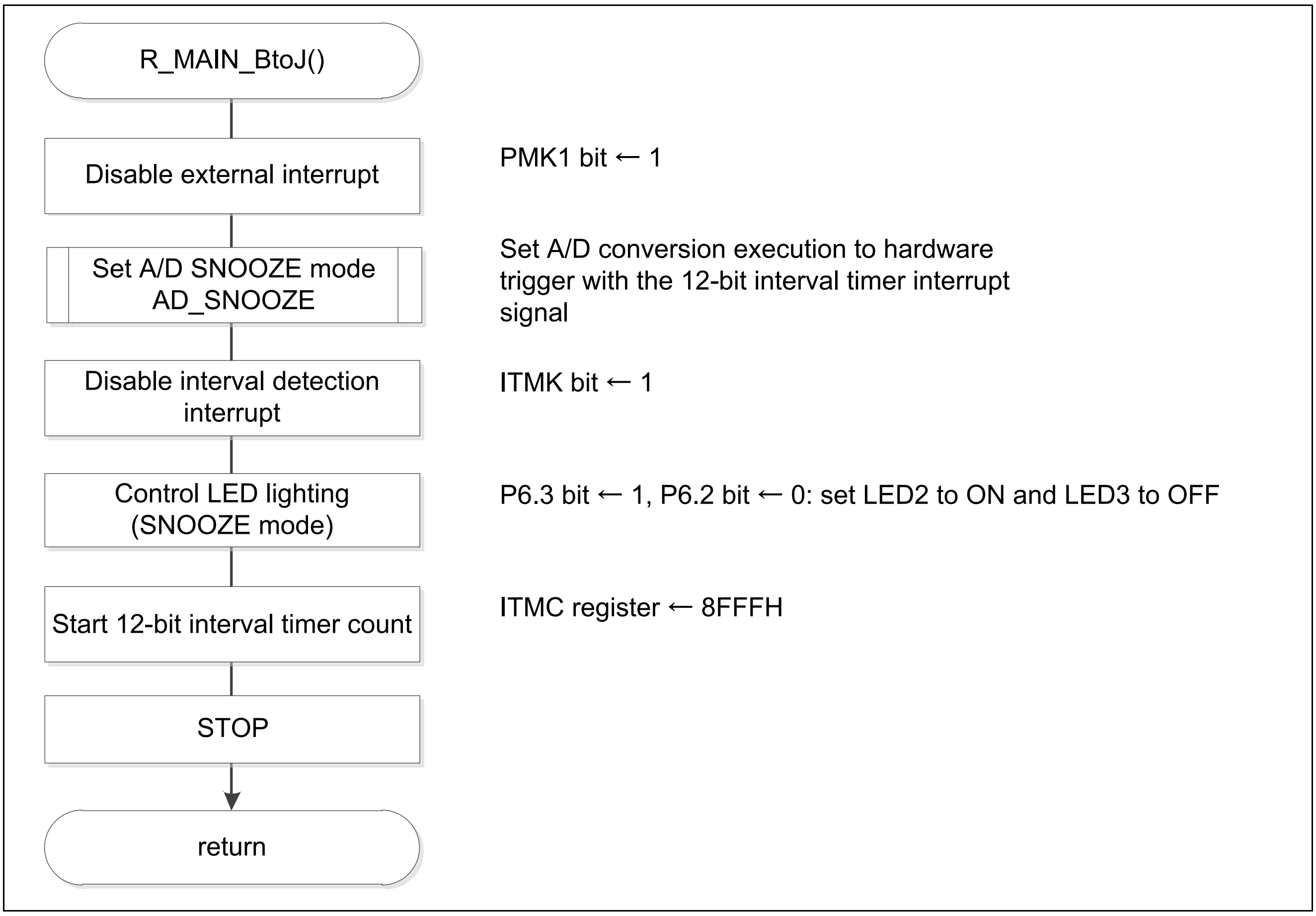

R_MAIN_BtoJ()
Disable external interrupt
PMK1 bit ← 1
Set A/D SNOOZE mode AD_SNOOZE
Set A/D conversion execution to hardware trigger with the 12-bit interval timer interrupt signal
Disable interval detection interrupt
ITMK bit ← 1
Control LED lighting (SNOOZE mode)
P6.3 bit ← 1, P6.2 bit ← 0: set LED2 to ON and LED3 to OFF
Start 12-bit interval timer count
ITMC register ← 8FFFH
STOP
return

Figure 5.35 Status Transition BtoJ

5.6.28 A/D Converter Setting

Figure5.36 shows the flowchart for setting the A/D converter.

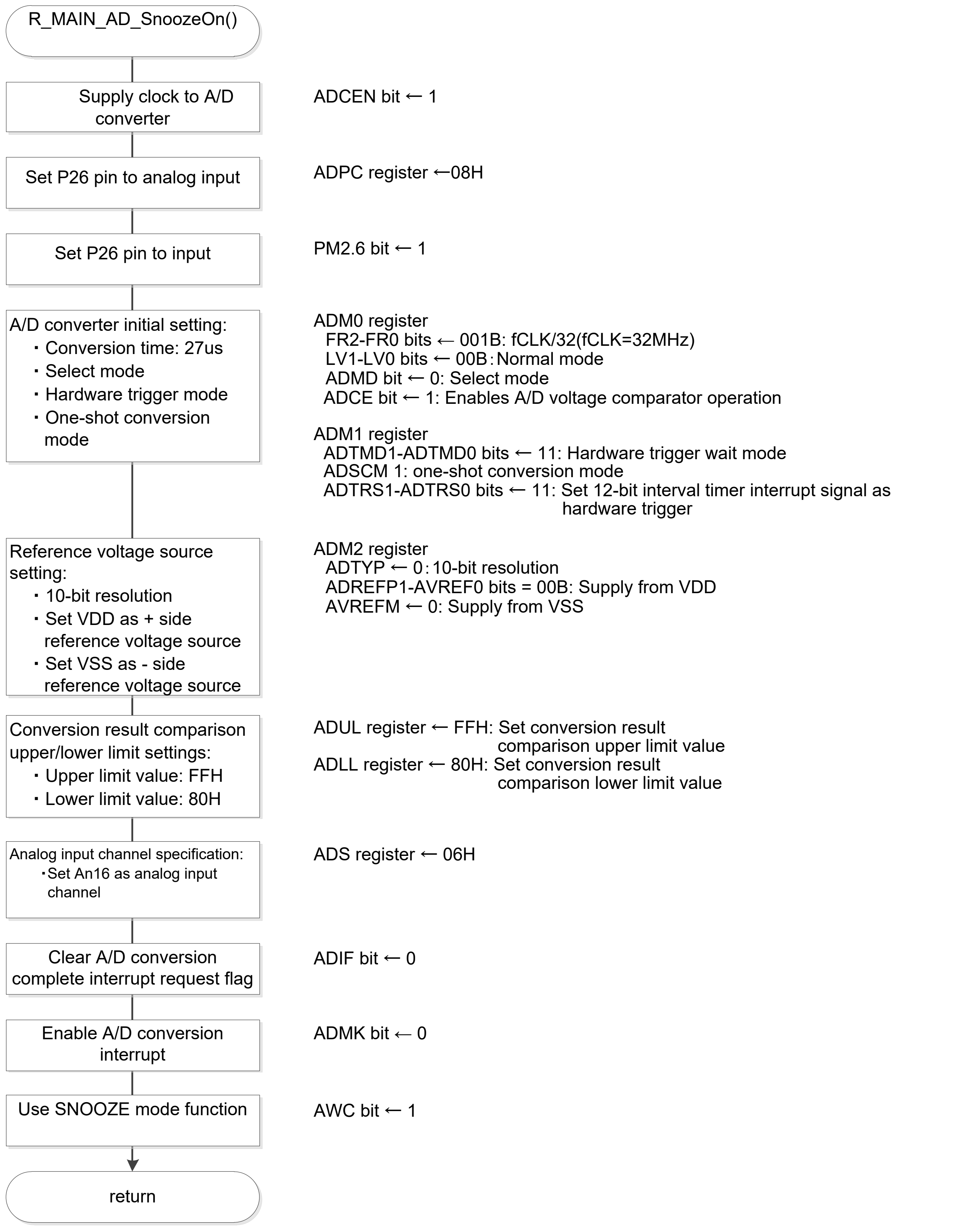

R_MAIN_AD_SnoozeOn()
Supply clock to A/D converter
ADCEN bit ← 1
Set P26 pin to analog input
ADPC register ←08H
Set P26 pin to input
PM2.6 bit ← 1
A/D converter initial setting:
・Conversion time: 27us
・Select mode
・Hardware trigger mode
・One-shot conversion mode
ADM0 register
FR2-FR0 bits ← 001B: fCLK/32(fCLK=32MHz)
LV1-LV0 bits ← 00B：Normal mode
ADMD bit ← 0: Select mode
ADCE bit ← 1: Enables A/D voltage comparator operation
ADM1 register
ADTMD1-ADTMD0 bits ← 11: Hardware trigger wait mode
ADSCM 1: one-shot conversion mode
ADTRS1-ADTRS0 bits ← 11: Set 12-bit interval timer interrupt signal as hardware trigger
Reference voltage source setting:
・10-bit resolution
・Set VDD as + side reference voltage source
・Set VSS as - side reference voltage source
ADM2 register
ADTYP ← 0：10-bit resolution
ADREFP1-AVREF0 bits = 00B: Supply from VDD
AVREFM ← 0: Supply from VSS
Conversion result comparison upper/lower limit settings:
・Upper limit value: FFH
・Lower limit value: 80H
ADUL register ← FFH: Set conversion result comparison upper limit value
ADLL register ← 80H: Set conversion result comparison lower limit value
Analog input channel specification:
・Set An16 as analog input channel
ADS register ← 06H
Clear A/D conversion complete interrupt request flag
ADIF bit ← 0
Enable A/D conversion interrupt
ADMK bit ← 0
Use SNOOZE mode function
AWC bit ← 1
return

Figure 5.36 A/D Converter Setting

A/D conversion time and operation mode settings

- A/D converter mode register 0 (ADM0)

Control the A/D conversion operation.
Specify the A/D conversion channel selection mode.

Symbol: ADM0

7	6	5	4	3	2	1	0
ADCS	ADMD	FR2	FR1	FR0	LV1	LV0	ADCE
x	**0**	**0**	**0**	**1**	**0**	**0**	**1**

Bit 6

ADMD	Specification of A/D channel selection mode
0	**Select mode**
1	Scan mode

Bits 5-1

ADM0					Mode	Conversion Time					Conversion clock (f_{AD})
FR2	FR1	FR0	LV1	LV0		f_{CLK}= 1MHz	f_{CLK}= 4MHz	f_{CLK}= 8MHz	f_{CLK}= 16MHz	**f_{CLK}= 32MHz**	
0	0	0	**0**	**0**	**Normal 1**	Setting prohibited	Setting prohibited	Setting prohibited	108 µs	54 µs	f_{CLK}/64
0	**0**	**1**						108 µs	54 µs	**27 µs**	f_{CLK}/32
0	1	0					108 µs	54 µs	27 µs	13.5 µs	f_{CLK}/16
0	1	1					54 µs	27 µs	13.5 µs	6.75 µs	f_{CLK}/8
1	0	0					40.5 µs	20.25 µs	10.125 µs	5.0625 µs	f_{CLK}/6
1	0	1				135 µ s	33.75 µs	16.875 µs	8.4375 µs	4.21875 µs	f_{CLK}/5
1	1	0				108 µ s	27 µs	13.5 µs	6.75 µs	3.375 µs	f_{CLK}/4
1	1	1				54 µ s	13.5 µs	6.75 µs	3.375 µs	Setting prohibited	f_{CLK}/2
0	0	0	0	1	Normal 2	Setting prohibited	Setting prohibited	Setting prohibited	100 µs	50 µs	f_{CLK}/64
0	0	1						100 µs	50 µs	25 µs	f_{CLK}/32
0	1	0					100 µs	50 µs	25 µs	12.5 µs	f_{CLK}/16
0	1	1					50 µs	25 µs	12.5 µs	6.25 µs	f_{CLK}/8
1	0	0					37.5 µs	18.75 µs	9.375 µs	4.6875 µs	f_{CLK}/6
1	0	1				125 µ s	31.25 µs	15.625 µs	7.8125 µs	3.90625 µs	f_{CLK}/5
1	1	0				100 µ s	25 µs	12.5 µs	6.25 µs	3.125 µs	f_{CLK}/4
1	1	1				50 µ s	12.5 µs	6.25 µs	3.125 µs	Setting prohibited	f_{CLK}/2

Bit 0

ADCE	A/D voltage comparator operation control
0	Stops A/D enables comparator operation
1	**Enables A/D voltage comparator operation**

Note: Refer to the RL78/G13 User's Manual (hardware version) for detailed explanations on how to set registers.

A/D conversion trigger mode setting

- A/D converter mode register 1 (ADM1)
  Select the A/D conversion trigger mode.
  Specify the A/D conversion operation mode.

Symbol: ADM1

7	6	5	4	3	2	1	0
ADTMD1	ADTMD0	ADSCM	0	0	0	ADTRS1	ADTRS0
1	**1**	**1**	0	0	0	**1**	**1**

Bits 7-6

ADTMD1	ADTMD0	Selection of the A/D conversion trigger mode
0	—	Software trigger mode
1	0	Hardware trigger no-wait mode
1	1	**Hardware trigger wait mode**

Bit 5

ADSCM	Specification of the A/D conversion mode
0	Sequential conversion mode
1	**One-shot conversion mode**

Bits 1-0

ADTRS1	ADTRS0	Selection of the hardware trigger signal
0	0	End of timer channel 01 count or capture interrupt signal (INTTM01)
0	1	Setting prohibited
1	0	Real-time clock interrupt signal (INTRTC)
1	**1**	**12-bit interval timer interrupt signal (INTIT)**

Note: Refer to the RL78/G13 User’s Manual (hardware version) for detailed explanations on how to set registers.

Reference voltage source setting

・A/D converter mode register 2 (ADM2)
Set the reference voltage source.

Symbol: ADM2

7	6	5	4	3	2	1	0
ADREFP1	ADREFP0	ADREFM	0	ADCRK	AWC	0	ADTYP
0	**0**	**0**	0	**0**	**0**	0	**0**

Bits 7-6

ADREFP1	ADREFP0	Selection of the + side reference voltage source of the A/D converter
0	**0**	**Supplied from V_{DD}**
0	1	Supplied fromP20/AV_{REFP}/ANI0
1	0	Supplied from the internal reference voltage (1.45 V)
1	1	Setting prohibited

Bit 5

ADREFM	Selection of the - side reference voltage source of the A/D converter
0	**Supplied from V_{SS}**
1	Supplied from P21/AV_{REFM}/ANI1

Bit 3

ADCRK	Checking the upper and lower limit conversion result values
0	**The interrupt signal (INTAD) is output when ADLL register ≤ ADCR register ≤ register ADUL.**
1	The interrupt signal (INTAD) is output when ADCR register ＜ADLL register, and ADUL register ＜ ADCR register.

Bit 2

AWC	Specification of the SNOOZE mode
0	**Do not use the SNOOZE mode function.**
1	Use the SNOOZE mode function.

Bit 0

ADTYP	Selection of the A/D conversion resolution
0	**10-bit resolution**
1	8-bit resolution

Conversion result comparison upper/lower limit settings

- Conversion result comparison upper limit setting register (ADUL)
- Conversion result comparison lower limit setting register (ADLL)

Set conversion result comparison upper/lower limit values.

Symbol: ADUL

7	6	5	4	3	2	1	0
ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Symbol: ADLL

7	6	5	4	3	2	1	0
ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0
1	**0**	**0**	**0**	**0**	**0**	**0**	**0**

Input channel specification

- Analog input channel specification register (ADS)

Specify the input channel of the analog voltage to be A/D converted.

Symbol: ADS

7	6	5	4	3	2	1	0
ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0
0	0	0	**0**	**0**	**1**	**1**	**0**

Bits 7, 4-0

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0 pin/AV$_{REFP}$ pin
0	0	0	0	0	1	ANI1	P21/ANI1 pin /AV$_{REFM}$ pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	0	0	1	0	0	ANI4	P24/ANI4 pin
0	0	0	1	0	1	ANI5	P25/ANI5 pin
0	**0**	**0**	**1**	**1**	**0**	**ANI6**	**P26/ANI6 pin**
0	0	0	1	1	1	ANI7	P27/ANI7 pin
0	1	0	0	0	0	ANI16	P03/ANI16 pin
0	1	0	0	0	1	ANI17	P02/ANI17 pin
0	1	0	0	1	0	ANI18	P147/ANI18 pin
0	1	0	0	1	1	ANI19	P120/ANI19 pin
1	0	0	0	0	0	—	Temperature sensor 0 output
1	0	0	0	0	1	—	Internal reference voltage output (1.45 V)
上記以外						Setting prohibited	

Note: Refer to the RL78/G13 User’s Manual (hardware version) for detailed explanations on how to set registers.

SNOOZE mode setting

・A/D converter mode register 2 (ADM2)
Set SNOOZE mode.

Symbol: ADM2

7	6	5	4	3	2	1	0
ADREFP1	ADREFP0	ADREFM	0	ADCRK	AWC	0	ADTYP
x	x	x	0	x	**1**	0	x

Bit 2

AWC	Specification of SNOOZE mode
0	Do not use the SNOOZE mode function
1	**Use the SNOOZE mode function**

Note: Refer to the RL78/G13 User's Manual (hardware version) for detailed explanations on how to set registers.

5.6.29 Status Transition JtoB

Figure5.37 shows the status transition JtoB.

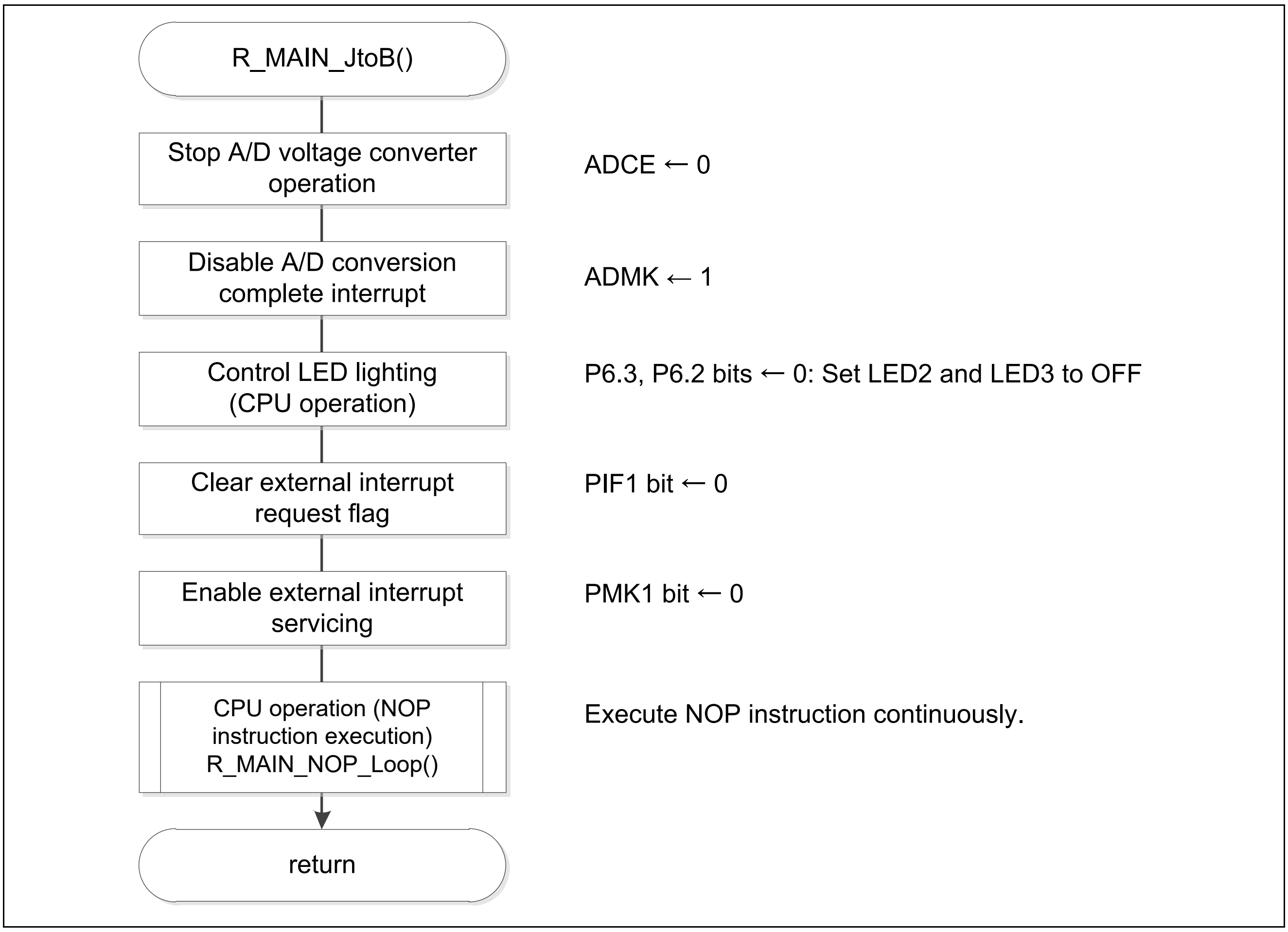

R_MAIN_JtoB()
Stop A/D voltage converter operation
ADCE ← 0
Disable A/D conversion complete interrupt
ADMK ← 1
Control LED lighting (CPU operation)
P6.3, P6.2 bits ← 0: Set LED2 and LED3 to OFF
Clear external interrupt request flag
PIF1 bit ← 0
Enable external interrupt servicing
PMK1 bit ← 0
CPU operation (NOP instruction execution) R_MAIN_NOP_Loop()
Execute NOP instruction continuously.
return

Figure 5.37 Status Transition JtoB

5.6.30 Status Transition End Processing

Figure 5.38 shows the flowchart for status transition end processing.

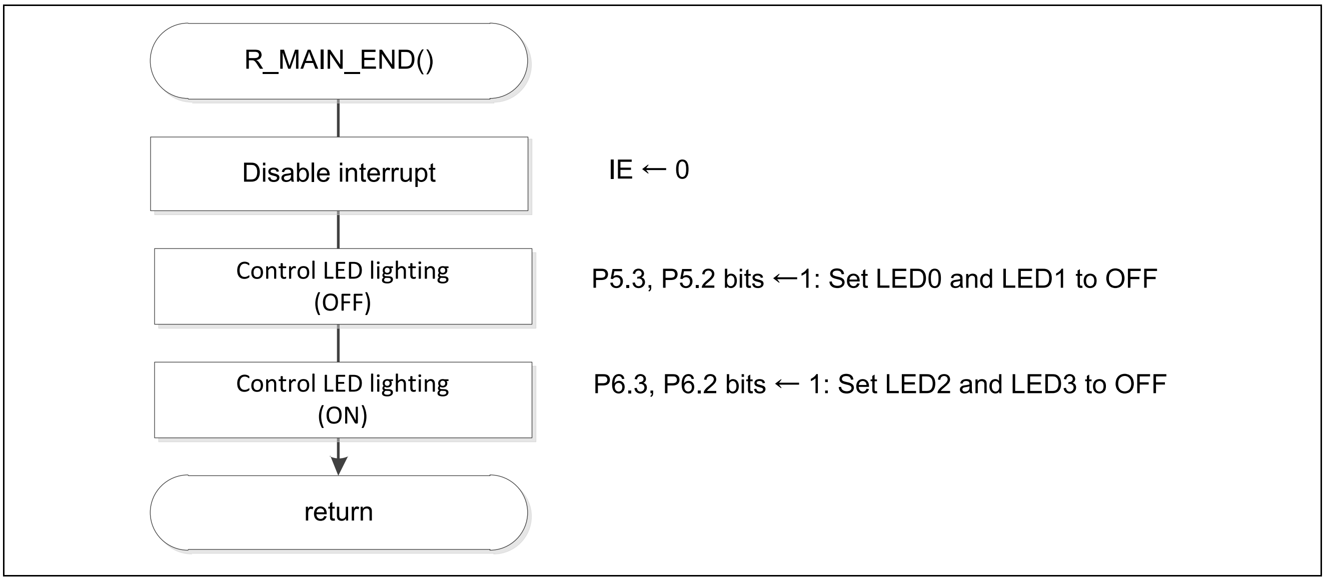

R_MAIN_END()
Disable interrupt
IE ← 0
Control LED lighting (OFF)
P5.3, P5.2 bits ←1: Set LED0 and LED1 to OFF
Control LED lighting (ON)
P6.3, P6.2 bits ← 1: Set LED2 and LED3 to OFF
return

Figure 5.38 Status Transition End Processing

5.6.31 External Interrupt Servicing

Figure 5.39 shows the flowchart for external interrupt servicing.

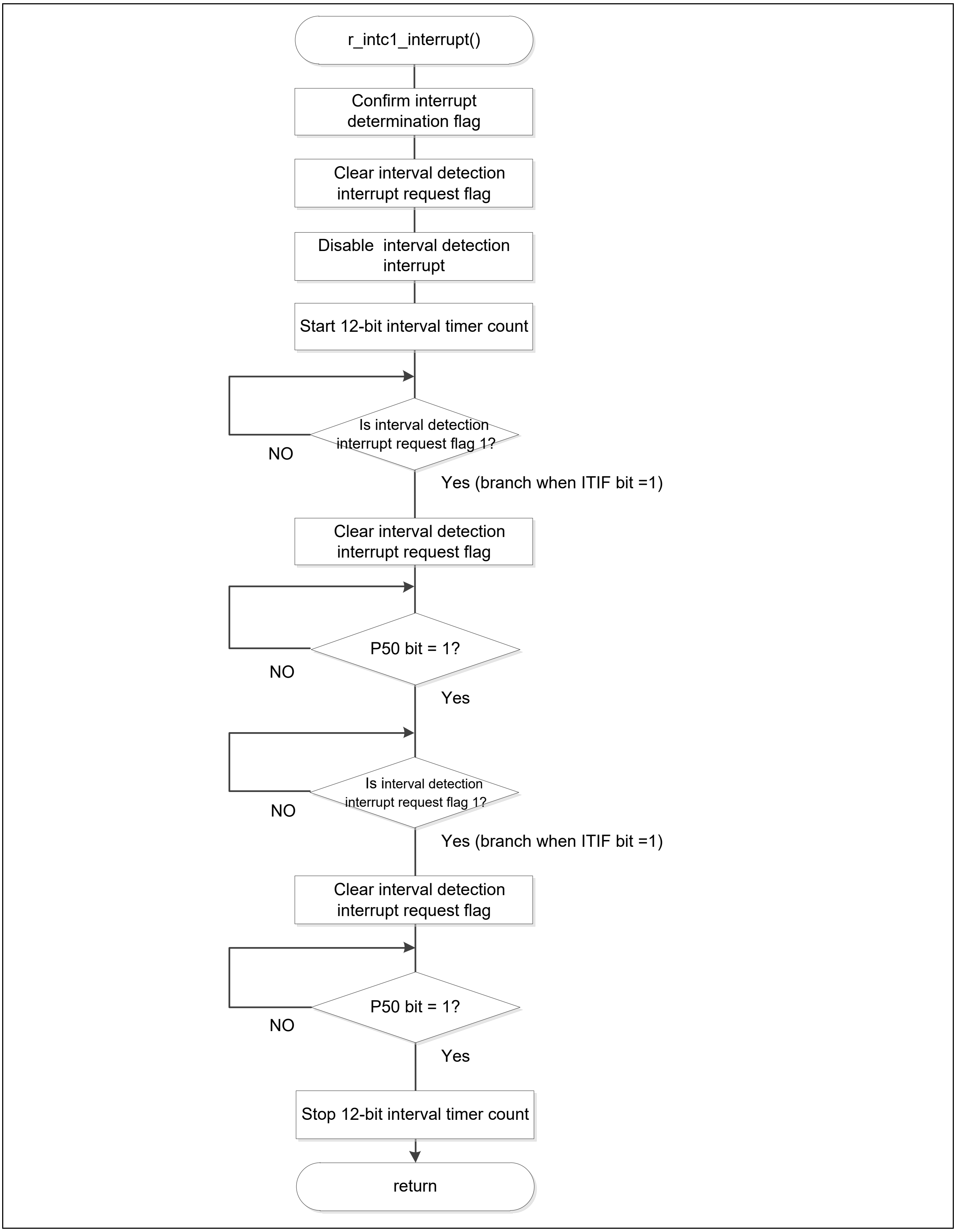

r_intc1_interrupt()
Confirm interrupt determination flag
Clear interval detection interrupt request flag
Disable interval detection interrupt
Start 12-bit interval timer count
Is interval detection interrupt request flag 1?
NO
Yes (branch when ITIF bit =1)
Clear interval detection interrupt request flag
P50 bit = 1?
NO
Yes
Is interval detection interrupt request flag 1?
NO
Yes (branch when ITIF bit =1)
Clear interval detection interrupt request flag
P50 bit = 1?
NO
Yes
Stop 12-bit interval timer count
return

Figure 5.39 External Interrupt Servicing

5.6.32 12-bit Interval Timer Interrupt Servicing

Figure 5.40 shows the flowchart for 12-bit interval timer interrupt servicing.

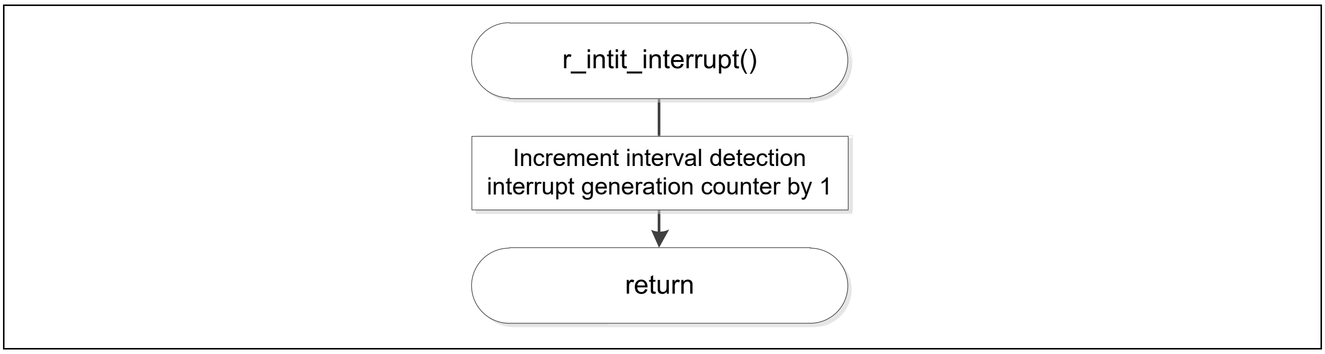

r_intit_interrupt()
Increment interval detection interrupt generation counter by 1
return

Figure 5.40 12-bit Interval Timer Interrupt Servicing

5.6.33 A/D Conversion Completion Interrupt Servicing

Figure 5.41 shows the flowchart for A/D conversion completion interrupt servicing.

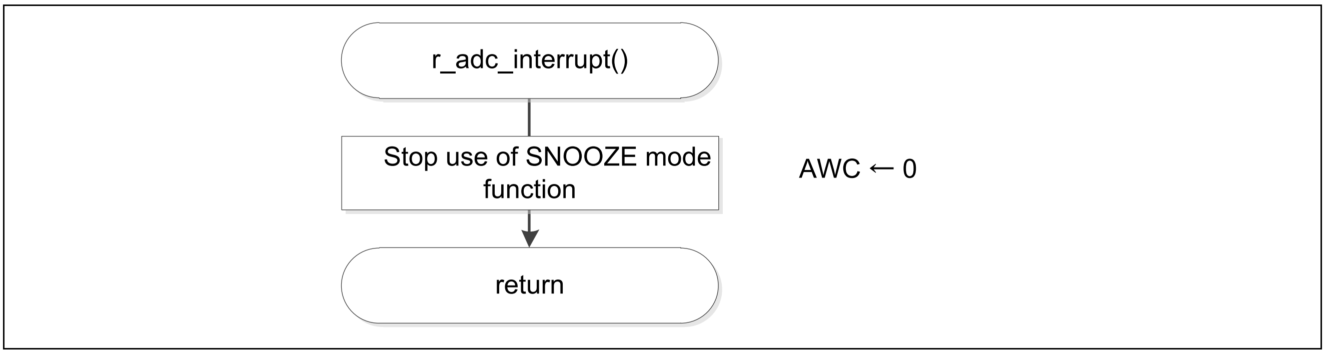


Figure 5.41 A/D Conversion Completion Interrupt Servicing

6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

7. Reference Documents

RL78/G13 User's Manual: Hardware (R01UH0146J)

RL78 Family User's Manual: Software (R01US0015J)

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

Technical Update/Technical News

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

Website and Support

Renesas Electronics website
http://www.renesas.com

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

s

REVISION HISTORY	RL78/G13 CPU Clock Changing and Standby Settings (C Language) CC-RL

Rev.	Date	Description	
		Page	Summary
1.00	Apr. 01, 2016	—	First edition issued
1.10	Aug. 26, 2016	5	Modify the Figure 1.1
		7,9,11,13	Add a description of the external clock
		12,13	Add a Note
		14	Change the version information of the CS+ and e^{2}studio
		18,20,25	Add a description about the status transition error processing
		43,45,49,51, 53,55,59	Modify the flow chart for the status transition error processing added
		46,56,57, 60,61	Add the process to clear the PIF1 bit
1.11	May. 20, 2019	5	Modify the Figure 1.1

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

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

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

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

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

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

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

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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, 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.

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

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

(Rev.4.0-1 November 2017)

Corporate Headquarters

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

Trademarks

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

Contact information

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