RL78/I1D

CPU Clock Changing and Standby Settings (Assembly) CC-RL

R01AN3344EJ0100
Rev.1.00
Aug. 29, 2016

Introduction

This application note describes how to change the RL78/I1D 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 5 LEDs to indicate the CPU clock status and the operation mode.

Target Device

RL78/I1D

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 5 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 (LED1-LED5) connected to pins P50, P51, P52, P60 and P61.
External interrupt	Interrupt (INTP0) 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 P20/ANI13 pin.

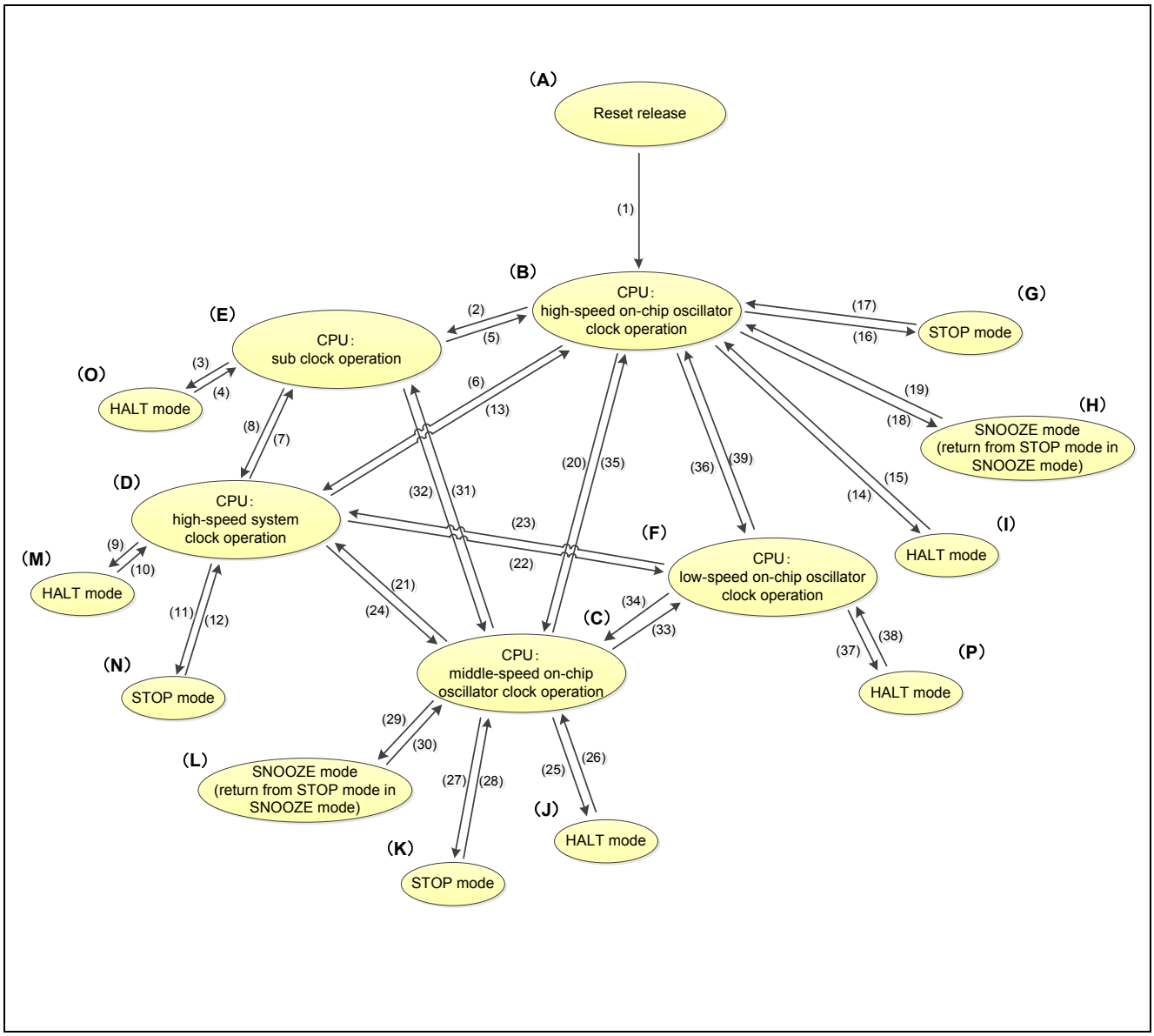

Figure 1.1 Operating Mode Status Transition Diagram

Table 1.2 Operation Modes and Corresponding LED Status

CPU/Peripheral Hardware Clock (f_{CLK})	Operation mode	LED Status				
		LED1	LED2	LED3	LED4	LED5
High-speed on-chip oscillator clock (f_{IH})	Normal operation mode	ON	ON	OFF	OFF	ON
	HALT mode	OFF	ON	OFF	OFF	ON
	SNOOZE mode	ON	OFF	OFF	OFF	ON
	STOP mode	OFF	OFF	OFF	OFF	ON
Middle-speed on-chip oscillator clock (f_{IM})	Normal operation mode	ON	ON	OFF	ON	OFF
	HALT mode	OFF	ON	OFF	ON	OFF
	SNOOZE mode	ON	OFF	OFF	ON	OFF
	STOP mode	OFF	OFF	OFF	ON	OFF
Low-speed on-chip oscillator clock (f_{IL})	Normal operation mode	ON	ON	OFF	ON	ON
	HALT mode	OFF	OFF	OFF	ON	ON
High-speed system clock (f_{MX})	Normal operation mode	ON	ON	ON	OFF	ON
	HALT mode	OFF	ON	ON	OFF	ON
	STOP mode	OFF	OFF	ON	OFF	ON
Sub clock (f_{SX})	Normal operation mode	ON	ON	ON	ON	OFF
	HALT mode	OFF	ON	ON	ON	OFF

Note: The current supplied to the 1pin in less than 8mA. Refer to the electrical specifications of the RL78/I1D User's Manual (Hardware version) for details of the current that can be supplied to the pin.

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 middle-speed on-chip oscillator clock
- Changing from high-speed on-chip oscillator clock to low-speed on-chip oscillator clock
- Changing from high-speed on-chip oscillator clock to high-speed system clock
- Changing from high-speed on-chip oscillator clock to sub clock
- Changing from middle-speed on-chip oscillator clock to high-speed on-chip oscillator clock
- Changing from middle-speed on-chip oscillator clock to low-speed on-chip oscillator clock
- Changing from middle-speed on-chip oscillator clock to high-speed system clock
- Changing from middle-speed on-chip oscillator clock to sub clock
- Changing from low-speed on-chip oscillator clock to high-speed on-chip oscillator clock
- Changing from low-speed on-chip oscillator clock to middle-speed on-chip oscillator clock
- Changing from low-speed on-chip oscillator clock to high-speed system clock
- Changing from high-speed system clock to high-speed on-chip oscillator clock
- Changing from high-speed system clock to middle-speed on-chip oscillator clock
- Changing from high-speed system clock to low-speed on-chip oscillator clock
- Changing from high-speed system clock to sub clock
- Changing from sub clock to high-speed on-chip oscillator clock
- Changing from sub clock to middle-speed on-chip oscillator clock
- Changing from sub clock to high-speed system clock

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

When changing the CPU clock from the high-speed on-chip oscillator clock to the middle-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 middle-speed on-chip oscillator clock to f_{CLK} using the system clock control register (CKC). Confirm that the status of the main on-chip oscillator clock status has switched to the middle-speed on-chip oscillator clock, and then stop the high-speed on-chip oscillator.

① Set (1) the MIOEN bit of the CSC register, and then start oscillating the middle-speed on-chip oscillator.

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

② Use a software wait to wait for the oscillation of the middle-speed on-chip oscillator to stabilize. Count the wait time (oscillation stabilization time: 4μs) using the timer function or another function.

③ Set(1) the MCM1 bit of the CKC register to specify the middle-speed on-chip oscillator clock as the main on-chip oscillator clock.

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	0	0	0	0	0	0	0	**1**

④ Confirm that the MCS1 bit of the CKC register has changed to 1, set (1) the HIOSTOP bit and stop the oscillating the high-speed on-chip oscillator.

	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
	x	x	0	0	0	0	1	**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 low-speed on-chip oscillator clock

When changing the CPU clock from the high-speed on-chip oscillator clock to the low-speed on-chip oscillator clock, start the oscillation using the subsystem clock select register (CKSEL), then wait for the oscillation to stabilize using the timer function or another function. After the oscillation stabilization time has elapsed, set the low-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 status has switched to the subsystem clock, and then stop the high-speed on-chip oscillator.

① Set (1) the SELLOSC bit of the CKSEL register, and then start oscillating the low-speed on-chip oscillator.

	7	6	5	4	3	2	1	0
CKSEL	0	0	0	0	0	0	0	SELLOSC
	0	0	0	0	0	0	0	**1**

② Use a software wait to wait for the oscillation of the low-speed on-chip oscillator to stabilize. Count the wait time (oscillation stabilization time: 210μs) using the timer function or another function. In this application note is always to operate the low-speed on-chip oscillator. For this reason, it does not perform oscillation stabilization wait of the low-speed on-chip oscillator.

③ Set(1) 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	MCS1	MCM1
	0	**1**	0	0	0	0	0	0

④ Confirm that the CLS bit of the CKC register has changed to 1, set (1) the HIOSTOP bit and stop the oscillating the high-speed on-chip oscillator.

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

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

1.1.3 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 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 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.

CMC

7	6	5	4	3	2	1	0
EXCLK	OSCSEL	EXCLKS	OSCSELS	0	AMPHS1	AMPHS0	AMPH
0/1	**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 102µs based on a 10 MHz resonator.

OSTS

7	6	5	4	3	2	1	0
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.

CSC

7	6	5	4	3	2	1	0
MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
0	x	0	0	0	0	x	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 102µs based on a 10 MHz resonator.

OSTC

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

Register setting values:

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

⑤ Set(1) the MCM0 bit of the CKC register to specify the high-speed system clock as the main system clock.

CKC

7	6	5	4	3	2	1	0
CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
0	0	0	**1**	0	0	0	0

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

CSC

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

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

1.1.4 Changing from high-speed on-chip oscillator clock to sub clock

When changing the CPU clock from the high-speed on-chip oscillator clock to the sub 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 sub 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.

OSMC

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

② Set(1) the OSCSELS bit of the CMC register to operate the XT1 oscillator. Set (1) the EXCLKS bit and OSCSELS bit when using the external clock.

CMC

7	6	5	4	3	2	1	0
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.

CSC

7	6	5	4	3	2	1	0
MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
x	**0**	0	0	0	0	x	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(1) the CSS bit of the CKC register to specify the subsystem clock as the CPU/peripheral hardware clock.

CKC

7	6	5	4	3	2	1	0
CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
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.

CSC

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

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

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

When changing the CPU clock from the middle-speed on-chip oscillator 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 on-chip oscillator clock status has switched to the high-speed on-chip oscillator clock, and then stop the middle-speed on-chip 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	MIOEN	HIOSTOP
	x	x	0	0	0	0	1	**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: 65μs) using the timer function or another function.

③ Clear(0) the MCM1 bit of the CKC register to specify the high-speed on-chip oscillator clock as the main on-chip oscillator clock.

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	0	0	0	0	0	0	1	**0**

④ Confirm that the MCS1 bit of the CKC register has changed to 0, clear (0) the MIOEN bit and stop the oscillating the middle-speed on-chip oscillator.

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

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

1.1.6 Changing from middle-speed on-chip oscillator clock to low-speed on-chip oscillator clock

When changing the CPU clock from the middle-speed on-chip oscillator clock to the low-speed on-chip oscillator clock, start the oscillation using the subsystem clock select register (CKSEL), then wait for the oscillation to stabilize using the timer function or another function. After the oscillation stabilization time has elapsed, set the low-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 status has switched to the subsystem clock, and then stop the middle-speed on-chip oscillator.

① Set (1) the SELLOSC bit of the CKSEL register, and then start oscillating the low-speed on-chip oscillator.

	7	6	5	4	3	2	1	0
CKSEL	0	0	0	0	0	0	0	SELLOSC
	0	0	0	0	0	0	0	**1**

② Use a software wait to wait for the oscillation of the low-speed on-chip oscillator to stabilize. Count the wait time (oscillation stabilization time: 210μs) using the timer function or another function. In this application note is always to operate the low-speed on-chip oscillator. For this reason, it does not perform oscillation stabilization wait of the low-speed on-chip oscillator.

③ Set(1) 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	MCS1	MCM1
	0	**1**	0	0	0	0	1	1

④ Confirm that the CLS bit of the CKC register has changed to 1, clear (0) the MIOEN bit and stop the oscillating the middle-speed on-chip oscillator.

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

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

1.1.7 Changing from middle-speed on-chip oscillator clock to high-speed system clock

When changing the CPU clock from the middle-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 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 middle-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.

CMC

7	6	5	4	3	2	1	0
EXCLK	OSCSEL	EXCLKS	OSCSELS	0	AMPHS1	AMPHS0	AMPH
0/1	**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 102µs based on a 10 MHz resonator.

OSTS

7	6	5	4	3	2	1	0
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.

CSC

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

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 102µs based on a 10 MHz resonator.

OSTC

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

Register setting values:

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

⑤ Set(1) the MCM0 bit of the CKC register to specify the high-speed system clock as the main system clock.

CKC

7	6	5	4	3	2	1	0
CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
0	0	0	**1**	0	0	1	1

⑥ Confirm that the MCS bit of the CKC register has changed to 1, clear (0) the MIOEN bit and stop the oscillating the middle-speed on-chip oscillator.

CSC

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

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

1.1.8 Changing from middle-speed on-chip oscillator clock to sub clock

When changing the CPU clock from the middle-speed on-chip oscillator clock to the sub 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 sub 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 middle-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.

OSMC

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

② Set(1) the OSCSELS bit of the CMC register to operate the XT1 oscillator. Set (1) the EXCLKS bit and OSCSELS bit when using the external clock.

CMC

7	6	5	4	3	2	1	0
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.

CSC

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

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(1) the CSS bit of the CKC register to specify the subsystem clock as the CPU/peripheral hardware clock.

CKC

7	6	5	4	3	2	1	0
CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
0	**1**	0	0	0	0	1	1

⑥ Confirm that the CLS bit of the CKC register has changed to 1, then clear (0) MIOEN and stop the middle-speed on-chip oscillator.

CSC

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

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

1.1.9 Changing from low-speed on-chip oscillator clock to high-speed on-chip oscillator clock

When changing the CPU clock from the low-speed on-chip oscillator 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 status has switched to the main system clock, and then stop the low-speed on-chip oscillator.

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

CSC

7	6	5	4	3	2	1	0
MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
x	x	0	0	0	0	x	**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: 65µs) using the timer function or another function.

③ Clear(0) the CSS bit of the CKC register to specify the main system clock as the CPU/peripheral hardware clock.

CKC

7	6	5	4	3	2	1	0
CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
1	**0**	0	0	0	0	0	0

④ Confirm that the CLS bit of the CKC register has changed to 0, clear (0) the SELLOSC bit of the subsystem clock select register(CKSEL) and stop the oscillating the low-speed on-chip oscillator. In this application note, for WUTMMCK0 bit of the subsystem clock supply mode control register(OSMC) is 1, does not stop the low-speed on-chip oscillator.

CKSEL

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SELLOSC
0	0	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

1.1.10 Changing from low-speed on-chip oscillator clock to middle-speed on-chip oscillator clock

When changing the CPU clock from the low-speed on-chip oscillator clock to the middle-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 middle-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 status has switched to the main system clock, and then stop the low-speed on-chip oscillator.

① Set (1) the MIOEN bit of the CSC register, and then start oscillating the middle-speed on-chip oscillator.

CSC

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

② Use a software wait to wait for the oscillation of the middle-speed on-chip oscillator to stabilize. Count the wait time (oscillation stabilization time: 4μs) using the timer function or another function.

③ Clear(0) the CSS bit of the CKC register to specify the main system clock as the CPU/peripheral hardware clock.

CKC

7	6	5	4	3	2	1	0
CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
1	**0**	0	0	0	0	1	1

④ Confirm that the CLS bit of the CKC register has changed to 0, clear (0) the SELLOSC bit of the subsystem clock select register(CKSEL) and stop the oscillating the low-speed on-chip oscillator. In this application note, for WUTMMCK0 bit of the subsystem clock supply mode control register(OSMC) is 1, does not stop the low-speed on-chip oscillator.

CKSEL

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SELLOSC
0	0	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

1.1.11 Changing from low-speed on-chip oscillator clock to high-speed system clock

When changing the CPU clock from the low-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 f_{CLK} using the system clock control register (CKC).

Confirm that the status of the CPU/peripheral hardware clock has changed to the main system clock, and then stop the low-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.

CMC

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

OSTS

7	6	5	4	3	2	1	0
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.

CSC

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

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.

OSTC

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

Register setting values:

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

⑤ Clear(0) the CSS bit of the CKC register to specify the main system clock as the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	1	**0**	1	1	0	0	0	0

⑥ Confirm that the CLS bit of the CKC register has changed to 0, clear (0) the SELLOSC bit of the subsystem clock select register(CKSEL) and stop the oscillating the low-speed on-chip oscillator. In this application note, for WUTMMCK0 bit of the subsystem clock supply mode control register(OSMC) is 1, does not stop the low-speed on-chip oscillator.

	7	6	5	4	3	2	1	0
CKSEL	0	0	0	0	0	0	0	SELLOSC
	0	0	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

1.1.12 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 main 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	MIOEN	HIOSTOP
	0	x	0	0	0	0	x	**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: 65µs) using the timer function or another function.

③ Clear(0) the MCM0 bit of the CKC register to specify the main on-chip oscillator clock as the main system clock.

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	0	0	1	**0**	0	0	0	0

④ Confirm that the MCS bit of the CKC register has changed to 0, 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	MIOEN	HIOSTOP
	1	x	0	0	0	0	x	0

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

1.1.13 Changing from high-speed system clock to middle-speed on-chip oscillator clock

When changing the CPU clock from the high-speed system clock to the middle-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 middle-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 main on-chip oscillator clock, and then stop the X1 oscillator.

① Set (1) the MIOEN bit of the CSC register, and then start oscillating the middle-speed on-chip oscillator.

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

② Use a software wait to wait for the oscillation of the middle-speed on-chip oscillator to stabilize. Count the wait time (oscillation stabilization time: 4μs) using the timer function or another function.

③ Clear(0) the MCM0 bit of the CKC register to specify the main on-chip oscillator clock as the main system clock.

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	0	0	1	**0**	0	0	1	1

④ Confirm that the MCS bit of the CKC register has changed to 0, 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	MIOEN	HIOSTOP
	1	x	0	0	0	0	1	x

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

1.1.14 Changing from high-speed system clock to low-speed on-chip oscillator clock

When changing the CPU clock from the high-speed system clock to the low-speed on-chip oscillator clock, start the oscillation using the subsystem clock select register (CKSEL), then wait for the oscillation to stabilize using the timer function or another function. After the oscillation stabilization time has elapsed, set the low-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 status has switched to the subsystem clock, and then stop the X1 oscillator.

① Set (1) the SELLOSC bit of the CKSEL register, and then start oscillating the low-speed on-chip oscillator.

	7	6	5	4	3	2	1	0
CKSEL	0	0	0	0	0	0	0	SELLOSC
	0	0	0	0	0	0	0	**1**

② Use a software wait to wait for the oscillation of the low-speed on-chip oscillator to stabilize. Count the wait time (oscillation stabilization time: 210μs) using the timer function or another function. In this application note is always to operate the low-speed on-chip oscillator. For this reason, it does not perform oscillation stabilization wait of the low-speed on-chip oscillator.

③ Set(1) 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	MCS1	MCM1
	0	**1**	1	1	0	0	0	0

④ Confirm that the CLS bit of the CKC register has changed to 1, 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	MIOEN	HIOSTOP
	1	x	0	0	0	0	x	x

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

1.1.15 Changing from high-speed system clock to sub clock

When changing the CPU clock from the high-speed system clock to the sub 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 sub 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 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.

OSMC

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

② 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.

CSC

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

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(1) the CSS bit of the CKC register to specify the subsystem clock as the CPU/peripheral hardware clock.

CKC

7	6	5	4	3	2	1	0
CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
0	**1**	1	1	0	0	0	0

⑤ Confirm that the CLS bit of the CKC register has changed to 1, then set (1) MSTOP and stop the X1 oscillator.

CSC

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

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

1.1.16 Changing from sub clock to high-speed on-chip oscillator clock

When changing the CPU clock from the sub 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 status has switched to the main system clock, and then stop the XT1 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	MIOEN	HIOSTOP
	x	0	0	0	0	0	x	**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: 65µs) using the timer function or another function.

③ Clear(0) the CSS bit of the CKC register to specify the main system clock as the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	1	**0**	0	0	0	0	0	0

④ Confirm that the CLS bit of the CKC register has changed to 0, set (1) the XTSTOP bit and stop the oscillating the XT1 oscillator.

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

1.1.17 Changing from sub clock to middle-speed on-chip oscillator clock

When changing the CPU clock from the sub clock to the middle-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 middle-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 status has switched to the main system clock, and then stop the XT1 oscillator.

① Set (1) the MIOEN bit of the CSC register, and then start oscillating the middle-speed on-chip oscillator.

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

② Use a software wait to wait for the oscillation of the middle-speed on-chip oscillator to stabilize. Count the wait time (oscillation stabilization time: 4μs) using the timer function or another function.

③ Clear(0) the CSS bit of the CKC register to specify the main system clock as the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
	1	**0**	0	0	0	0	1	1

④ Confirm that the CLS bit of the CKC register has changed to 0, set (1) the XTSTOP bit and stop the oscillating the XT1 oscillator.

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

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

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

1.1.18 Changing from sub clock to high-speed system clock

When changing the CPU clock from the sub 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 f_{CLK} using the system clock control register (CKC).

Confirm that the status of the CPU/peripheral hardware clock has changed to the main system clock, and then stop the XT1 oscillator.

① 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 102μs based on a 10 MHz resonator.

OSTS

7	6	5	4	3	2	1	0
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 to start oscillating the X1 oscillator. After the external clock is input to the external clock signal, to clear (0) the MSTOP bit.

CSC

7	6	5	4	3	2	1	0
MSTOP	XTSTOP	0	0	0	0	MIOEN	HIOSTOP
0	0	0	0	0	0	x	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 102μs based on a 10 MHz resonator.

OSTC

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

④ Clear(0) the CSS bit of the CKC register to specify the main system clock as the CPU/peripheral hardware clock.

CKC

7	6	5	4	3	2	1	0
CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1
1	**0**	1	1	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

⑤ Confirm that the CLS bit of the CKC register has changed to 0, then set (1) XTSTOP and stop the XT1 oscillator.

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

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/I1D (R5F117GC)
Operating frequencies	● High-speed on-chip oscillator clock: 24MHz ● Middle-speed on-chip oscillator clock: 4MHz ● Low-speed on-chip oscillator clock: 15kHz ● High-speed system clock: 20MHz ● Sub clock: 32.768kHz ● CPU/peripheral hardware clock: 24MHz/20MHz/4MHz/32.768kHz/15kHz[Note]
Operating voltage	3.3V (operating range 2.9V to 3.6V) 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.
Assembler (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.
Assembler (e^2 studio)	CC-RL V1.02.00 from Renesas Electronics Corp.
Board used	Renesas Electronics Corp. RL78/I1D Target board (RTE5117GC0TGB0000R)

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/I1D Operation State Switching CC-RL (R01AN3095E) 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.

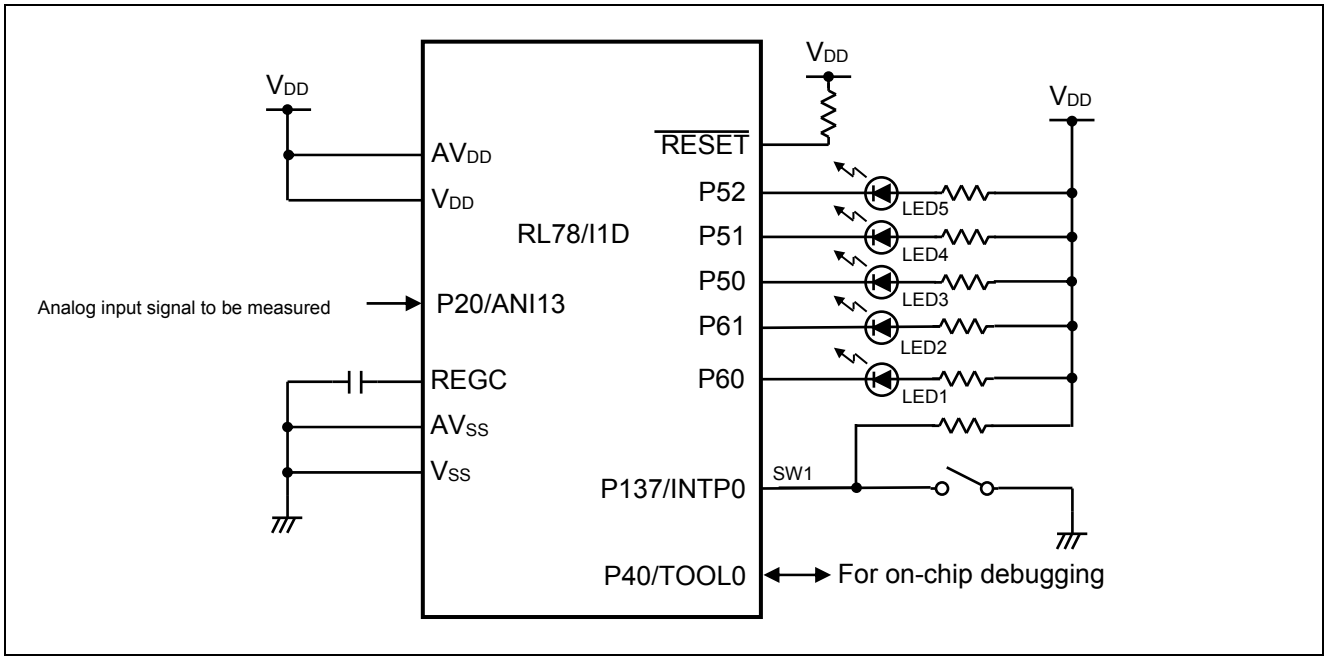

Figure 4.1 Hardware Configuration

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. provides List of Pins and Functions

Table 4.1 List of Pins and Functions

Pin Name	Input/Output	Function
P137/INTP0	Input	Switch (SW1) input port
P20/AIN13	Input	A/D converter analog input port
P60	Output	LED (LED1) control port
P61	Output	LED (LED2) control port
P50	Output	LED (LED3) control port
P51	Output	LED (LED4) control port
P52	Output	LED (LED5) 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. The CPU clock and the operating mode is changed in the order of 1 to 39 of Figure 1.1 operation mode status transition diagram.

(1) Input/output port initialization

- P50-P52 and P60-P61 pins: set as output ports (use to control LEDs)
- P20/ANI13 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 24 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 INTP0 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 5ms. 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 P137/INTP0 pin is detected. The following is the CPU clock and operation mode after the switch is pressed.

Table 5.1 LED status (after the switch is pressed) (1/2)

	CPU clock	Operation mode	LED1	LED2	LED3	LED4	LED5
(1)	High-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	OFF	ON
(2)	Sub clock	Normal operation mode	ON	ON	ON	ON	OFF
(3)	Sub clock	HALT mode	OFF	ON	ON	ON	OFF
(4)	Sub clock	Normal operation mode	ON	ON	ON	ON	OFF
(5)	High-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	OFF	ON
(6)	High-speed system clock	Normal operation mode	ON	ON	ON	OFF	ON
(7)	Sub clock	Normal operation mode	ON	ON	ON	ON	OFF
(8)	High-speed system clock	Normal operation mode	ON	ON	ON	OFF	ON
(9)	High-speed system clock		OFF	ON	ON	OFF	ON
(10)	High-speed system clock	Normal operation mode	ON	ON	ON	OFF	ON
(11)	High-speed system clock		OFF	OFF	ON	OFF	ON
(12)	High-speed system clock	Normal operation mode	ON	ON	ON	OFF	ON
(13)	High-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	OFF	ON
(14)	High-speed on-chip oscillator clock	HALT mode	OFF	ON	OFF	OFF	ON
(15)	High-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	OFF	ON
(16)	High-speed on-chip oscillator clock	STOP mode	OFF	OFF	OFF	OFF	ON
(17)	High-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	OFF	ON
(18)	High-speed on-chip oscillator clock	SNOOZE mode	ON	OFF	OFF	OFF	ON
(19)	High-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	OFF	ON
(20)	Middle-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	ON	OFF
(21)	High-speed system clock	Normal operation mode	ON	ON	ON	OFF	ON
(22)	Low-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	ON	ON
(23)	High-speed system clock	Normal operation mode	ON	ON	ON	OFF	ON
(24)	Middle-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	ON	OFF

Table 5.2 LED Status (after the switch is pressed) (2/2)

	CPU clock	Operation mode	LED1	LED2	LED3	LED4	LED5
(25)	Middle-speed on-chip oscillator clock	HALT mode	OFF	ON	OFF	ON	OFF
(26)	Middle-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	ON	OFF
(27)	Middle-speed on-chip oscillator clock	STOP mode	OFF	OFF	OFF	ON	OFF
(28)	Middle-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	ON	OFF
(29)	Middle-speed on-chip oscillator clock	SNOOZE mode	ON	OFF	OFF	ON	OFF
(30)	Middle-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	ON	OFF
(31)	Sub clock	Normal operation mode	ON	ON	ON	ON	OFF
(32)	Middle-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	ON	OFF
(33)	Low-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	ON	ON
(34)	Middle-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	ON	OFF
(35)	High-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	OFF	ON
(36)	Low-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	ON	ON
(37)	Low-speed on-chip oscillator clock	HALT mode	OFF	ON	OFF	ON	ON
(38)	Low-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	ON	ON
(39)	High-speed on-chip oscillator clock	Normal operation mode	ON	ON	OFF	OFF	ON

After the CPU clock and operation mode have been changed according to steps 1 to 39 above, the falling edge of a signal (switch) input to the P137/INTP0 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/I1D User's Manual for usage notes concerning this device.

5.2 Option Byte Settings

Table 5.3 lists the option byte settings.

Table 5.3 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: reset mode Detection voltage: Rising edge 2.81 V/falling edge 2.75 V
000C2H/010C2H	11100000B	HS mode, HOCO: 24 MHz
000C3H/010C3H	10000100B	On-chip debugging enabled

5.3 Variables

Table 5.4 lists the global variables.

Table 5.4 Variables

Type	Variable Name	Contents	Function Used
8-bit	RITCOUNT	Number of interval signal detection interrupts for12-bit interval timer	BtoE, EtoB, BtoD, DtoE, EtoD, DtoB, BtoC, CtoD, DtoF, FtoD, DtoC, CtoE, EtoC, CtoB, BtoF, FtoB, CtoF, FtoC, IINTIT
8-bit	RINTFLAG	Confirm the external interrupt generation detection flag	_NOP, IINTP0

5.4 Functions (subroutines)

Table 5.5 and Table 5.6 lists the functions (subroutines).

Table 5.5 Functions(1/2)

Function Name	Outline	Number of operating mode status transition diagram
AtoB	Status transition processing from (A) to (B)	(1)
BtoE	Status transition processing from (B) to (E)	(2)
EtoO	Status transition processing from (E) to (O)	(3)
OtoE	Status transition processing from (O) to (E)	(4)
EtoB	Status transition processing from (E) to (B)	(5)
BtoD	Status transition processing from (B) to (D)	(6)
DtoE	Status transition processing from (D) to (E)	(7)
EtoD	Status transition processing from (E) to (D)	(8)
DtoM	Status transition processing from (D) to (M)	(9)
MtoD	Status transition processing from (M) to (D)	(10)
DtoN	Status transition processing from (D) to (N)	(11)
NtoD	Status transition processing from (N) to (D)	(12)
DtoB	Status transition processing from (D) to (B)	(13)
BtoI	Status transition processing from (B) to (I)	(14)
ItoB	Status transition processing from (I) to (B)	(15)
BtoG	Status transition processing from (B) to (G)	(16)
GtoB	Status transition processing from (G) to (B)	(17)
BtoH	Status transition processing from (B) to (H)	(18)
HtoB	Status transition processing from (H) to (B)	(19)
BtoC	Status transition processing from (B) to (C)	(20)
CtoD	Status transition processing from (C) to (D)	(21)
DtoF	Status transition processing from (D) to (F)	(22)
FtoD	Status transition processing from (F) to (D)	(23)
DtoC	Status transition processing from (D) to (C)	(24)
CtoJ	Status transition processing from (C) to (J)	(25)
JtoC	Status transition processing from (J) to (C)	(26)
CtoK	Status transition processing from (C) to (K)	(27)
KtoC	Status transition processing from (K) to (C)	(28)
CtoL	Status transition processing from (C) to (L)	(29)
LtoC	Status transition processing from (L) to (C)	(30)
CtoE	Status transition processing from (C) to (E)	(31)
EtoC	Status transition processing from (E) to (C)	(32)
CtoF	Status transition processing from (C) to (F)	(33)
FtoC	Status transition processing from (F) to (C)	(34)
CtoB	Status transition processing from (C) to (B)	(35)
BtoF	Status transition processing from (B) to (F)	(36)
FtoP	Status transition processing from (F) to (P)	(37)
PtoF	Status transition processing from (P) to (F)	(38)
FtoB	Status transition processing from (F) to (B)	(39)

Table 5.6 Functions(2/2)

Function Name	Outline
_NOP	Continuous NOP instruction execution processing
AD_SNOOZE	A/D converter setting
END	End processing of status transition
ERROR	Error processing of status transition
IINTP0	Confirm the external interrupt generation detection flag update processing
IINTIT	12-bit interval timer interval signal detection interrupt count processing
IINTAD	SNOOZE mode release processing

5.5 Function (subroutine) Specifications

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

[Function Name] AtoB

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

[Function Name] BtoE

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

[Function Name] EtoO

Outline	Status transition processing from (E) to (O)
Declaration	—
Description	Control LED lighting, transition to HALT mode. (CPU clock stopped (when using sub clock))
Argument	None
Return Value	None
Notes	None

[Function Name] OtoE

Outline	Status transition processing from (O) to (E)
Declaration	—
Description	Control LED lighting (CPU clock: sub clock)
Argument	None
Return Value	None
Notes	None

[Function Name] EtoB

Outline	Status transition processing from (E) to (B)
Declaration	—
Description	Change the CPU clock from sub 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] BtoD

Outline	Status transition processing from (B) to (D)
Declaration	—
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] DtoE

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

[Function Name] EtoD

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

[Function Name] DtoM

Outline	Status transition processing from (D) to (M)
Declaration	—
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] MtoD

Outline	Status transition processing from (M) to (D)
Declaration	—
Description	Control LED lighting. (CPU clock: high-speed system clock)
Argument	None
Return Value	None
Notes	None

[Function Name] DtoN

Outline	Status transition processing from (D) to (N)
Declaration	—
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] NtoD

Outline	Status transition processing from (N) to (D)
Declaration	—
Description	Control LED lighting. (CPU clock: high-speed system clock)
Argument	None
Return Value	None
Notes	None

[Function Name] DtoB

Outline	Status transition processing from (D) to (B)
Declaration	—
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] BtoI

Outline	Status transition processing from (B) to (I)
Declaration	—
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] ItoB

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

[Function Name] BtoG

Outline	Status transition processing from (B) to (G)
Declaration	—
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] GtoB

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

[Function Name] BtoH

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

[Function Name] HtoB

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

[Function Name] BtoC

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

[Function Name] CtoD

Outline	Status transition processing from (C) to (D)
Declaration	—
Description	Change the CPU clock from middle-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] DtoF

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

[Function Name] FtoD

Outline	Status transition processing from (F) to (D)
Declaration	—
Description	Change the CPU clock from low-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] DtoC

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

[Function Name] CtoJ

Outline	Status transition processing from (C) to (J)
Declaration	—
Description	Control LED lighting, then transition to HALT mode. (CPU clock stopped (when using middle-speed on-chip oscillator clock))
Argument	None
Return Value	None
Notes	None

[Function Name] JtoC

Outline	Status transition processing from (J) to (C)
Declaration	—
Description	Control LED lighting. (CPU clock: middle-speed on-chip oscillator clock)
Argument	None
Return Value	None
Notes	None

[Function Name] CtoK

Outline	Status transition processing from (C) to (K)
Declaration	—
Description	Control LED lighting, then transition to STOP mode. (CPU clock stopped (when using middle-speed on-chip oscillator clock))
Argument	None
Return Value	None
Notes	None

[Function Name] KtoC

Outline	Status transition processing from (K) to (C)
Declaration	—
Description	Control LED lighting. (CPU clock: middle-speed on-chip oscillator clock)
Argument	None
Return Value	None
Notes	None

[Function Name] CtoL

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

[Function Name] LtoC

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

[Function Name] CtoE

Outline	Status transition processing from (C) to (E)
Declaration	—
Description	Change the CPU clock from middle-speed on-chip oscillator clock to sub clock. After the clock is switched, control LED lighting.
Argument	None
Return Value	None
Notes	None

[Function Name] EtoC

Outline	Status transition processing from (E) to (C)
Declaration	—
Description	Change the CPU clock from sub clock to middle-speed on-chip oscillator clock. After the clock is switched, control LED lighting.
Argument	None
Return Value	None
Notes	None

[Function Name] CtoF

Outline	Status transition processing from (C) to (F)
Declaration	—
Description	Change the CPU clock from middle-speed on-chip oscillator clock to low-speed on-chip oscillator clock. After the clock is switched, control LED lighting.
Argument	None
Return Value	None
Notes	None

[Function Name] FtoC

Outline	Status transition processing from (F) to (C)
Declaration	—
Description	Change the CPU clock from low-speed on-chip oscillator clock to middle-speed on-chip oscillator clock. After the clock is switched, control LED lighting.
Argument	None
Return Value	None
Notes	None

[Function Name] CtoB

Outline	Status transition processing from (C) to (B)
Declaration	—
Description	Change the CPU clock from middle-speed on-chip oscillator 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] BtoF

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

[Function Name] FtoP

Outline	Status transition processing from (F) to (P)
Declaration	—
Description	Control LED lighting, then transition to HALT mode. (CPU clock stopped (when using low-speed on-chip oscillator clock))
Argument	None
Return Value	None
Notes	None

[Function Name] PtoF

Outline	Status transition processing from (P) to (F)
Declaration	—
Description	Control LED lighting. (CPU clock: low-speed on-chip oscillator clock)
Argument	None
Return Value	None
Notes	None

[Function Name] FtoB

Outline	Status transition processing from (F) to (B)
Declaration	—
Description	Change the CPU clock from low-speed on-chip oscillator 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] _NOP

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

[Function Name] AD_SNOOZE

Outline	A/D converter setting
Declaration	—
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] END

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

[Function Name] ERROR

Outline	Error processing of status transition
Declaration	—
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] IINTP0

Outline	External interrupt generation detection flag confirmation processing
Declaration	—
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] IINTIT

Outline	12-bit interval timer interval signal detection interrupt count processing
Declaration	—
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] IINTAD

Outline	SNOOZE mode release processing
Declaration	—
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 5.2 shows the entire flow for this application note.

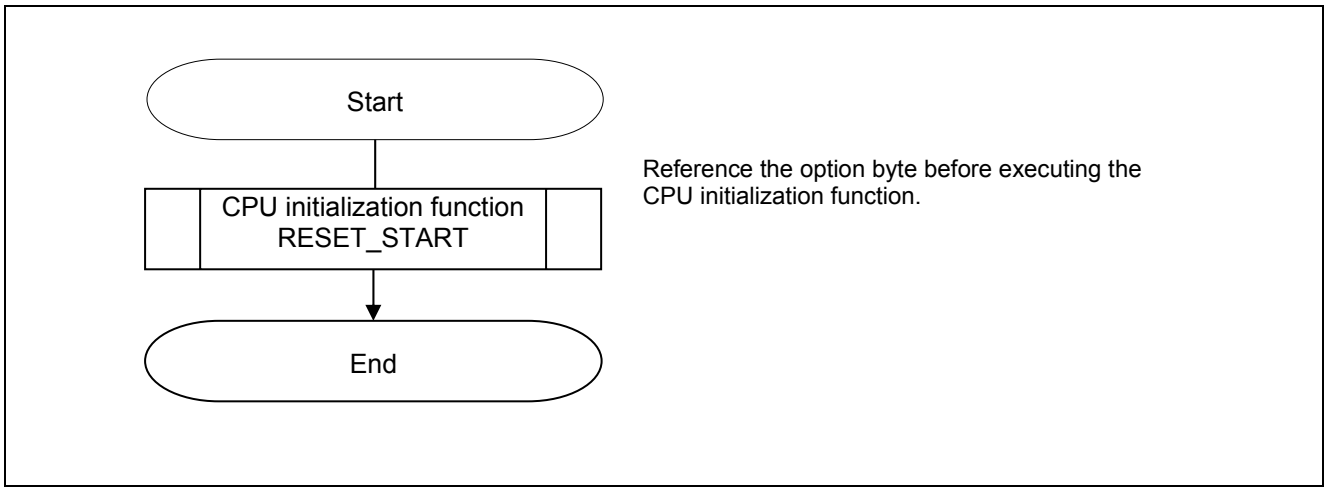

Figure 5.2 Overall Flowchart

5.6.1 CPU Initialization Function

Figure 5.3 shows the flowchart for the CPU initialization function.

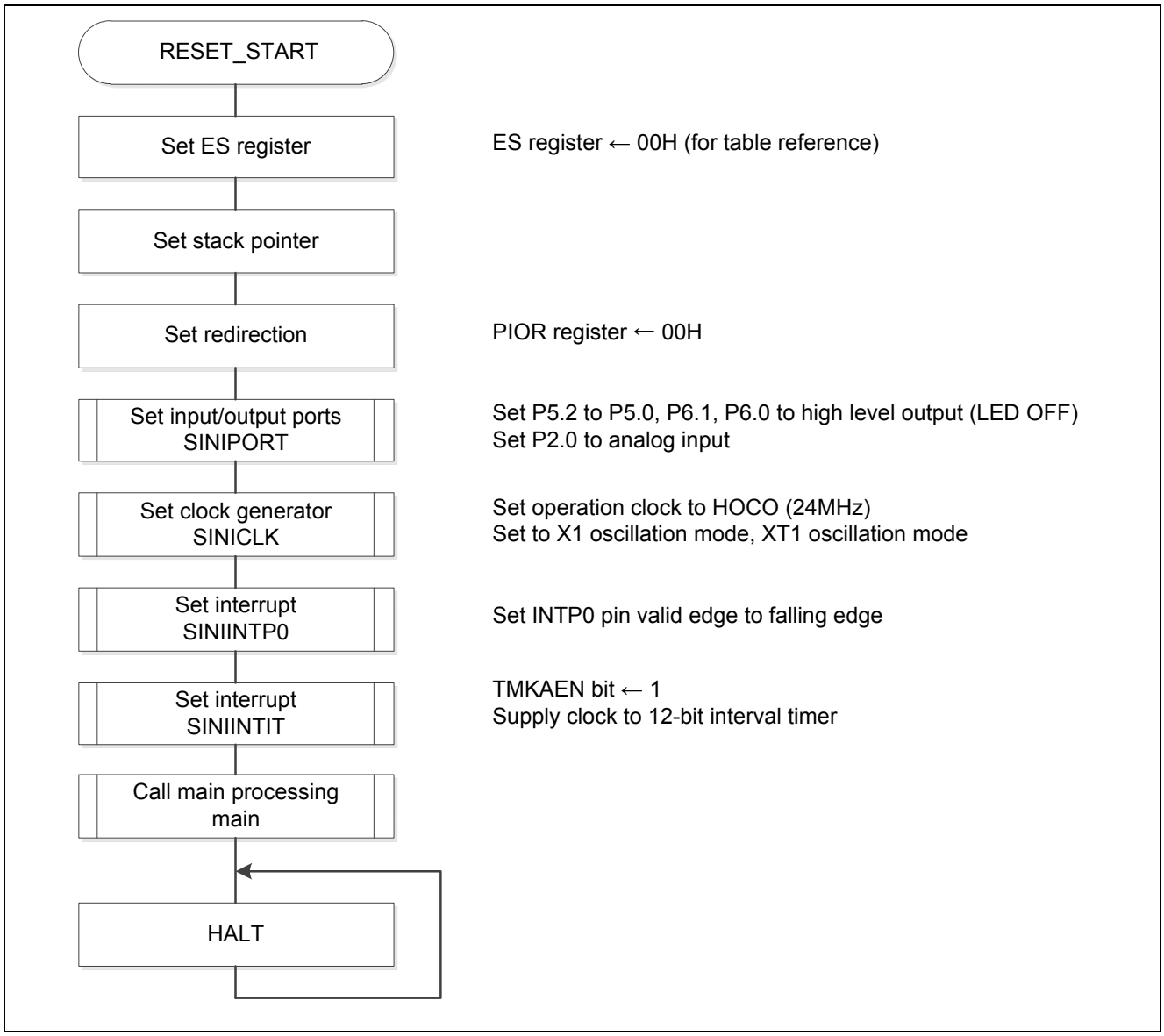

Figure 5.3 CPU Initialization Function

5.6.2 Input/Output Port Settings

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

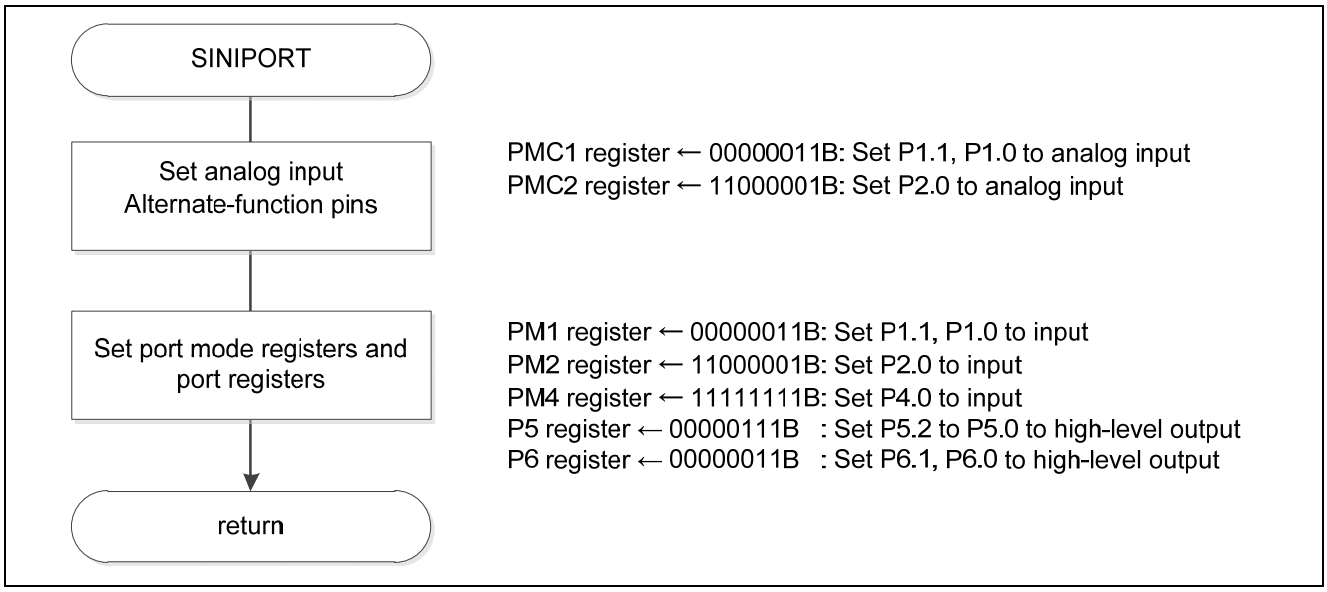

Figure 5.4 Input/Output Port Settings

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.

5.6.3 Clock Generator Setting

Figure 5.5 shows the flowchart for setting the clock generator.

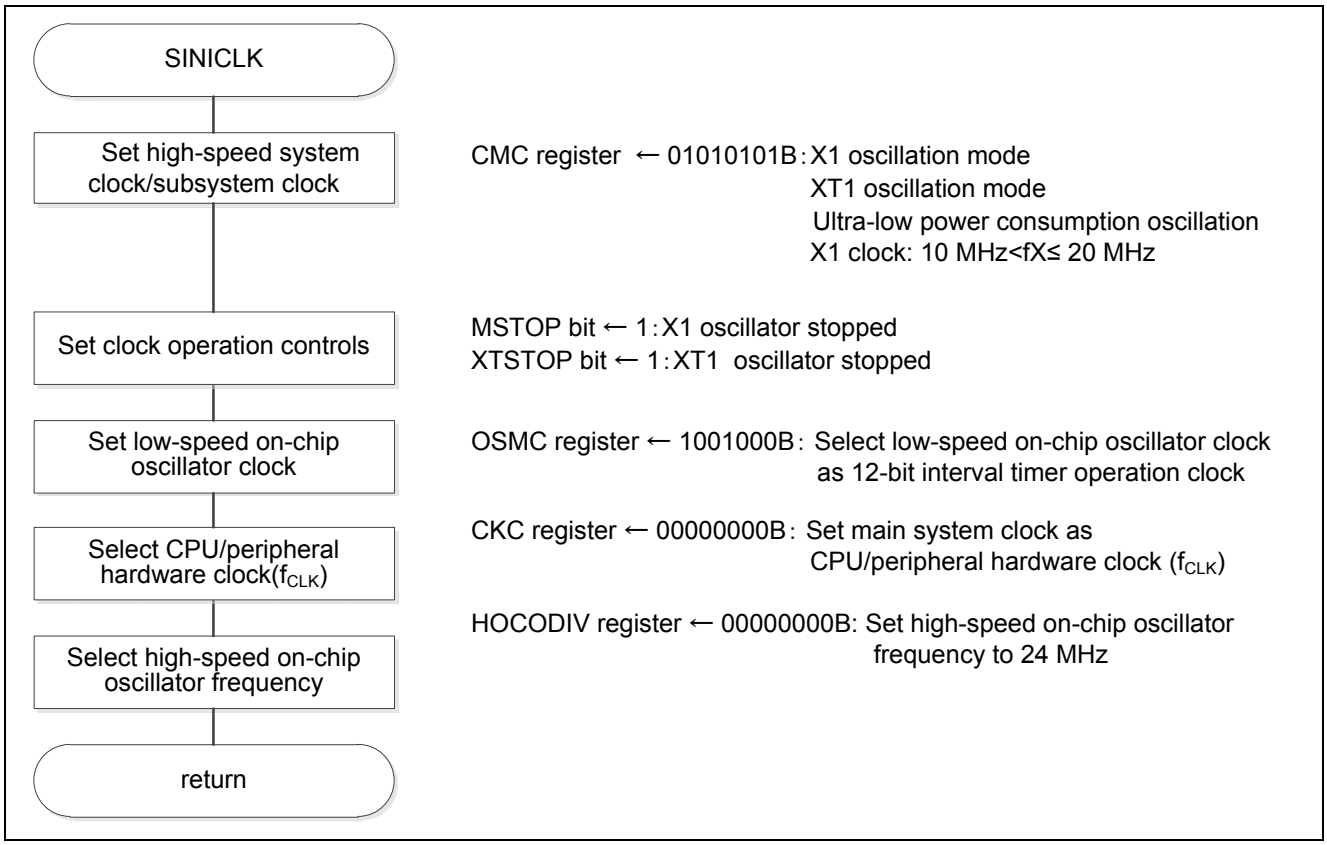

Figure 5.5 Clock Generator Setting

Clock operation mode setting

- Clock operation mode control register (CMC)

High-speed system clock pin operation mode: X1 oscillation mode
Subsystem clock pin operation mode: XT1 oscillation mode
XT1 oscillator oscillation mode: ultra-low power consumption oscillation
X1 clock oscillation frequency control: 1MHz ≤ f_X ≤ 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 Pin	X2/EXCLK/P122 Pin
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 Pin	XT2/EXCLKS/P124 Pin
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/I1D 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

Middle-speed on-chip oscillator clock operation control: Middle-speed on-chip oscillator stopped

High-speed on-chip oscillator clock operation control: High-speed on-chip oscillator operating

Symbol: CSC

7	6	5	4	3	2	1	0
MSTOP	XTSTOP	0	0	0	0	MIOEN	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 1

MIOEN	Middle-speed on-chip oscillator clock operation control
0	**Middle-speed on-chip oscillator stopped**
1	Middle-speed on-chip oscillator operating

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/I1D 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	MCS1	MCM1
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	**Main on-chip oscillator clock f_{oco})**
1	High-speed system clock (f_{MX})

Bit 4

MCM0	Main system clock (f_{MAIN}) operation control
0	**Selects main on-chip oscillator clock (f_{oco}) as main system clock (f_{MAIN})**
1	Selects high-speed system clock (f_{MX}) as main system clock (f_{MAIN}).

Bit 1

MCS1	Status of main on-chip oscillator clock (f_{oco})
0	**High-speed on-chip oscillator clock (f_{IH})**
1	Middle-speed on-chip oscillator clock (f_{IM})

Bit 0

MCS1	Main on-chip oscillator clock (f_{oco}) operation control
0	**High-speed on-chip oscillator clock (f_{IH})**
1	Middle-speed on-chip oscillator clock (f_{IM})

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

: Stops supply of subsystem clock to peripheral functions other than the real-time clock 2, 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output controller.

Selection of count clock for real-time clock and 12-bit interval timer: low-speed on-chip oscillator clock

Symbol: OSMC

7	6	5	4	3	2	1	0
RTCLPC	0	0	WUTMMCK0	0	0	0	0
1	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 the real-time clock 2, 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output controller.**

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 clock**

Note: Refer to the RL78/I1D User's Manual (Hardware version) for details on how to set registers.

5.6.4 External Interrupt Setting

Figure 5.6 shows the flowchart for setting the external interrupt.

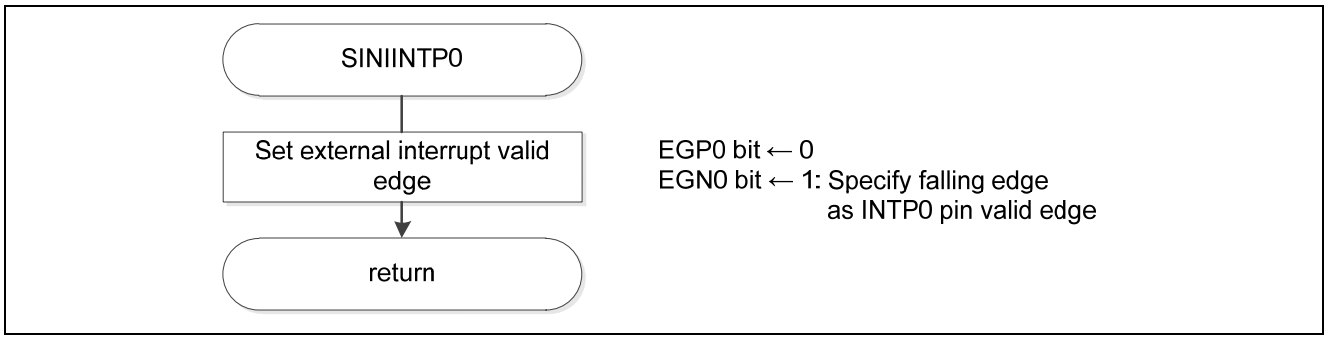

Figure 5.6 External Interrupt Setting

Control of external interrupt valid edge

・External interrupt rising edge enable register (EGP0)
Select valid edge for INTP0 pin: falling edge

Symbol: EGP0

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

Symbol: EGN0

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

Bit 1

EGP0	EGN0	INTP0pin 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/I1D User's Manual (Hardware version) for details on how to set registers.

5.6.5 12-bit Interval Timer Setting

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

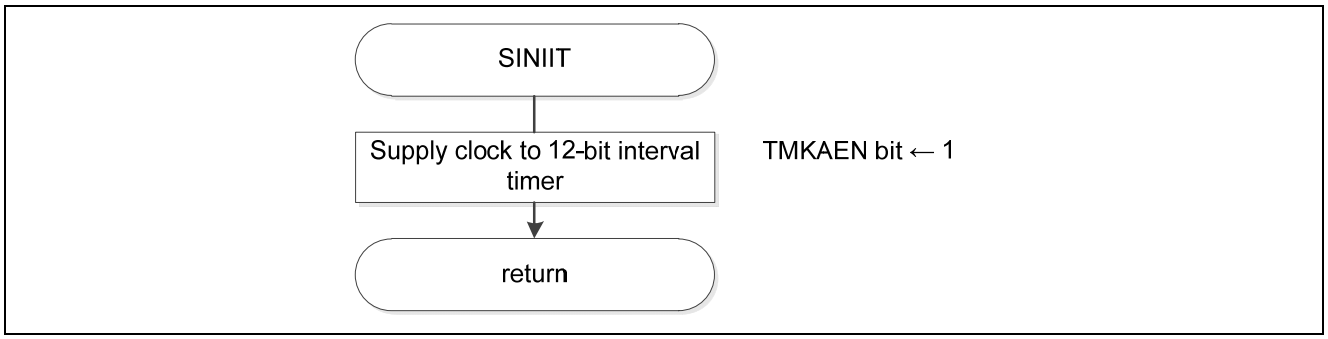

Figure 5.7 12-bit Interval Timer Setting

12-bit interval timer clock supply setting

- Peripheral enable register 2 (PER2)
  Enable clock supply to 12-bit interval timer.

Symbol: PER2

7	6	5	4	3	2	1	0
TMKAEN	FMCEN	DOCEN	0	0	0	0	0
1	x	x	0	0	0	0	0

Bit 7

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

Note: Refer to the RL78/I1D User's Manual (Hardware version) for details on how to set registers.

5.6.6 Main Processing

Figure 5.8, Figure 5.9, Figure 5.10, Figure 5.11 shows the flowchart for the main processing.

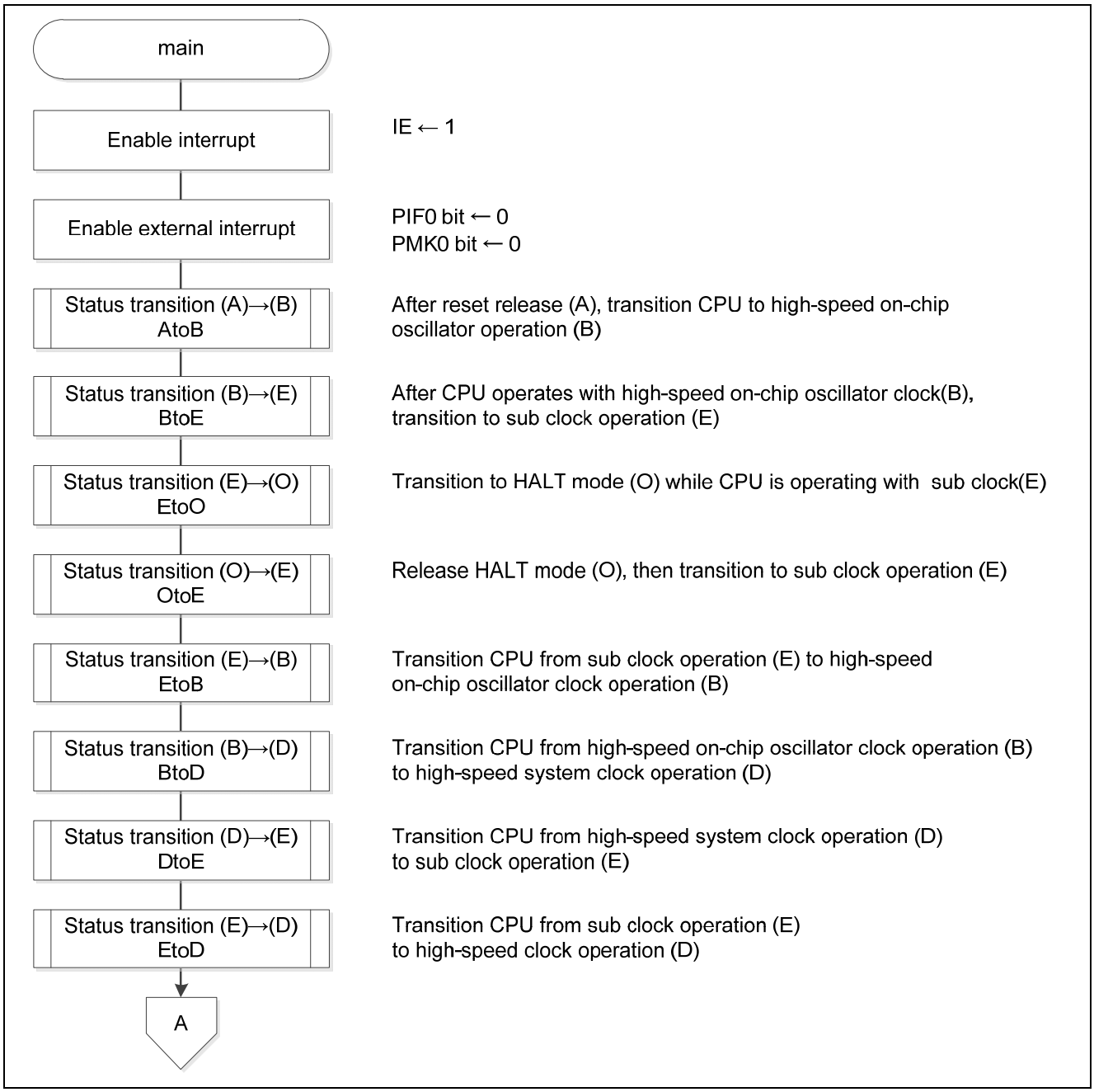

Figure 5.8 Main Processing (1/4)

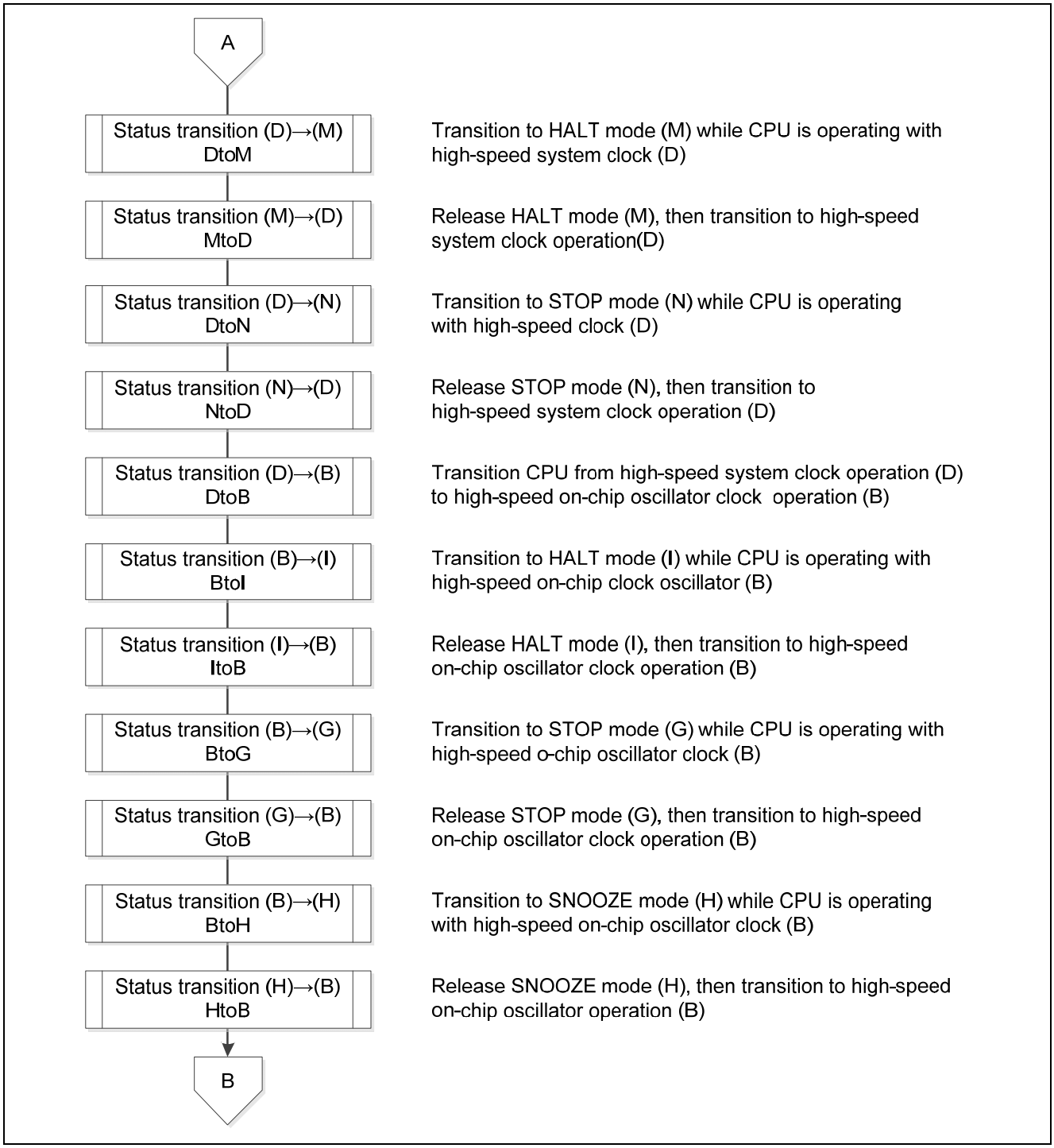

Figure 5.9 Main Processing (2/4)

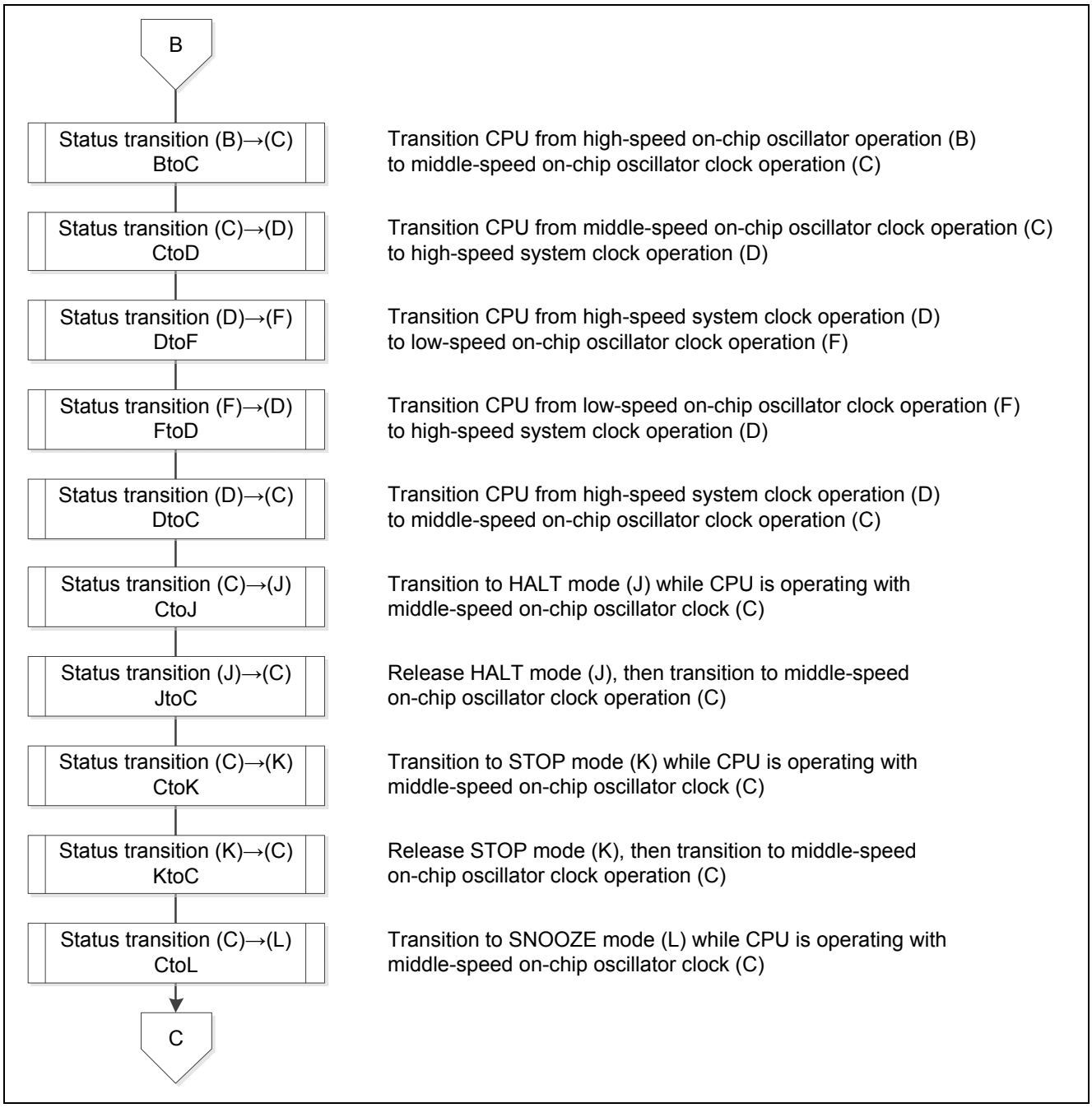

Figure 5.10 Main Processing (3/4)

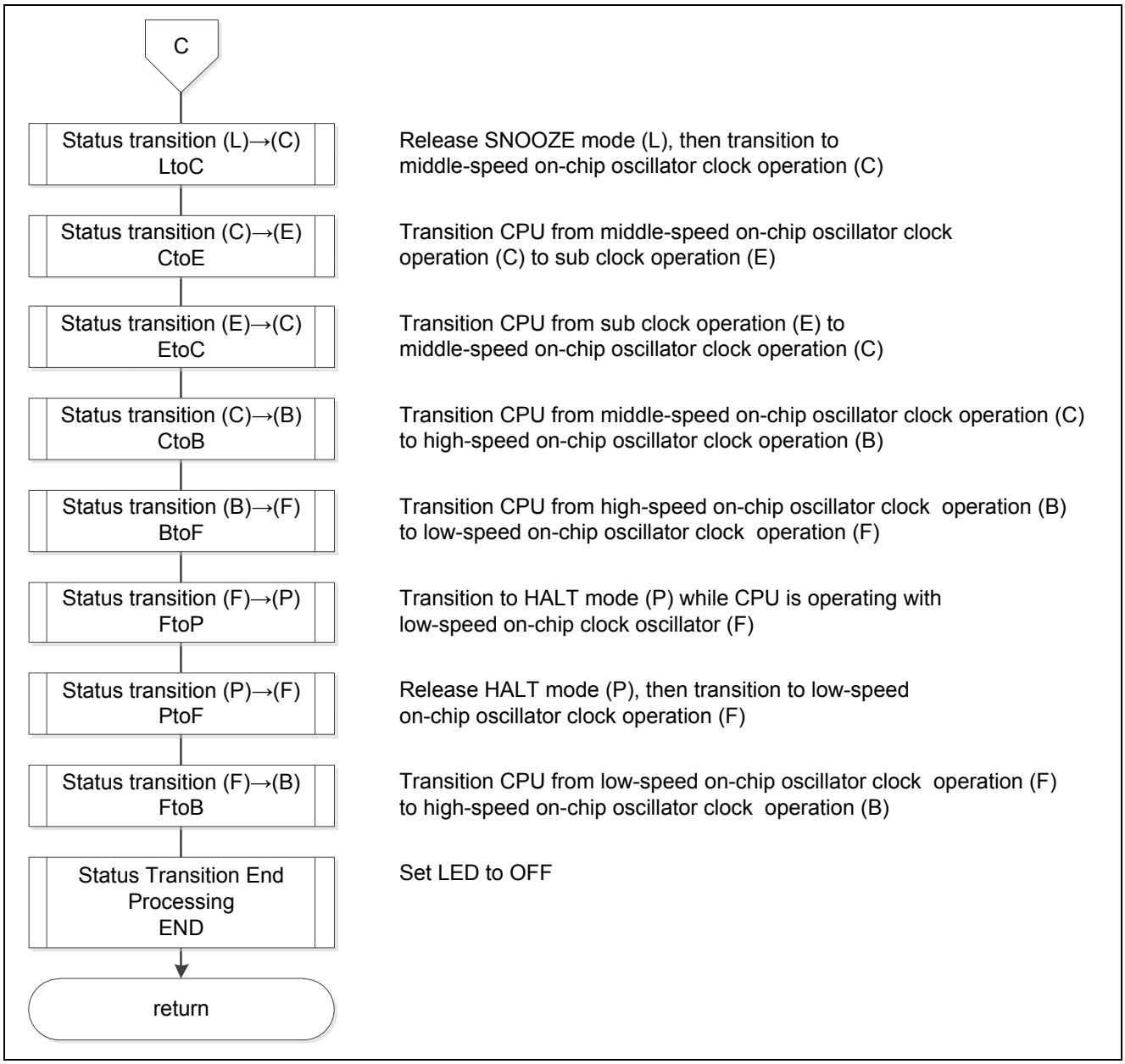

Figure 5.11 Main Processing (4/4)

Pin input edge detection interrupt (INTP0) setting

- Interrupt request flag register (IF0L)
  Clear the PIF0 interrupt source flag.
- Interrupt mask flag register (MK0L)
  Set PMK0 interrupt mask.

Symbol: IF0L

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

Bit 2

PIF0	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	x	**0**	x	x

Bit 2

PMK0	Interrupt servicing control
0	**Interrupt servicing enabled**
1	Interrupt servicing disabled

Note: Refer to the RL78/I1D User's Manual (Hardware version) for details on how to set registers.

5.6.7 Status Transition AtoB

Figure 5.12 shows the flowchart for status transition AtoB.

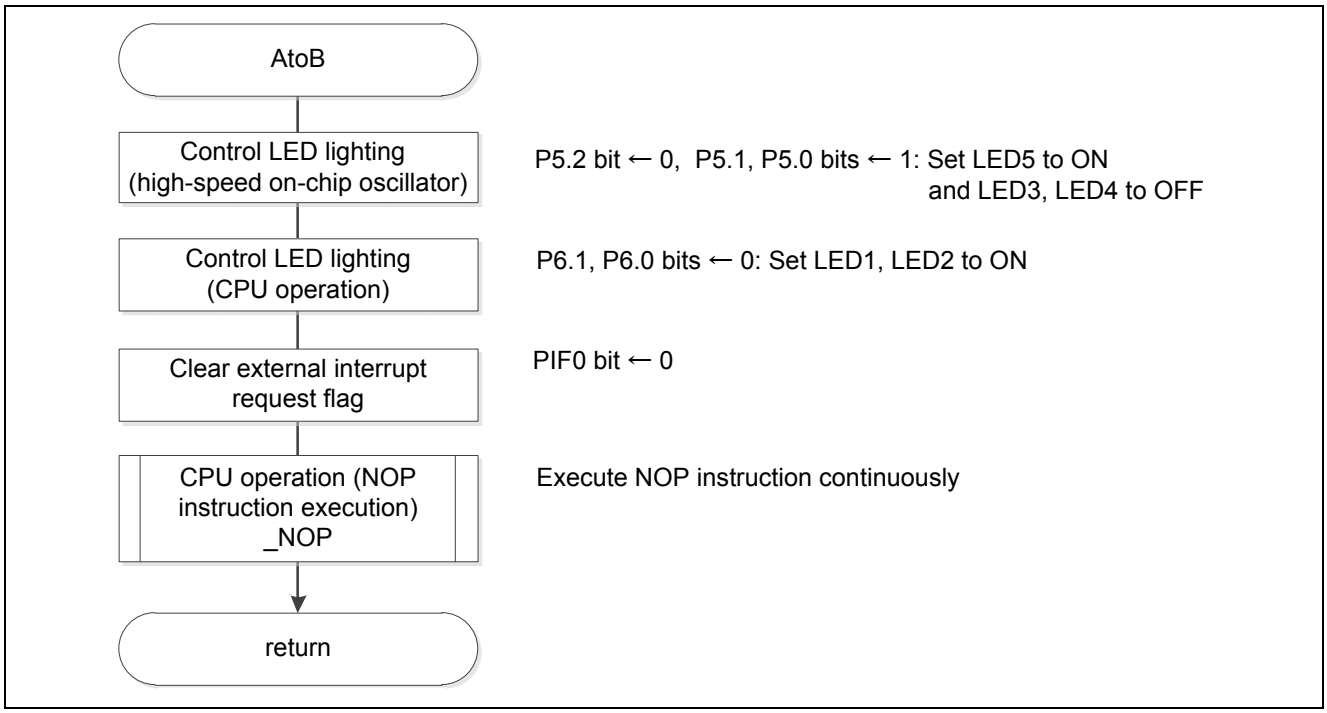

Figure 5.12 Status Transition AtoB

5.6.8 CPU operation (NOP instruction execution)

Figure 5.13 shows the flowchart for the CPU operation (NOP instruction execution)

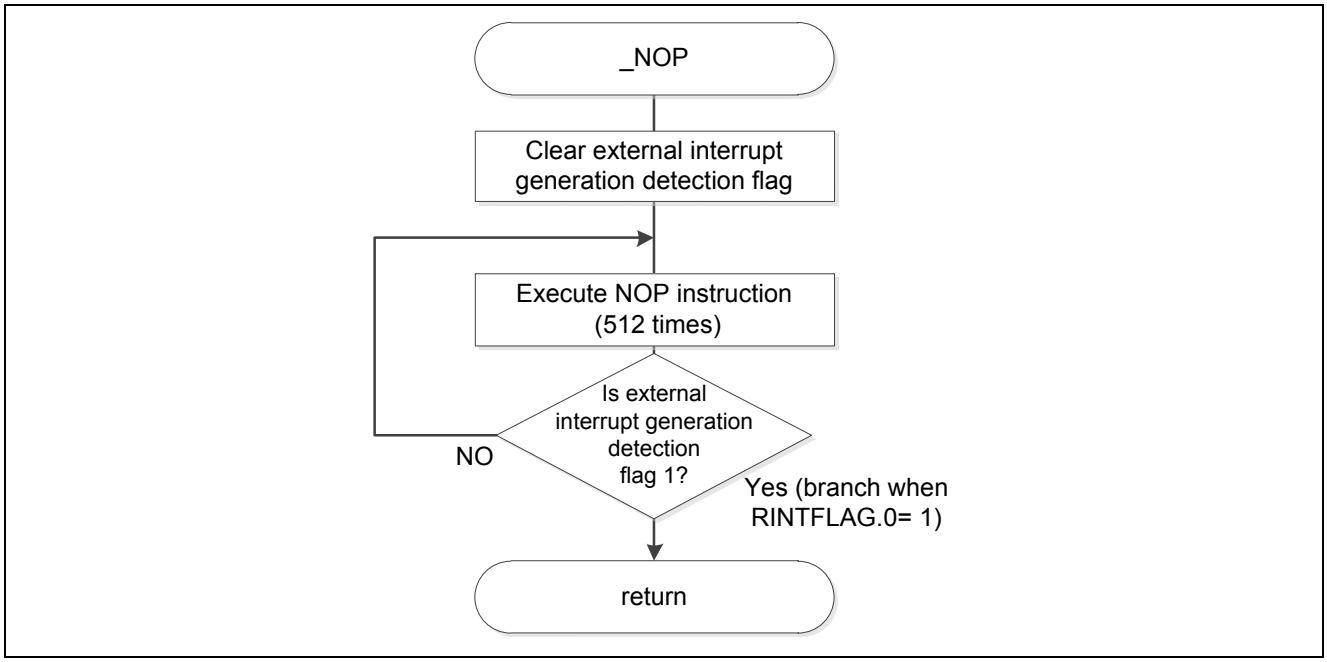

Figure 5.13 CPU Operation (NOP instruction execution)

5.6.9 Status Transition BtoE

Figure 5.14 and Figure 5.15 shows the flowchart for status transition BtoE.

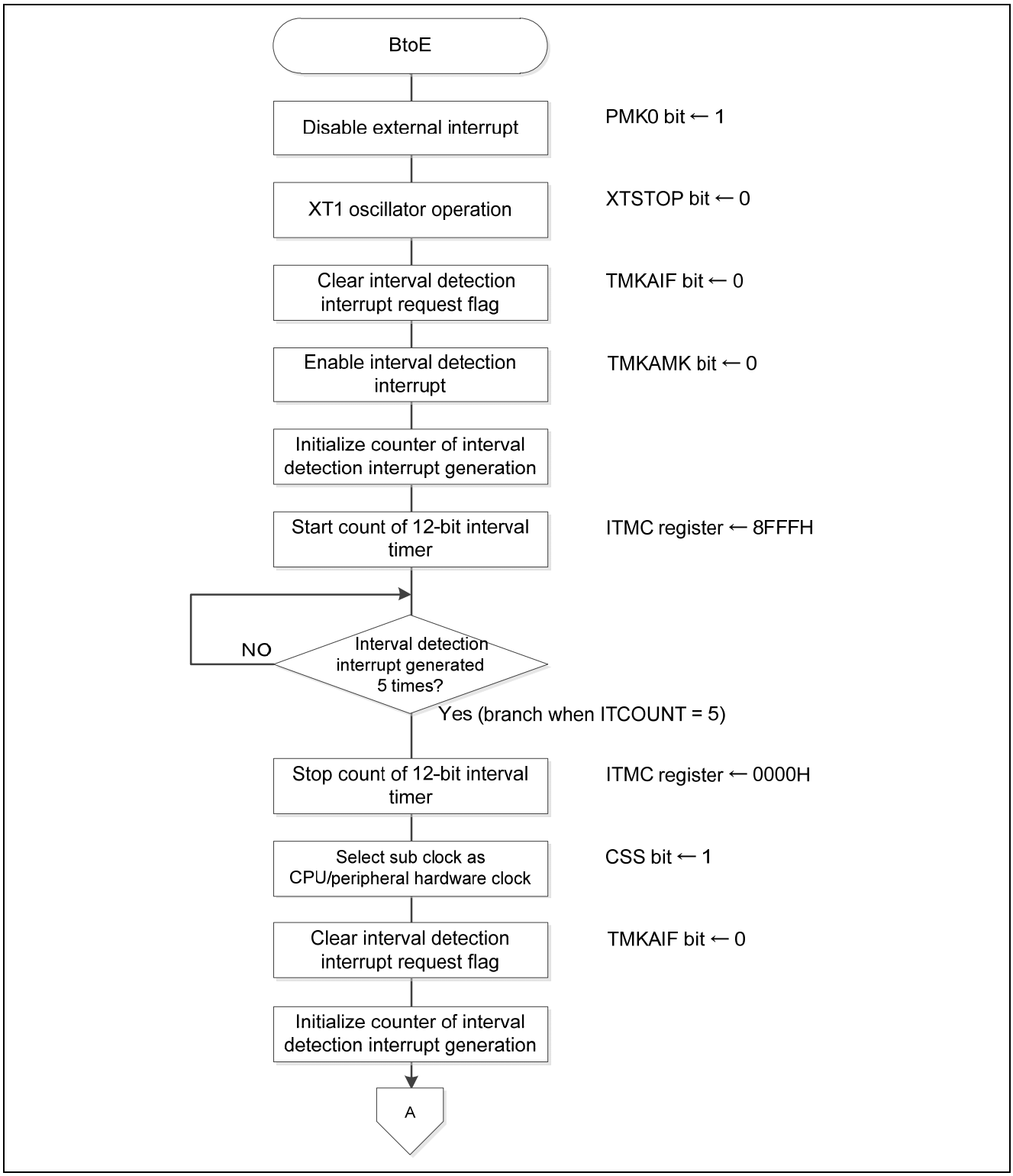

Figure 5.14 Status Transition BtoE (1/2)

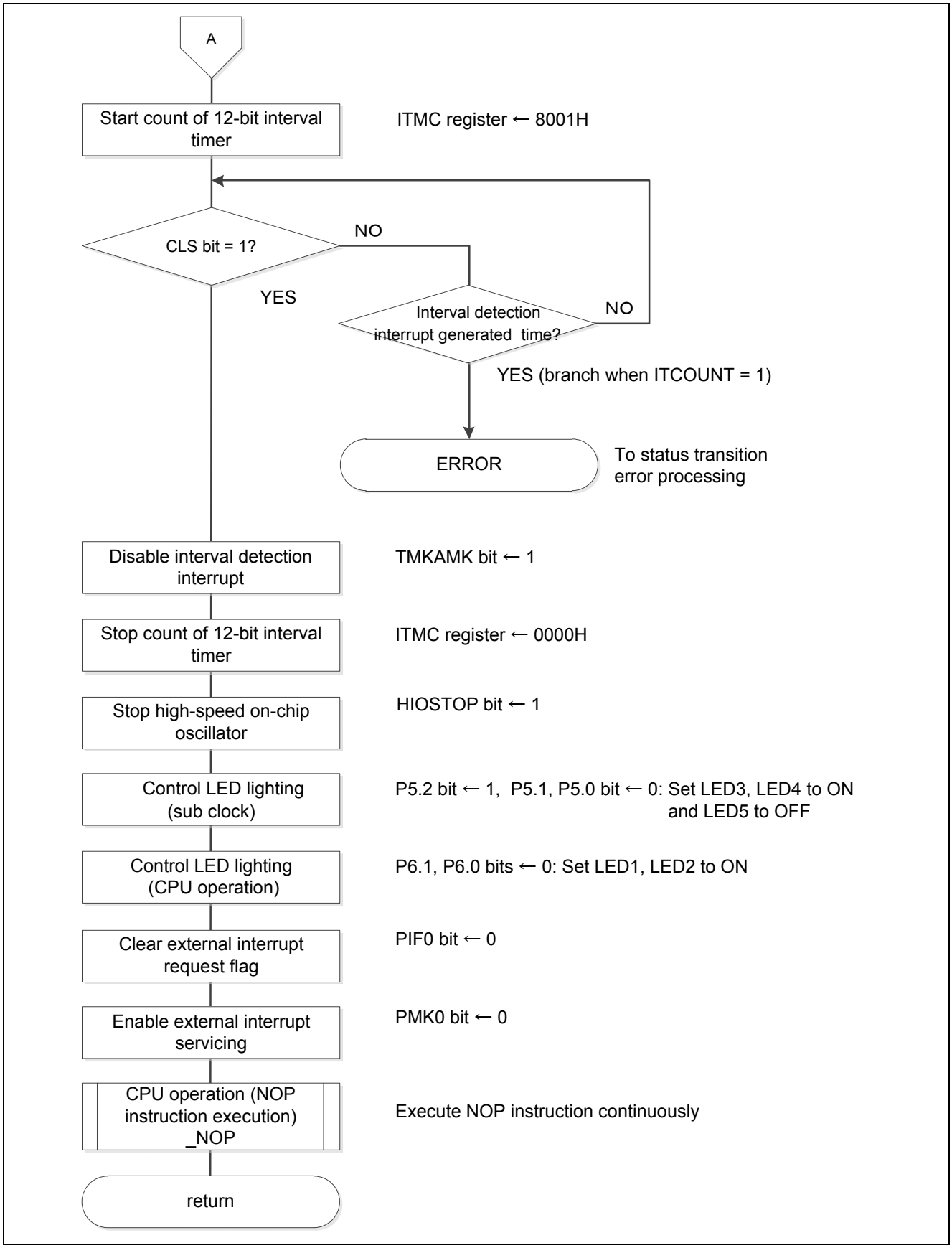

Figure 5.15 Status Transition BtoE (2/2)

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

- Interrupt request flag register (IF1H)
  Clear TMKAIF interrupt source flag.
- Interrupt mask flag register (MK1H)
  Set TMKAMK interrupt mask.

Symbol: IF1H

7	6	5	4	3	2	1	0
0	DOCIF	CMPIF1	CMPIF0	KRIF	TMKAIF	RTCIF	ADIF
0	x	x	x	x	**0**	x	x

Bit 2

TMKAIF	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
0	DOCMK	CMPMK1	CMPMK0	KRMK	TMKAMK	RTCMK	ADMK
x	x	x	x	x	**0**	x	x

Bit 2

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

Note: Refer to the RL78/I1D 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/I1D User's Manual (Hardware version) for details on how to set registers.

5.6.10 Error Processing of Status Transition

Figure 5.16 shows the flowchart for error processing of status transition.

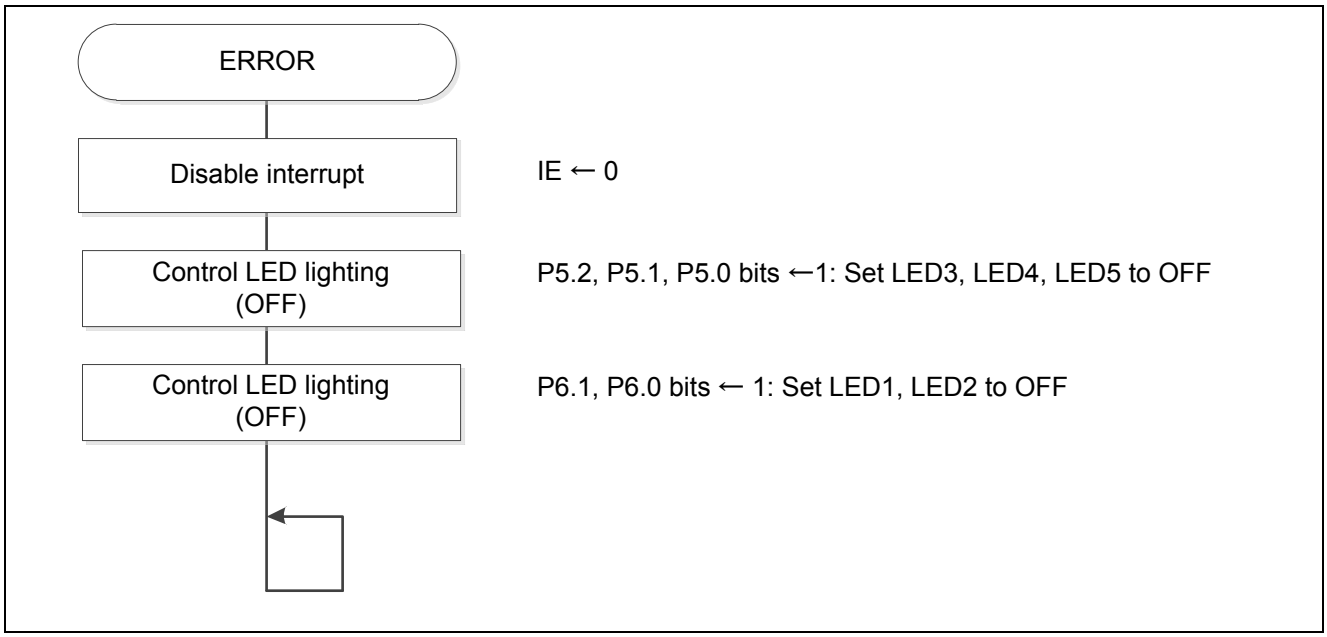

Figure 5.16 Error Processing of Status Transition

5.6.11 Status Transition EtoO

Figure 5.17 shows the flowchart for status transition EtoO.

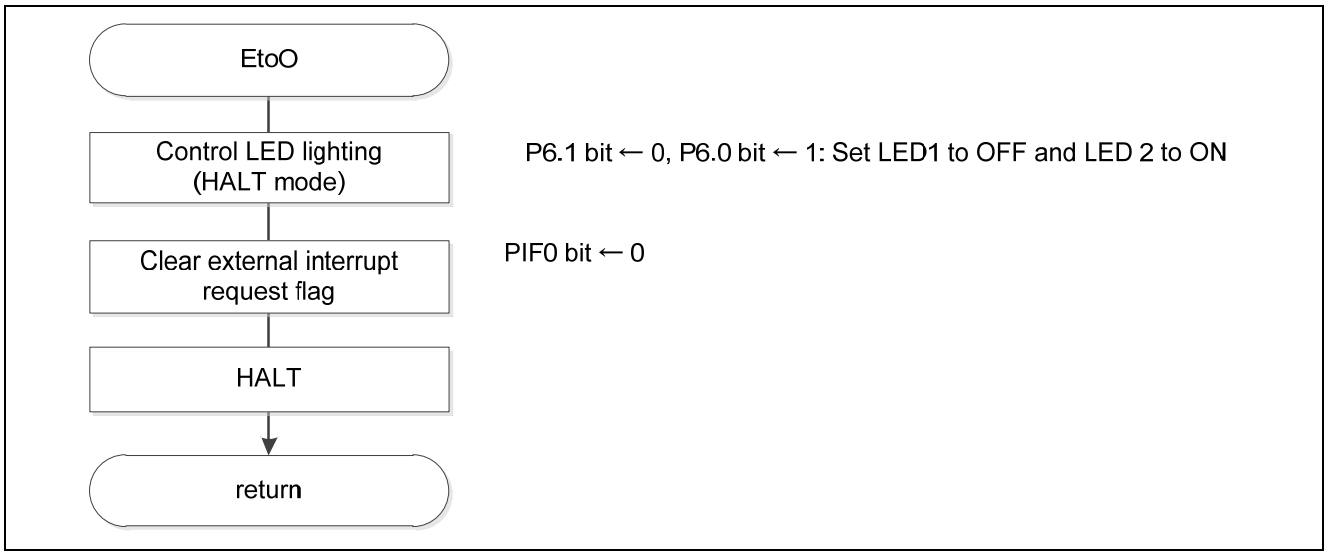

Figure 5.17 Status Transition EtoO

5.6.12 Status Transition OtoE

Figure 5.18 shows the flowchart for status transition OtoE.

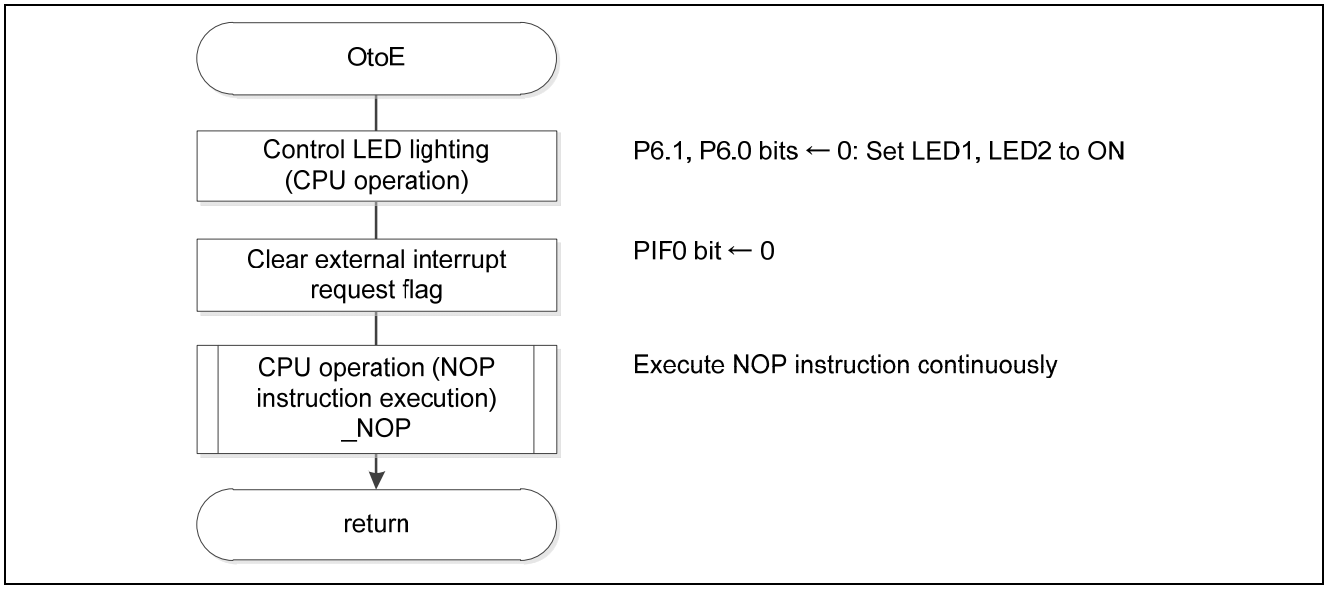

Figure 5.18 Status Transition OtoE

5.6.13 Status Transition EtoB

Figure 5.19 and Figure 5.20 shows the flowchart for status transition EtoB.

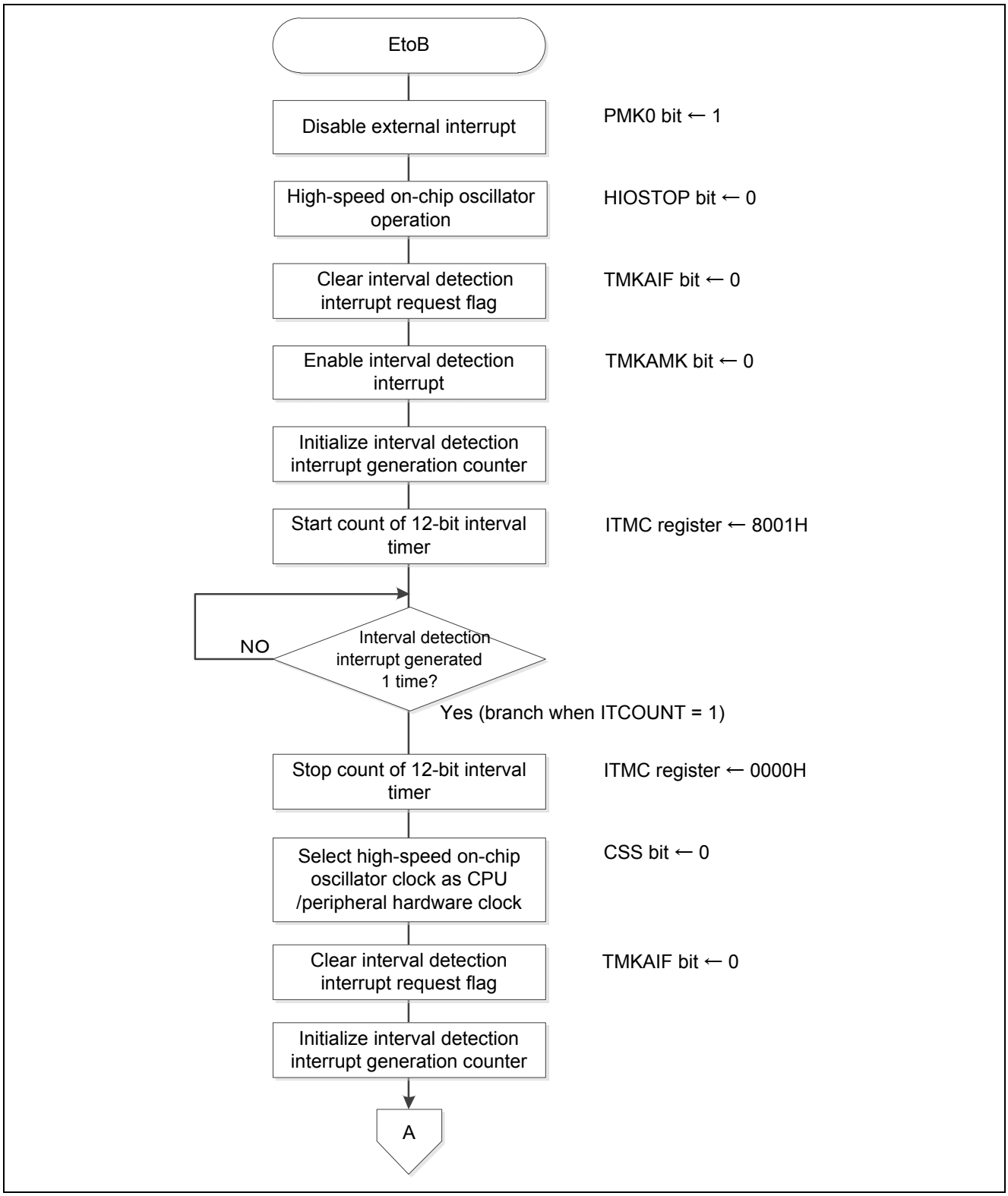

Figure 5.19 Status Transition EtoB (1/2)

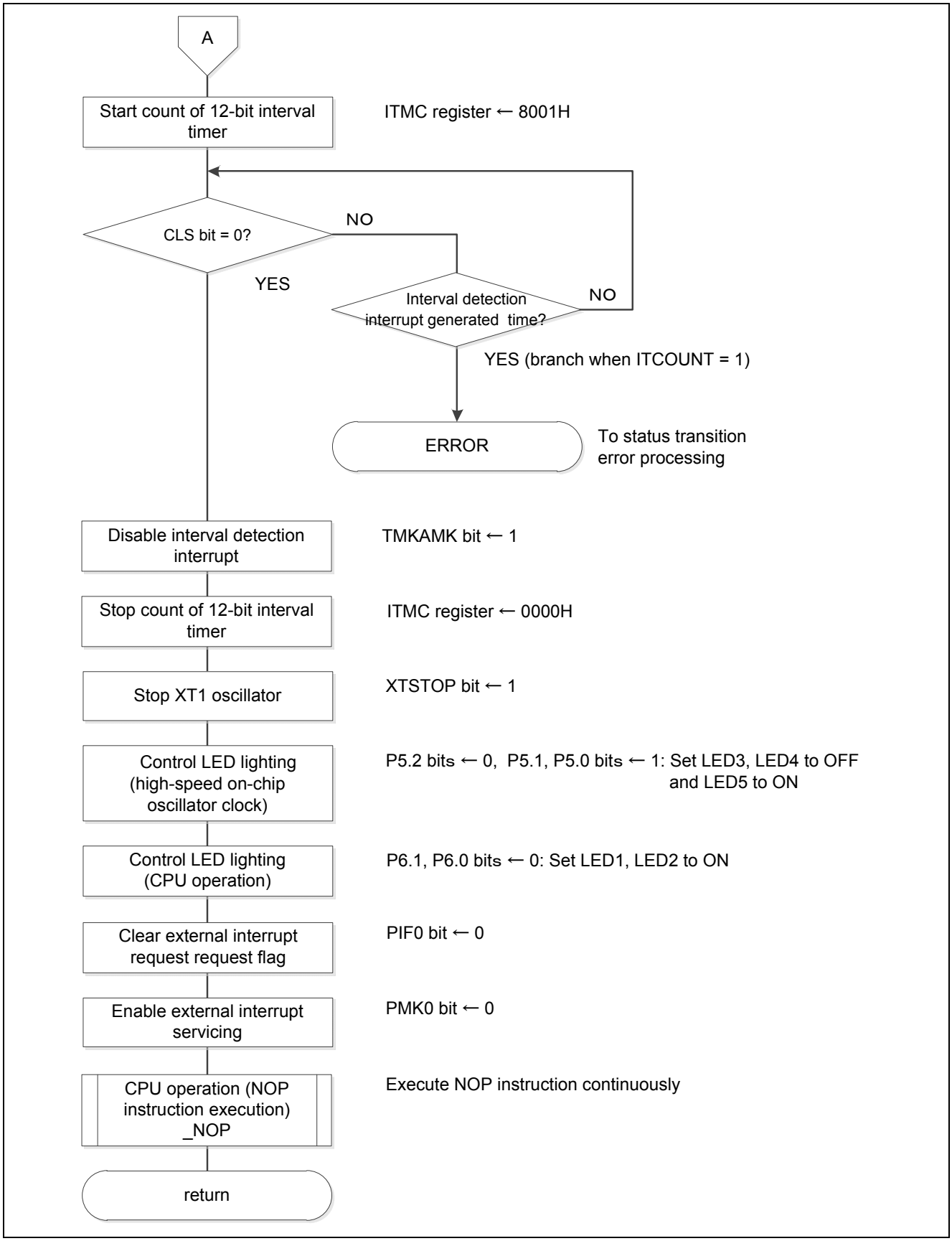

Figure 5.20 Status Transition EtoB (2/2)

5.6.14 Status Transition BtoD

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

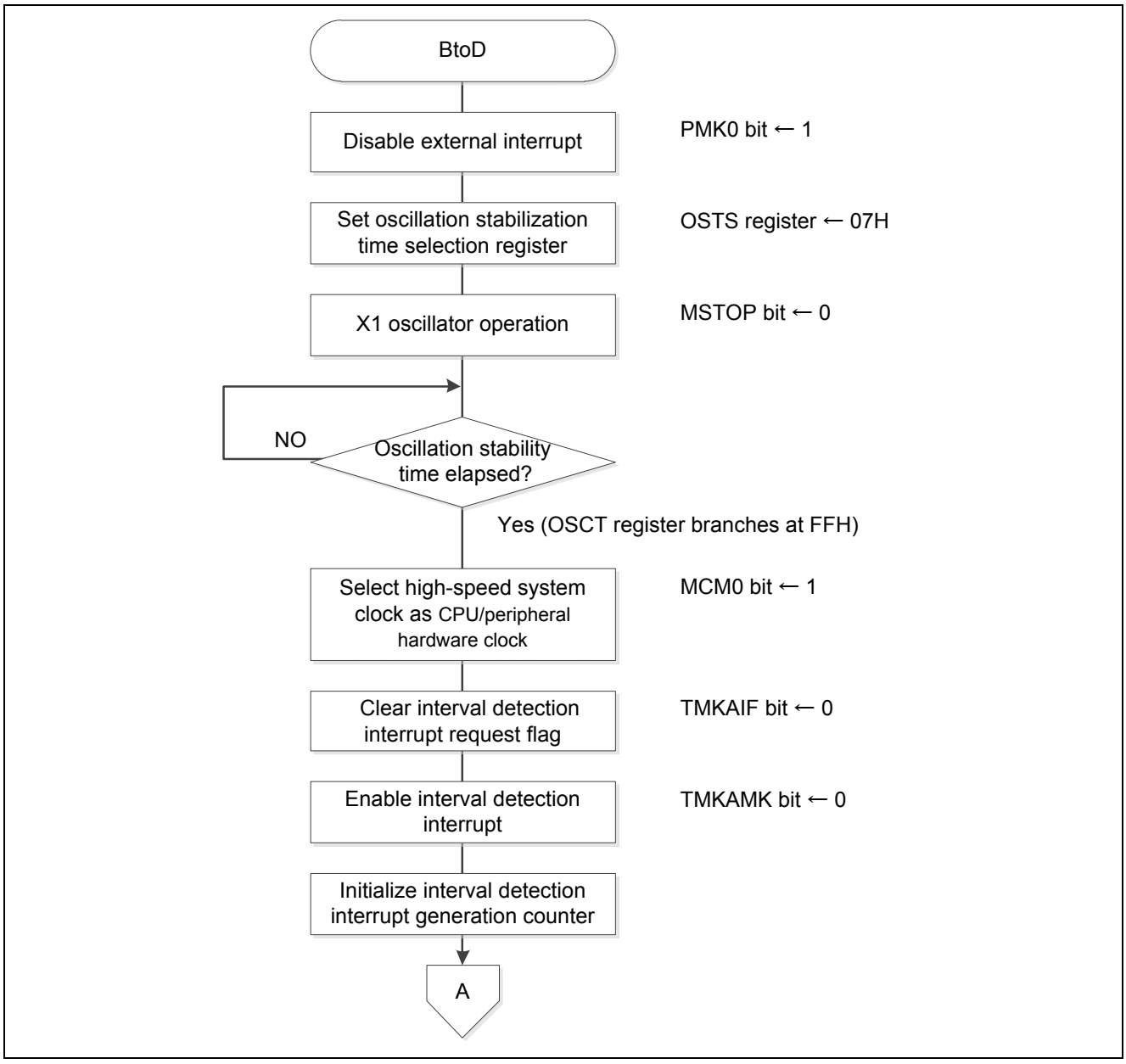

Figure 5.21 Status Transition BtoD (1/2)

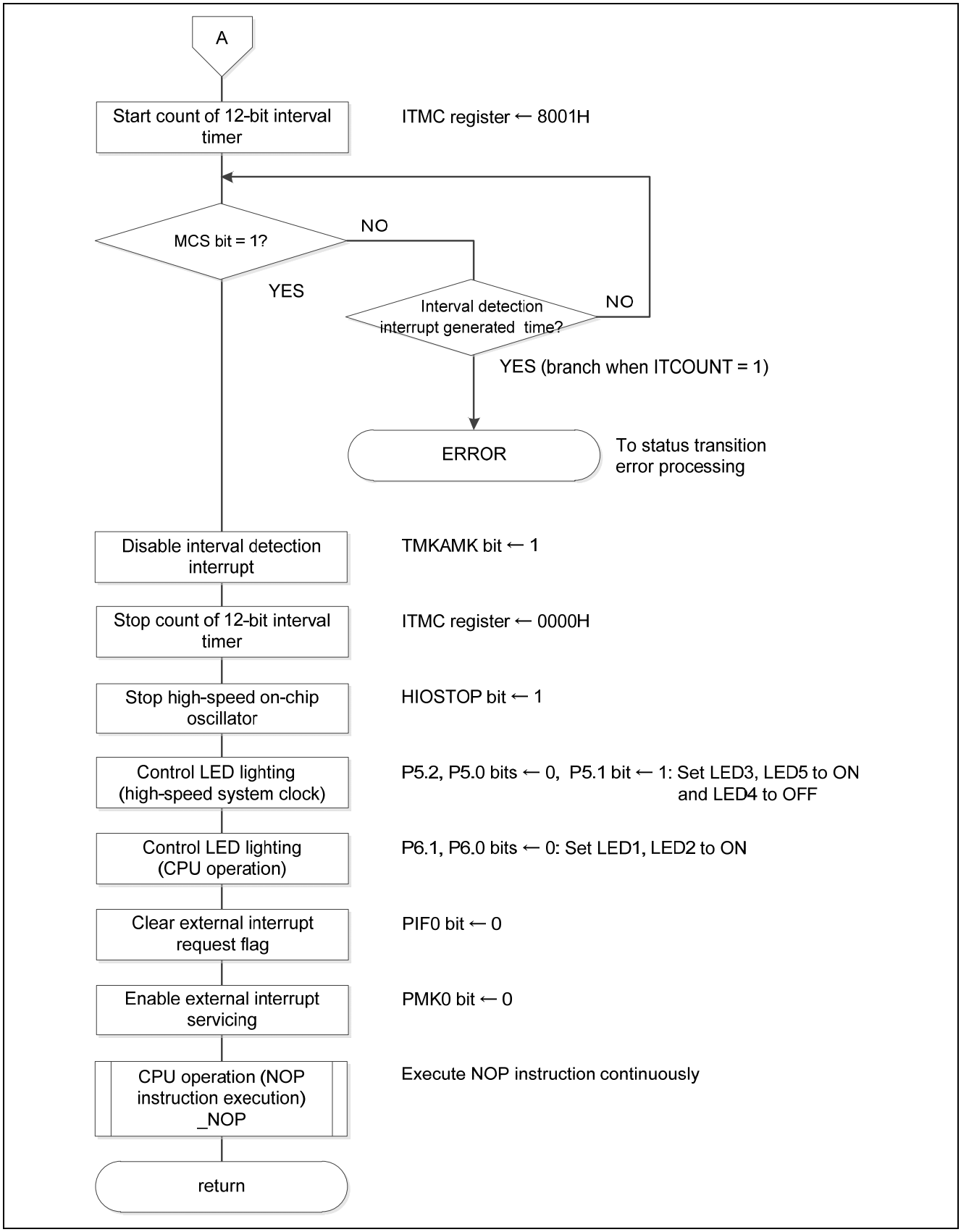

Figure 5.22 Status Transition BtoD (2/2)

5.6.15 Status Transition DtoE

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

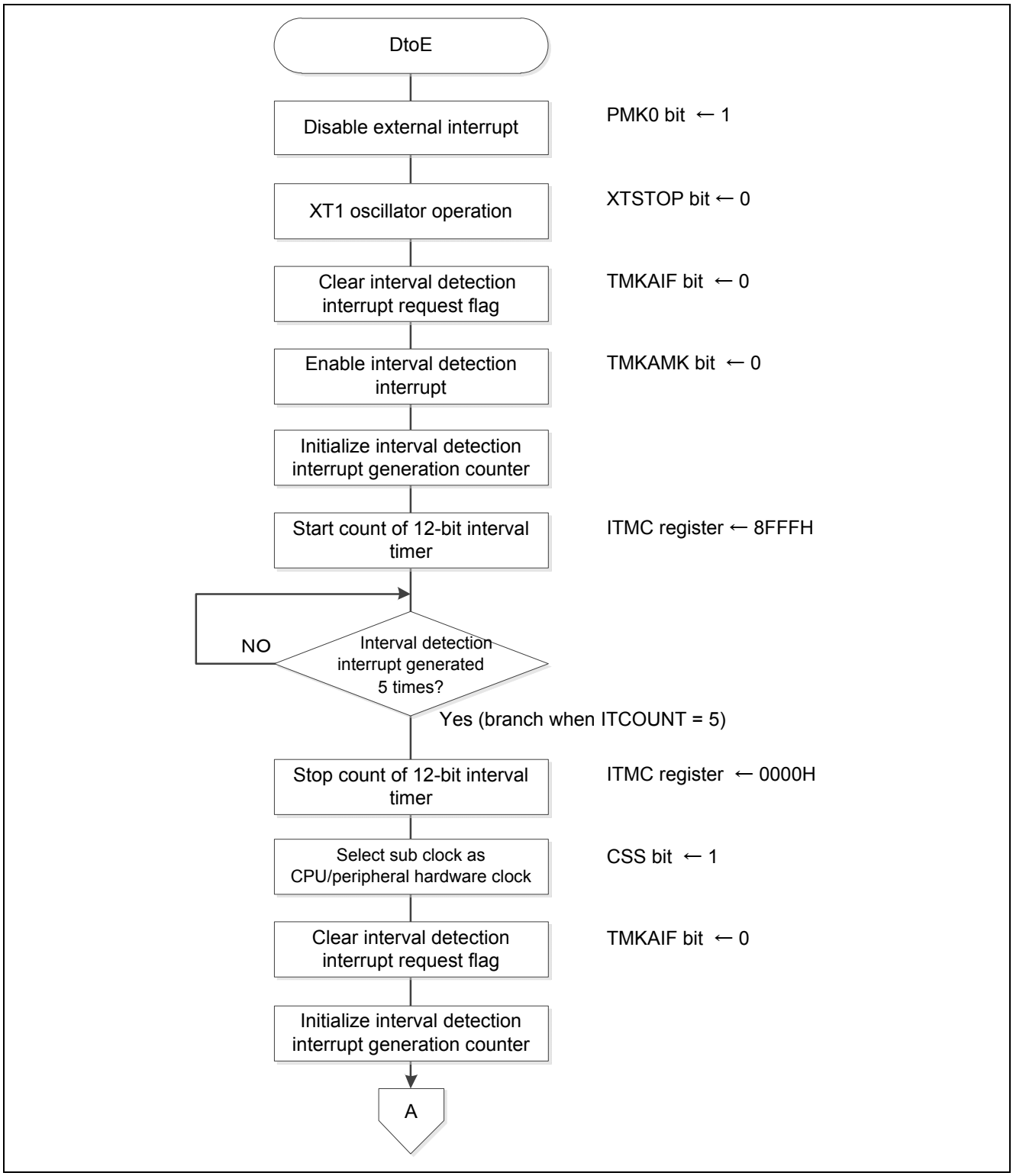

Figure 5.23 Status Transition DtoE (1/2)

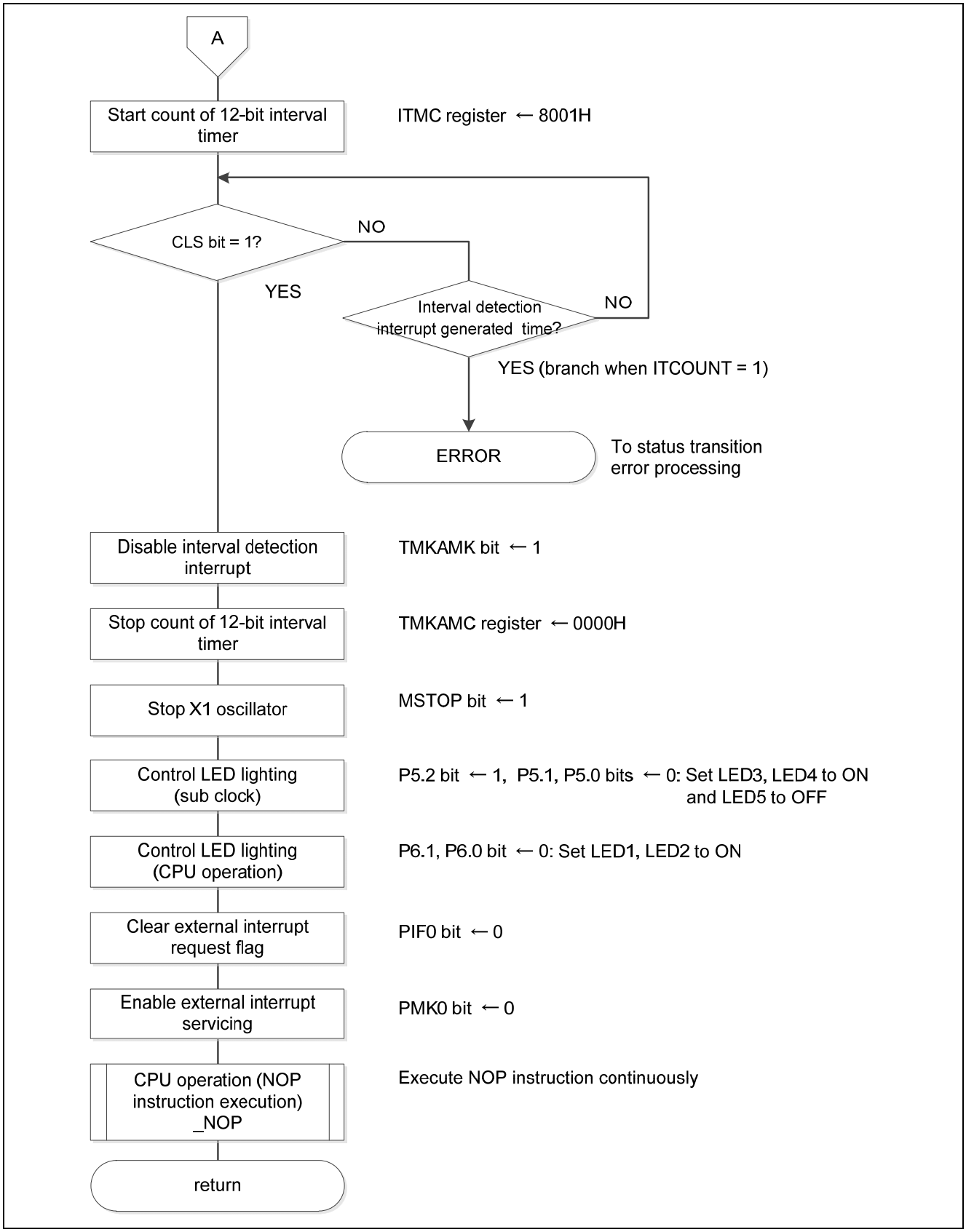

Figure 5.24 Status Transition DtoE (2/2)

5.6.16 Status Transition EtoD

Figure 5.25 and Figure 5.26 shows the flowchart for status transition EtoD.

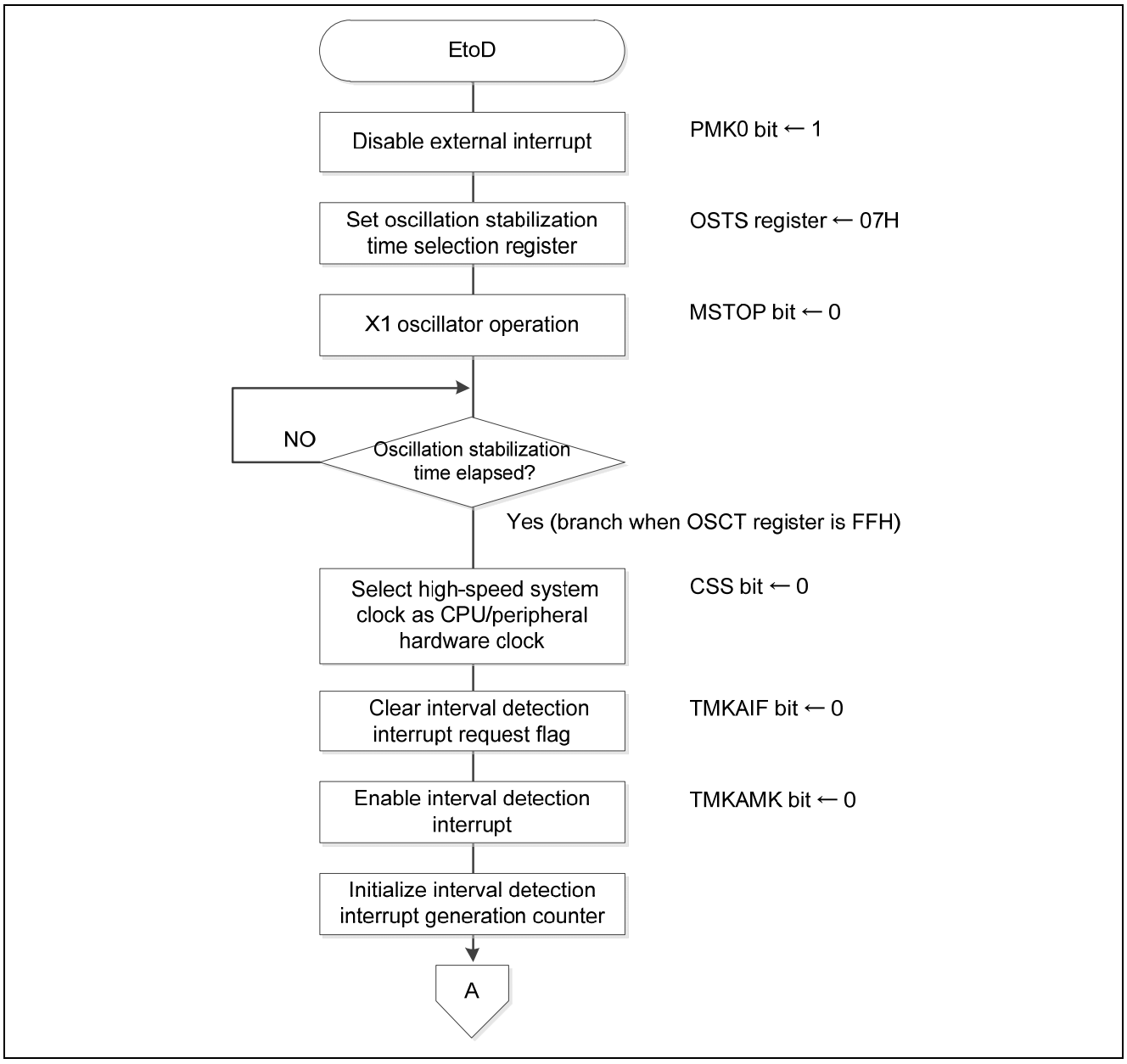

Figure 5.25 Status Transition EtoD (1/2)

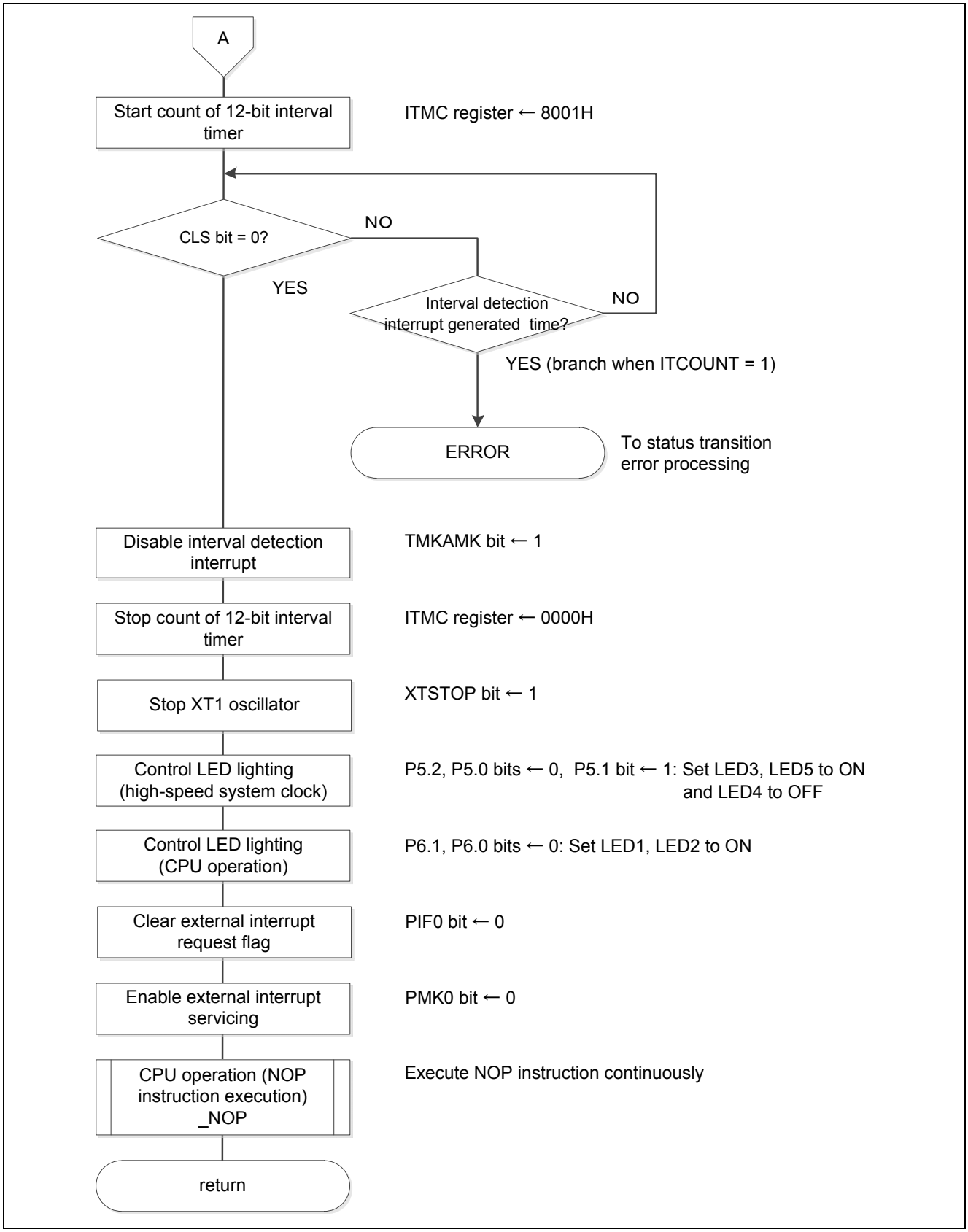

Figure 5.26 Status Transition EtoD (2/2)

5.6.17 Status Transition DtoM

Figure 5.27 shows the flowchart for status transition DtoM.

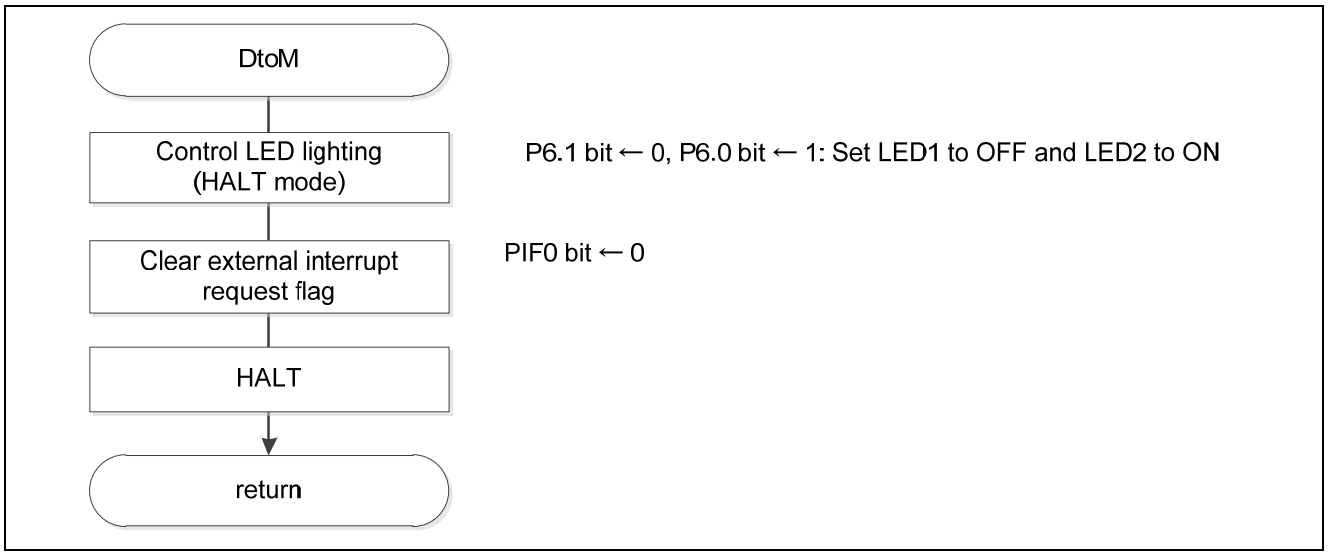

Figure 5.27 Status Transition DtoM

5.6.18 Status Transition MtoD

Figure 5.28 shows the flowchart for status transition MtoD.

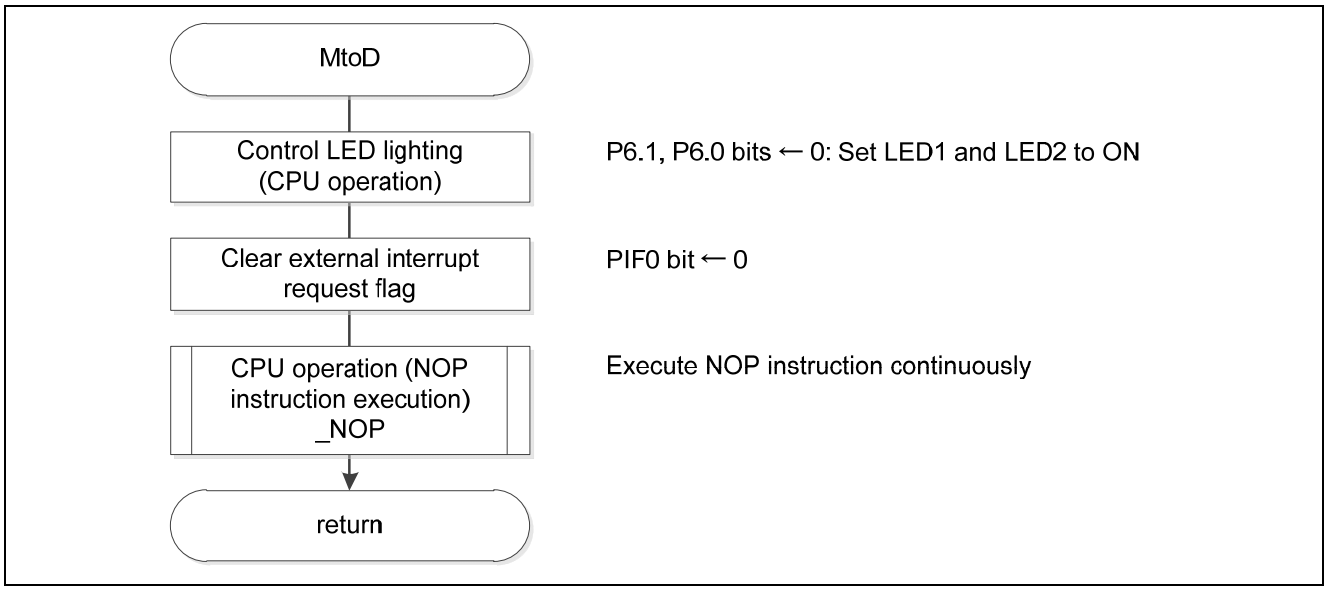

Figure 5.28 Status Transition MtoD

5.6.19 Status Transition DtoN

Figure 5.29 shows the flowchart for status transition DtoN.

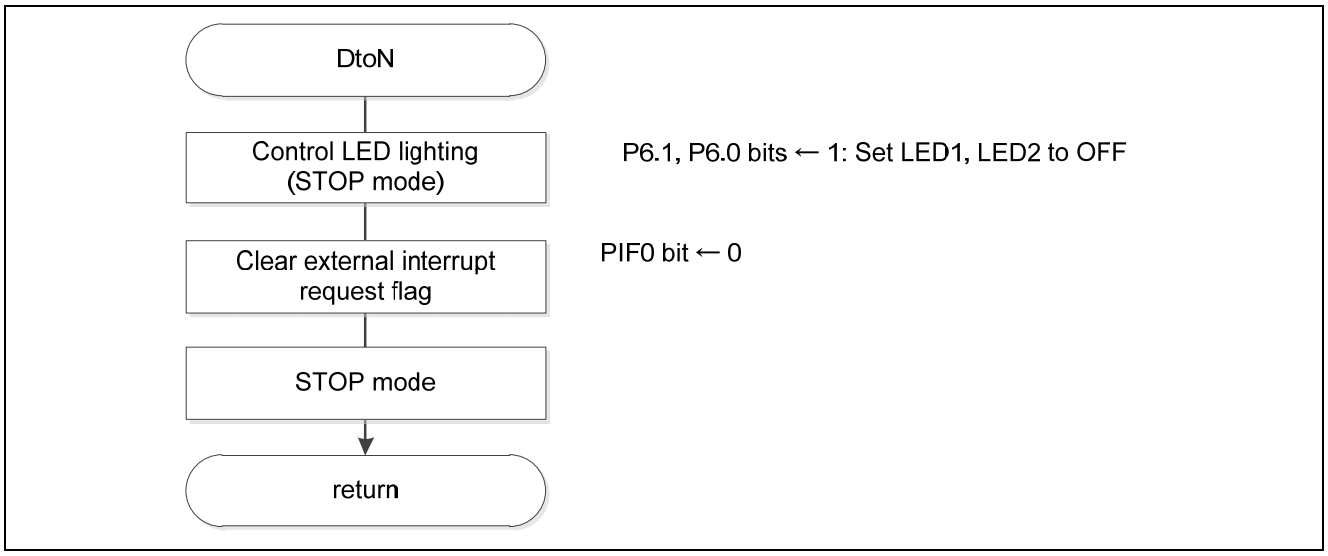

Figure 5.29 Status Transition DtoN

5.6.20 Status Transition NtoD

Figure 5.30 shows the flowchart for status transition NtoD.

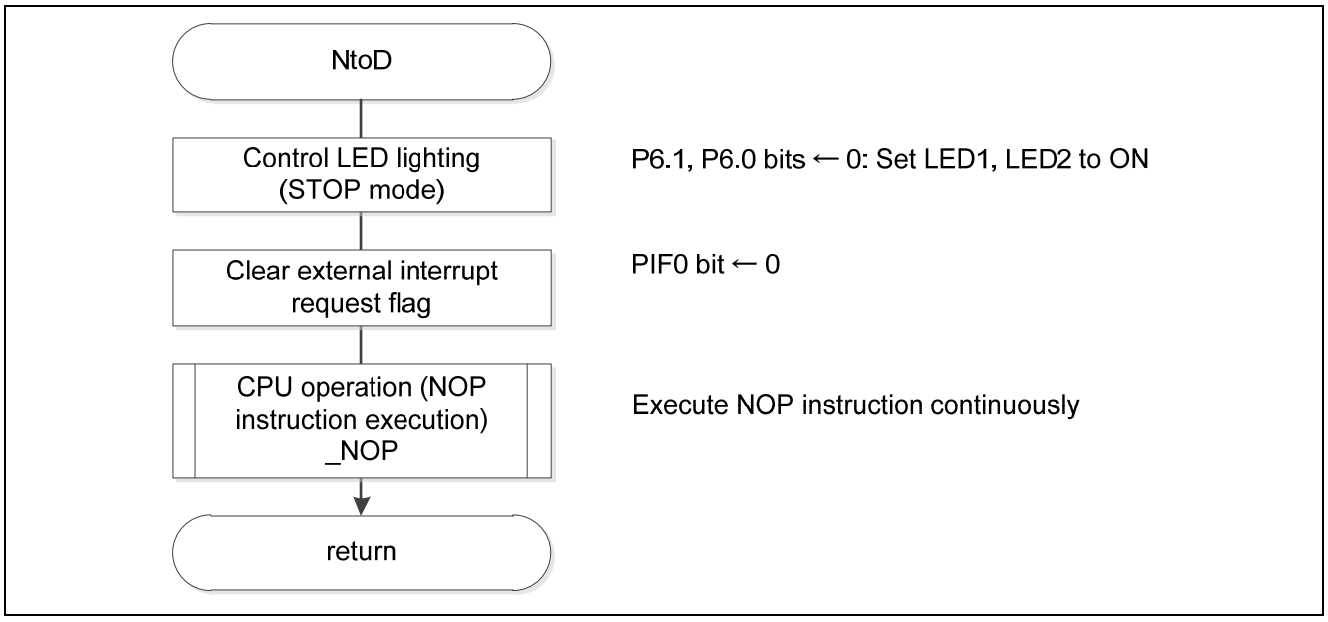

Figure 5.30 Status Transition NtoD

5.6.21 Status Transition DtoB

Figure 5.31 and Figure 5.32 shows the flowchart for status transition DtoB.

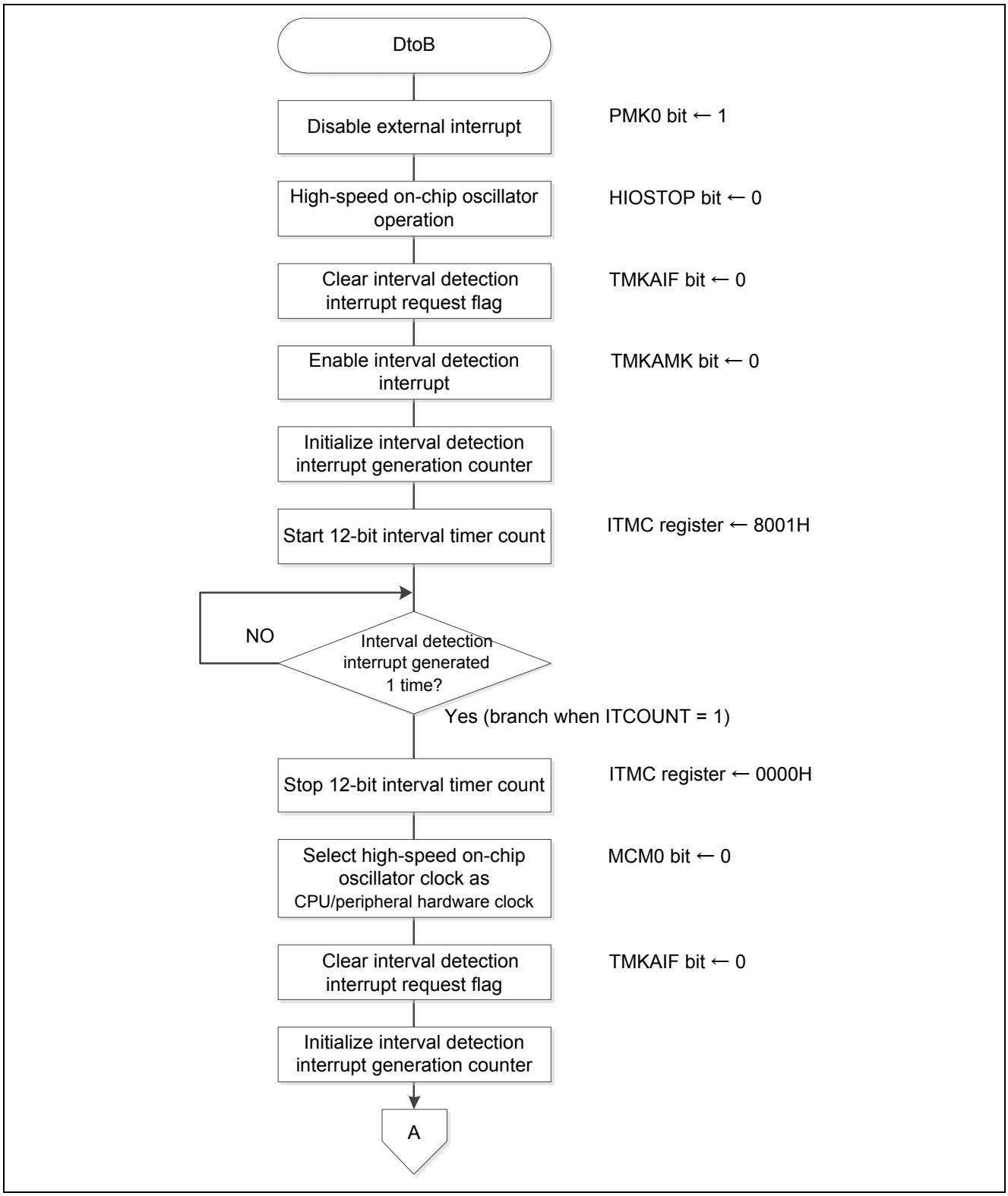

Figure 5.31 Status Transition DtoB (1/2)

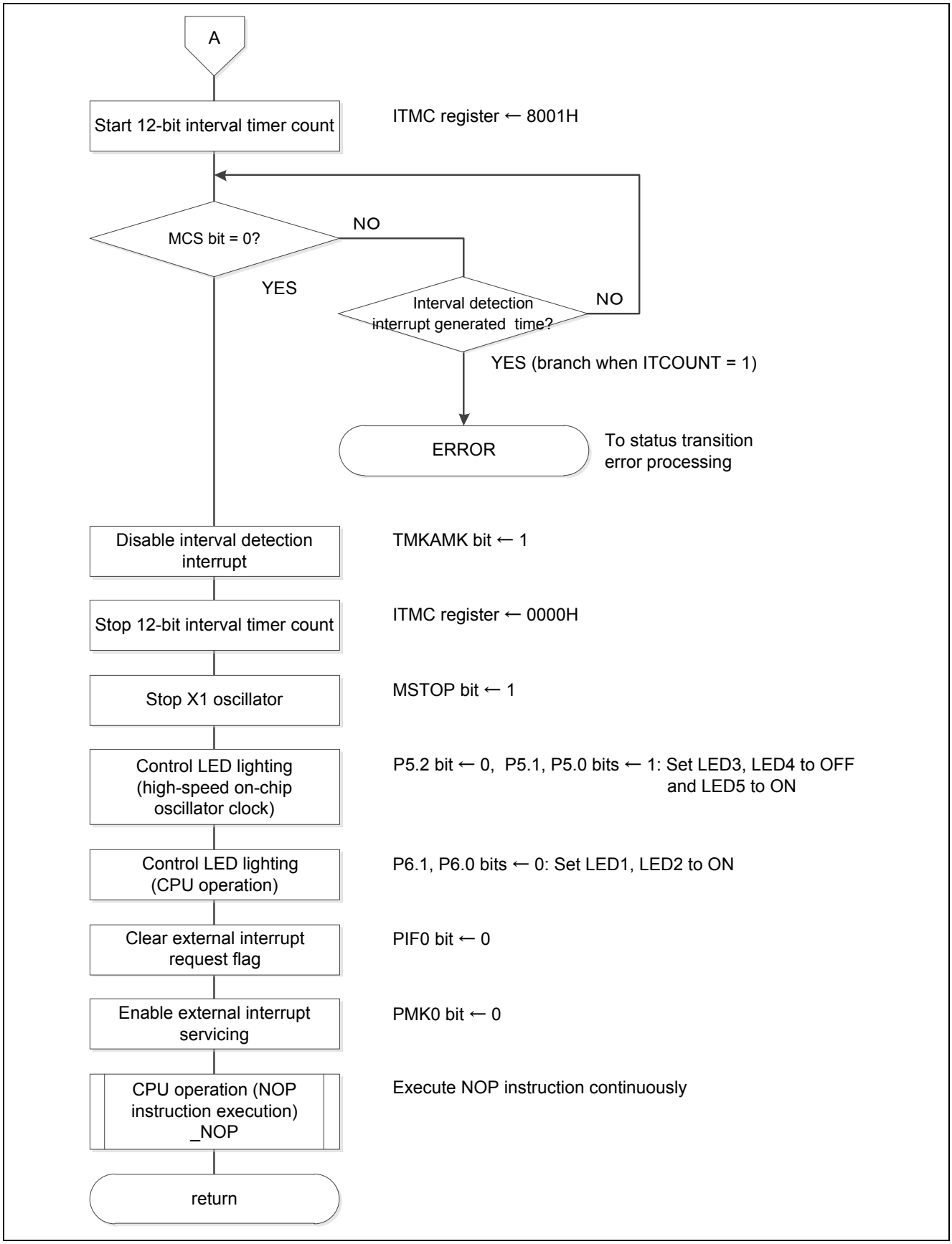

Figure 5.32 Status Transition DtoB(2/2)

5.6.22 Status Transition BtoI

Figure 5.33 shows the flowchart for status transition BtoI.

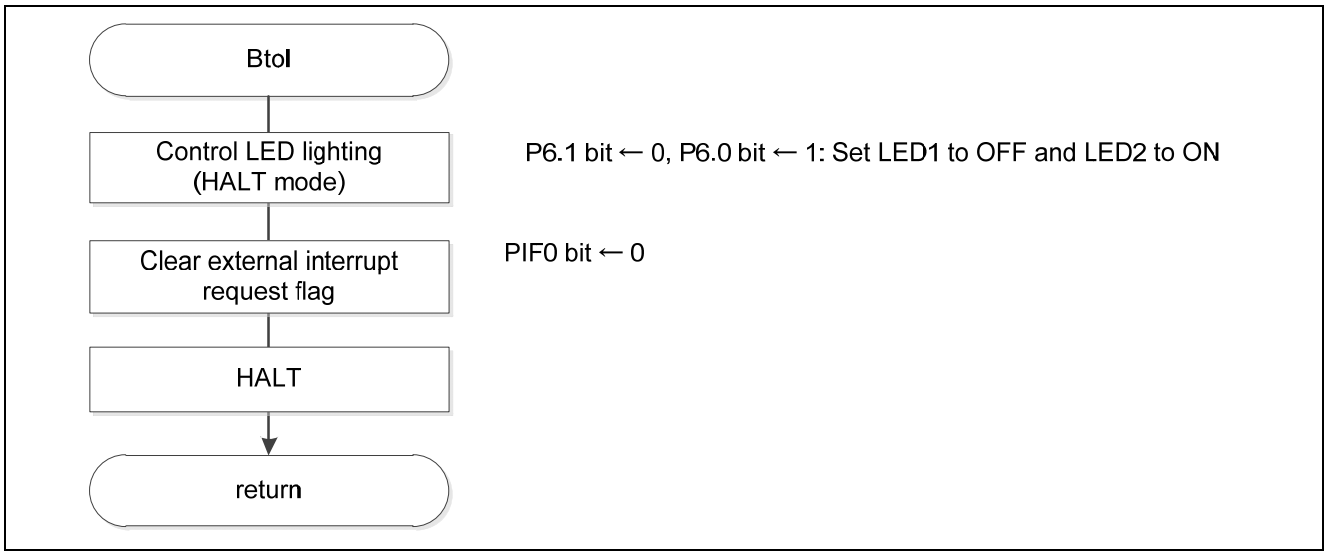

Figure 5.33 Status Transition BtoI

5.6.23 Status Transition ItoB

Figure 5.34 shows the flowchart for status transition ItoB.

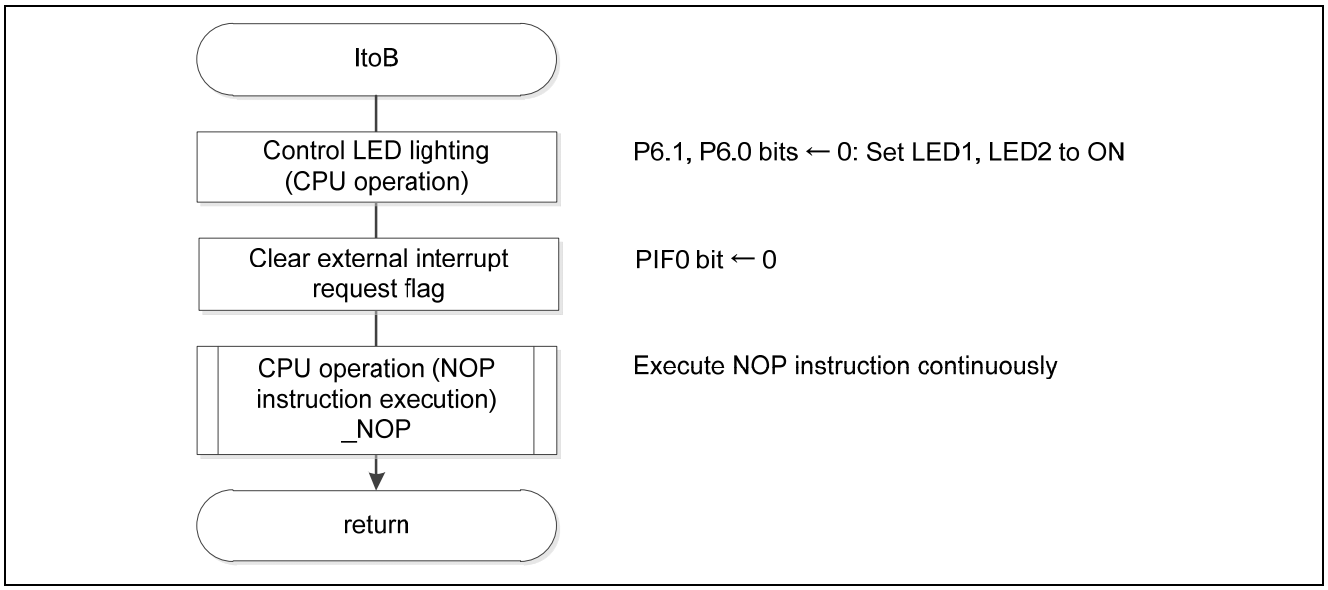

Figure 5.34 Status Transition ItoB

5.6.24 Status Transition BtoG

Figure 5.35 shows the flowchart for status transition BtoG.

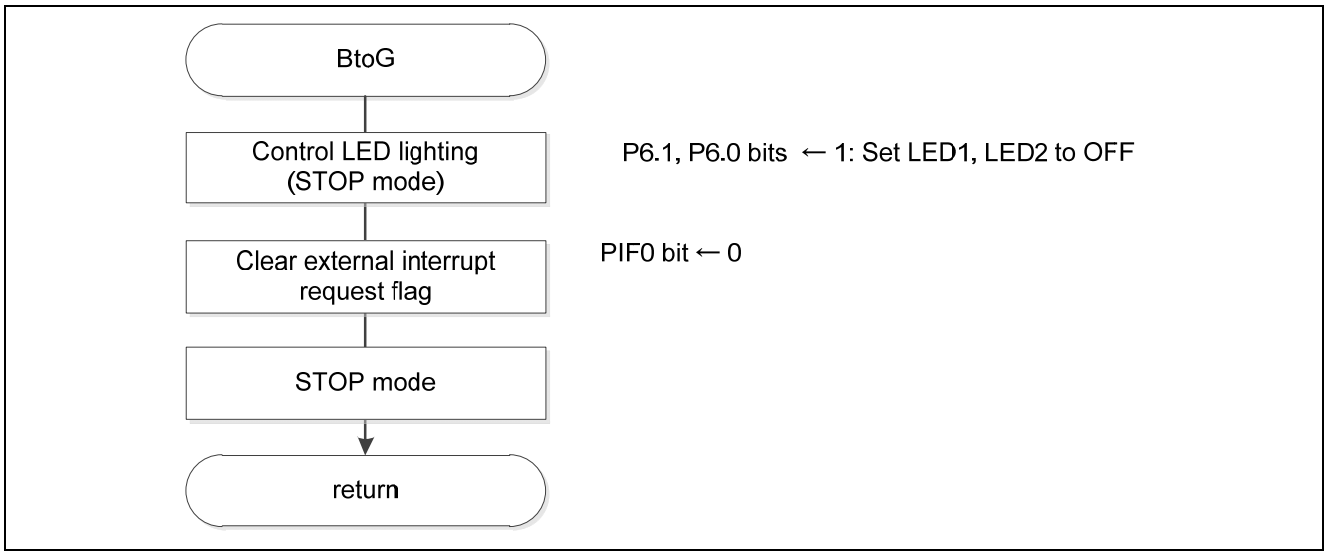

Figure 5.35 Status Transition BtoG

5.6.25 Status Transition GtoB

Figure 5.36 shows the flowchart for status transition GtoB.

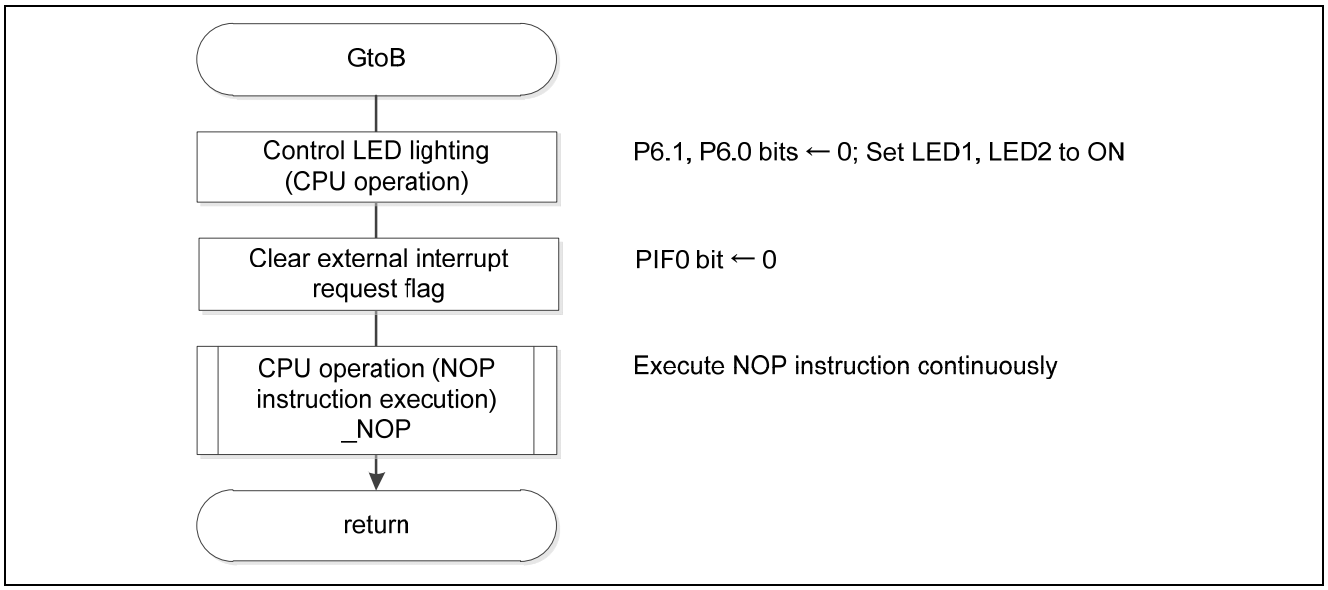

Figure 5.36 Status Transition GtoB

5.6.26 Status Transition BtoH

Figure 5.37 shows the flowchart for status transition BtoH.

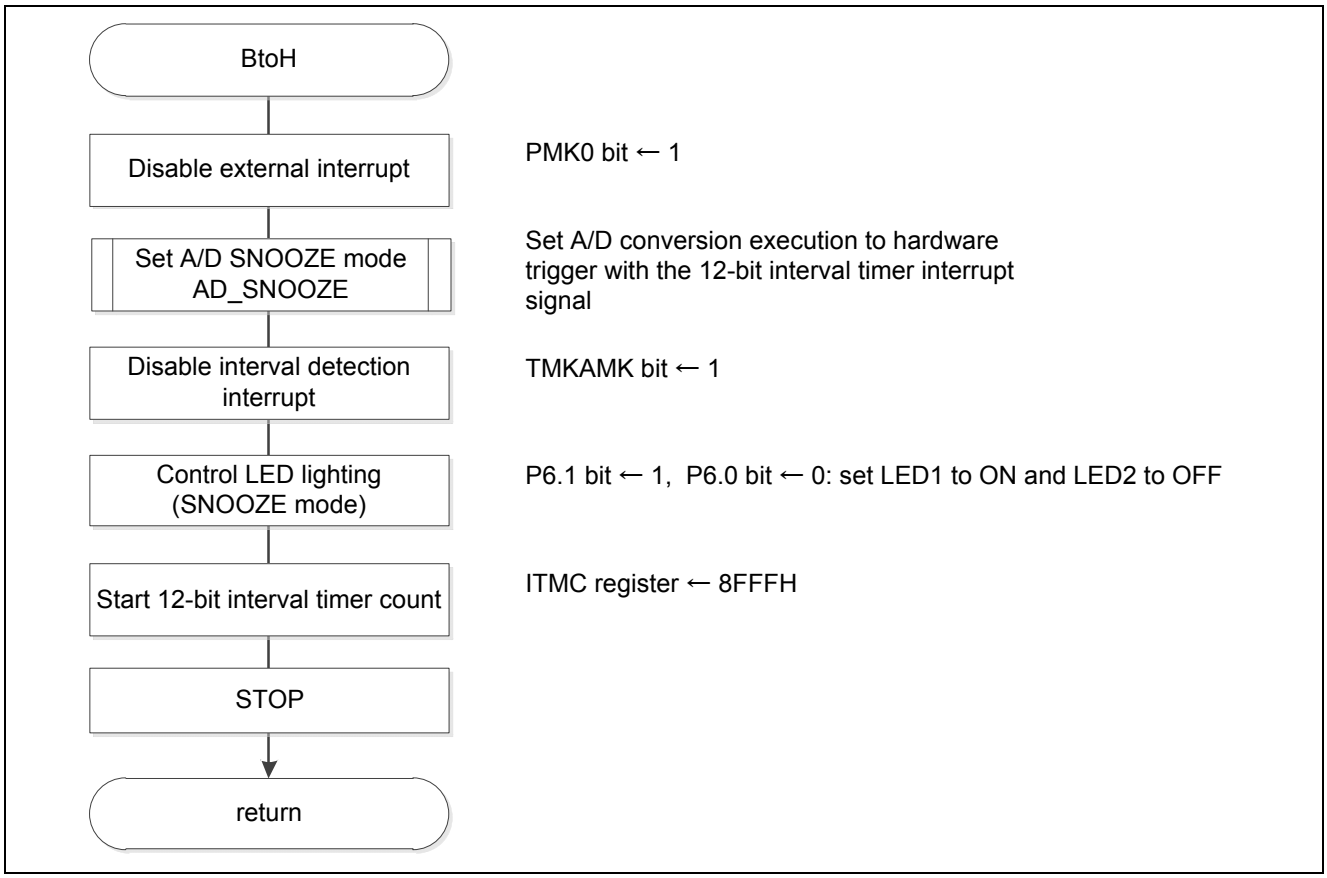

Figure 5.37 Status Transition BtoH

5.6.27 A/D Converter Setting

Figure 5.38 shows the flowchart for setting the A/D converter.

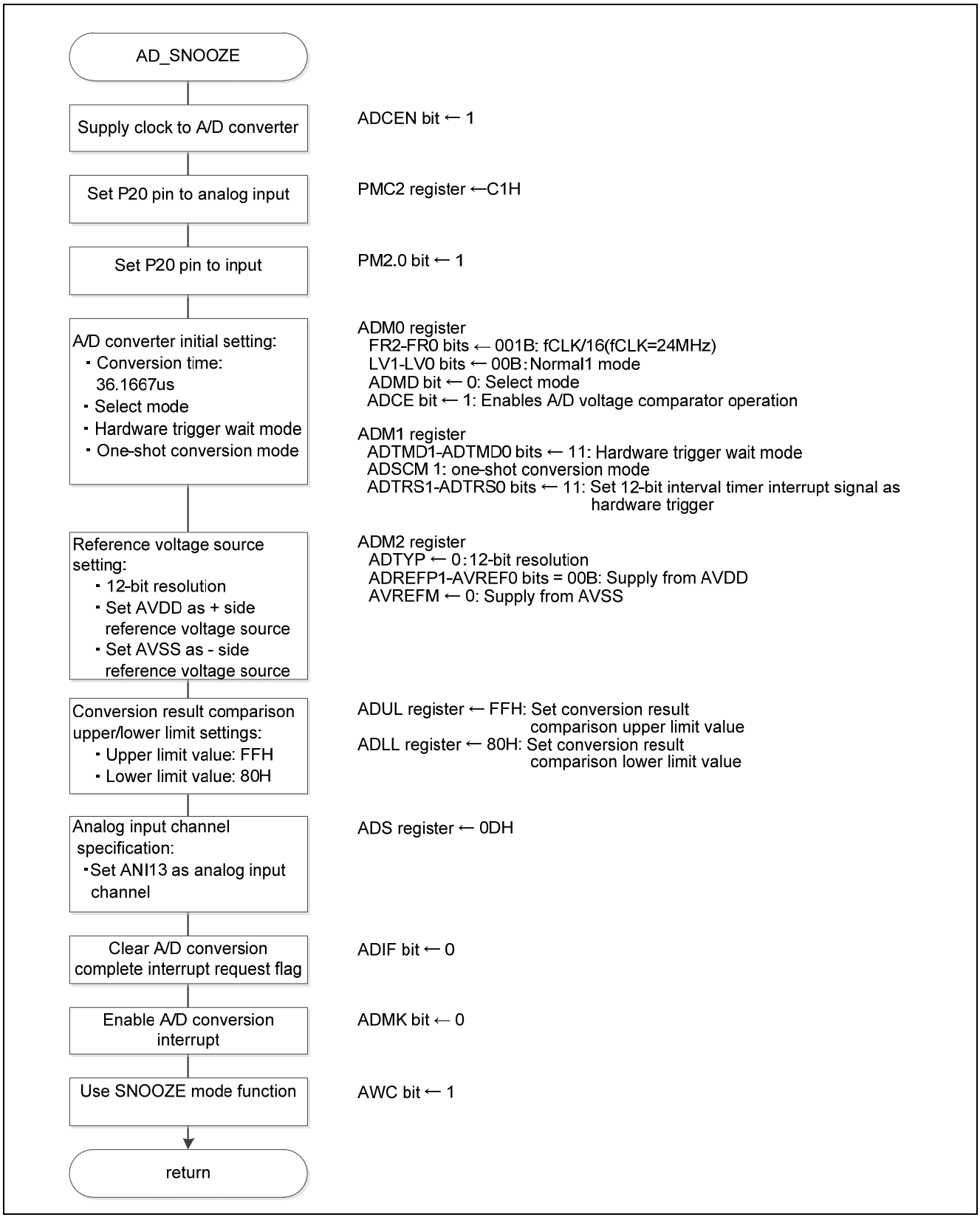

Figure 5.38 A/D Converter Setting

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}= 24MHz**	
0	0	0	**0**	**0**	**Normal 1**	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	72.1667 µs	f_{CLK}/32
0	**0**	**1**							54.25 µs	**36.1667 µs**	f_{CLK}/16
0	1	0						54.5 µs	27.25 µs	18.1667 µs	f_{CLK}/8
0	1	1						41 µs	20.5 µs	13.6667 µs	f_{CLK}/6
1	0	0						34.25 µs	17.125 µs	11.4167 µs	f_{CLK}/5
1	0	1					55 µs	27.5 µs	13.75 µs	9.1667 µs	f_{CLK}/4
1	1	0					28 µs	14 µs	7 µs	4.6667 µs	f_{CLK}/2
1	1	1				56 μ s	14 µs	7 µs	3.5 µs	Setting prohibited	f_{CLK}/1

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/I1D 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/I1D 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 AV_{DD}**
0	1	Supplied from 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 AV_{SS}**
1	Supplied from AV_{REFM}/ANI1

Bit 3

ADCRK	Checking the upper and lower limit conversion result values
0	**The A/D conversion end interrupt request signal (INTAD) is output when the ADLL register ▪ the ADCR register ▪ the ADUL register (AREA1).**
1	The A/D conversion end interrupt request signal (INTAD) is output when the ADCR register < the ADLL register (AREA2) or the ADUL register < the ADCR register (AREA3).

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	**12-bit resolution**
1	8-bit resolution

Note: Refer to the RL78/I1D User's Manual (hardware version) for detailed explanations on how to set registers.

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**	**1**	**1**	**0**	**1**

Bits 7, 4-0

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P10/ANI0 /AV_{REFP} pin
0	0	0	0	0	1	ANI1	P11/ANI1 /AV_{REFM} pin
0	0	0	0	1	0	ANI2	P12/ANI2 pin
0	0	0	0	1	1	ANI3	P13/ANI3 pin
0	0	0	1	0	0	ANI4	P14/ANI4 pin
0	0	0	1	0	1	ANI5	P15/ANI5 pin
0	0	0	1	1	0	ANI6	P16/ANI6 pin
0	0	0	1	1	1	ANI7	P17/ANI7 pin
0	0	1	0	0	0	ANI8	P25/ANI8 pin
0	0	1	0	0	1	ANI9	P24/ANI9 pin
0	0	1	0	1	0	ANI10	P23/ANI10 pin
0	0	1	0	1	1	ANI11	P22/ANI11 pin
0	0	1	1	0	0	ANI12	P21/ANI12 pin
0	**0**	**1**	**1**	**0**	**1**	**ANI13**	**P20/ANI13 pin**
0	1	0	0	0	0	ANI16	P02/ANI16 pin
0	1	0	0	0	1	ANI17	P03/ANI17 pin
0	1	0	0	1	0	ANI18	P04/ANI18 pin
1	0	0	0	0	0	—	Temperature sensor 0 output
1	0	0	0	0	1	—	Internal reference voltage output (1.45 V)
Other than the above						Setting prohibited	

Note: Refer to the RL78/I1D 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/I1D User's Manual (hardware version) for detailed explanations on how to set registers.

5.6.28 Status Transition HtoB

Figure 5.39 shows the status transition HtoB.

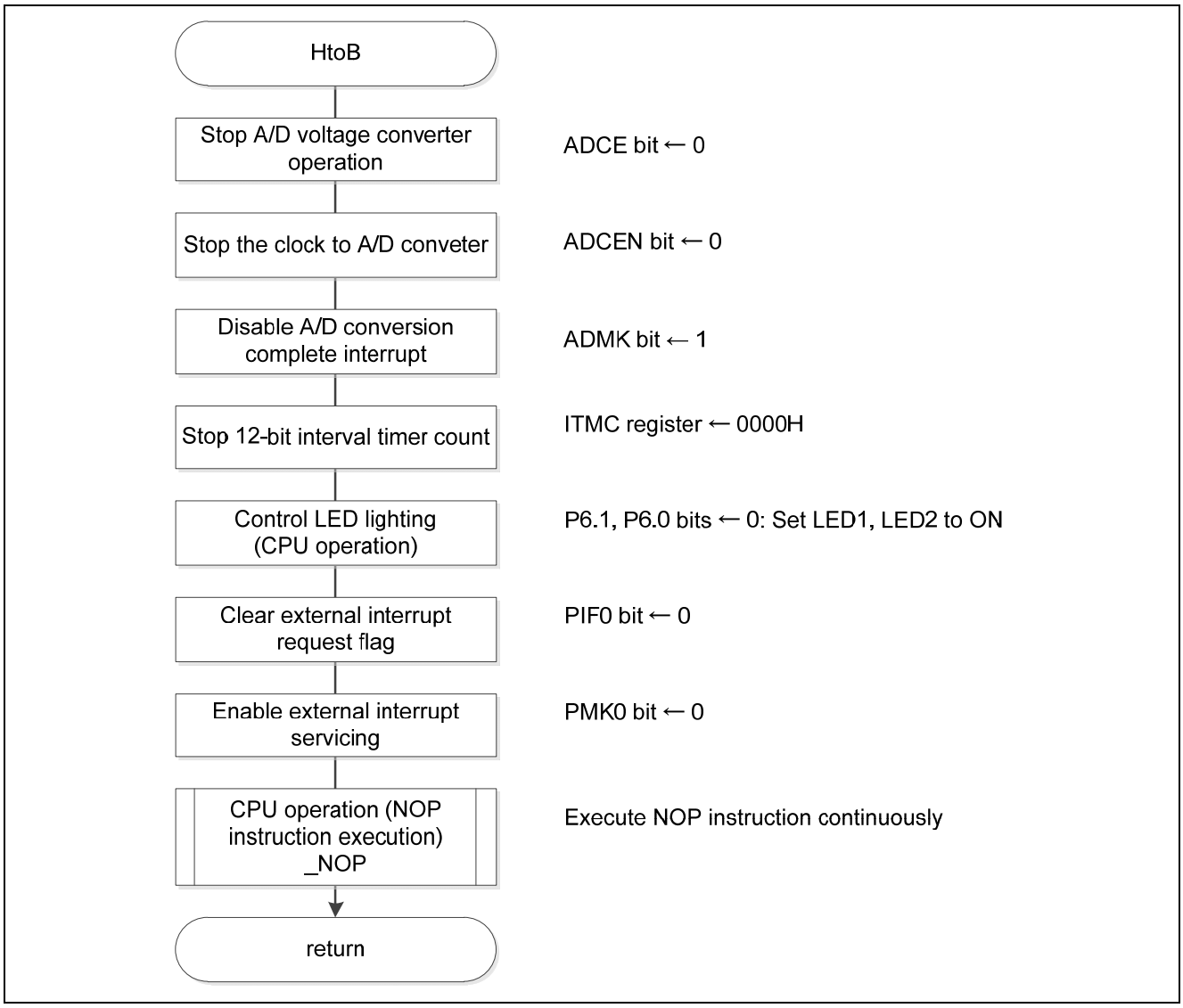

Figure 5.39 Status Transition HtoB

5.6.29 Status Transition BtoC

Figure 5.40 and Figure 5.41 shows the status transition BtoC.

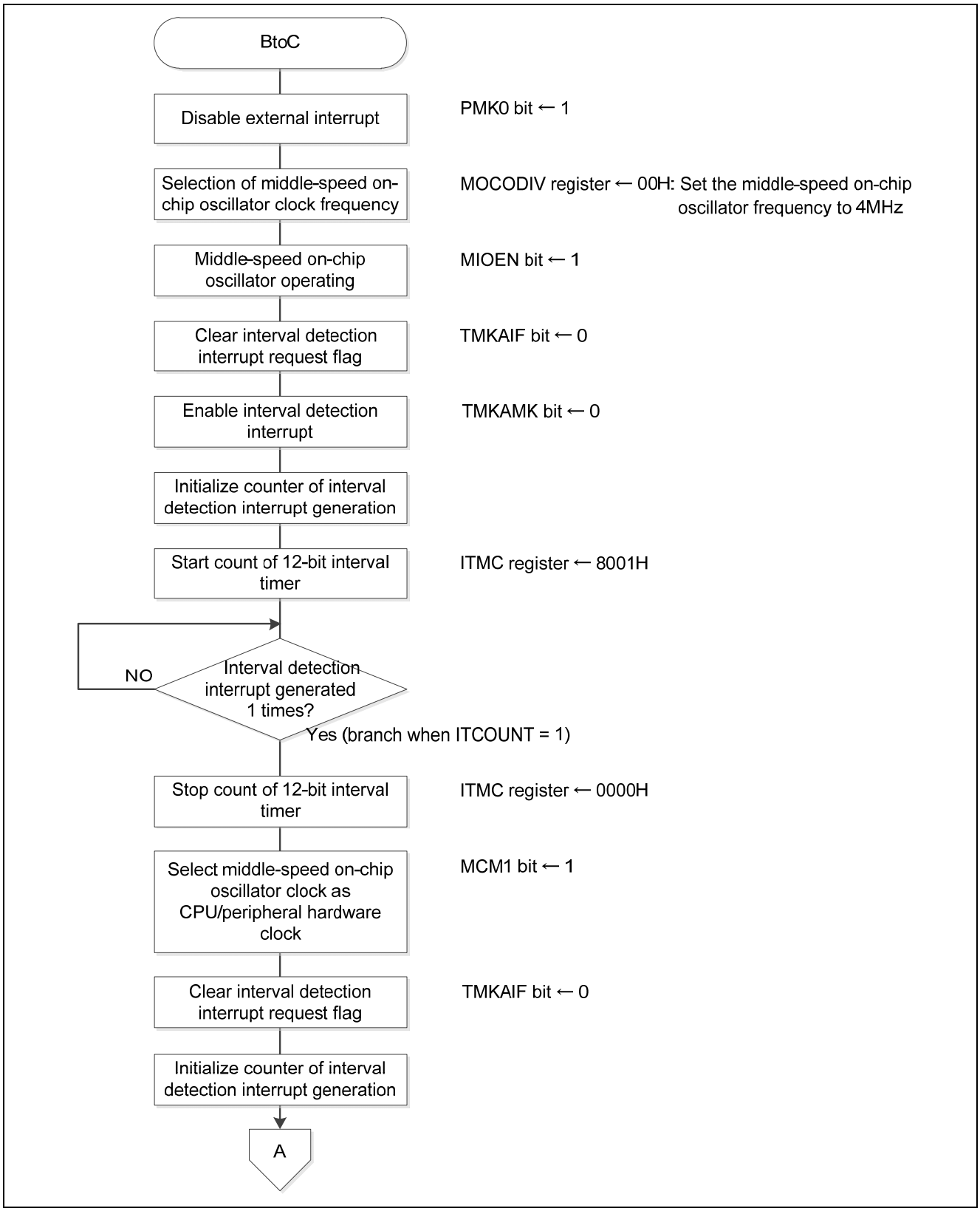

Figure 5.40 Status Transition BtoC(1/2)

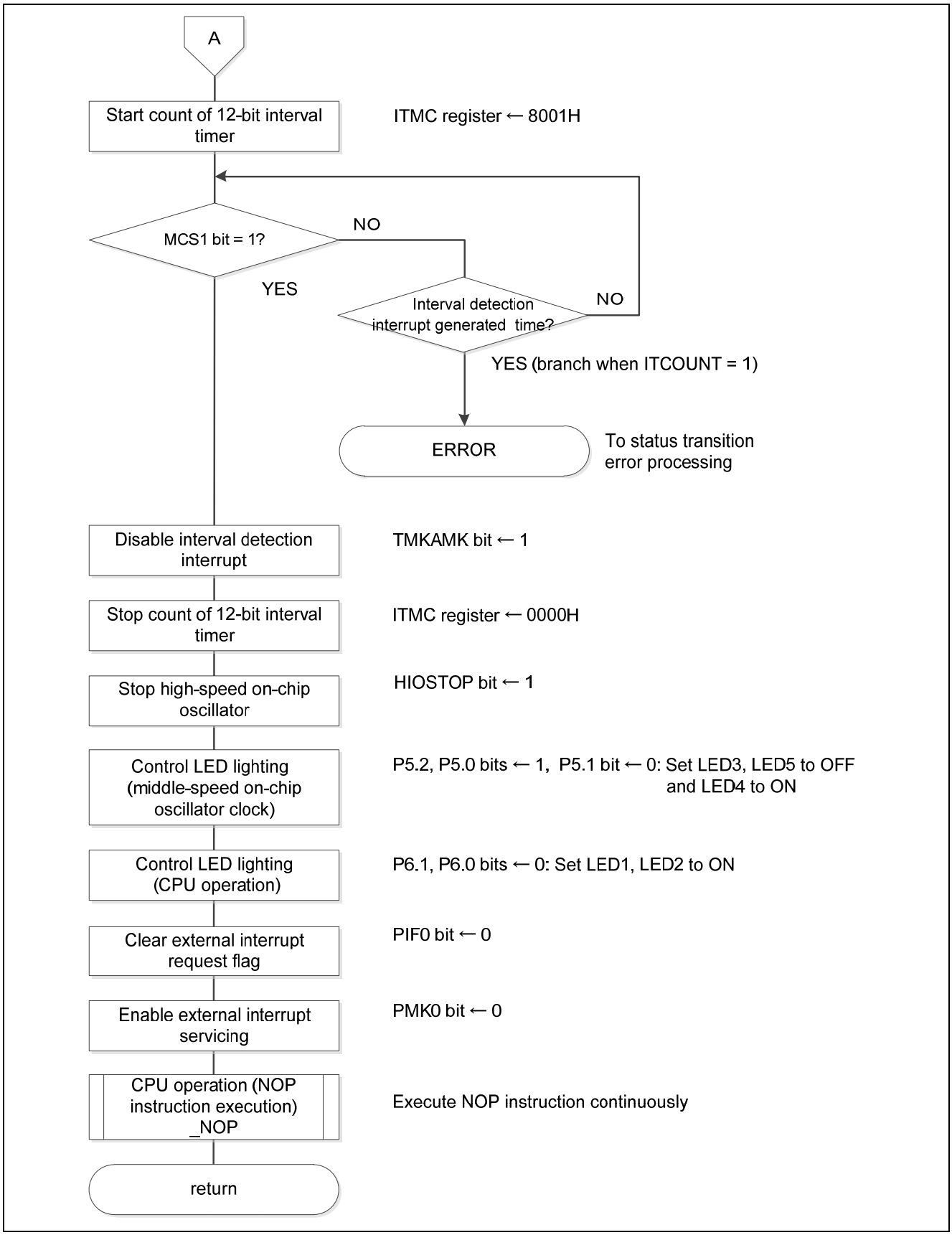

Figure 5.41 Status Transition BtoC(2/2)

5.6.30 Status Transition CtoD

Figure 5.42 and Figure 5.43 shows the status transition CtoD.

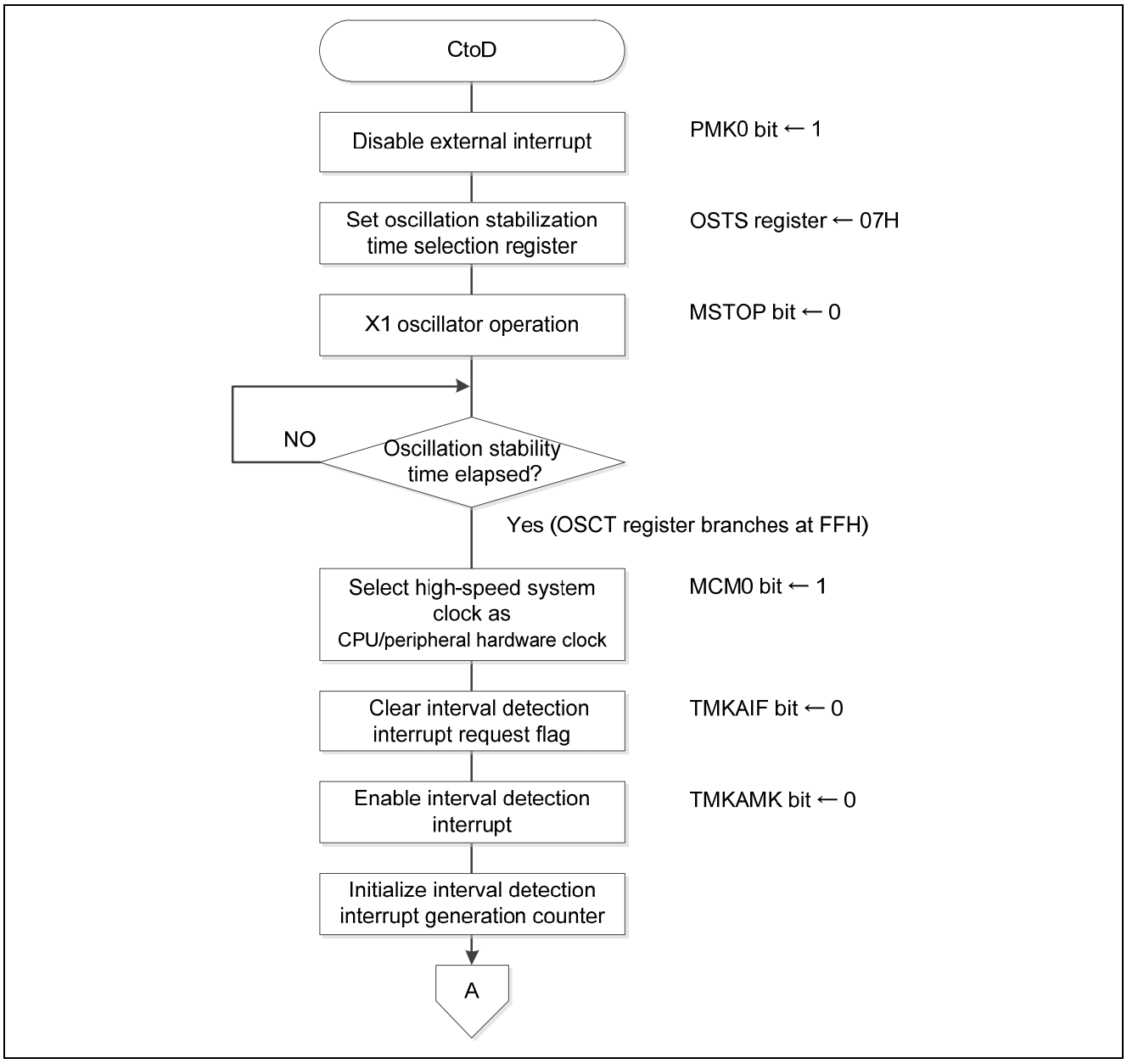

Figure 5.42 Status Transition CtoD(1/2)

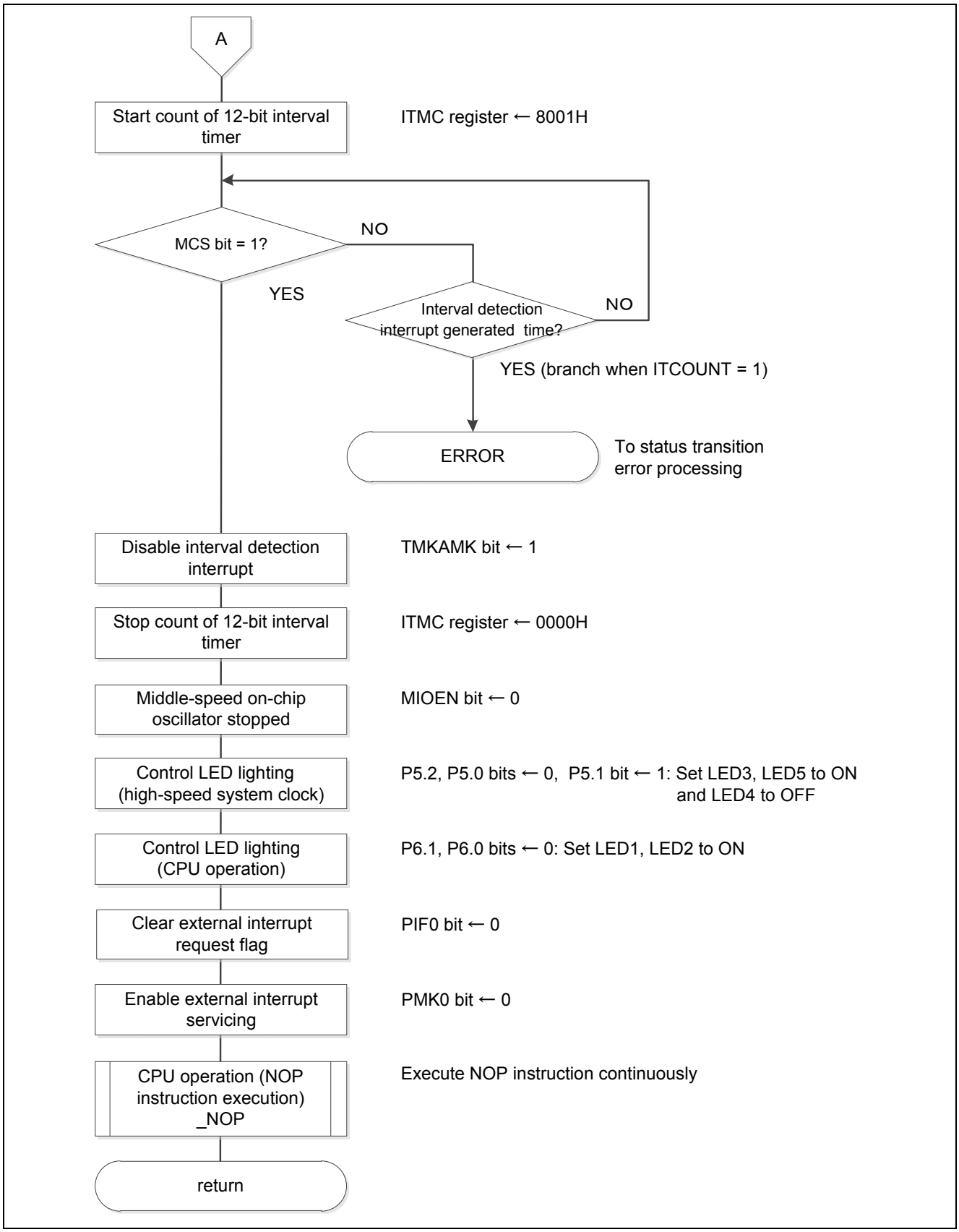

Figure 5.43 Status Transition CtoD(2/2)

5.6.31 Status Transition DtoF

Figure 5.44 and Figure 5.45 shows the status transition DtoF.

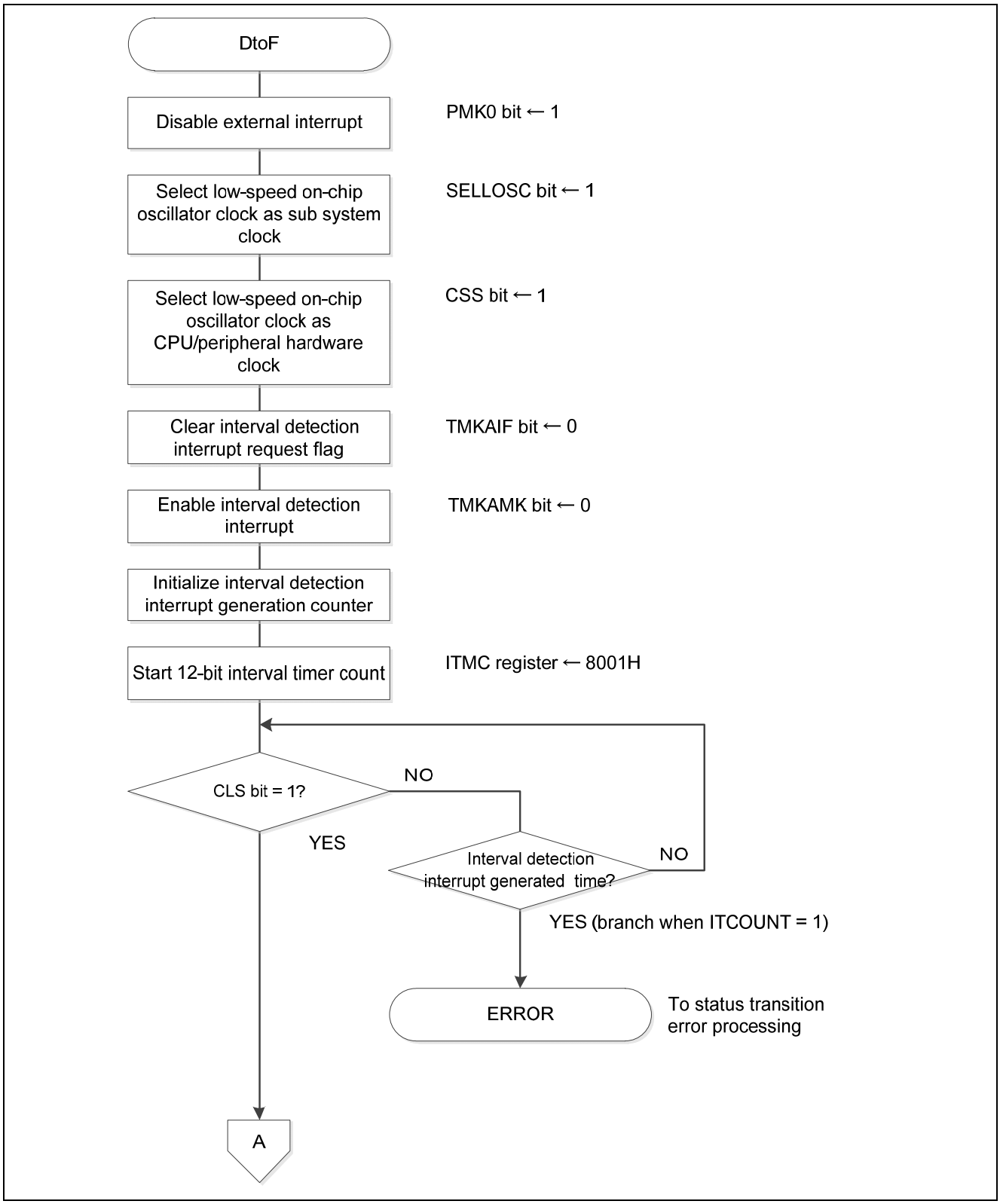

Figure 5.44 Status Transition DtoF(1/2)

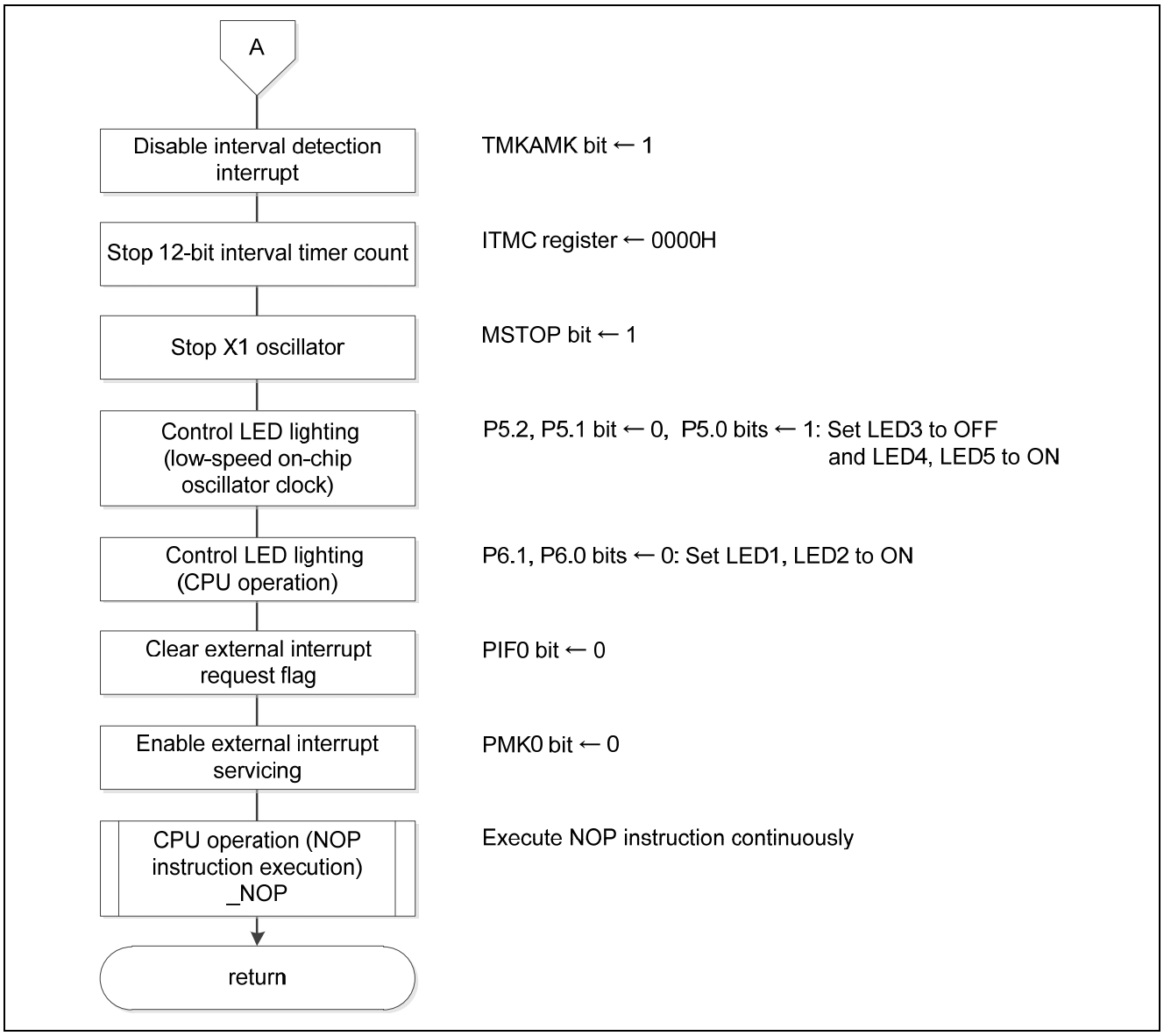

Figure 5.45 Status Transition DtoF(2/2)

5.6.32 Status Transition FtoD

Figure 5.46 and Figure 5.47 shows the status transition FtoD.

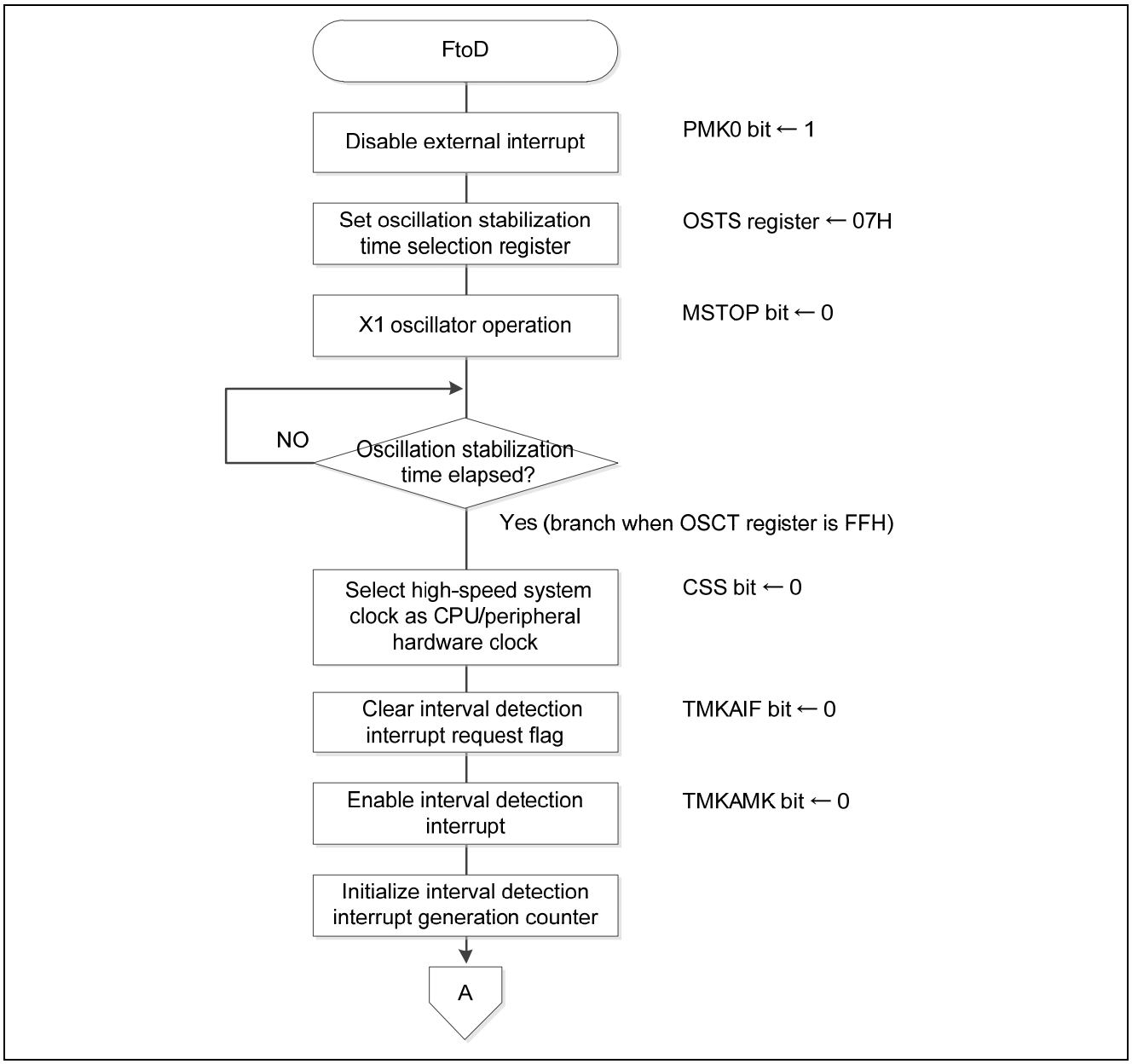

Figure 5.46 Status Transition FtoD(1/2)

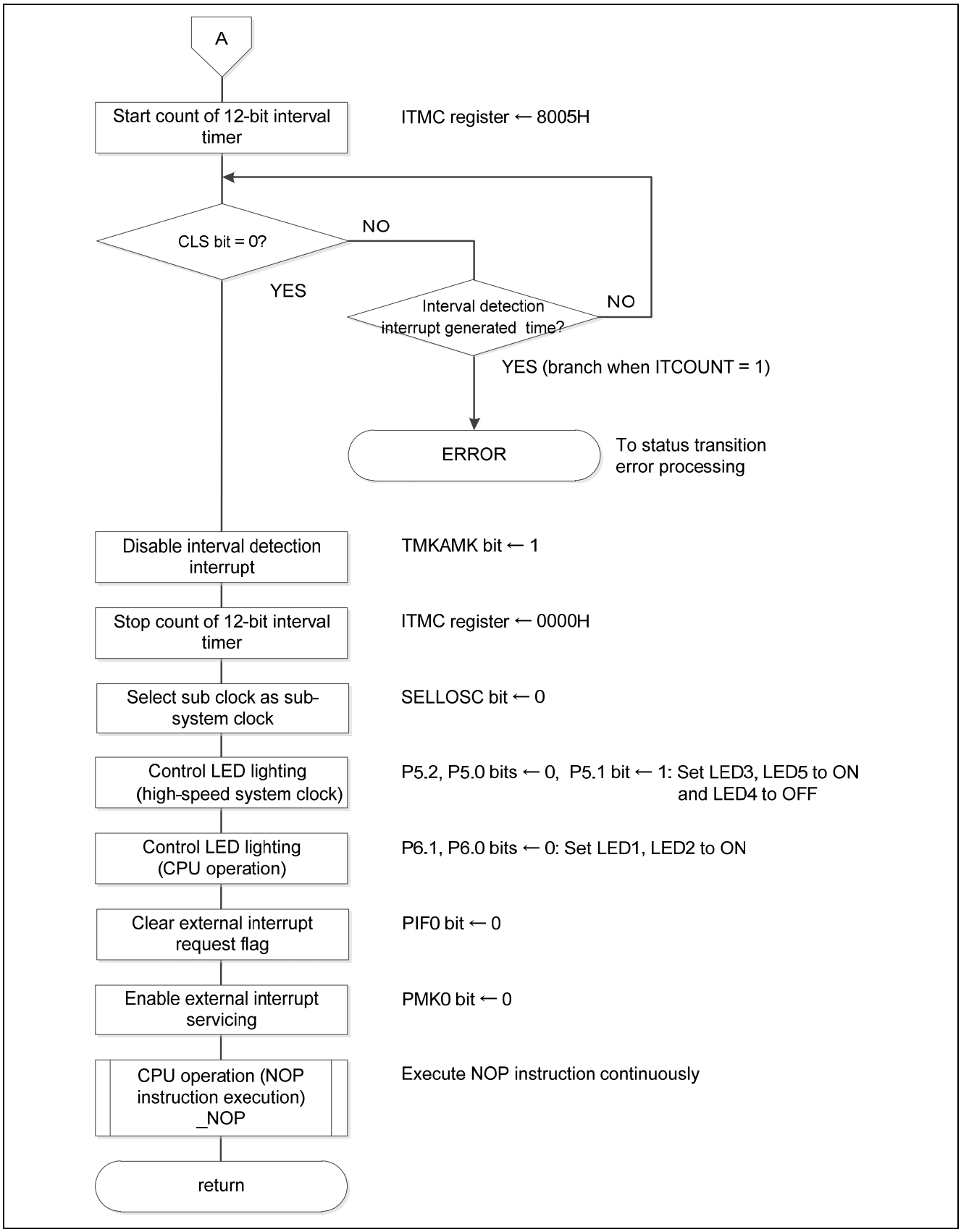

Figure 5.47 Status Transition FtoD(2/2)

5.6.33 Status Transition DtoC

Figure 5.48 and Figure 5.49 shows the status transition DtoC.

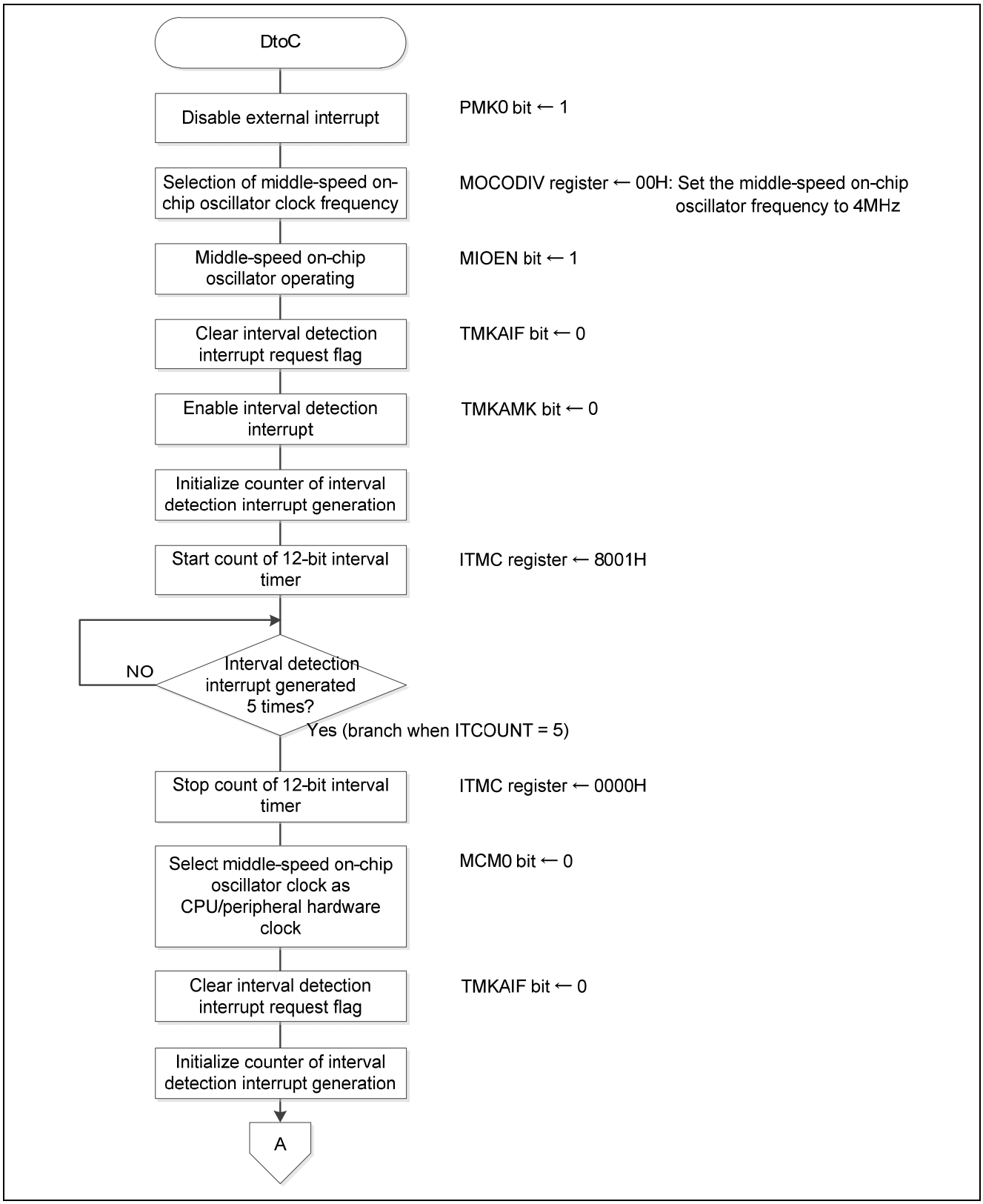

Figure 5.48 Status Transition DtoC(1/2)

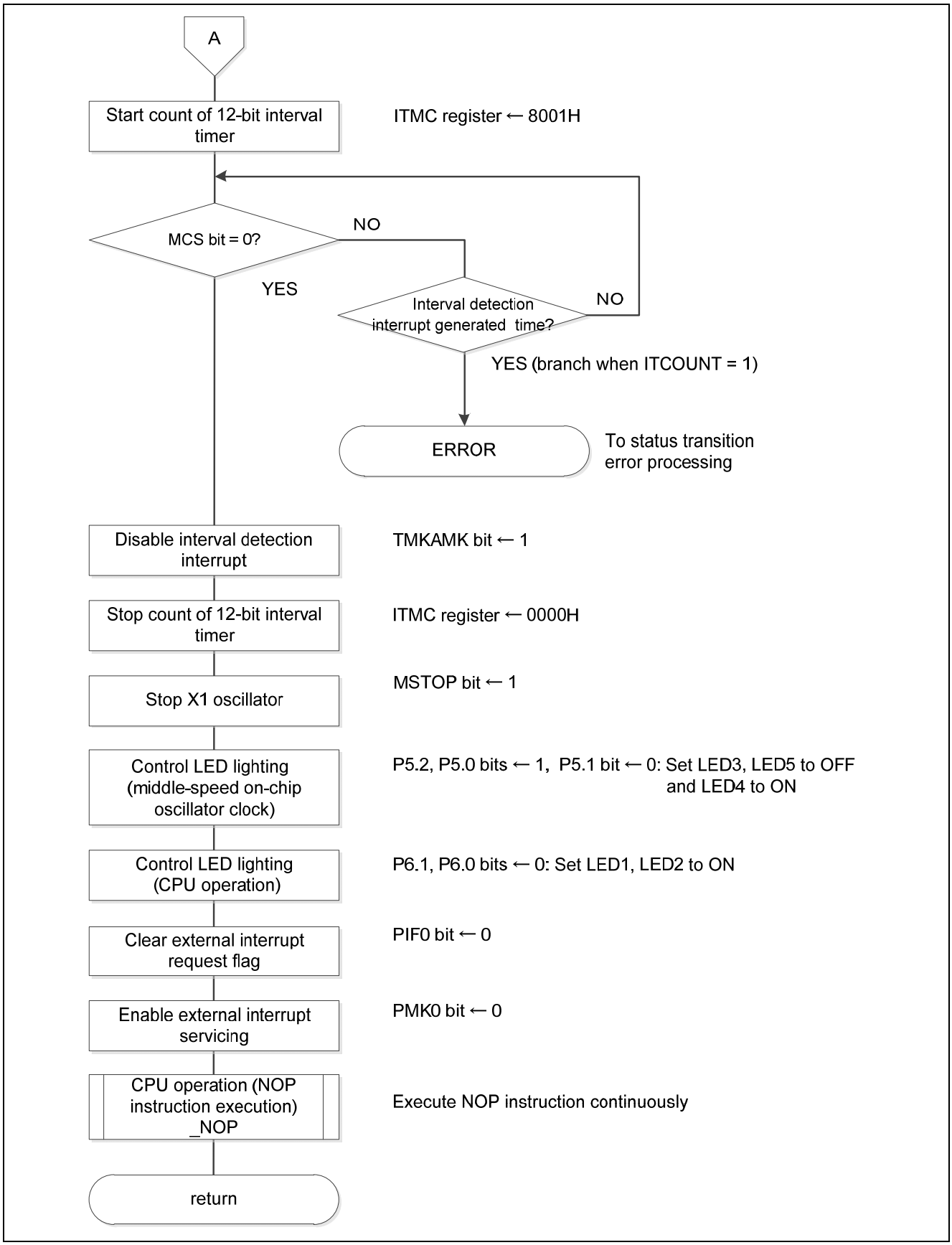

Figure 5.49 Status Transition DtoC(2/2)

5.6.34 Status Transition CtoJ

Figure 5.50 shows the status transition CtoJ.

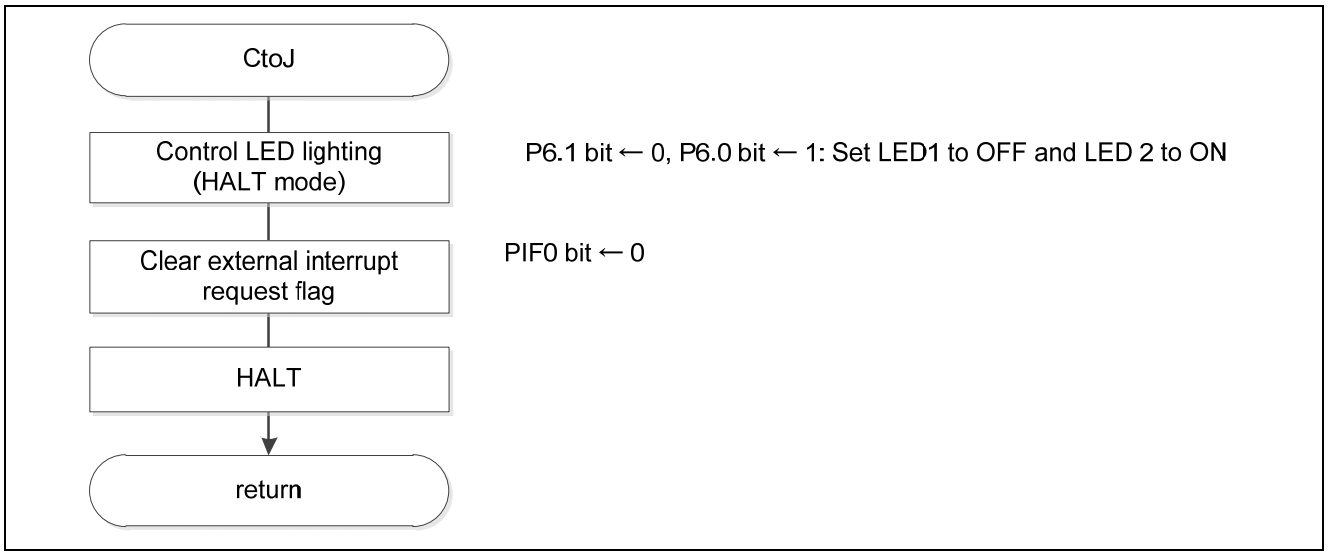

Figure 5.50 Status Transition CtoJ

5.6.35 Status Transition JtoC

Figure 5.51 shows the status transition JtoC.

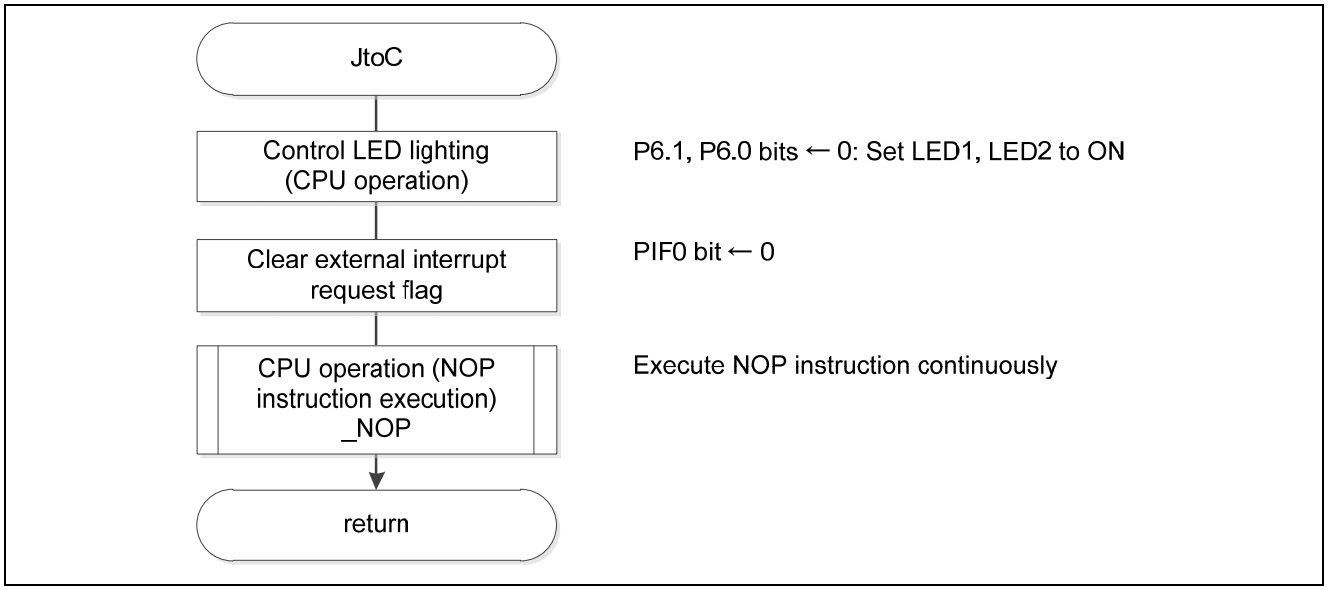

Figure 5.51 Status Transition JtoC

5.6.36 Status Transition CtoK

Figure 5.52 shows the status transition CtoK.

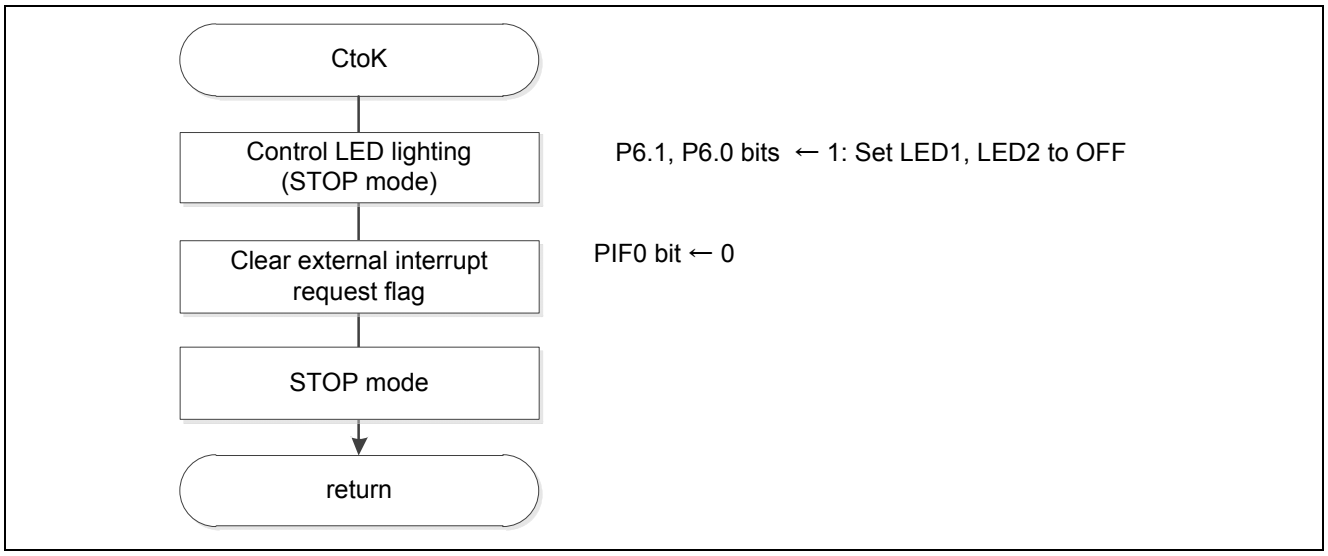

Figure 5.52 Status Transition CtoK

5.6.37 Status Transition KtoC

Figure 5.53 shows the status transition KtoC.

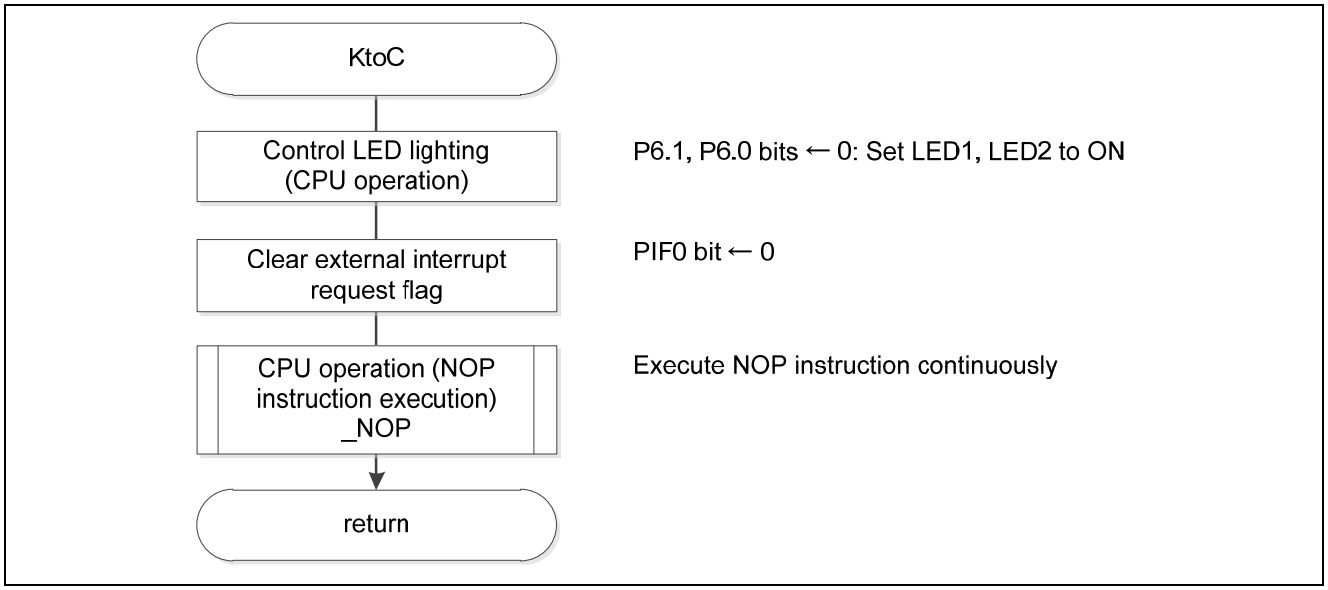

Figure 5.53 Status Transition KtoC

5.6.38 Status Transition CtoL

Figure 5.54 shows the status transition CtoL.

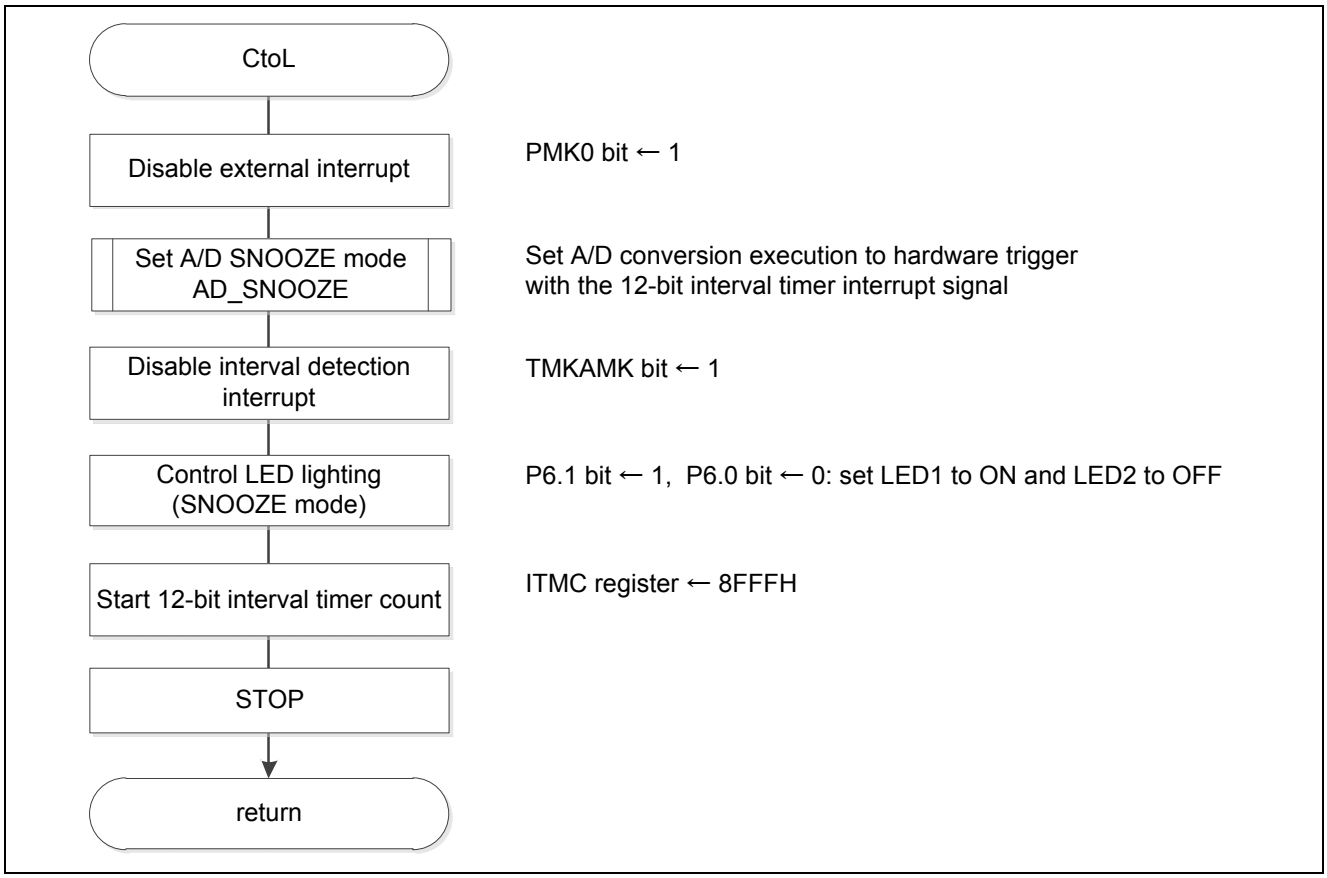

Figure 5.54 Status Transition CtoL

5.6.39 Status Transition LtoC

Figure 5.55 shows the status transition LtoC.

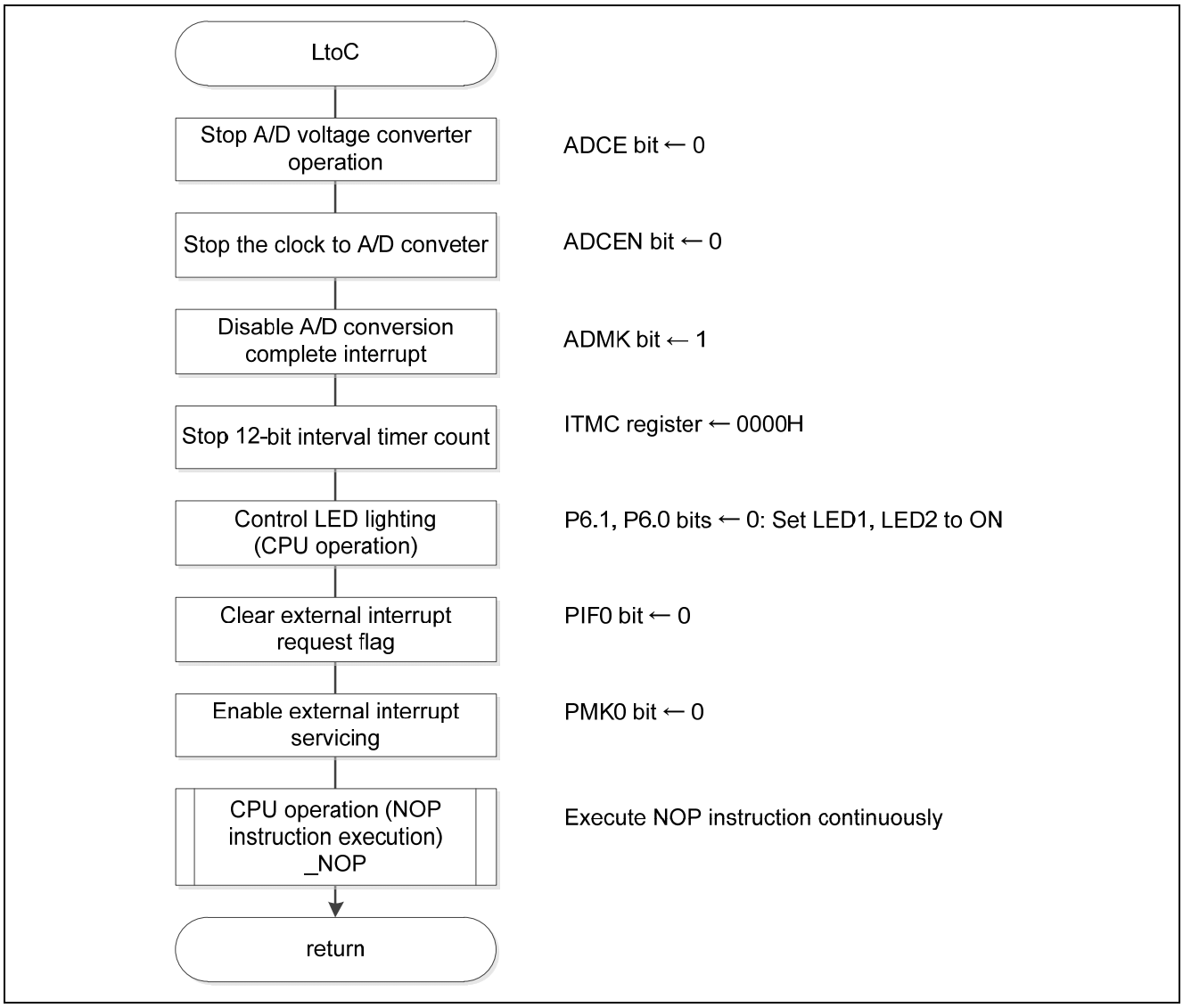

Figure 5.55 Status Transition LtoC

5.6.40 Status Transition CtoE

Figure 5.56 and Figure 5.57 shows the status transition CtoE.

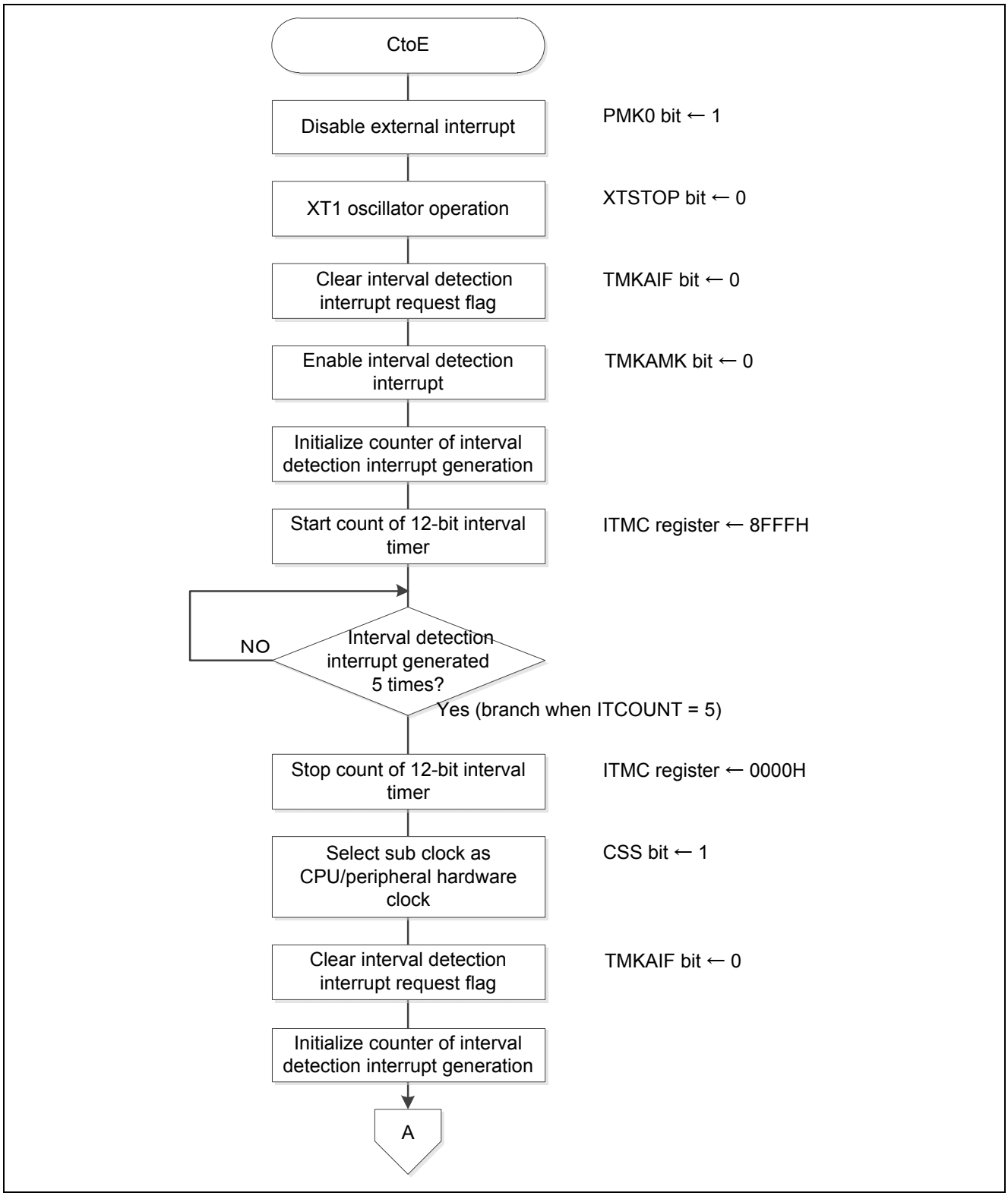

Figure 5.56 Status Transition CtoE(1/2)

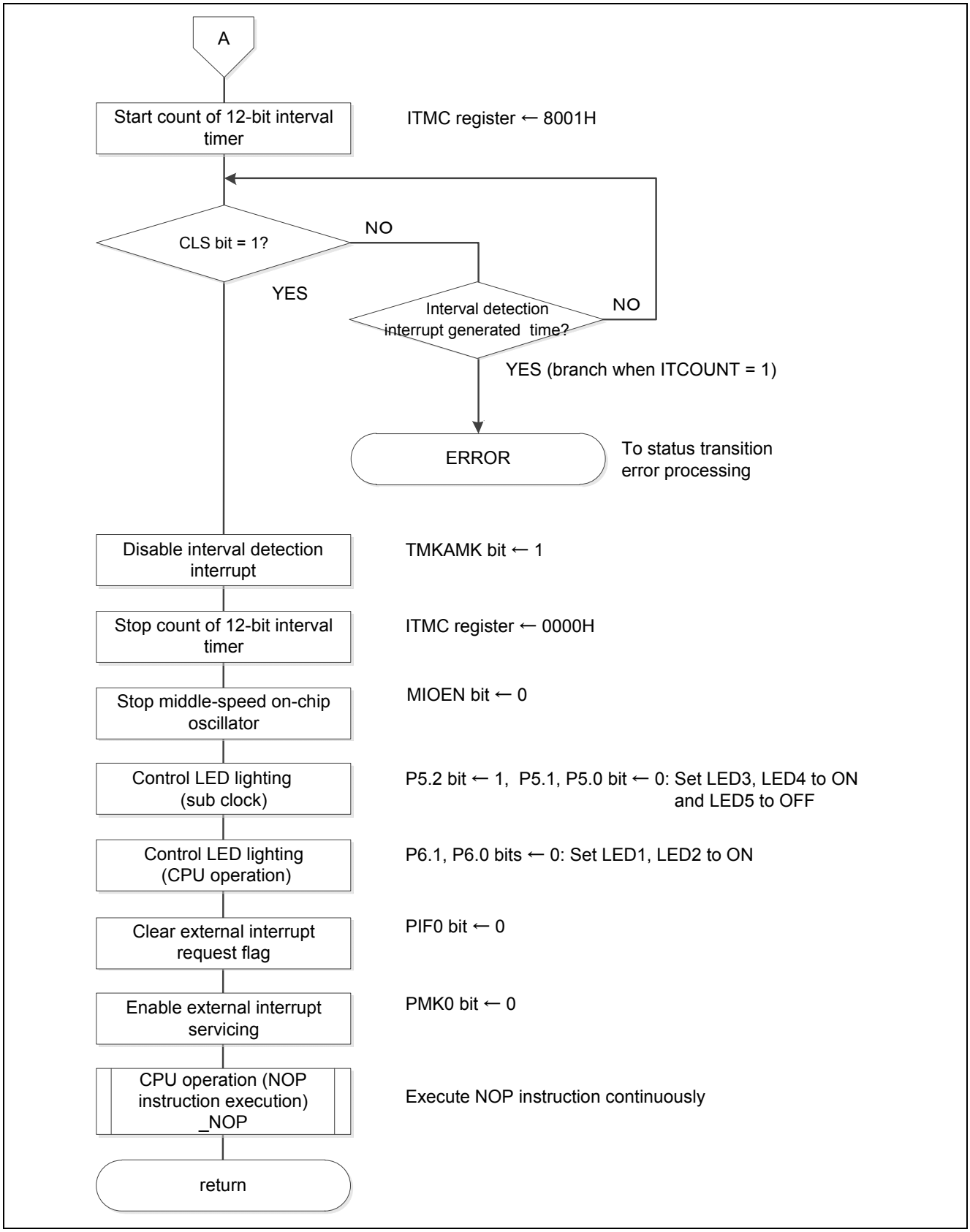

Figure 5.57 Status Transition CtoE(2/2)

5.6.41 Status Transition EtoC

Figure 5.58 and Figure 5.59 shows the status transition EtoC.

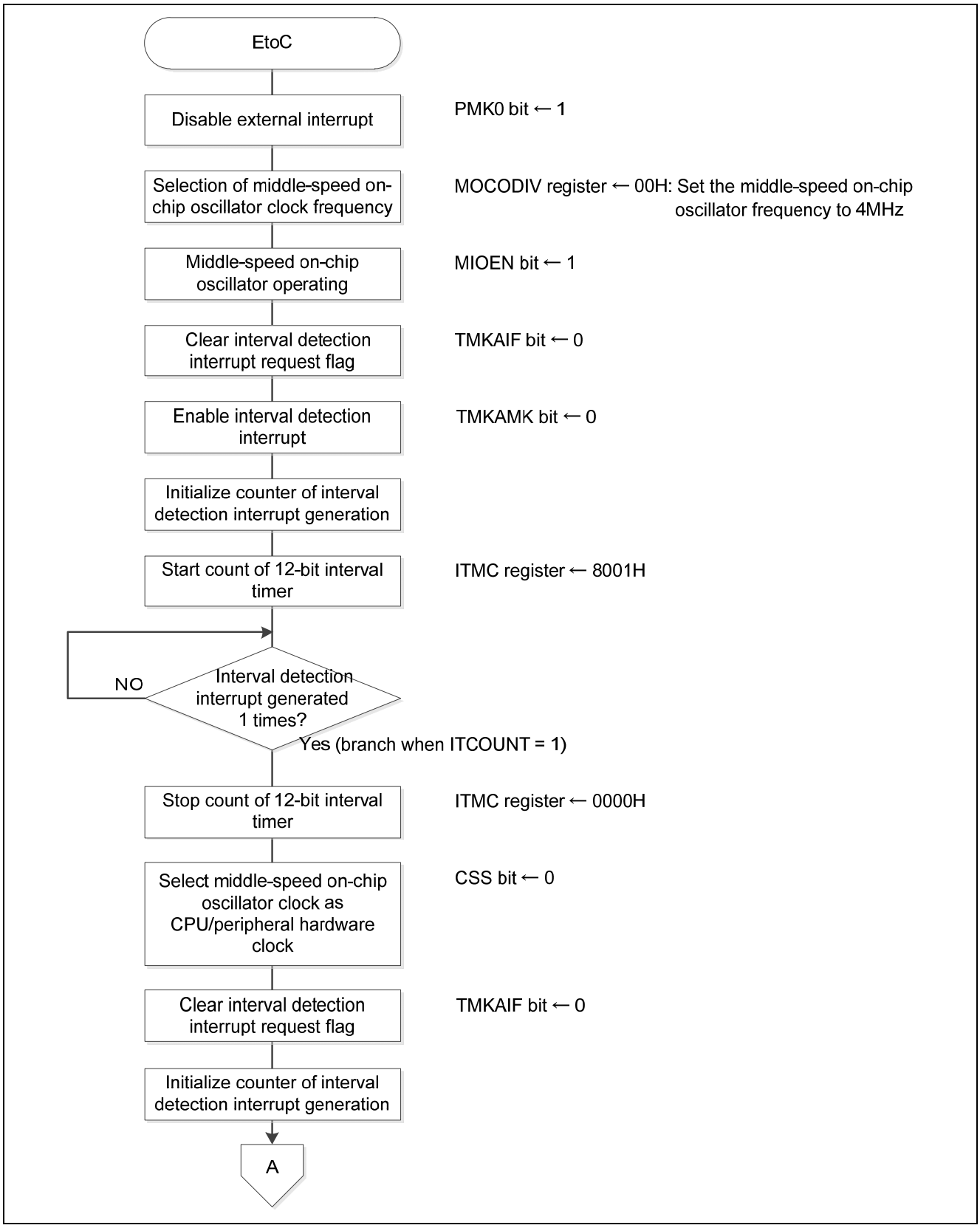

Figure 5.58 Status Transition EtoC(1/2)

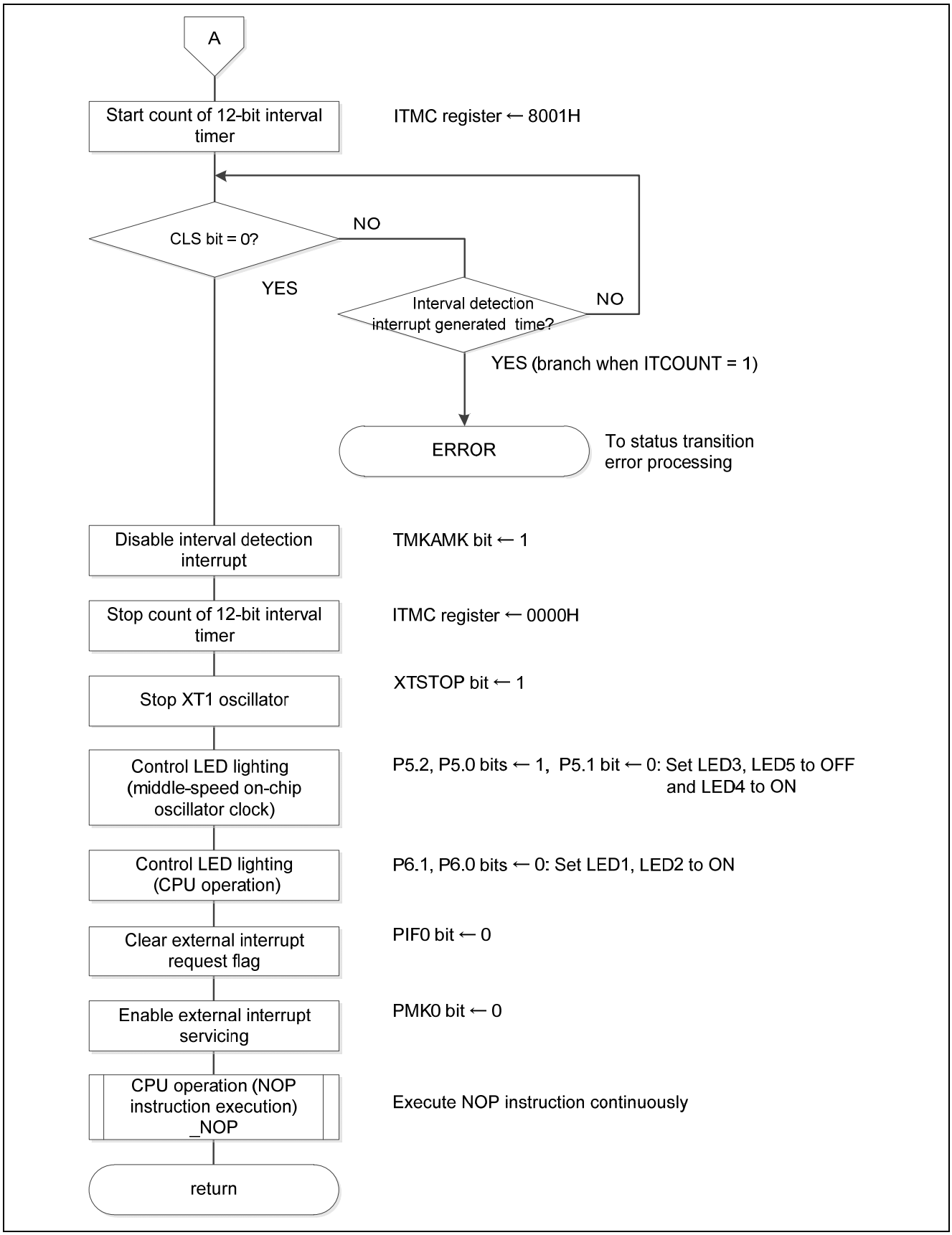

Figure 5.59 Status Transition EtoC(2/2)

5.6.42 Status Transition CtoF

Figure 5.60 and Figure 5.61 shows the status transition CtoF.

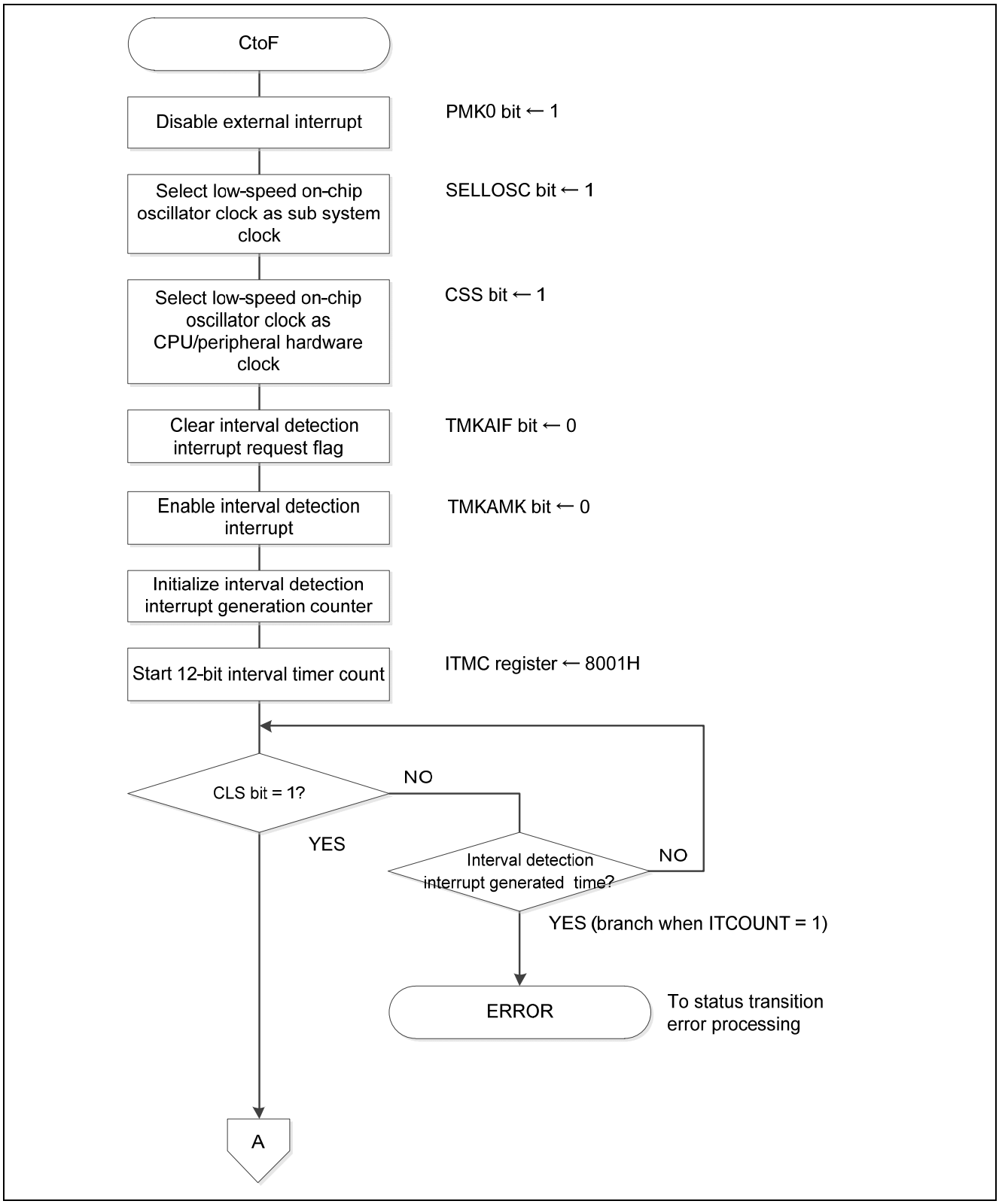

Figure 5.60 Status Transition CtoF(1/2)

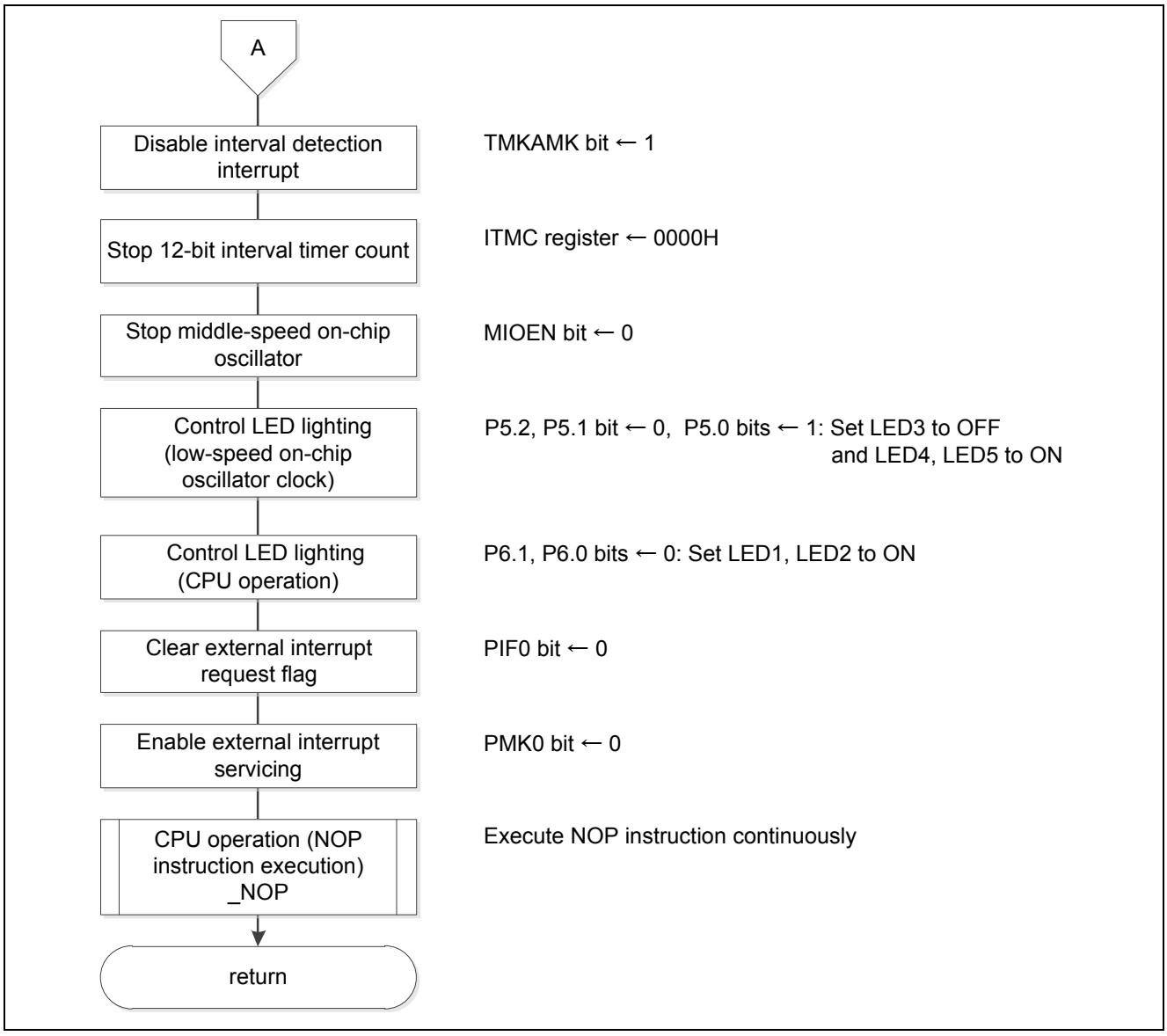

Figure 5.61 Status Transition CtoF(2/2)

5.6.43 Status Transition FtoC

Figure 5.62 and Figure 5.63 shows the status transition FtoC.

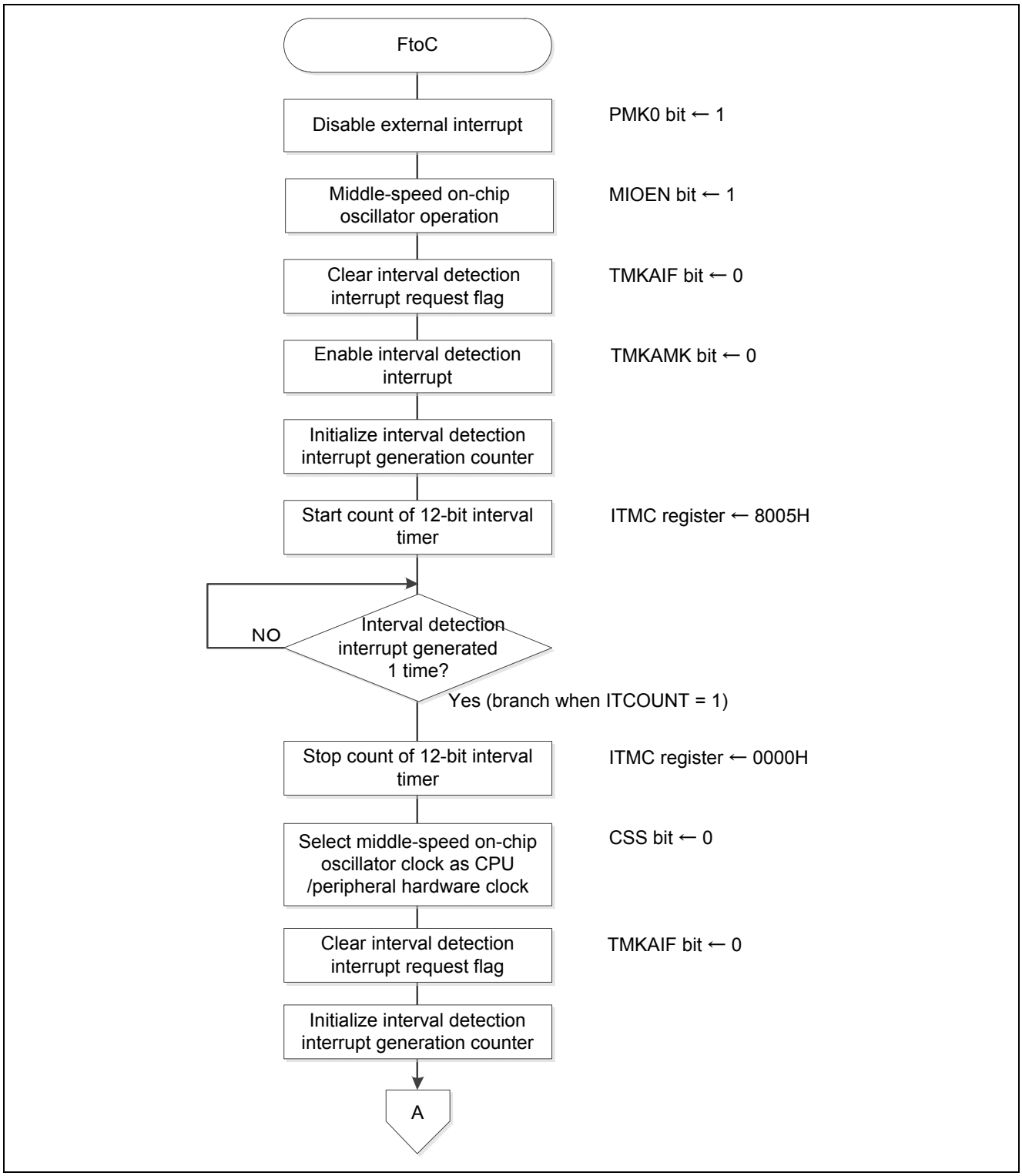

Figure 5.62 Status Transition FtoC(1/2)

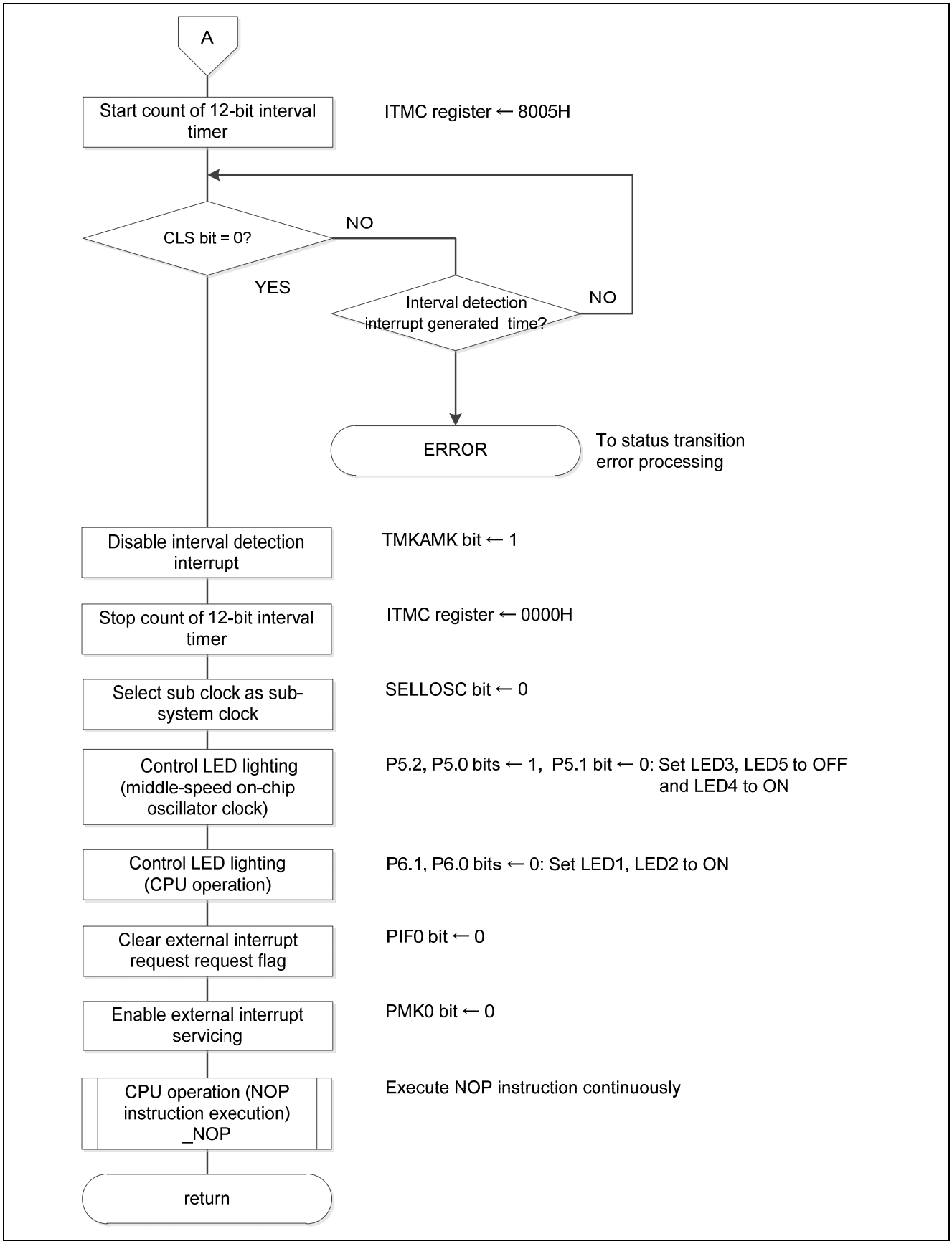

Figure 5.63 Status Transition FtoC(2/2)

5.6.44 Status Transition CtoB

Figure 5.64 and Figure 5.65 shows the status transition CtoB.

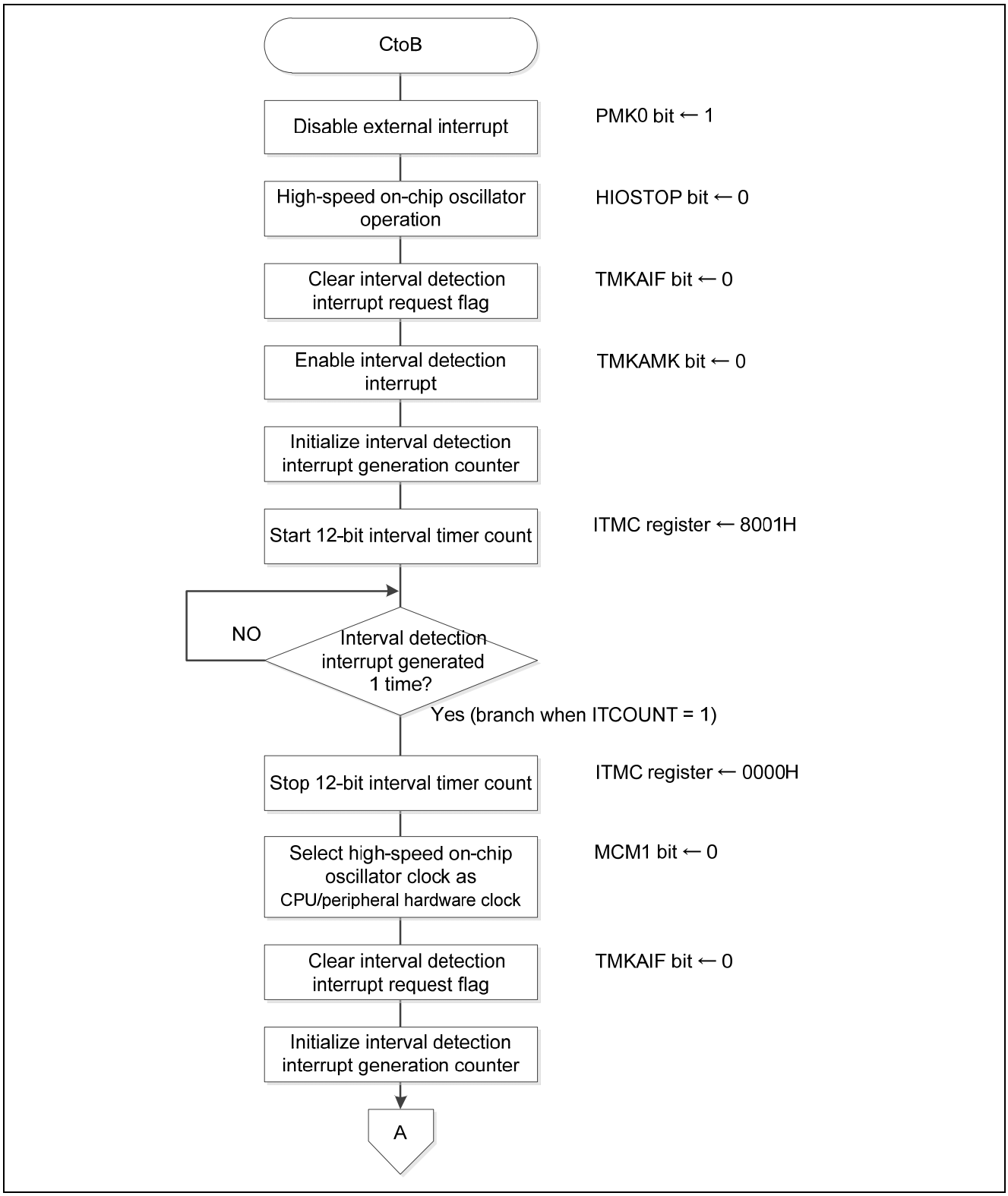

Figure 5.64 Status Transition CtoB(1/2)

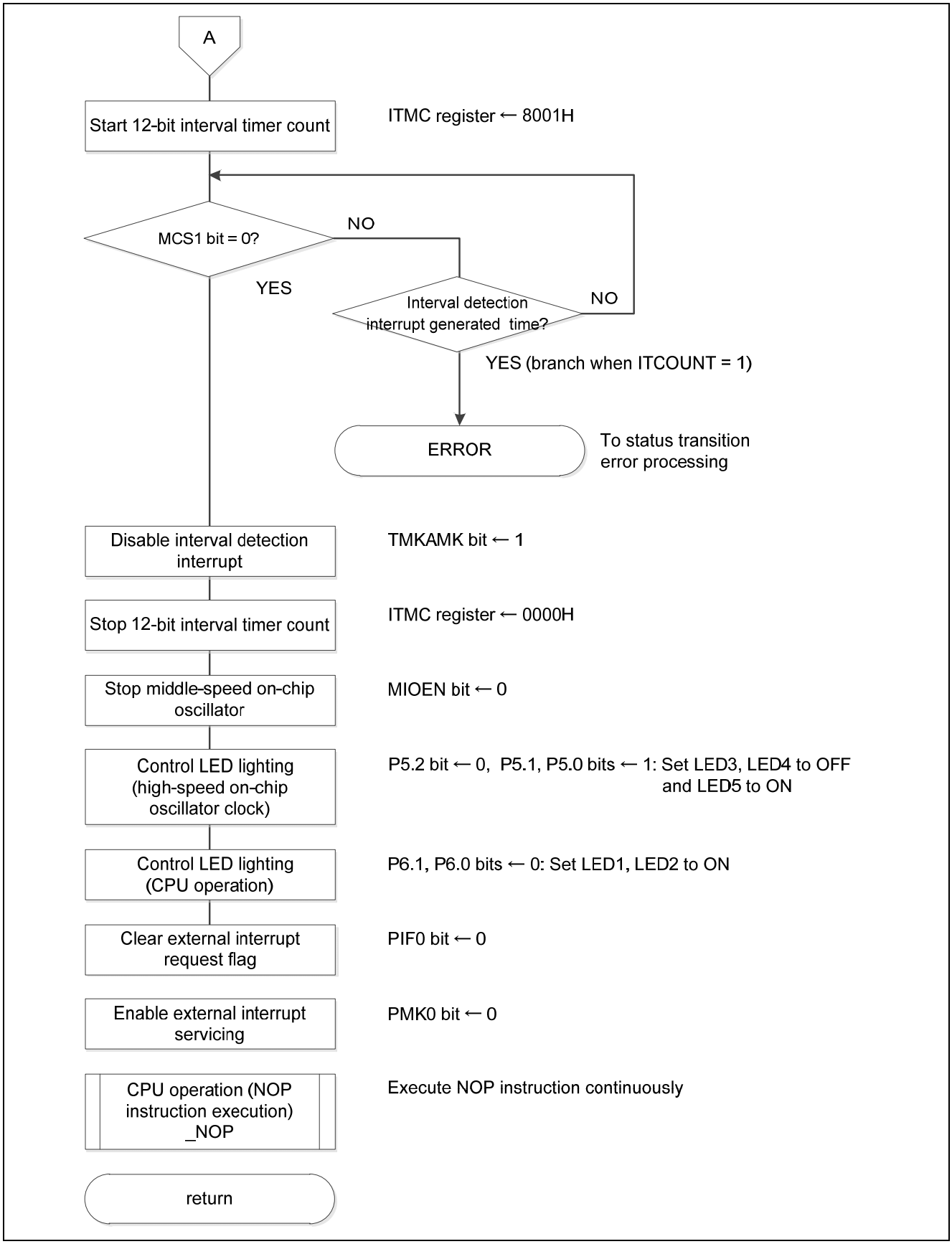

Figure 5.65 Status Transition CtoB(2/2)

5.6.45 Status Transition BtoF

Figure 5.66 and Figure 5.67 shows the status transition BtoF.

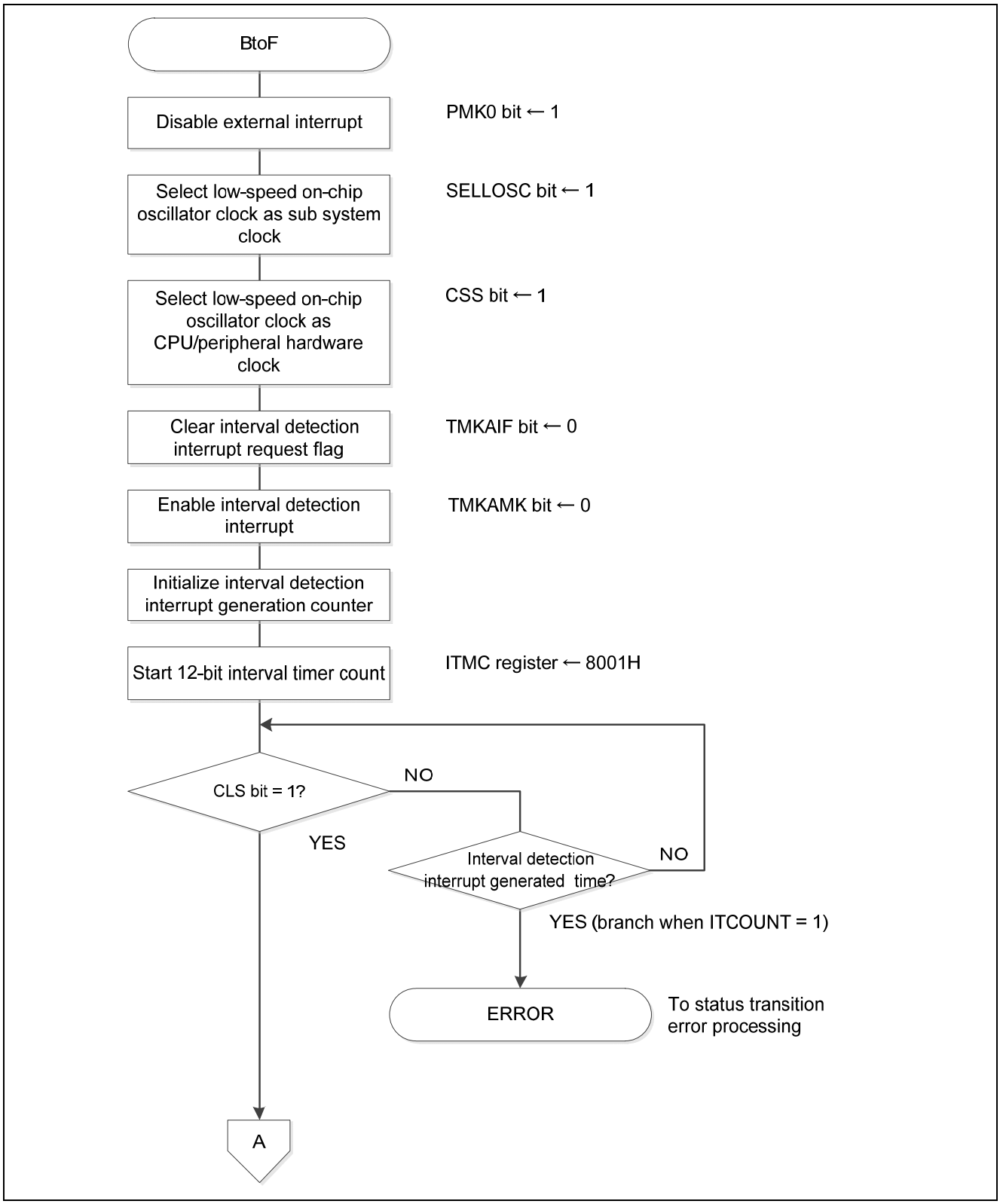

Figure 5.66 Status Transition BtoF

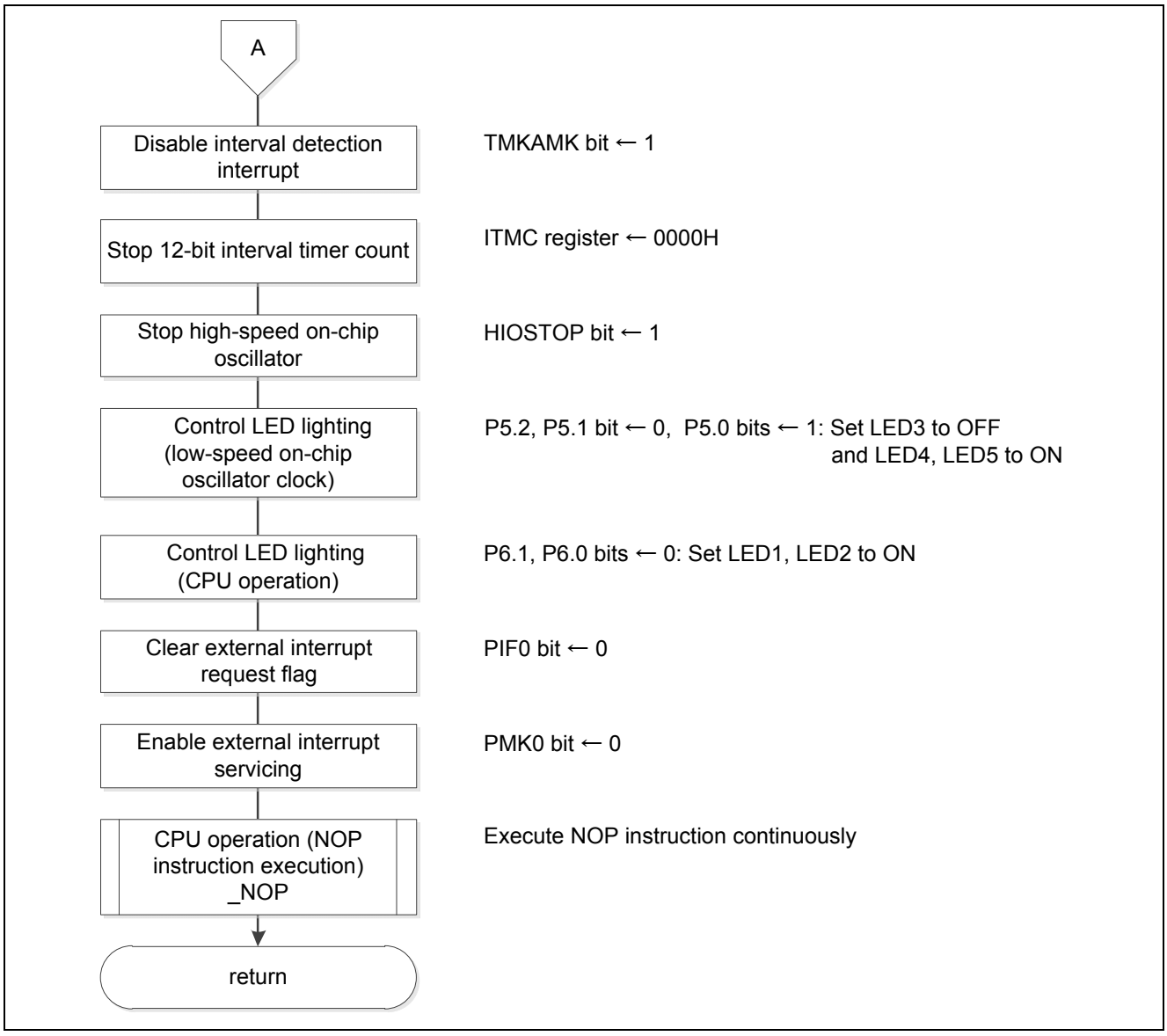

Figure 5.67 Status Transition BtoF

5.6.46 Status Transition FtoP

Figure 5.68 shows the status transition FtoP.

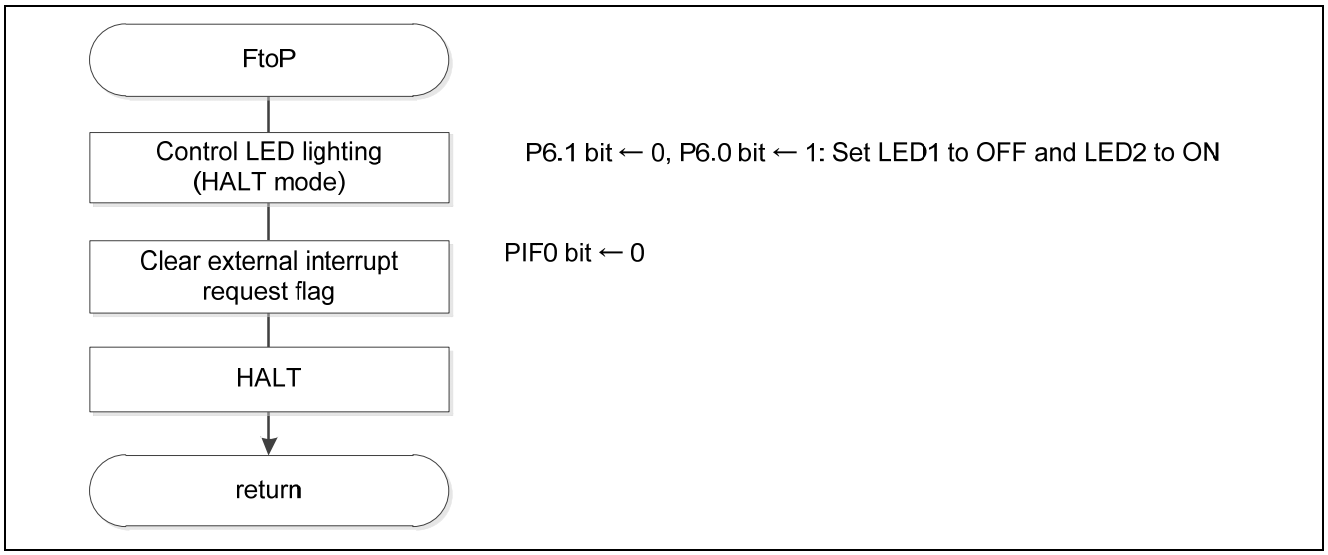

Figure 5.68 Status Transition FtoP

5.6.47 Status Transition PtoF

Figure 5.69 shows the status transition PtoF.

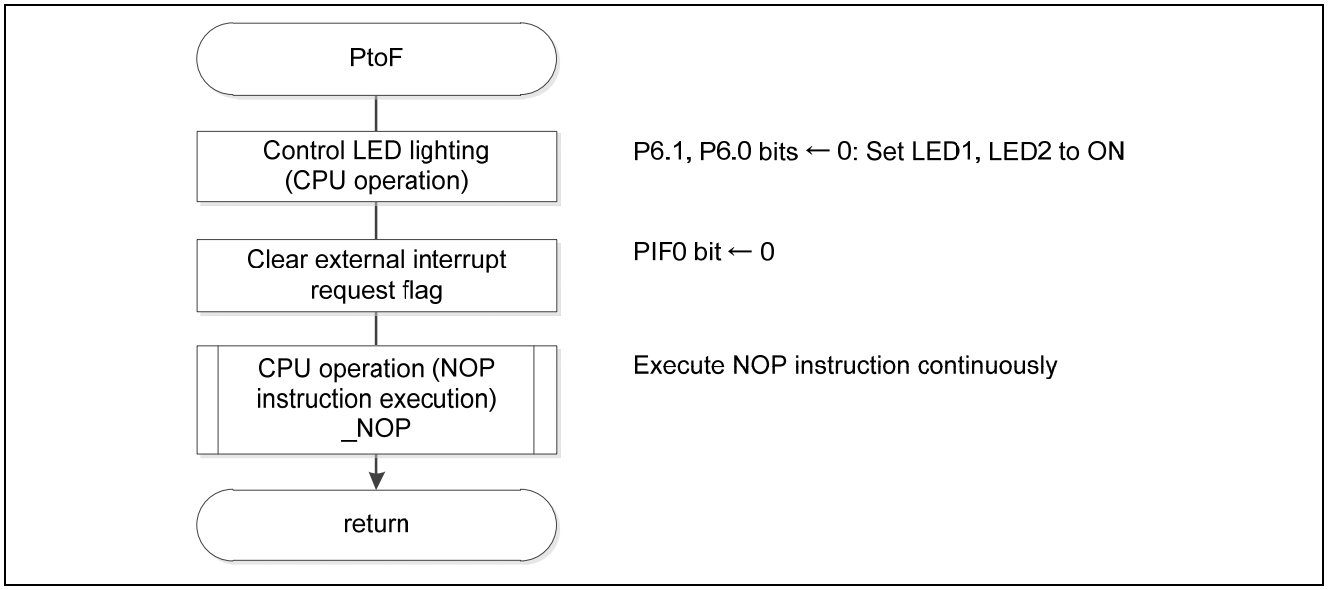

Figure 5.69 Status Transition PtoF

5.6.48 Status Transition FtoB

Figure 5.70 and Figure 5.71 shows the status transition FtoB.

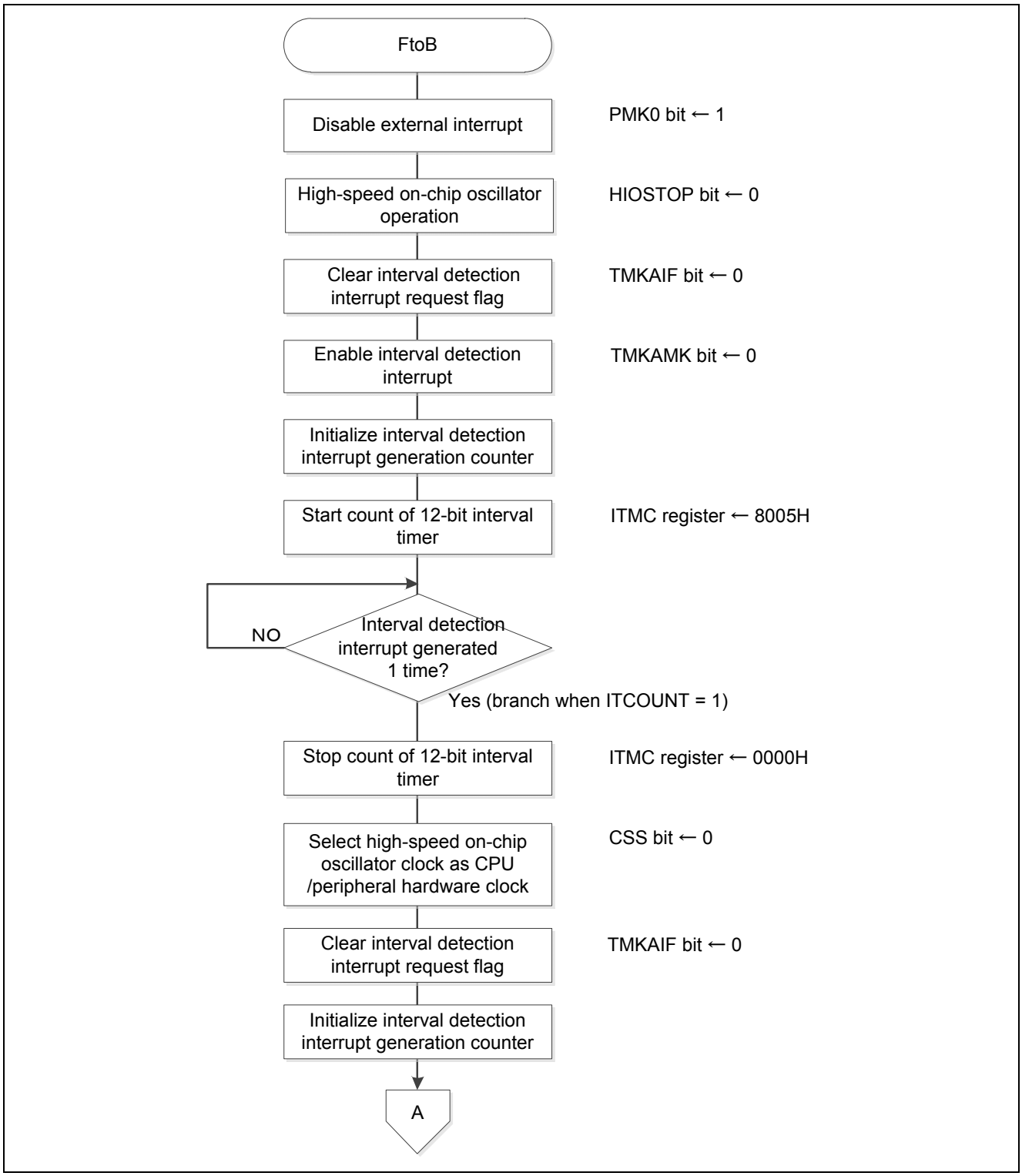

Figure 5.70 Status Transition FtoB(1/2)

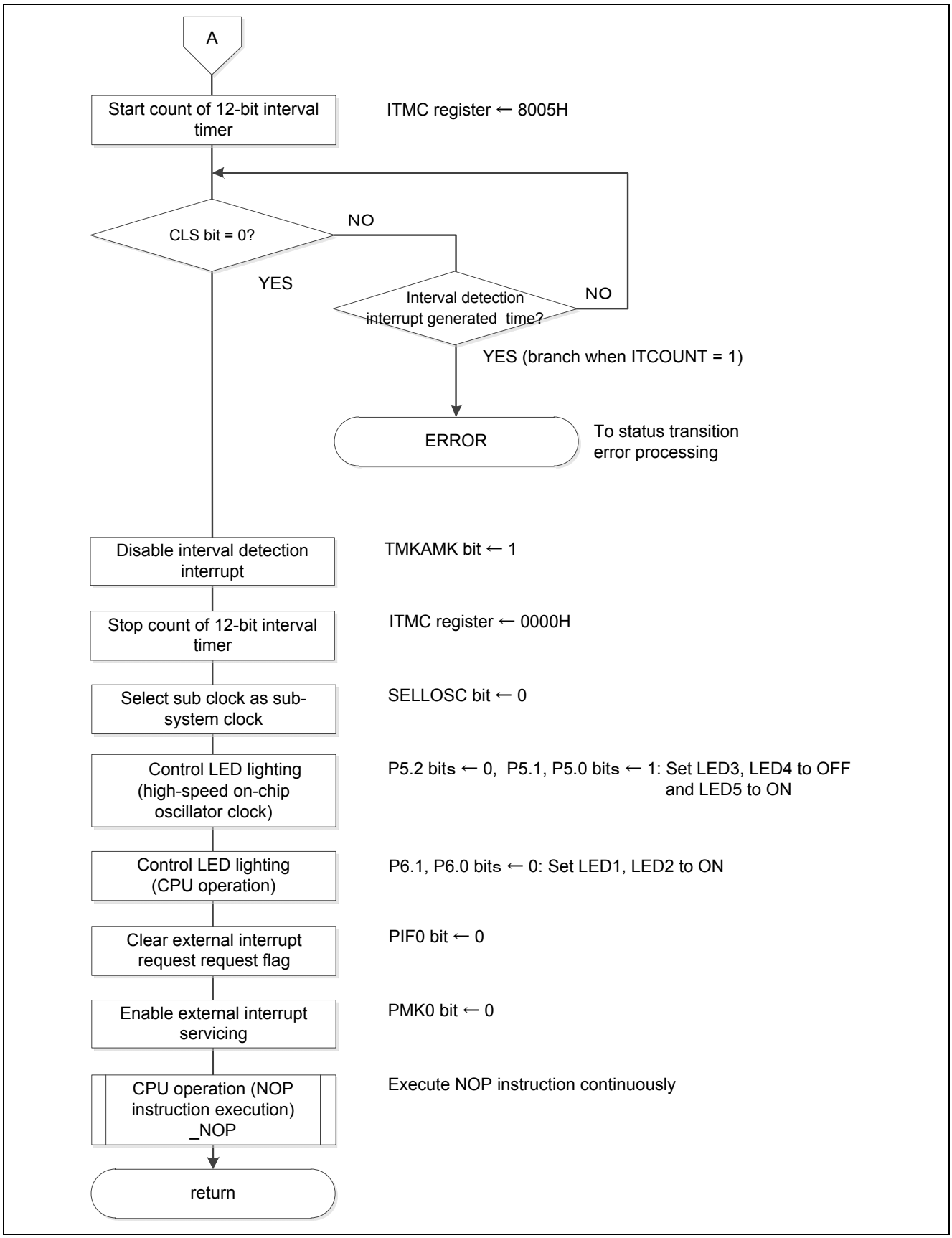

Figure 5.71 Status Transition FtoB(2/2)

5.6.49 Status Transition End Processing

Figure 5.72 shows the flowchart for status transition end processing.

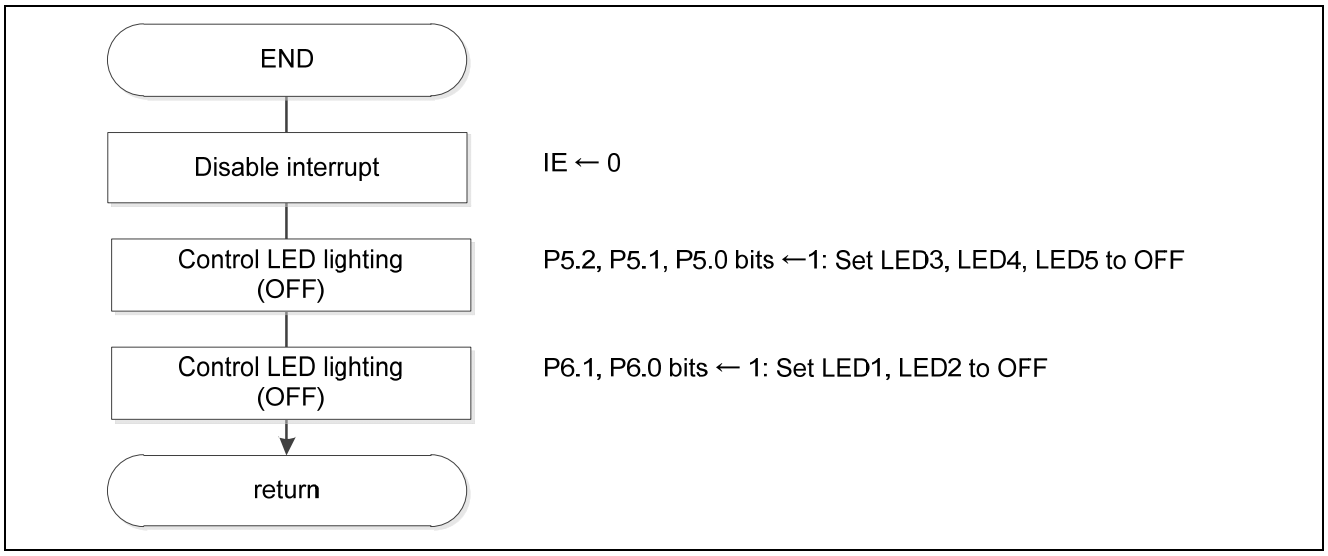

Figure 5.72 Status Transition End Processing

5.6.50 External Interrupt Servicing

Figure 5.73 shows the flowchart for external interrupt servicing.

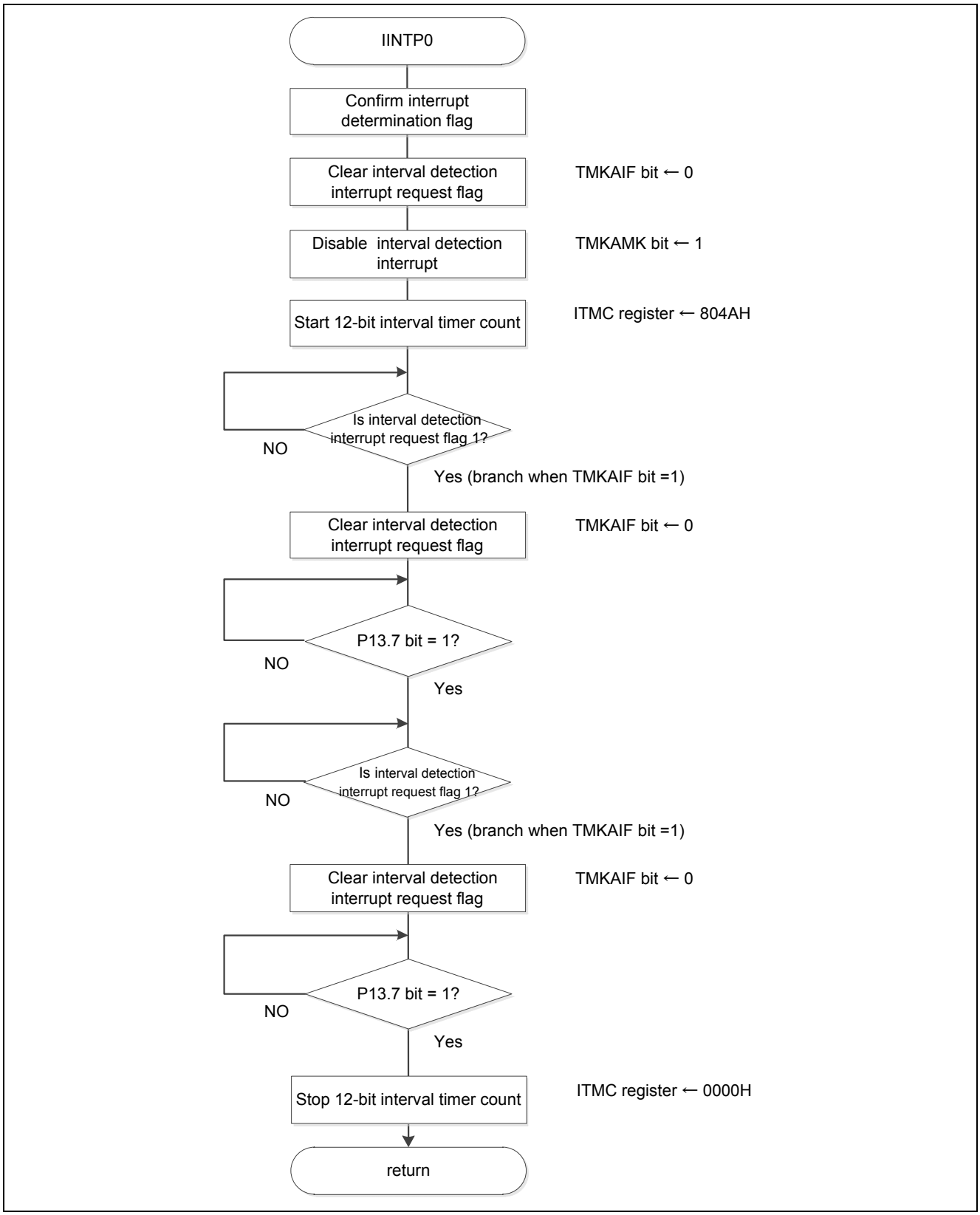

Figure 5.73 External Interrupt Servicing

5.6.51 12-bit Interval Timer Interrupt Servicing

Figure 5.74 shows the flowchart for 12-bit interval timer interrupt servicing.

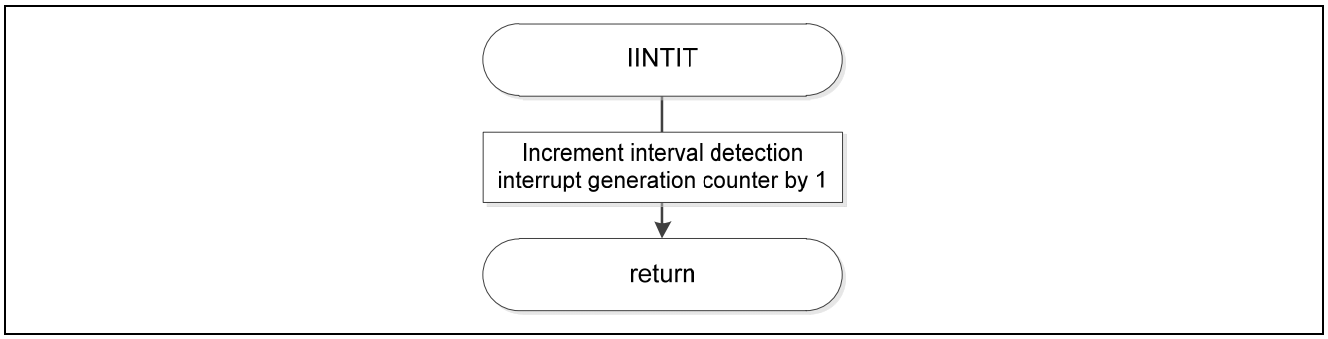

Figure 5.74 12-bit Interval Timer Interrupt Servicing

5.6.52 A/D Conversion Completion Interrupt Servicing

Figure 5.75 shows the flowchart for A/D conversion completion interrupt servicing.

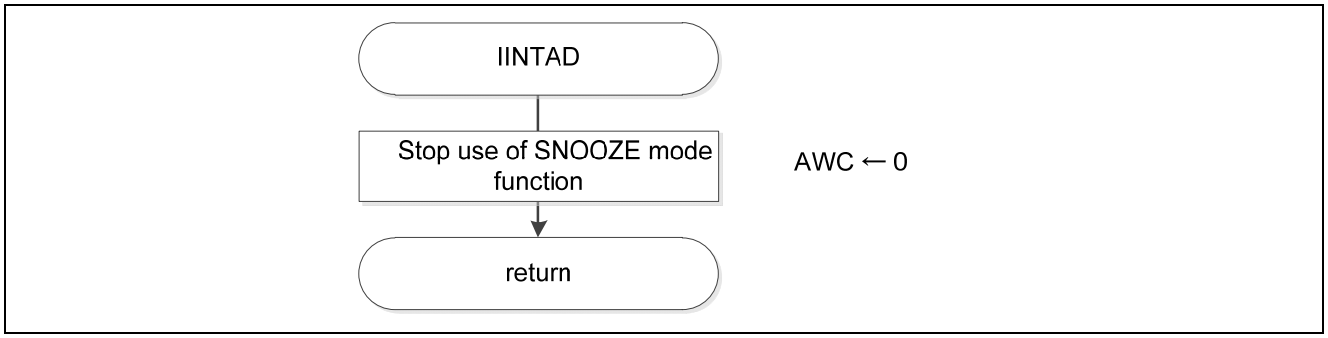

Figure 5.75 A/D Conversion Completion Interrupt Servicing

6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

7. Reference Documents

RL78/I1D User's Manual: Hardware Rev.2.10 (R01UH0474J)

RL78 Family User's Manual: Software Rev.1.00 (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/

REVISION HISTORY	RL78/I1D CPU Clock Changing and Standby Settings (Assembly) CC-RL

Rev.	Date	Description	
		Page	Summary
1.00	Aug. 29, 2016	—	First edition issued

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

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

1. Handling of Unused Pins

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

   ¾ The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

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

   ¾ The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   ¾ The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

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

   ¾ When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

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

   ¾ The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

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

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

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

RENESAS

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

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

California Eastern Laboratories, Inc.
4590 Patrick Henry Drive, Santa Clara, California 95054-1817, U.S.A.
Tel: +1-408-919-2500, Fax: +1-408-988-0279

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

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

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

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

Colophon 5.0