RL78/G13

Clock Generator (Clock Switching)

R01AN0811EJ0100
Rev. 1.00
Feb. 08, 2012

Introduction

This application note explains how to use the clock generator of the RL78/G13.

The clock from the clock generator is switched when a switch is pressed. The clock generator uses the high-speed on-chip oscillator clock, (32 MHz), X1 oscillation clock (20 MHz), or XT1 oscillation clock (32.768 kHz) as the CPU/peripheral hardware clock (f_{CLK}).

Target Device

RL78/G13 (40 pins or more)

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

The sample code covered in this application note switches its operating clock when the switch on the target board is pressed according to the sequence below.

(1) High-speed on-chip oscillator clock (32 MHz) → X1 oscillation clock (20 MHz)
(2) X1 oscillation clock (20 MHz) → XT1 oscillation clock (32.768 kHz)
(3) XT1 oscillation clock (32.768 kHz) → High-speed on-chip oscillator clock (32 MHz)
Subsequently, steps (1) to (3) are repeated.

The sample code takes the following actions according to its operating state:

- When the high-speed on-chip oscillator clock (HOCO clock) operates: Stops the X1 oscillation clock.
- When the X1 oscillation clock operates: Stops the HOCO clock.
- When the XT1 oscillation clock operates: Stops the X1 oscillation clock and the HOCO clock.

The XT1 oscillation clock is always generated.

The sample code also changes the LED blinking period on the target board as shown below according to the operating clock. This allows the operating clock to be visually checked.

LED blinking period of the HOCO clock (32 MHz): 0.5 seconds

LED blinking period of the X1 oscillation clock (20 MHz): 1 seconds

LED blinking period of the XT1 oscillation clock (32.768 kHz): 2 seconds

Table 1.1 summarizes the peripheral functions to be used and their uses. Figure 1.1 shows the outline of the clock switching.

Table 1.1 Peripheral Functions to be Used and their Uses

Peripheral Function	Use
Clock generator	Generates oscillation clocks and switches the operating clocks.
External interrupt input (INTP0)	Detects the press of the switch.
Timer array unit 0 channel 0	Generates the timing signal to determine the LED blinking period.
12-bit interval timer	Generates the wait time to deal with chattering.
P62	Generates the output signal to the LED.

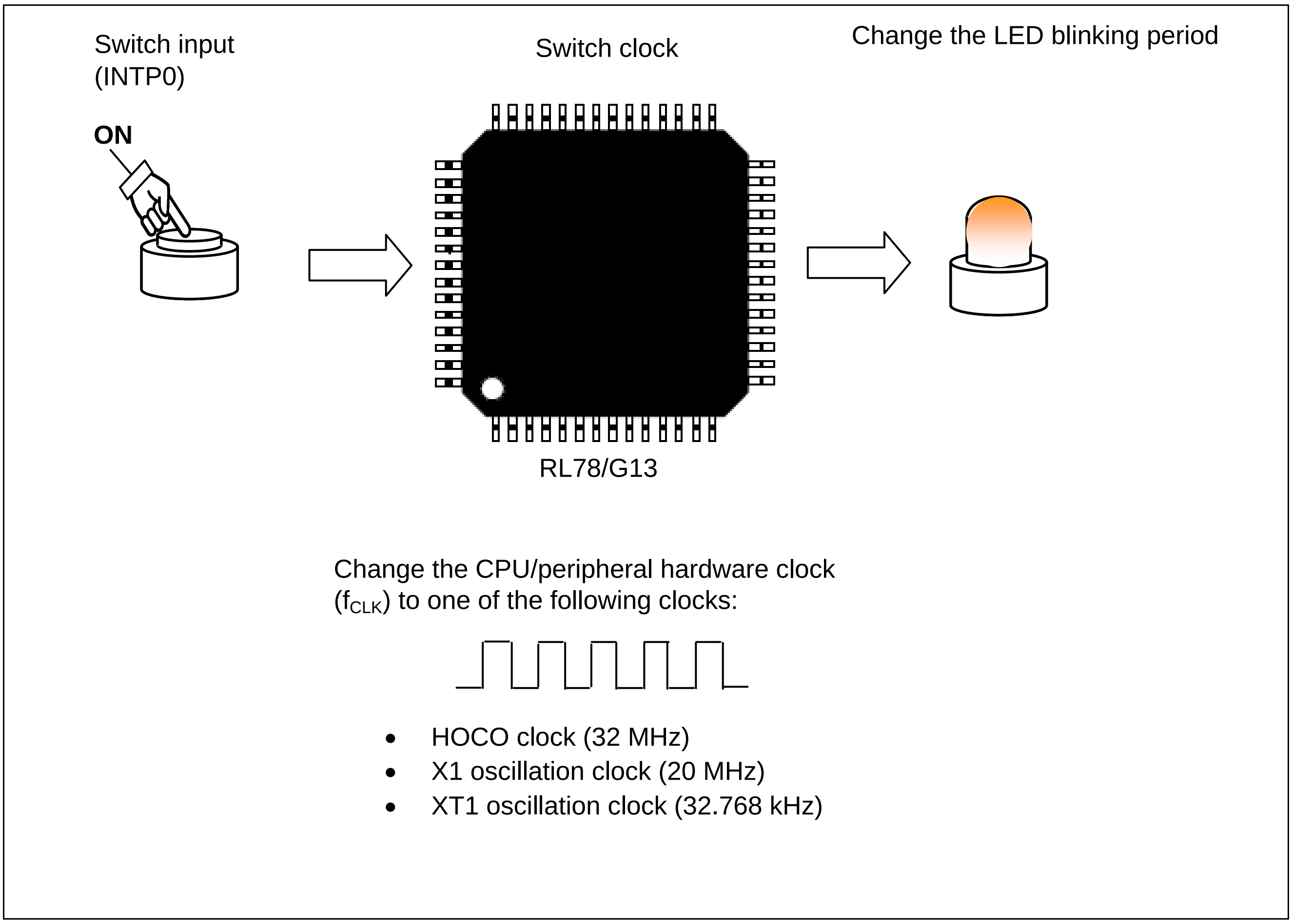

Figure 1.1 Outline of Clock Switching

2. Operation Check Conditions

The sample code contained in this application note has been checked under the conditions listed in the table below.

Table 2.1 Operation Check Conditions

Item	Description
Microcontroller used	RL78/G13 (R5F100LEA)
Operating frequency	• CPU/peripheral hardware clock: Switches the operating clock when the switch on the target board is pressed. When the HOCO clock is selected: 32 MHz When the X1 oscillation clock is selected: 20 MHz When the XT1 oscillation clock is selected: 32.768 kHz
Operating voltage	5.0 V (can run on a voltage range of 2.9 V to 5.5 V.) LVD operation (V_{LVI}): Reset mode 2.81 V (2.76 V to 2.87 V)
Integrated development environment	CubeSuite + V1.00.01 from Renesas Electronics Corp.
C compiler	CA78K0R V1.20 from Renesas Electronics Corp.
Board used	RL78/G13 target board (QB-R5F100LE-TB)

3. Related Application Note

The application note that is related to this application note is listed below for reference.

RL78/G13 Initialization (R01AN0451E) Application Note

4. Description of the Hardware

4.1 Hardware Configuration Example

Figure 4.1 shows an example of hardware configuration that is used for this application note.

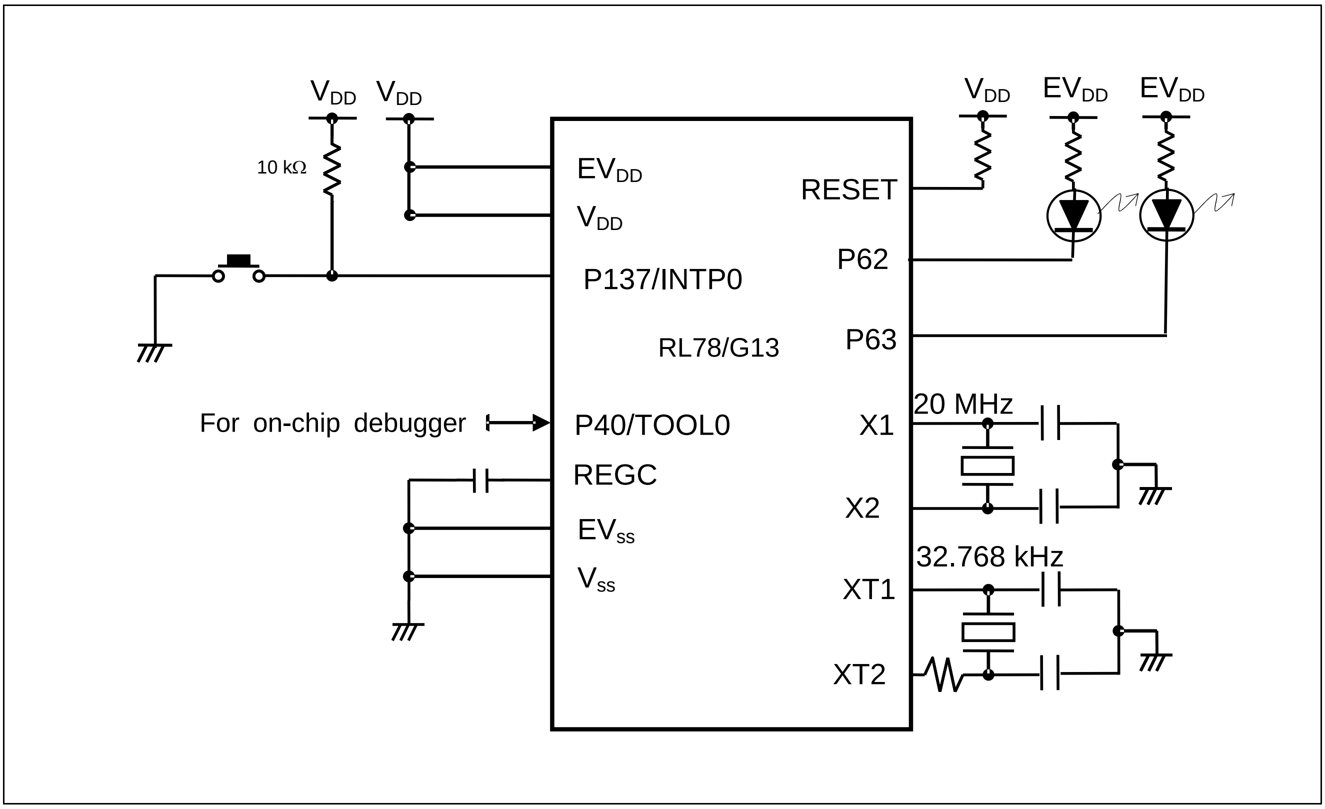

Figure 4.1 Hardware Configuration

Cautions: 1. The purpose of this circuit is only to provide the connection outline and the circuit is simplified accordingly. When designing and implementing an actual circuit, provide proper pin treatment and make sure that the hardware's electrical specifications are met (connect the input-only ports separately to V_{DD} or V_{SS} via a resistor).

2. Connect any pins whose name begins with EV_{SS} to V_{SS} and any pins whose name begins with EV_{DD} to V_{DD}, respectively.
3. V_{DD} must be held at not lower than the reset release voltage (V_{LVI}) that is specified as LVD.
4. The LED connected to P63 is always off.

4.2 List of Pins to be Used

Table 4.1 lists the pins to be used and their functions.

Table 4.1 Pins to be Used and their Functions

Pin Name	I/O	Description
P137/INTP0	Input	Switch input
P62	Output	LED output

5. Software Description

5.1 Operation Overview

The sample code covered in this application note switches its operating clock when the switch on the target board is pressed.

(1) Initialize the clock generator.
The sample code initializes the I/O ports, clock generator, timer array unit 0 (TAU0), 12-bit interval timer, and external interrupt input hardware. It enables interrupt processing after the initialization.

The LED blinks at the interval of the TAU0 interval interrupts corresponding to the selected operating clock.

(2) Get the switch state.
The sample code gets the switch state. It switches the operating clock when the press of the switch is detected. The switch state is tested when an INTP0 external interrupt occurs. If the press of the switch is not detected, the sample code places the CPU into the HALT mode.

(3) Switch the clock.
The CPU/peripheral hardware clock (f_{CLK}) is switched when the switch is pressed.

The CPU/peripheral hardware clock (f_{CLK}) is switched according to the sequence below.

1) HOCO clock (32 MHz) → X1 oscillation clock (20 MHz)

2) X1 oscillation clock (20 MHz) → XT1 oscillation clock (32.768 kHz)

3) XT1 oscillation clock (32.768 kHz) → HOCO clock (32 MHz)

Subsequently, steps (1) to (3) are repeated.

(4) Get the clock status.
The sample code gets the clock status. If the clock status is found to have been changed, the sample code takes one of the following actions according to the clock operating state:

- When the high-speed on-chip oscillator clock (HOCO clock) operates: Stops the X1 oscillation clock.
- When the X1 oscillation clock operates: Stops the HOCO clock.
- When the XT1 oscillation clock operates: Stops the X1 oscillation clock and the HOCO clock.

The XT1 oscillation clock is always generated.

(5) Change the LED blinking period.
The sample code changes the TAU0 interrupt interval as follows according to the operating CPU/peripheral hardware clock (f_{CLK}):

LED blinking period of the HOCO clock (32 MHz):	0.5 seconds
LED blinking period of the X1 oscillation clock (20 MHz):	1 seconds
LED blinking period of the XT1 oscillation clock (32.768 kHz):	2 seconds

(6) Transition to the HALT mode.
The sample code transitions to the HALT mode. It returns from the HALT mode by TAU0 interval interrupt or external interrupt generated by the switch. After returning from the HALT mode, the sample code performs step (2). Subsequently, it repeats a cycle of steps (2) to (6).

5.2 File Configuration

Table 5.1 lists the files that are used in this sample code. This table excludes files which are automatically generated by the integrated development environment.

Table 5.1 File Configuration

File Name	Description	Remarks
r_cgc.c	Clock generator module	CPU clock initialization
r_cg_cgc.h	External reference header file for the clock generator module	
r_cgc_user.c	Processing specific to the clock generator sample code	Additional functions: R_CGC_ChangeClock, R_CGC_HOCOToX1, R_CGC_X1ToXT1, R_CGC_XT1ToHOCO, R_CGC_GetClockStatus, R_CGC_Get_X1_Status R_CGC_Get_XT1_Status, R_CGC_Get_HOCO_Status, R_CGC_StopClock
r_intc.c	External interrupt input module	
r_cg_intc.h	External reference header file for the external interrupt input module	
r_intc_user.c	External interrupt input module INTP0 external interrupt	
r_it.c	12-bit interval timer module	
r_cg_it.h	External reference header file for the 12-bit interval timer module	
r_it_user.c	Processing specific to the 12-bit interval timer module sample code	Additional function: R_IT_Wait_ms
r_main.c	Main processing	
r_cg_macrodriver.h	Common header file	Type definitions, Macro definitions about error status
r_cg_userdefine.h	Macro definitions specific to the sample code	
r_port.c	Port function module	I/O port setting
r_cg_port.h	External reference header file for the port function module	
r_systeminit.c	System module	Initialization and system functions
r_timer.c	Timer module	
r_cg_timer.h	External reference header file for the timer module	
r_timer_user.c	Processing specific to the timer module sample code TAU0 channel 0 interrupt	Additional functions: R_TAU0_Channel0_GetParameter, R_TAU0_Channel0_Restart, R_TAU0_Channel0_ChangeInterval, R_TAU0_Channel0_Interrupt

5.3 List of Option Byte Settings

Table 5.2 summarizes the settings of the option bytes.

Table 5.2 Option Byte Settings

Address	Value	Description
000C0H/010C0H	11101111B	Disables the watchdog timer. (Stops counting after the release from the reset status.)
000C1H/010C1H	01111111B	LVD reset mode, 2.81 V (2.76 V to 2.87 V)
000C2H/010C2H	11101000B	HS mode HOCO: 32 MHz
000C3H/010C3H	10000100B	Enables the on-chip debugger. Erases the data in the flash memory when on-chip debugging security ID authentication fails.

5.4 List of Constants

Table 5.3 lists the constants that are used in this sample program.

Table 5.3 Constants for Sample Program

Constant	Setting	Description
HOCO_NEXT_STATUS_X1	1	Clock status : Operating on the HOCO clock and the next clock is the X1 oscillation clock.
X1_NEXT_STATUS_XT1	2	Clock status : Operating on the X1 oscillation clock and the next clock is the XT1 oscillation clock.
XT1_NEXT_STATUS_HOCO	3	Clock status : Operating on the XT1 oscillation clock and the next clock is the HOCO clock.
X1_STATUS	1	The current status is operation on X1 oscillation clock.
XT1_STATUS	2	The current status is operation on XT1 oscillation clock.
HOCO_STATUS	3	The current status is operation on HOCO clock.
CHATTERING_WAIT	10	Wait time to deal with chattering is 10 ms.
HOCO_LED_SETTING_CHANNEL_PRESCALER	9	Frequency division ratio of channel 0 of TAU0 when HOCO clock is selected
HOCO_LED_SETTING_CHANNEL_COUNT	15625	Count value of channel 0 of TAU0 when HOCO clock is selected
X1_LED_SETTING_CHANNEL_PRESCALER	9	Frequency division ratio of channel 0 of TAU0 when X1 oscillation clock is selected
X1_LED_SETTING_CHANNEL_COUNT	19531	Count value of channel 0 of TAU0 when X1 oscillation clock is selected
XT1_LED_SETTING_CHANNEL_PRESCALER	9	Frequency division ratio of channel 0 of TAU0 when XT1 oscillation clock is selected
XT1_LED_SETTING_CHANNEL_COUNT	64	Count value of channel 0 of TAU0 when XT1 oscillation clock is selected
SWITCH_OFF	0	Switch is not pressed
SWITCH_ON	1	Switch is pressed
SWITCH_ON_PORT_LEVEL	0	Input port level when switch is on
CLOCK_NOT_OSCILLATING	0	Clock is not oscillating.
CLOCK_OSCILLATING	1	Clock is oscillating.

5.5 List of Variables

Table 5.4 lists the global variables that are used in this sample program.

Table 5.4 Global Variable

Type	Variable Name	Contents	Function Used
uint8_t	g_ClockStatus	Clock status	main() R_INTC0_Interrupt
uint8_t	g_TAU0_Channel0_Clkdiv	Frequency division ratio of channel 0 of TAU0	main() R_TAU0_Channel0_GetParameter()
uint16_t	g_TAU0_Channel0_Count	Counter value of channel 0 of TAU0	main() R_TAU0_Channel0_GetParameter()
uint8_t	g_SwitchStatus	Switch status	main() R_INTC0_Interrupt

5.6 List of Functions

Table 5.5 lists the functions that are used in this sample program.

Table 5.5 Functions

Function Name	Outline
R_INTC0_Start	Sets start of INTP0 external interrupt processing.
R_TAU0_Channel0_Start	Sets start of channel 0 of TAU0.
R_CGC_ChangeClock	Switches clocks.
R_CGC_HOCOToX1	Switches from HOCO clock to X1 oscillation clock.
R_CGC_X1ToXT1	Switches from X1 oscillation clock to XT1 oscillation clock.
R_CGC_XT1ToHOCO	Switches from XT1 oscillation clock to HOCO clock.
R_CGC_GetClockStatus	Gets clock status.
R_CGC_Get_X1_Status	Gets X1 oscillation clock status.
R_CGC_Get_XT1_Status	Gets XT1 oscillation clock status.
R_CGC_Get_HOCO_Status	Gets HOCO clock status.
R_CGC_StopClock	Stops clock.
R_TAU0_Channel0_GetParameter	Gets parameters of channel 0 of TAU0.
R_TAU0_Channel0_Restart	Restarts channel 0 of TAU0.
R_TAU0_Channel0_ChangeInterval	Changes interval of channel 0 of TAU0.
R_TAU0_Channel0_Stop	Sets stop of channel 0 of TAU0.
R_TAU0_Channel0_Interrupt	Processes an interval timer interrupt of channel 0 of TAU0.
R_INTC0_Interrupt	Processes an INTP0 external interrupt.
R_IT_Wait_ms	Waits in units of 1 ms.

5.7 Function Specifications

This section describes the specifications for the functions that are used in the sample code.

[Function Name R_INTC0_Start

Synopsis	Sets start of INTP0 external interrupt processing.
Header	#include "r_cg_macrodriver.h" #include "r_cg_intc.h" #include "r_cg_userdefine.h"
Declaration	void R_INTC0_Start(void)
Explanation	Releases mask of INTP0 interrupts to enables them.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_TAU0_Channel0_Start

Synopsis	Sets start of channel 0 of TAU0.
Header	#include "r_cg_macrodriver.h" #include "r_cg_timer.h" #include "r_cg_userdefine.h"
Declaration	void R_TAU0_Channel0_Start(void)
Explanation	Releases mask of INTP0 interrupts to start counting.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_CGC_ChangeClock

Synopsis	Switches clocks.	
Header	#include "r_cg_macrodriver.h" #include "r_cg_cgc.h" #include "r_cg_userdefine.h"	
Declaration	void R_CGC_ChangeClock(uint8_t status)	
Explanation	Switches clocks.	
Arguments	• First argument: status	: Clock status (1 to 3) Set one of the following constants: HOCO_NEXT_STATUS_X1 → Switches to X1 oscillation clock. X1_NEXT_STATUS_XT1 → Switches to XT1 oscillation clock. XT1_NEXT_STATUS_HOCO → Switch to HOCO clock.
Return value	None	
Remarks	None	

[Function Name] R_CGC_HOCOToX1

Synopsis	Switches from HOCO clock to X1 oscillation clock.
Header	#include "r_cg_macrodriver.h" #include "r_cg_cgc.h" #include "r_cg_userdefine.h"
Declaration	void R_CGC_HOCOToX1 (void)
Explanation	Switches the CPU/peripheral hardware clock (f_{CLK}) from the HOCO clock to the X1 oscillation clock.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_CGC_X1ToXT1

Synopsis	Switches from X1 oscillation clock to XT1 oscillation clock.
Header	#include "r_cg_macrodriver.h" #include "r_cg_cgc.h" #include "r_cg_userdefine.h"
Declaration	void R_CGC_X1ToXT1(void)
Explanation	Switches the CPU/peripheral hardware clock (f_{CLK}) from X1 oscillation clock to XT1 oscillation clock.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_CGC_XT1ToHOCO

Synopsis	Switches from XT1 oscillation clock to HOCO clock.
Header	#include "r_cg_macrodriver.h" #include "r_cg_cgc.h" #include "r_cg_userdefine.h"
Declaration	void R_CGC_XT1ToHOCO(void)
Explanation	Switches the CPU/peripheral hardware clock (f_{CLK}) from XT1 oscillation clock to HOCO clock.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_CGC_GetClockStatus

Synopsis	Gets clock status.	
Header	#include "r_cg_macrodriver.h" #include "r_cg_cgc.h" #include "r_cg_userdefine.h"	
Declaration	uint8_t R_CGC_GetClockStatus(uint8_t status)	
Explanation	Gets the clock status. Gets the status of the clock specified in the argument so that the user can check if the clock oscillates.	
Arguments	First argument: status	: Clock status (1 to 3) Set one of the following constants: X1_STATUS → Gets the status of X1 oscillation clock. XT1_STATUS → Gets the status of XT1 oscillation clock. HOCO_STATUS → Gets the status of HOCO clock.
Return value	• If the clock has not been switched: CLOCK_NOT_OSCILLATING (0x00) • If the clock has been switched: CLOCK_OSCILLATING (0x01)	
Remarks	None	

[Function Name] R_CGC_Get_X1_Status

Synopsis	Gets X1 oscillation clock status.
Header	#include "r_cg_macrodriver.h" #include "r_cg_cgc.h" #include "r_cg_userdefine.h"
Declaration	uint8_t R_CGC_Get_X1_Status(void)
Explanation	Gets the X1 oscillation clock status.
Arguments	None
Return value	• When the clock has not been switched: CLOCK_NOT_OSCILLATING (0x00) • When the clock has been switched: CLOCK_OSCILLATING (0x01)
Remarks	None

[Function Name] R_CGC_Get_XT1_Status

Synopsis	Gets XT1 oscillation clock status.
Header	#include "r_cg_macrodriver.h" #include "r_cg_cgc.h" #include "r_cg_userdefine.h"
Declaration	uint8_t R_CGC_Get_XT1_Status(void)
Explanation	Gets the XT1 oscillation clock status.
Arguments	None
Return value	• When the clock has not been switched: CLOCK_NOT_OSCILLATING (0x00) • When the clock has been switched: CLOCK_OSCILLATING (0x01)
Remarks	None

[Function Name] R_CGC_Get_HOCO_Status

Synopsis	Gets HOCO clock status.
Header	#include "r_cg_macrodriver.h" #include "r_cg_cgc.h" #include "r_cg_userdefine.h"
Declaration	uint8_t R_CGC_Get_HOCO_Status(void)
Explanation	Gets the HOCO clock status.
Arguments	None
Return value	• When the clock has not been switched: CLOCK_NOT_OSCILLATING (0x00) • When the clock has been switched: CLOCK_OSCILLATING (0x01)
Remarks	None

[Function Name] R_CGC_StopClock

Synopsis	Stops clock.
Header	#include "r_cg_macrodriver.h" #include "r_cg_cgc.h" #include "r_cg_userdefine.h"
Declaration	void R_CGC_StopClock(uint8_t status)
Explanation	Stops the clock specified in the argument.
Arguments	• First argument: status : Clock to stop (1 to 3) Specify one of the following constants: X1_STATUS XT1_STATUS → Stops HOCO. HOCO_STATUS → Does nothing (because the old XT1 oscillation clock always oscillates.)
Return value	None
Remarks	None

[Function Name] R_TAU0_Channel0_GetParameter

Synopsis	Gets parameters of channel 0 of TAU0.
Header	#include "r_cg_macrodriver.h" #include "r_cg_timer.h" #include "r_cg_userdefine.h"
Declaration	void R_TAU0_Channel0_GetParameter(uint8_t status, uint8_t *p_clkdiv, uint16_t *p_count)
Explanation	• Gets and stores the parameters associated with the clock status specified in an argument in p_clkdiv, p_count. p_clkdiv ← Loaded with the frequency division ratio of the operation clock for channel 0 of TAU0. p_count ← Loaded with the count value of channel 0 of TAU0.
Arguments	• First argument: status : Clock status (1 to 3) Specify one of the following constants: X1_STATUS → Gets parameters for X1 oscillation clock. XT1_STATUS → Gets parameters for XT1 oscillation clock. HOCO_STATUS → Gets parameters for HOCO clock. • Second argument: *p_clkdiv : Loaded with the frequency division ratio of the operation clock for channel 0 of TAU0. • Third argument: *p_count : Loaded with the count value of channel 0 of TAU0.
Return value	None
Remarks	• p_clkdiv and p_count are the first and second arguments of R_TAU0_Channel0_Restart, respectively.

[Function Name] R_TAU0_Channel0_Restart

Synopsis	Restarts channel 0 of TAU0.
Header	#include "r_cg_macrodriver.h" #include "r_cg_timer.h" #include "r_cg_userdefine.h"
Declaration	void R_TAU0_Channel0_Restart(uint8_t clkdiv, uint16_t count)
Explanation	• Stops the interval timer for channel 0 of TAU0 temporarily. • Call R_TAU0_Channel0_ChangeInterval specified in the argument to change the interval of channel 0 of TAU0. • Restarts the interval timer for channel 0 of TAU0.
Arguments	• First argument: clkdiv : Frequency division ratio of the operation clock for channel 0 of TAU0
Return value	• Second argument: count : Count value of channel 0 of TAU0
Remarks	None
Synopsis	None

[Function Name] R_TAU0_Channel0_ChangeInterval

Synopsis	Changes interval of channel 0 of TAU0.
Header	#include "r_cg_macrodriver.h" #include "r_cg_timer.h" #include "r_cg_userdefine.h"
Declaration	void R_TAU0_Channel0_ChangeInterval(uint8_t clkdiv, uint16_t count)
Explanation	• Changes the frequency division ratio of the operation clock for channel 0 of TAU0 to clkdiv. • Changes the count value of channel 0 of TAU0 to count.
Arguments	• First argument: clkdiv : Frequency division ratio of operation clock (0 to 15) 0: $f_{CLK} / 2^0$ 15: $f_{CLK} / 2^{15}$ f_{CLK} : Frequency of CPU/peripheral hardware clock
Return value	• Second argument: count : Count value (0 to 65535)
Remarks	None
Synopsis	• 15 is assumed if clkdiv in the first argument is greater than 15.

[Function Name] R_TAU0_Channel0_Stop

Synopsis	Sets stop of channel 0 of TAU0.
Header	#include "r_cg_macrodriver.h" #include "r_cg_timer.h" #include "r_cg_userdefine.h"
Declaration	void R_TAU0_Channel0_Stop(void)
Explanation	Masks the interrupts of channel 0 of TAU0 to stop counting.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_TAU0_Channel0_Interrupt

Synopsis	Processes an interval timer interrupt of channel 0 of TAU0.
Header	#include "r_cg_macrodriver.h" #include "r_cg_timer.h" #include "r_cg_userdefine.h"
Declaration	__interrupt void R_TAU0_Channel0_Interrupt(void)
Explanation	Inverts the LED output (P62) at transition to this interrupt processing.
Arguments	None
Return value	None
Remarks	None

[Function Name] R_INTC0_Interrupt

Synopsis	Processes an INTP0 external interrupt.
Header	#include "r_cg_macrodriver.h" #include “r_cg_intc.h #include "r_cg_userdefine.h"
Declaration	__interrupt void R_INTC0_Interrupt(void)
Explanation	• Updating the switch status. This function checks the switch status twice. It waits (for 10 ms) to deal with chattering between the first and second switch checking operations. The function sets g_SwitchStatus to SWITCH_OFF (0x00) if it fails to detect two switch presses in succession. The function sets g_SwitchStatus to SWITCH_ON (0x01) if it detects two switch presses in succession. • Updating the clock status This function sets g_ClockStatus to one of the following constants according to the number of switch presses: HOCO_NEXT_STATUS_X1 X1_NEXT_STATUS_XT1 XT1_NEXT_STATUS_HOCO
Arguments	None
Return value	None
Remarks	None

[Function Name] R_IT_Wait_ms

Synopsis	Waits in units of 1 ms.
Header	#include "r_cg_macrodriver.h" #include "r_cg_it.h" #include "r_cg_userdefine.h"
Declaration	void R_IT_Wait_ms(uint16_t wait_count)
Explanation	Waits for a period of wait_count * 1 ms using a 12-bit interval timer.
Arguments	• First argument: wait_count : Wait count
Return value	None
Remarks	• This function does not use the vector interrupt function of the 12-bit interval timer. It waits for the period specified in the argument while polling the interrupt request flag after the 12-bit interval timer is started. • The function does not wait if the first argument wait_count is set to 0.

5.8 Flowcharts

Figure 5.1 shows the overall flow of the sample program described in this application note.

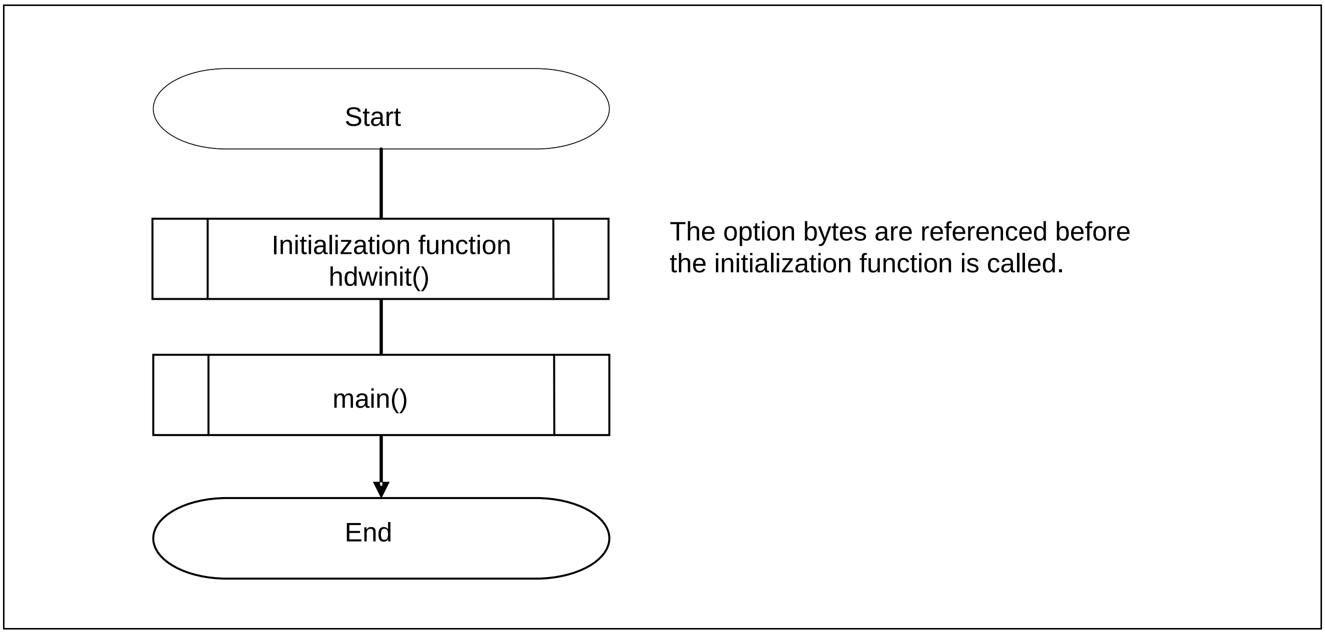

Figure 5.1 Overall Flow

5.8.1 Initialization Function

Figure 5.2 shows the flowchart for the initialization function.

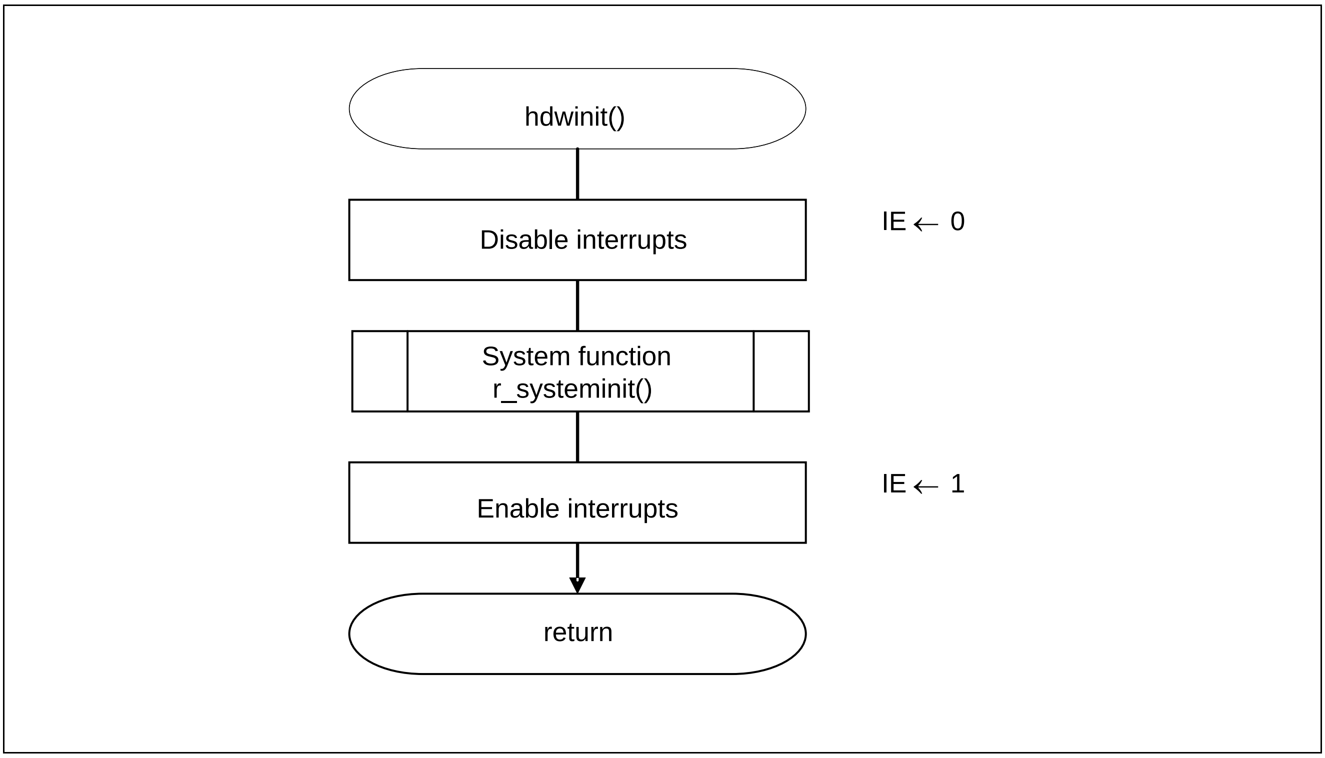

Figure 5.2 Initialization Function

5.8.2 System Function

Figure 5.3 shows the flowchart for the system function.

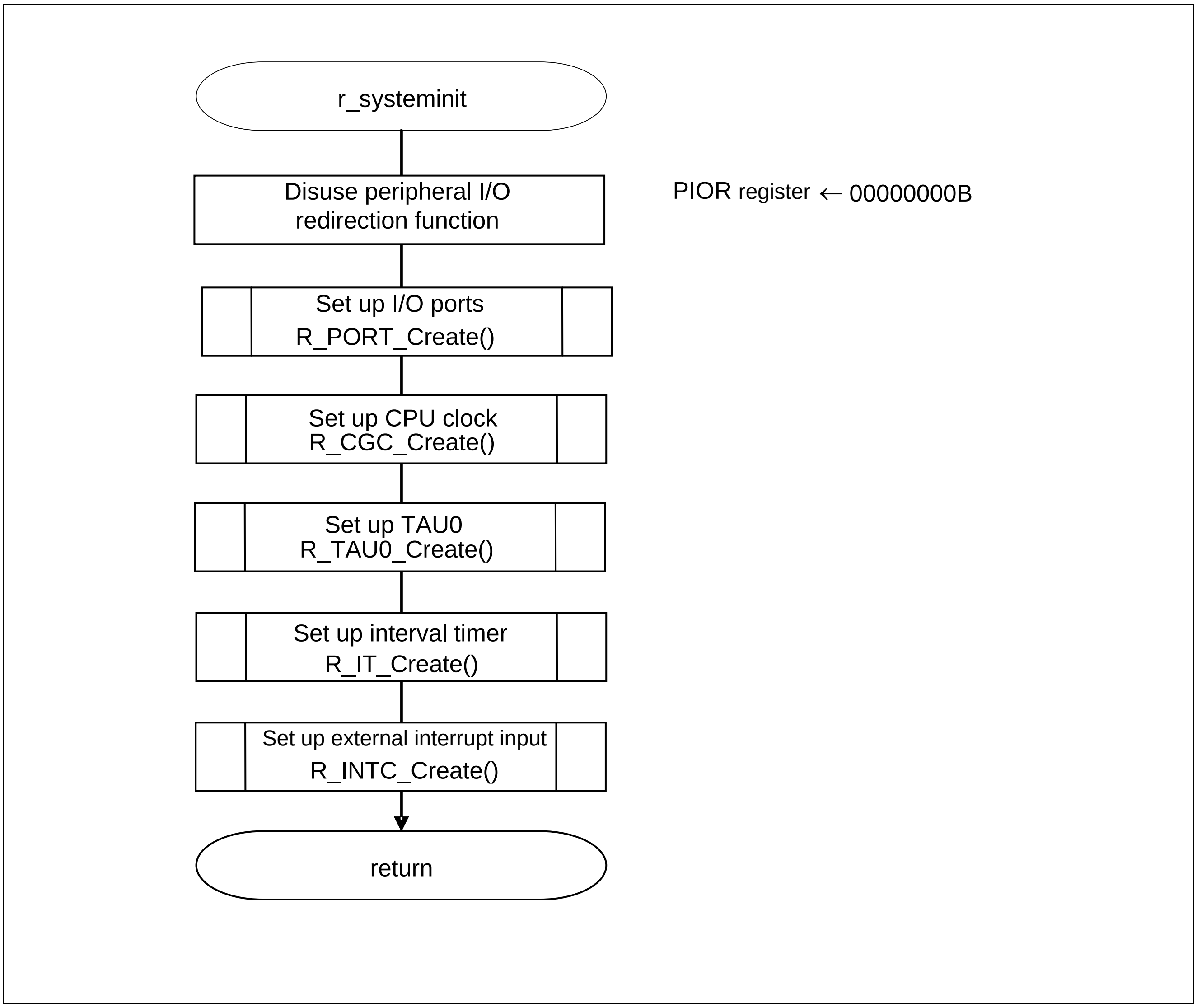

Figure 5.3 System Function

5.8.3 I/O Port Setup

Figure 5.4 shows the flowchart for setting up the I/O port.

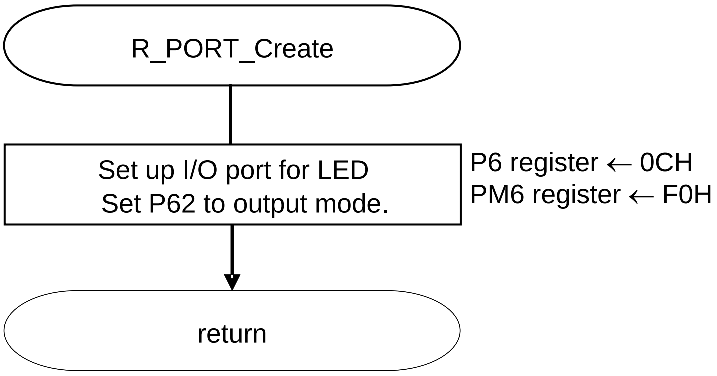

Figure 5.4 I/O Port Setup

Cautions: 1. Refer to the section entitled "Flowcharts" in RL78/G13 Initialization Application Note (R01AN0451E) for the configuration of the unused ports.

2. Provide proper treatment for unused pins so that their electrical specifications are met. Connect each of any unused input-only ports to V_{DD} or V_{SS} via a separate resistor.

3. Set the I/O port to a high level because the LED connected to P63 is always off.

5.8.4 CPU Clock Setup

Figure 5.5 shows the flowchart for setting up the CPU clock.

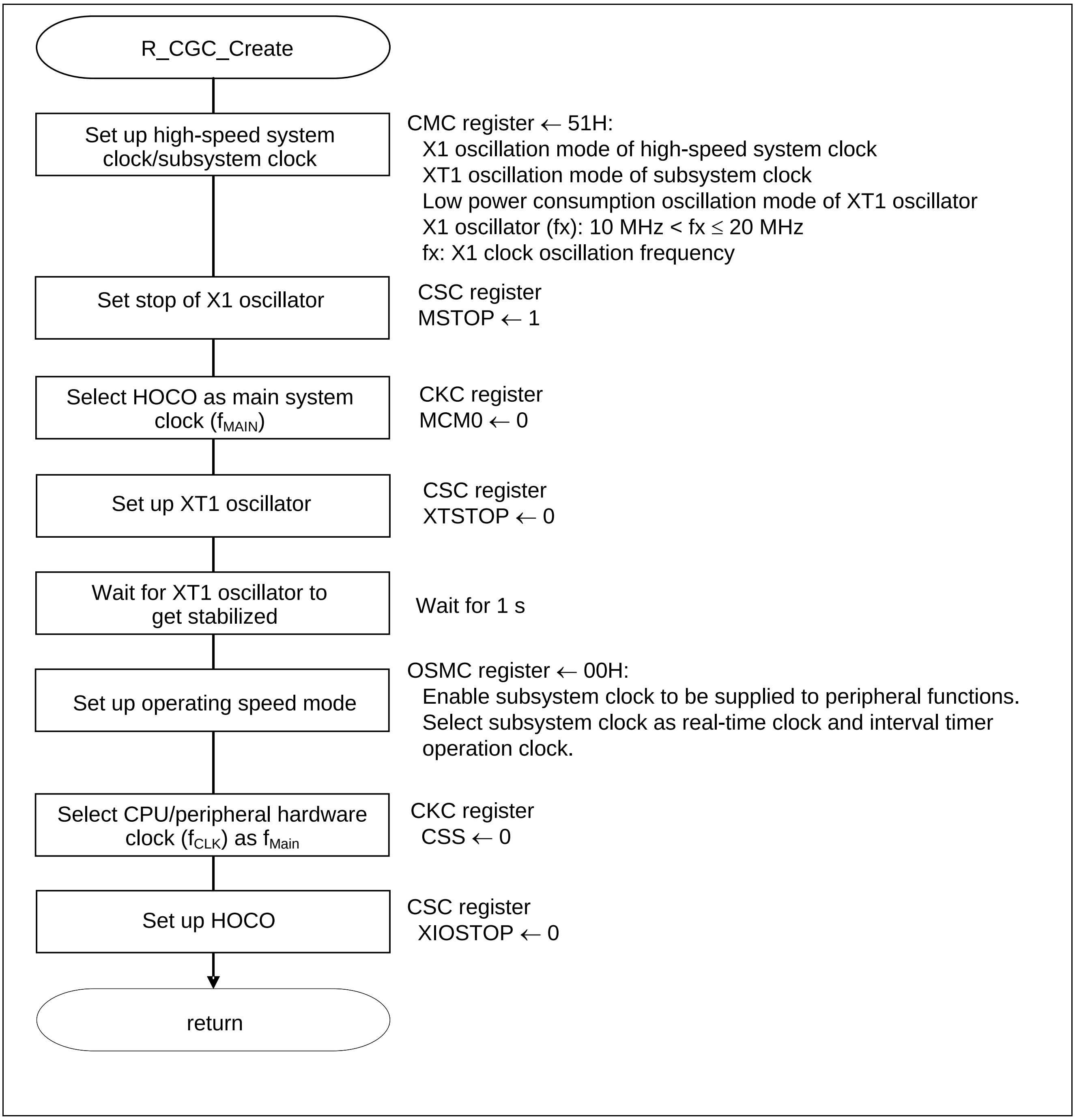

Figure 5.5 CPU Clock Setup

Remarks: CPU clock setup (R_CGC_Create()) determines whether stabilization of subsystem clock oscillation is completed within a waiting period (about 1 second). This period is specified with constant CGC_SUBWAITTIME in r_cg_cgc.h.

Caution: For details on the procedure for setting up the CPU clock (R_CGC_Create ()), refer to the section entitled "Flowcharts" in RL78/G13 Initialization Application Note (R01AN0451E).

Setting up clock generator operation mode

- Clock operation mode control register (CMC)
  : Select high-speed system clock pin operation mode.
  Select subsystem clock pin operation mode.
  Select XT1 oscillator oscillation mode.
  Control X1 clock frequency.

Symbol: CMC

	7	6	5	4	3	2	1	0
	EXCLK	OSCSEL	EXCLKS	OSCSELS	0	AMPHS1	AMPHS0	AMPH
When HOCO clock is used	**0**	**1**	**0**	**1**	0	**0**	**0**	**1**
When X1 oscillation clock is used								
When XT1 oscillation clock is used								

Bits 7 and 6

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

Bits 5 and 4

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

Bits 2 and 1

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

Bit 0

AMPHS	Control of X1 clock oscillation frequency
0	1 MHz ≤ fx ≤ 10 MHz
1	**10 MHz < fx ≤ 20 MHz**

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Selecting clocks

- System clock control register (CKC)
  : Select CPU/peripheral hardware clock (f_{CLK}) status,
  CPU/peripheral hardware clock (f_{CLK}),
  main system clock (f_{MAIN}), and
  control main system clock (f_{MAIN}) operation.

Symbol: CKC

	7	6	5	4	3	2	1	0
	CLS	CSS	MCS	MCM0	0	0	0	0
When HOCO clock is used	**0**	**0**	**0**	**0**	0	0	0	0
When X1 oscillation clock is used	**0**	**0**	**1**	**1**	0	0	0	0
When XT1 oscillation clock is used	**1**	**1**	0	0	0	0	0	0

Bit 7

CLS Note 1	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 Note2	Subsystem clock (f_{SUB})

Bit 5

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

Bit 4

MCM0 Note 2	Main system clock (f_{MAIN}) operation control
0	Selects the high-speed on-chip oscillator clock (f_{IH}) as the main system clock (f_{MAIN})
1	Selects the high-speed system clock (f_{MX}) as the main system clock (f_{MAIN})

Notes: 1. Bits 7 and 5 are read-only.
2. Changing the value of the MCM0 bit is prohibited while the CSS bit is set to 1.

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

Controlling clock operation

- Clock operation status control register (CSC)
  : Control operation of high-speed system clock,
  Subsystem clock, and
  high-speed on-chip oscillator clock.

Symbol: CSC

	7	6	5	4	3	2	1	0
	MSTOP	XTSTOP	0	0	0	0	0	HIOSTOP
When HOCO clock is used	**1**	**0**			0			**0**
When X1 oscillation clock is used	**0**							**1**
When XT1 oscillation clock is used	**1**							**1**

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 mode
1	**X1 oscillator stopped**	External clock from EXCLK pin is invalid.	

Bit 6

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

Bit 0

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

Caution: For details on the register setup procedures, refer to RL78/G13 User's Manual: Hardware.

5.8.5 Timer Array Unit 0 (TAU0) Setup

Figure 5.6 shows the flowchart for setting up the timer array unit 0 (TAU0).

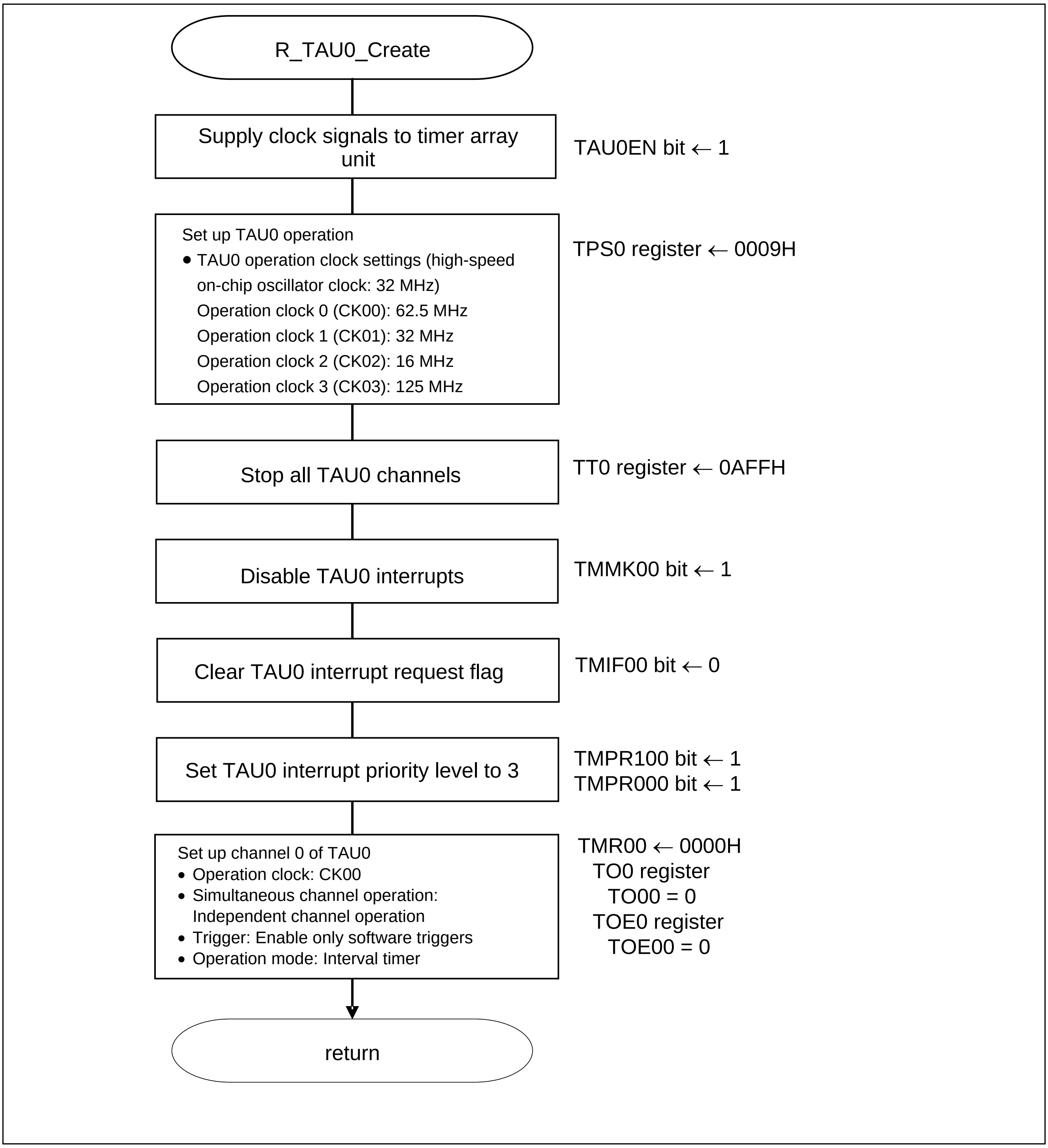

Figure 5.6 Timer Array Unit 0 (TAU0) Setup

5.8.6 Setting up 12-bit Interval Timer

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

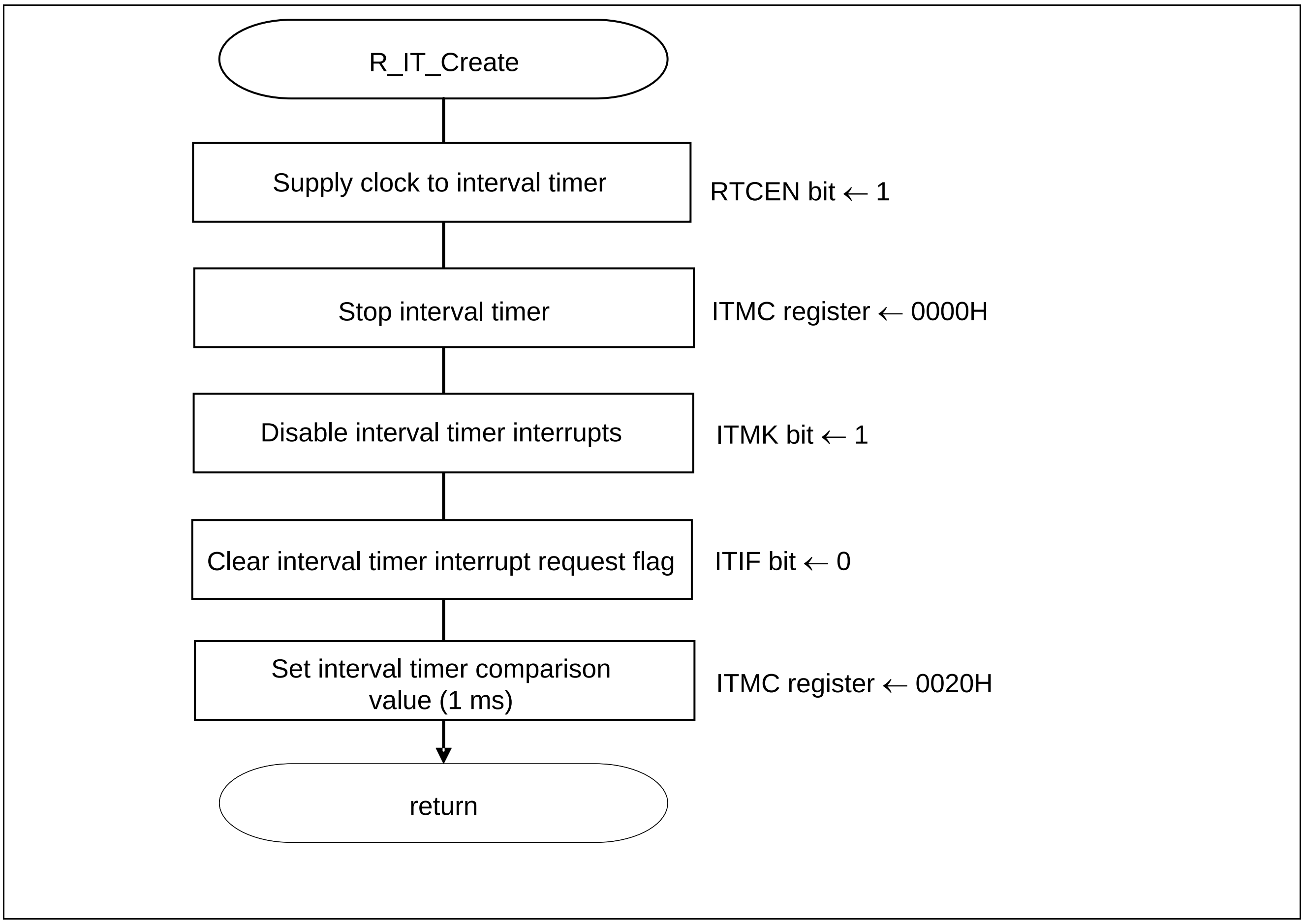

Figure 5.7 12-bit Interval Timer Setup

5.8.7 Setting up External Interrupt Input

Figure 5.8 shows the flowchart for setting up the external interrupt input.

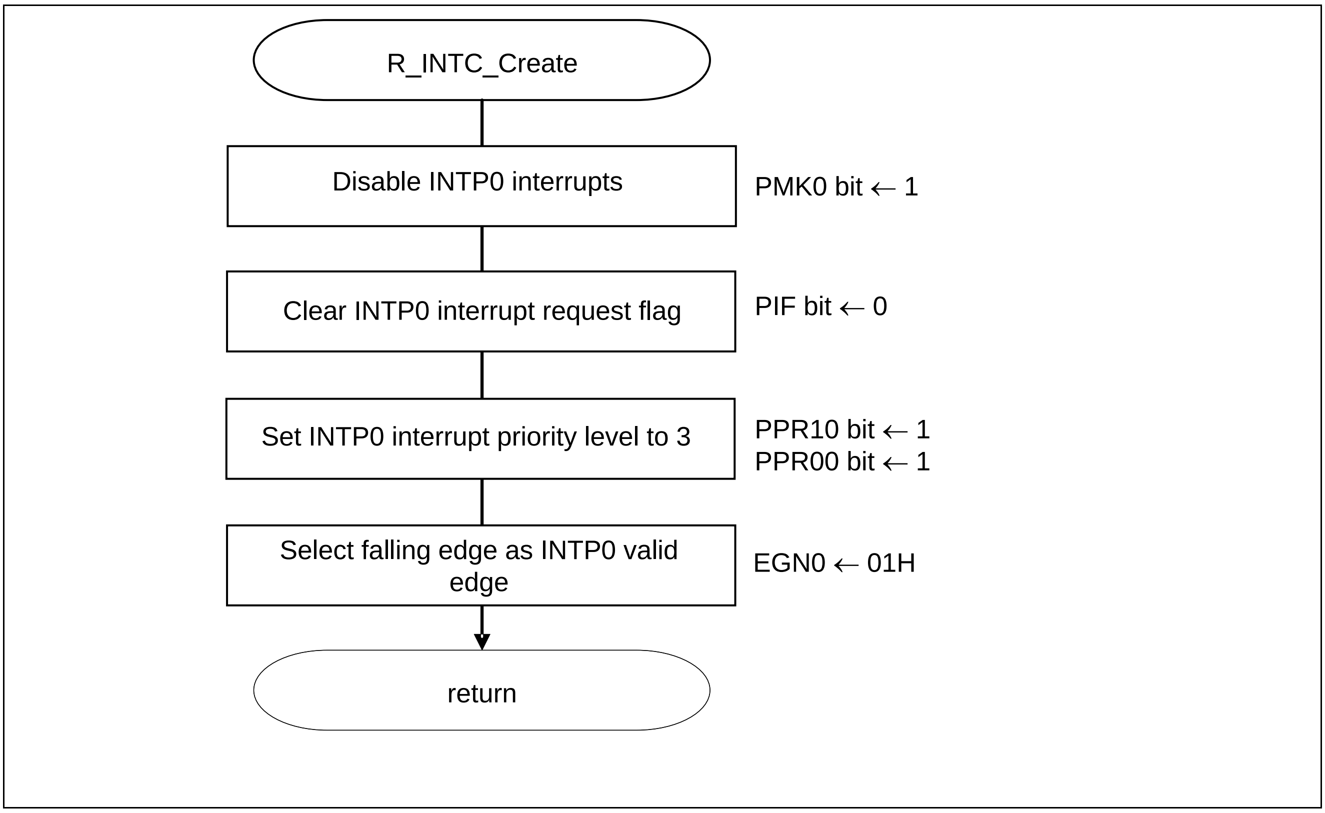

Figure 5.8 External Interrupt Input Setup

5.8.8 Main Processing

Figure 5.9 shows the flowchart for the main processing.

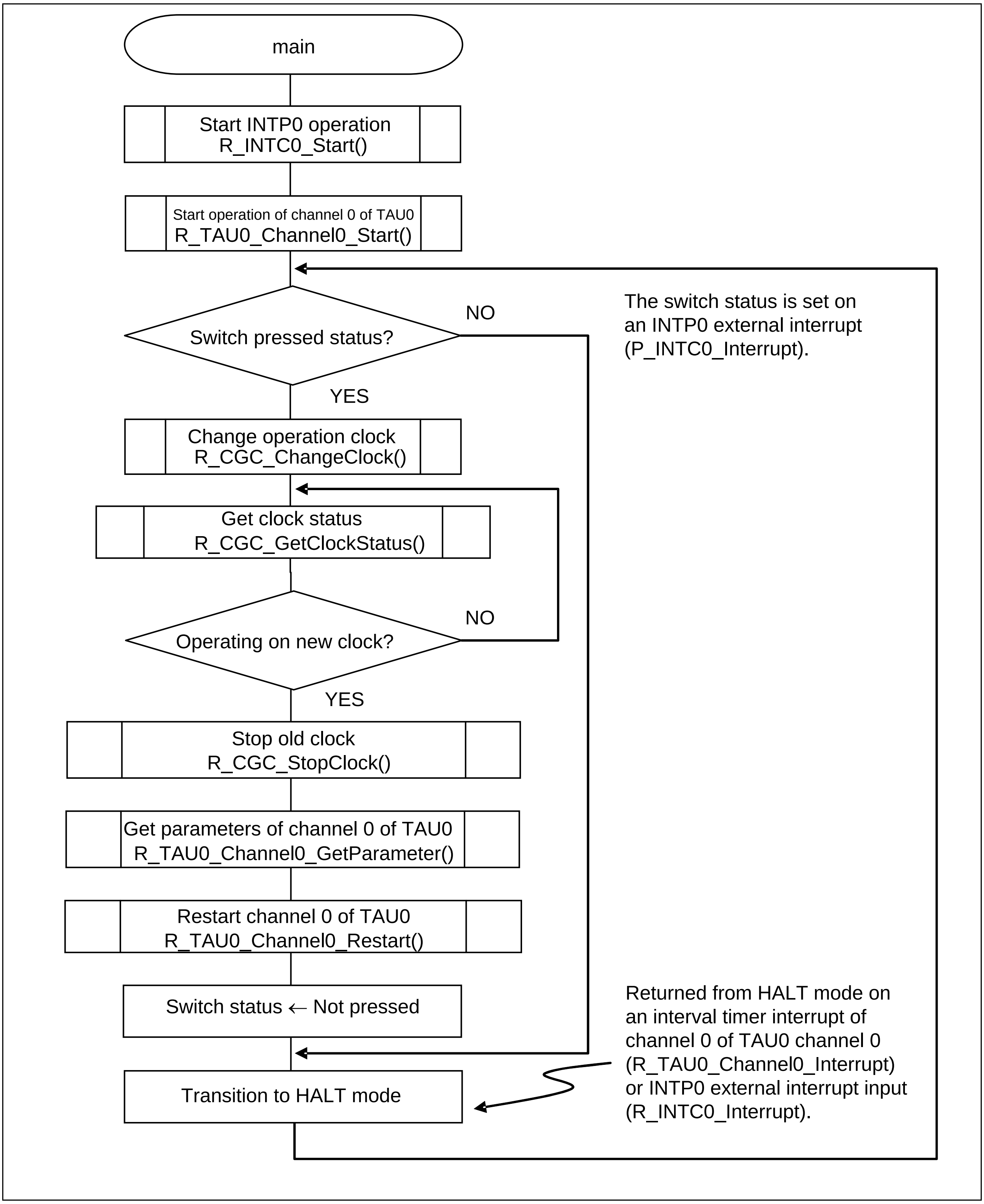

Figure 5.9 Main Processing

5.8.9 Setting to Start INTP0 External Interrupt Processing.

Figure 5.10 shows the flowchart for setting to start INTP0 external interrupt processing.

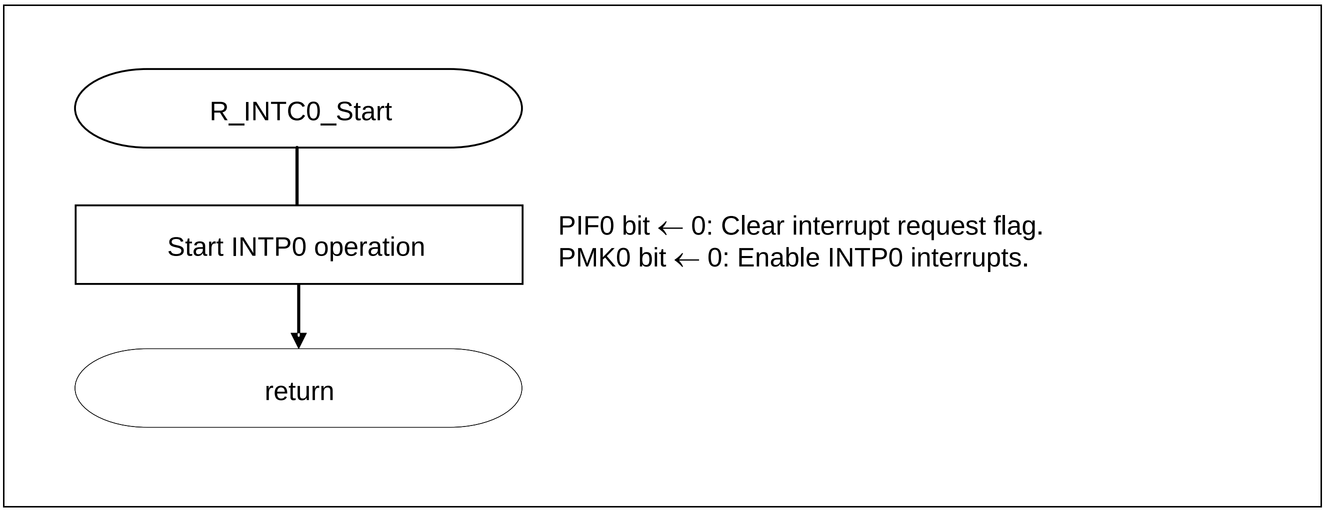

Figure 5.10 Setting to Start INTP0 External Interrupt Processing

5.8.10 Setting to Start Operation of Channel 0 of TAU0

Figure 5.11 shows the flowchart for setting to start the operation of channel 0 of TAU0.

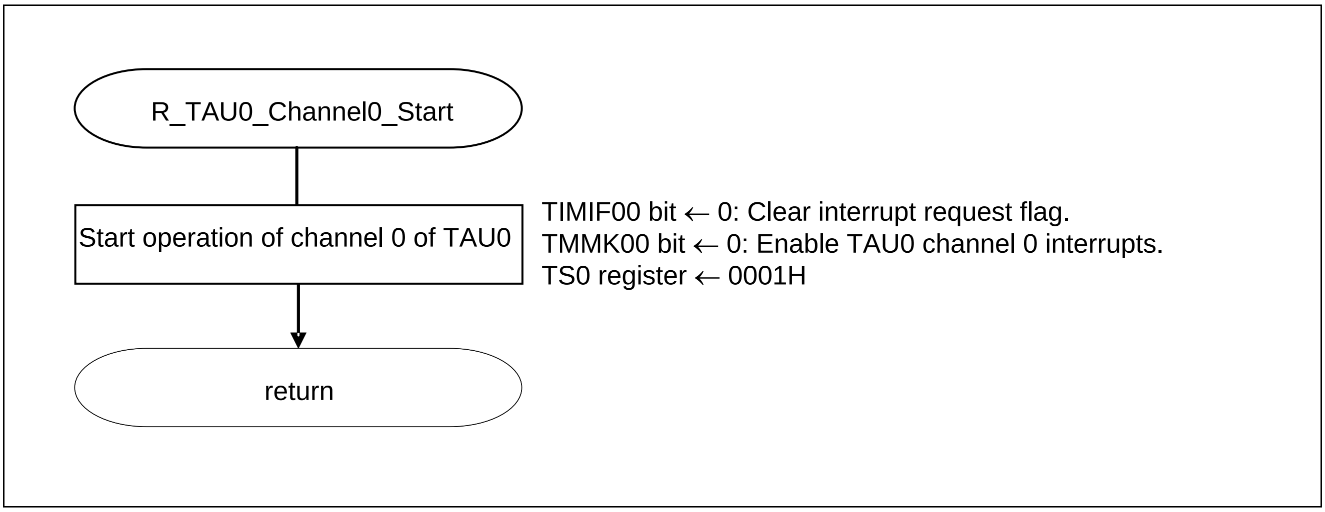

Figure 5.11 Setting to Start Operation of Channel 0 of TAU0

5.8.11 Switching Clock

Figure 5.12 shows the flowchart for switching clocks.

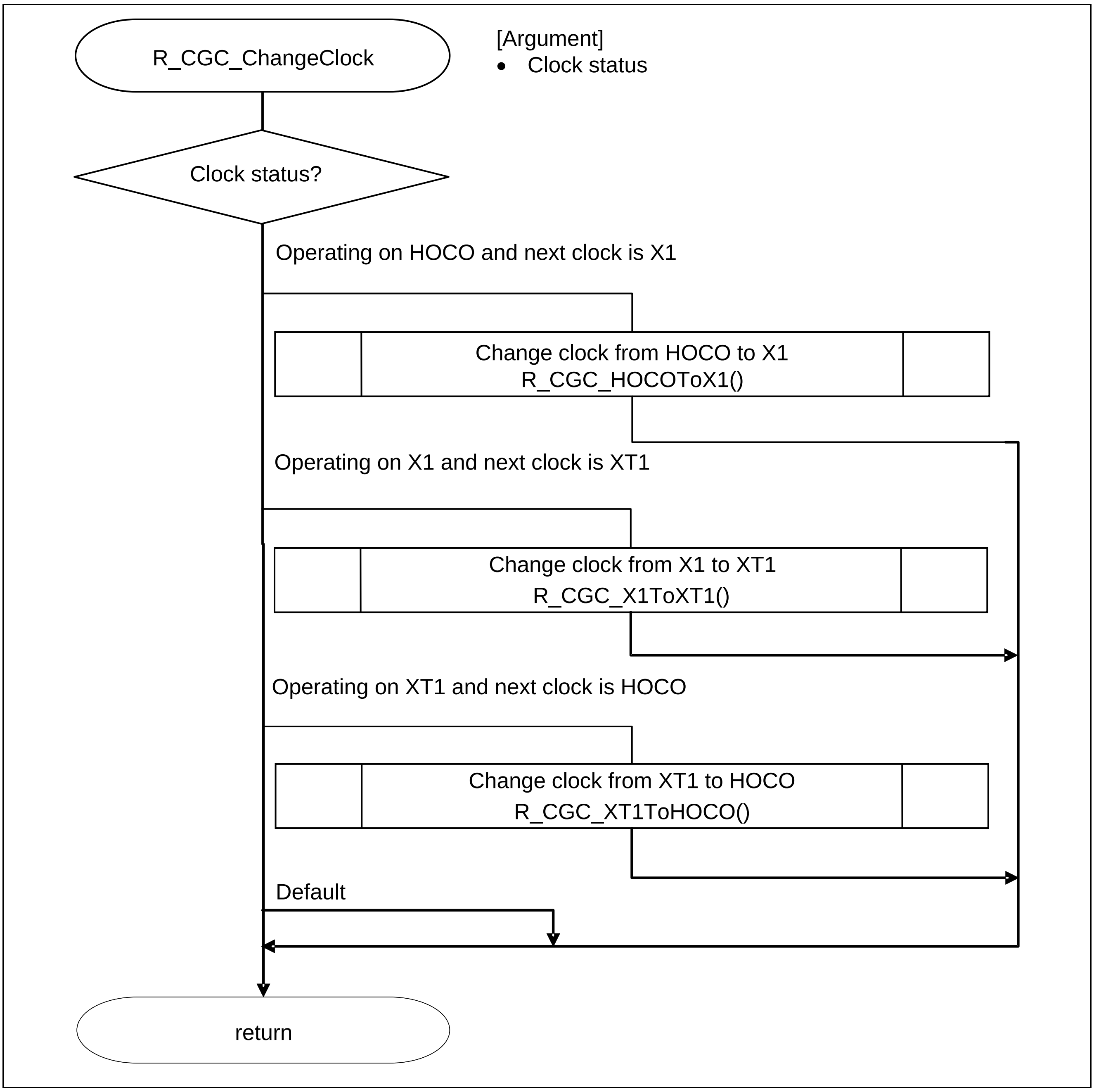

Figure 5.12 Switching Clocks

5.8.12 Switching from HOCO Clock to X1 Oscillation Clock.

Figure 5.13 shows the flowchart for switching from HOCO clock to X1 oscillation clock.

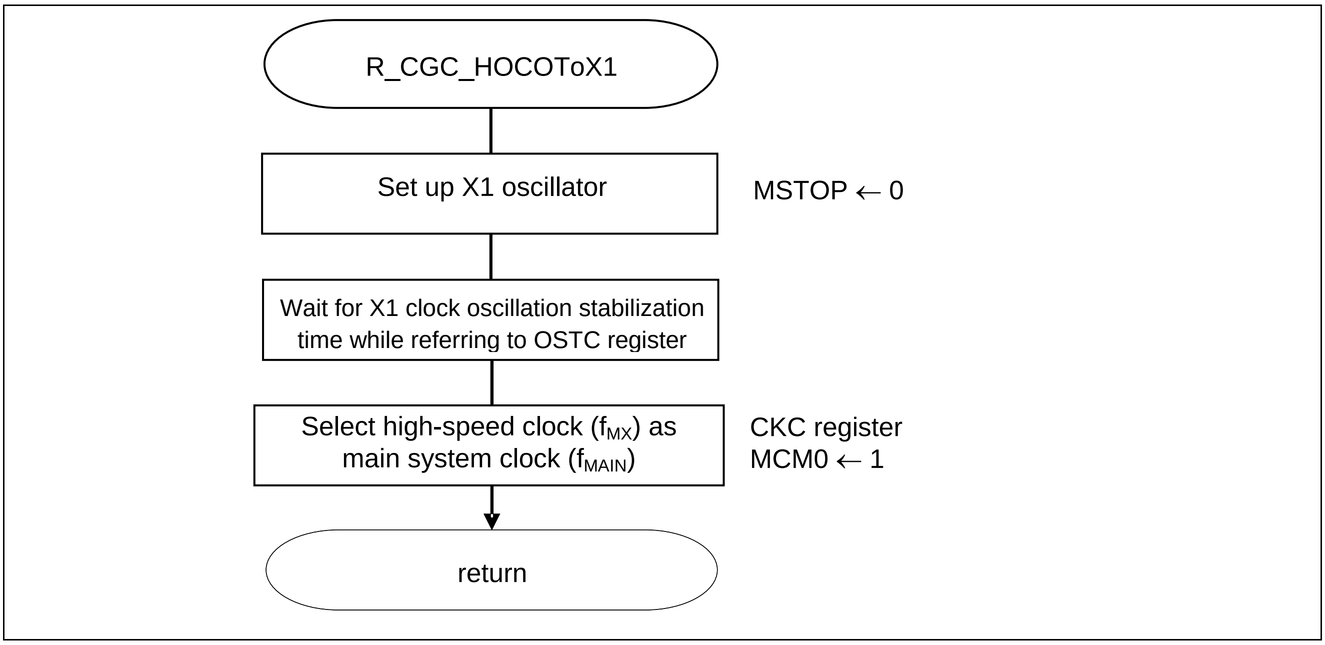

Figure 5.13 Switching from HOCO Clock to X1 Oscillation Clock

5.8.13 Switching from X1 Oscillation Clock to XT1 Oscillation Clock

Figure 5.14 shows the flowchart for switching from X1 oscillation clock to XT1 oscillation clock.

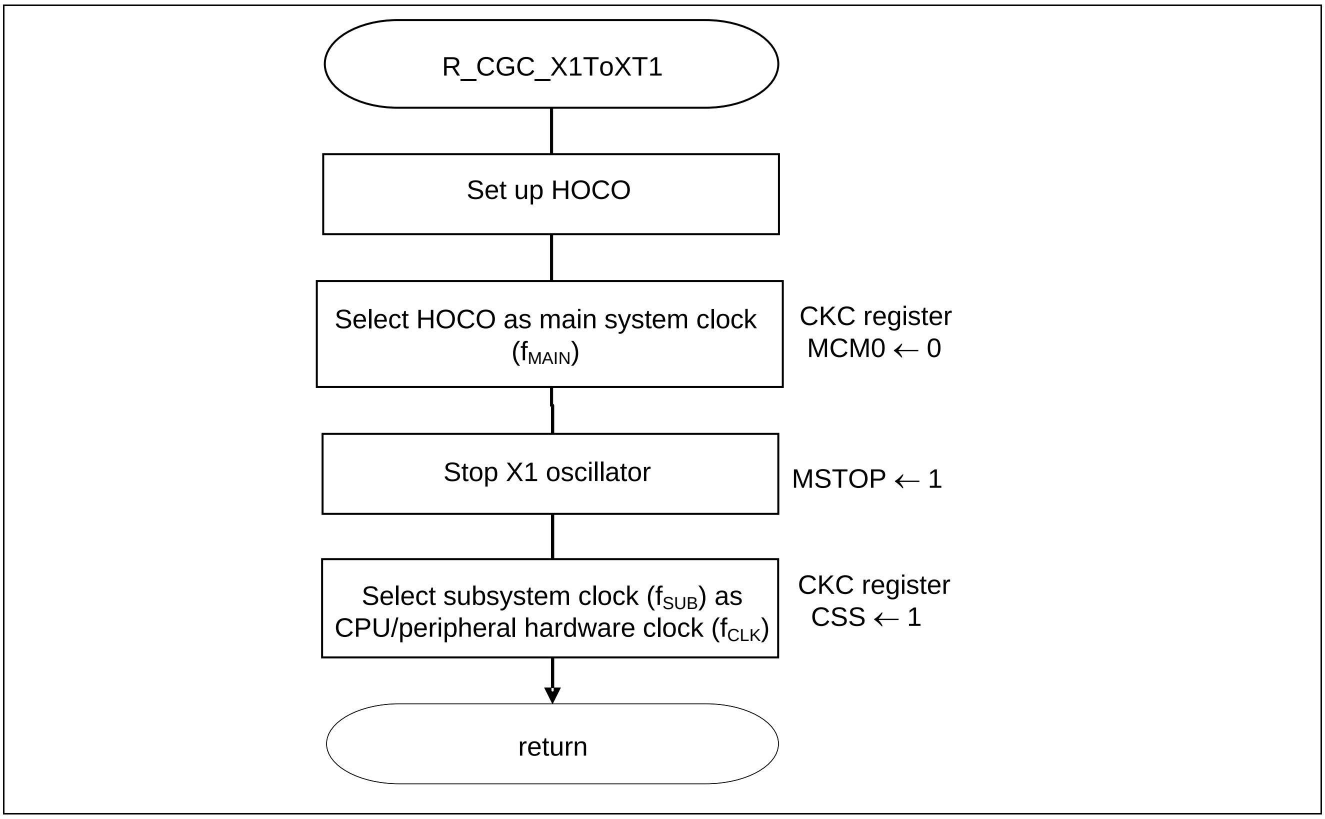

Figure 5.14 Switching from X1 Oscillation Clock to XT1 Oscillation Clock

5.8.14 Switching from XT1 Oscillation Clock to HOCO Clock

Figure 5.15 shows the flowchart for switching from XT1 oscillation clock to HOCO clock.

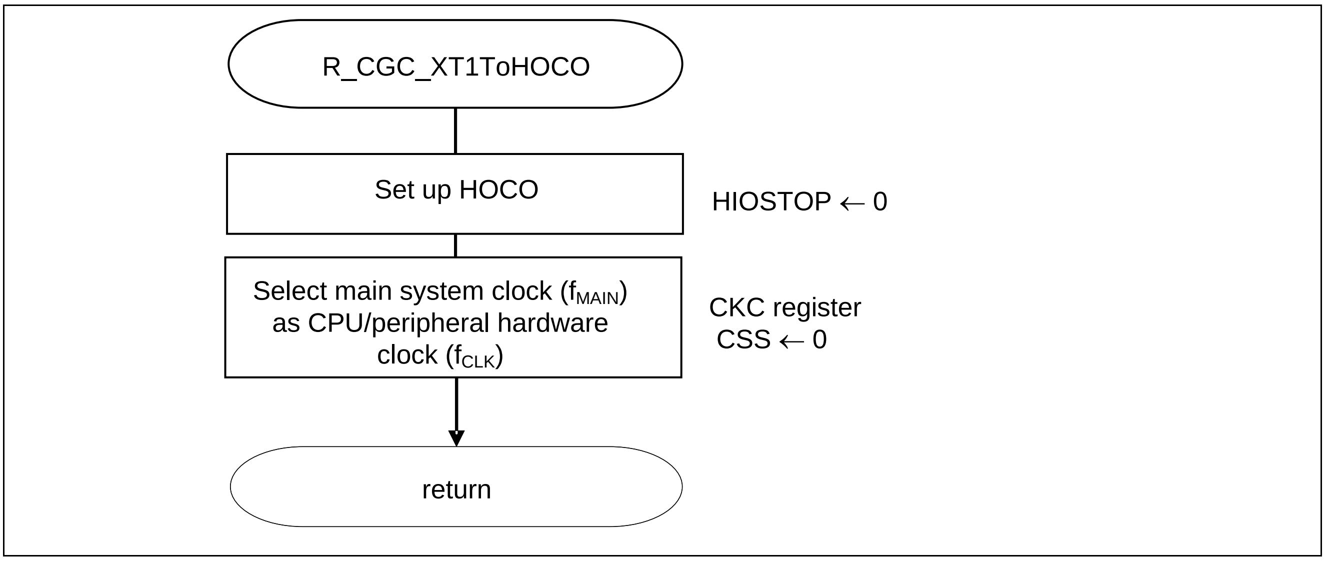

Figure 5.15 Switching from XT1 Oscillation Clock to HOCO Clock

5.8.15 Getting Clock Status

Figure 5.16 shows the flowchart for getting clock status.

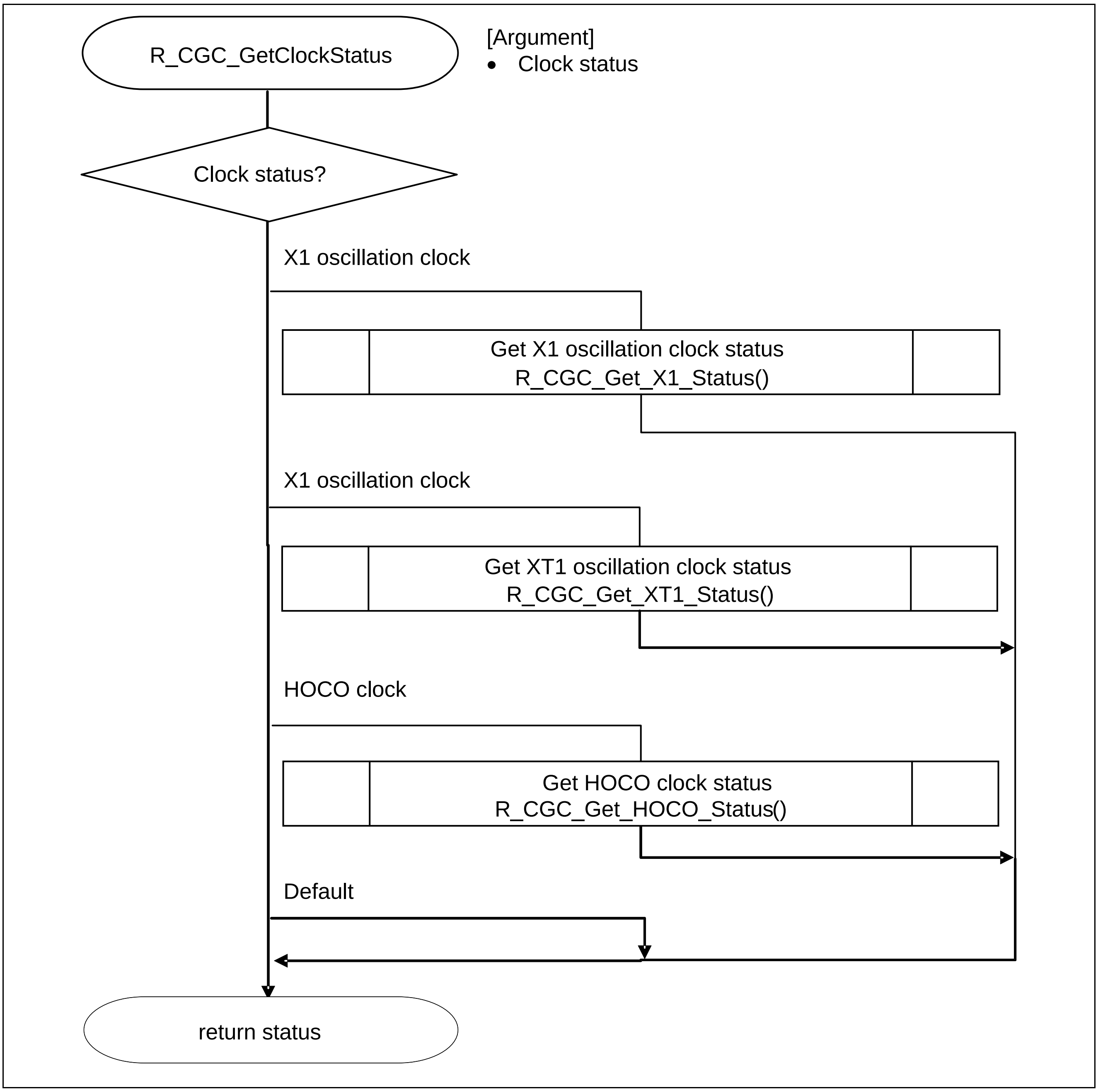

Figure 5.16 Getting Clock Status

5.8.16 Getting X1 Oscillation Clock Status

Figure 5.17 shows the flowchart for getting X1 oscillation clock status.

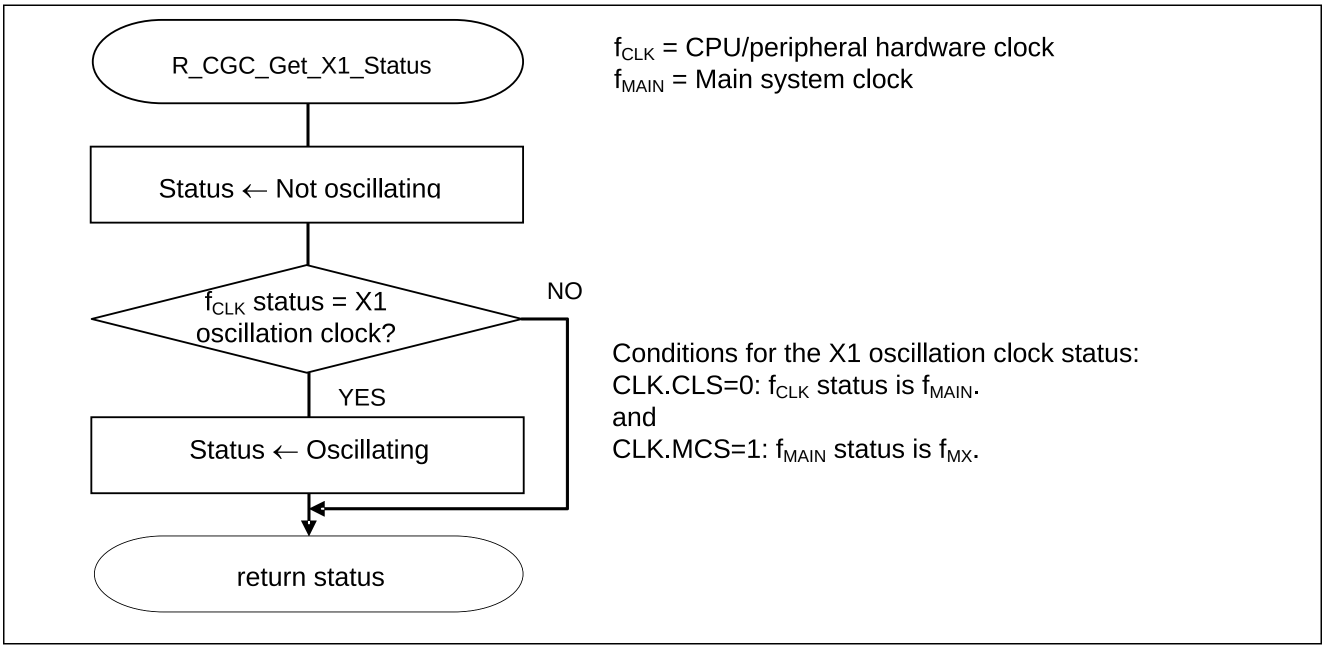

Figure 5.17 Getting X1 Oscillation Clock Status

5.8.17 Getting XT1 Oscillation Clock Status

Figure 5.18 shows the flowchart for getting XT1 oscillation clock status.

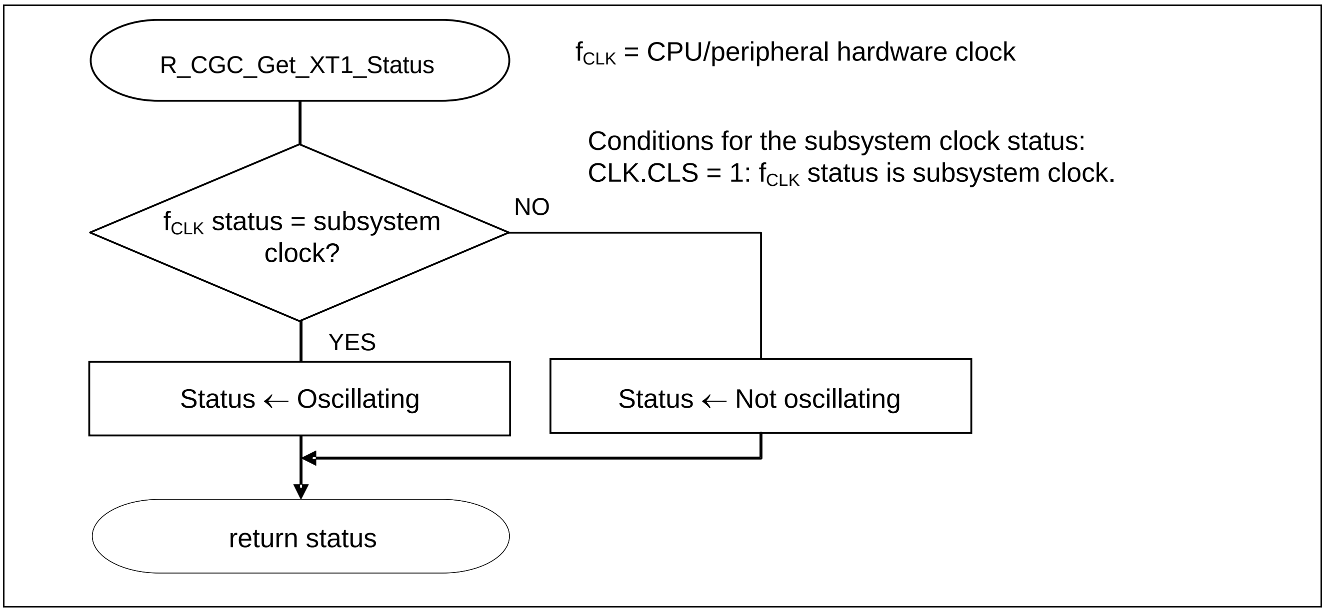

Figure 5.18 Getting XT1 Oscillation Clock Status

5.8.18 Getting HOCO Clock Status

Figure 5.19 shows the flowchart for getting HOCO clock status.

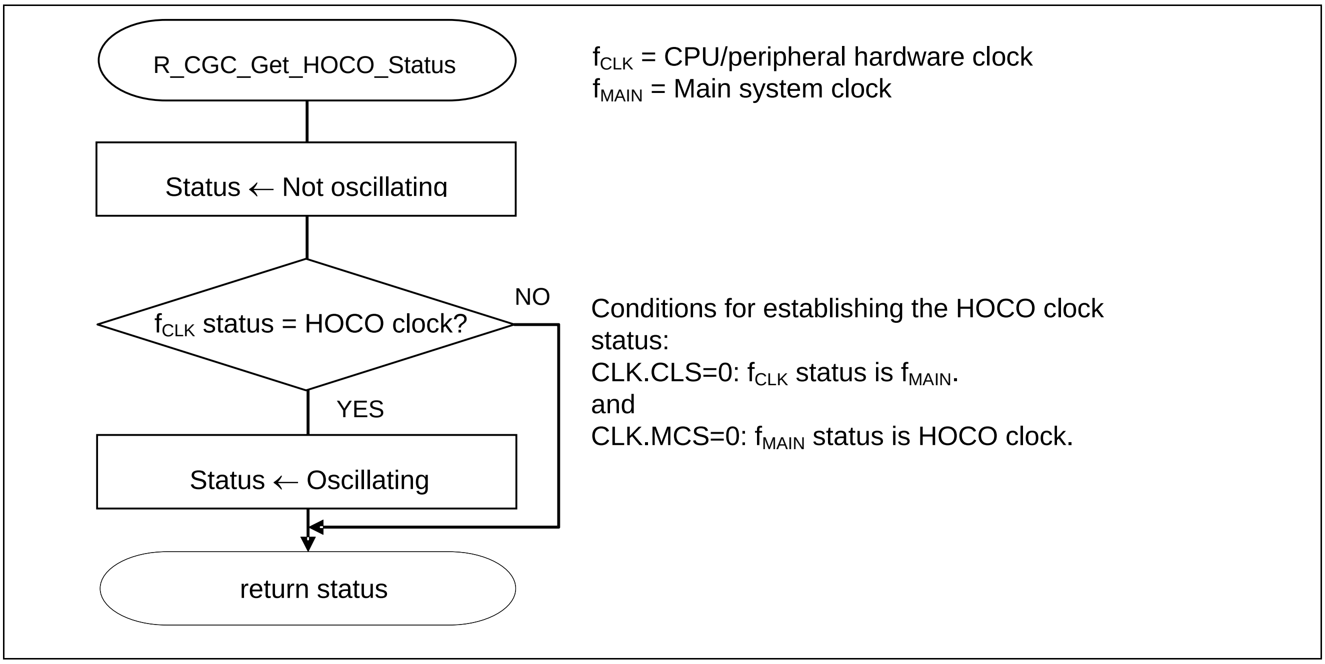

Figure 5.19 Getting HOCO Clock Status

5.8.19 Stopping Clock

Figure 5.20 shows the flowchart for stopping clock.

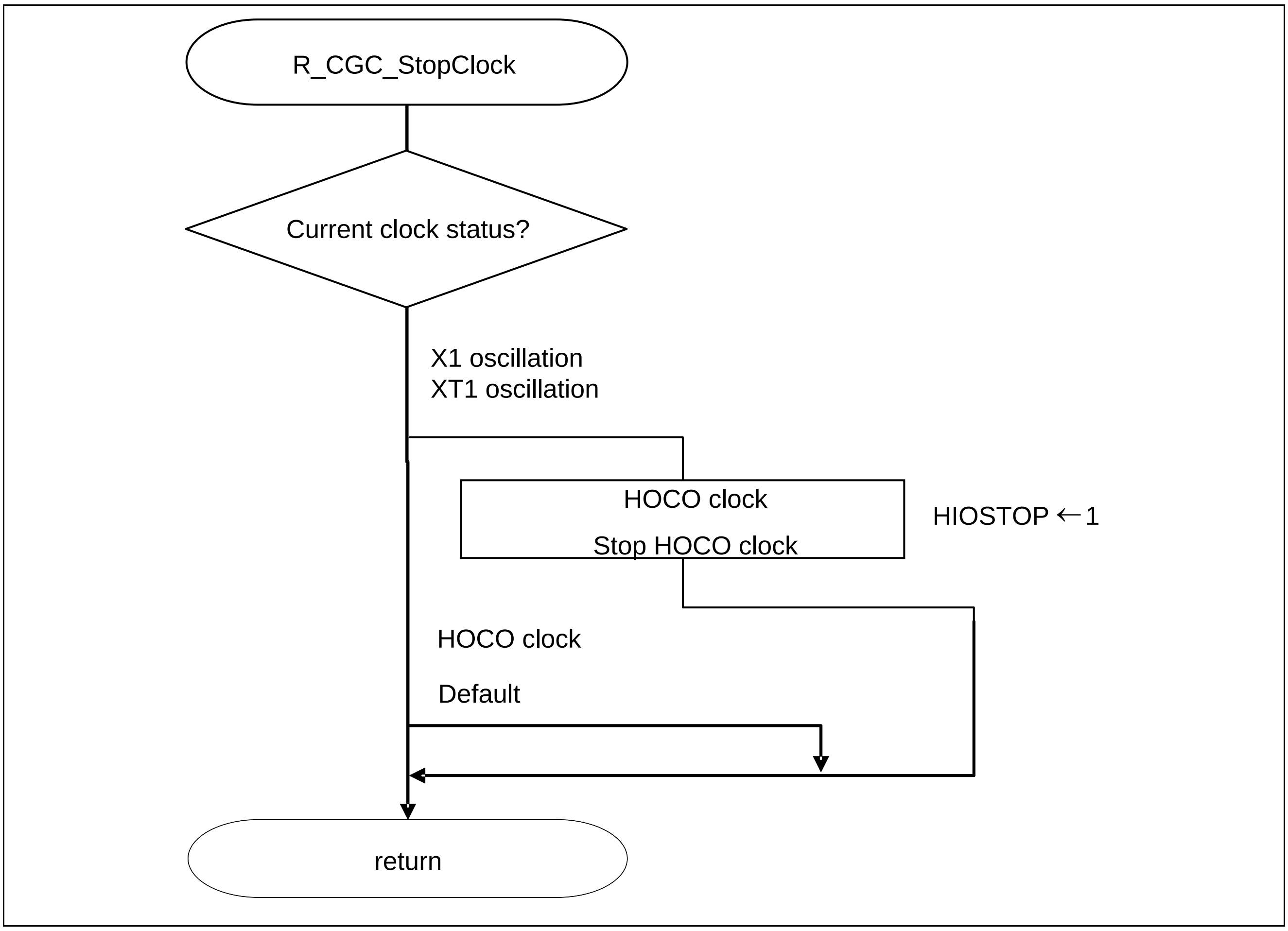

Figure 5.20 Stopping Clock

5.8.20 Getting Parameters of Channel 0 of TAU0

Figure 5.21 shows the flowchart for getting parameters of channel 0 of TAU0.

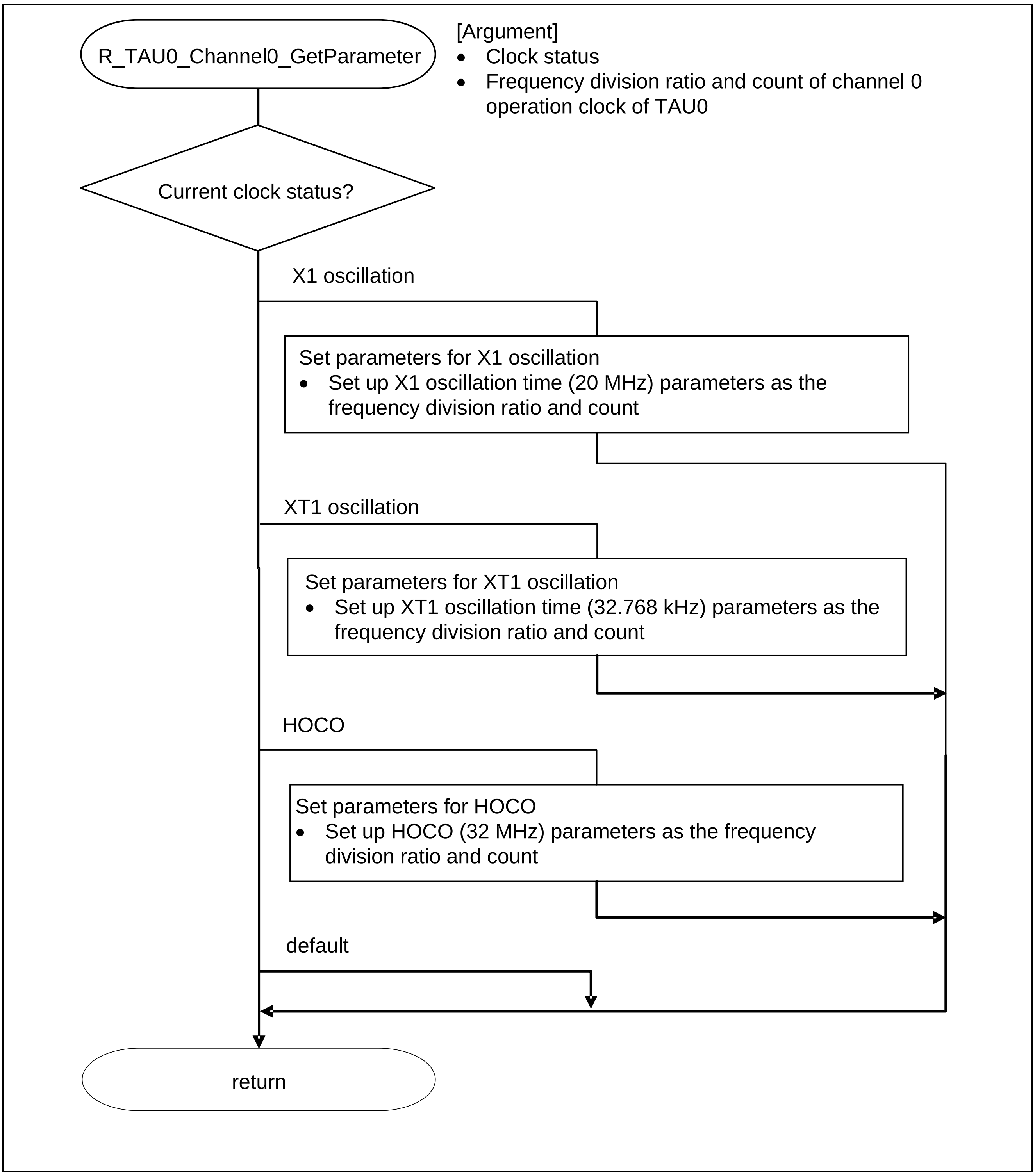

Figure 5.21 Getting Parameters of Channel of TAU0

5.8.21 Restarting Channel 0 of TAU0

Figure 5.22 shows the flowchart for restarting channel 0 of TAU0.

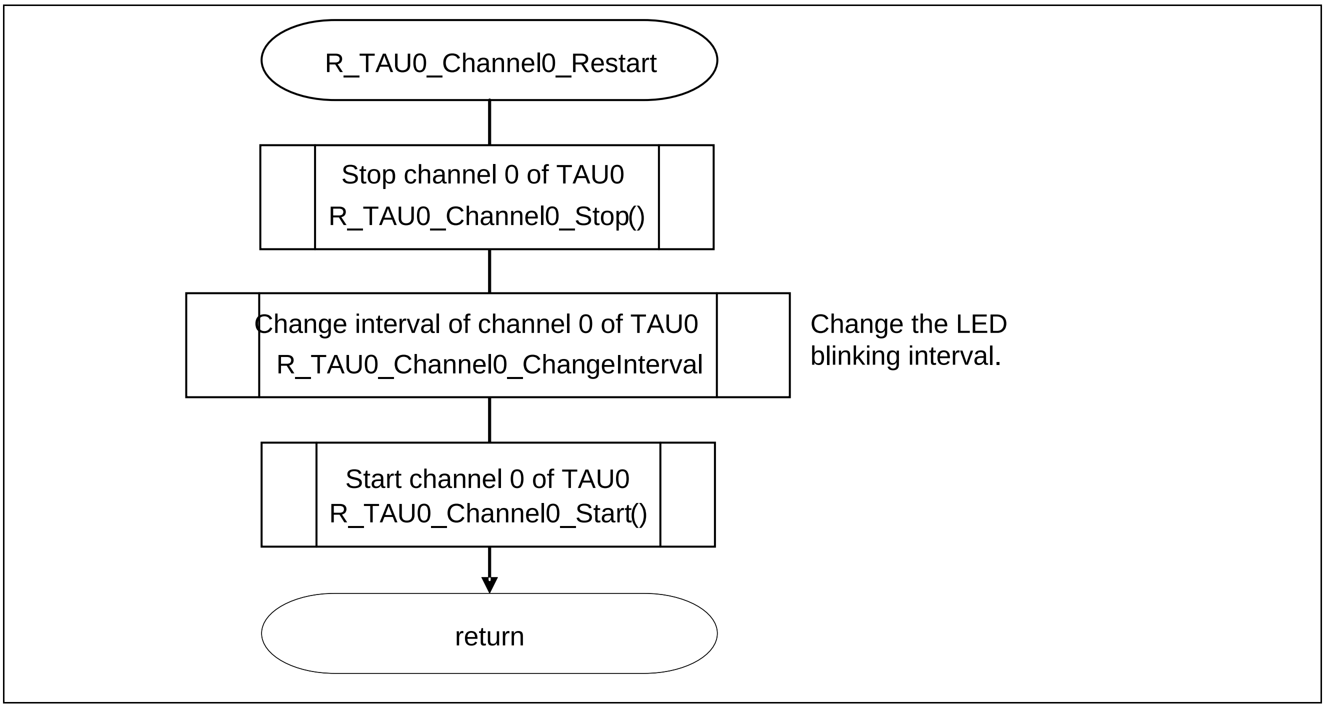

Figure 5.22 Restarting Channel 0 of TAU0

5.8.22 Changing Interval of Channel 0 of TAU0

Figure 5.23 shows the flowchart for changing the interval of channel 0 of TAU0.

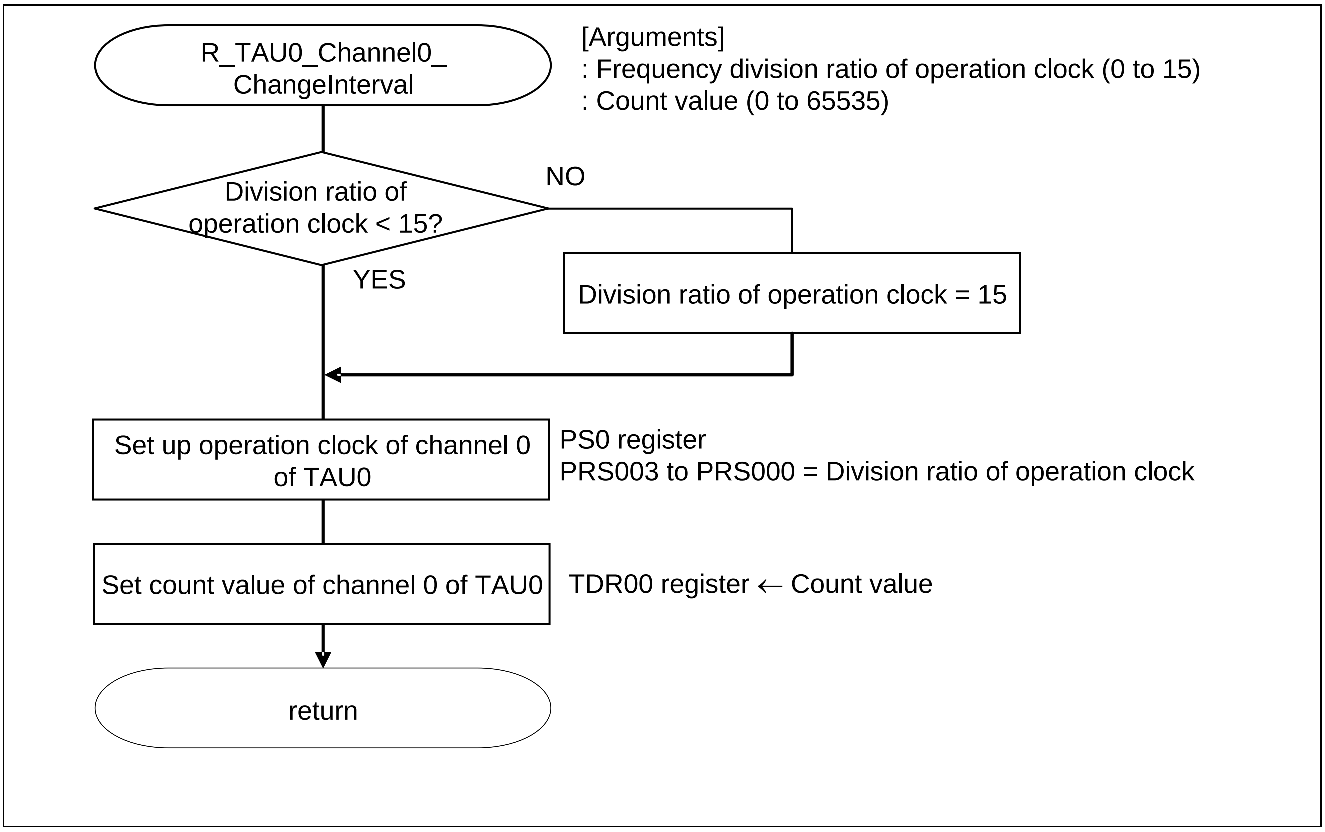

Figure 5.23 Changing Interval of Channel 0 of TAU0

5.8.23 Setting to Stop Channel 0 of TAU0

Figure 5.24 shows the flowchart for setting to stopping channel 0 of TAU0.

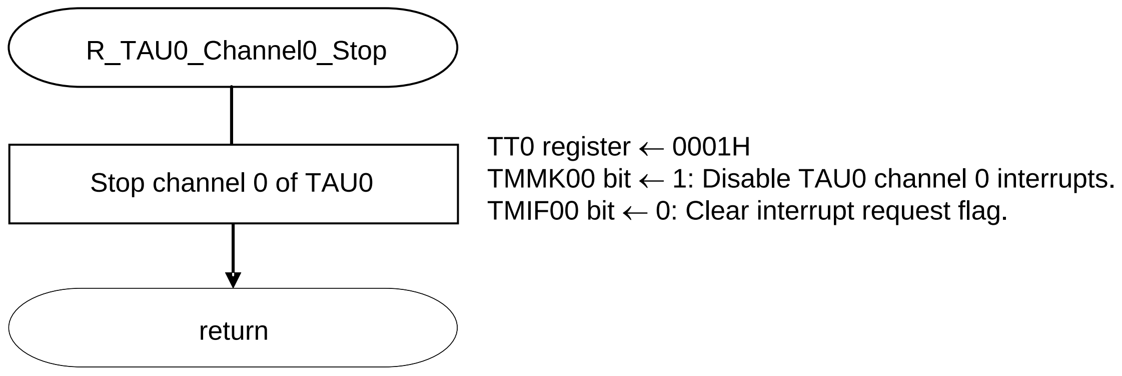

Figure 5.24 Setting to Stop Channel 0 of TAU0

5.8.24 Processing of Interval Timer Interrupt of Channel 0 of TAU0

Figure 5.25 shows the flowchart for processing an interval timer interrupt of channel 0 of TAU0.

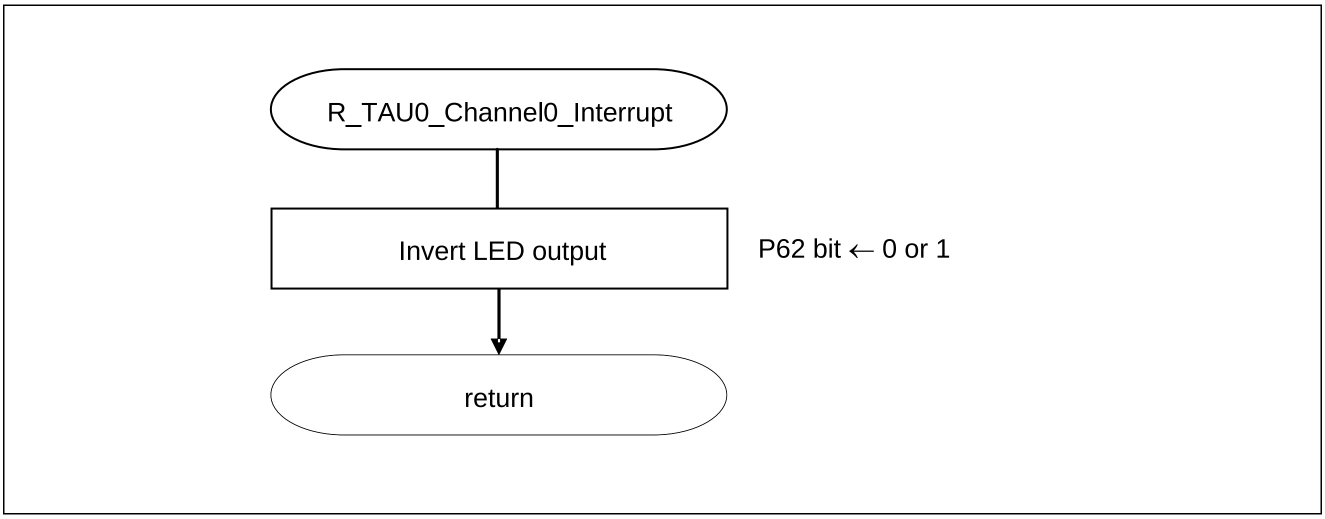

Figure 5.25 Processing Interval Timer Interrupt of Channel 0 of TAU0

5.8.25 Processing of INTP0 External Interrupt

Figure 5.26 shows the flowchart for processing INTP0 external interrupt.

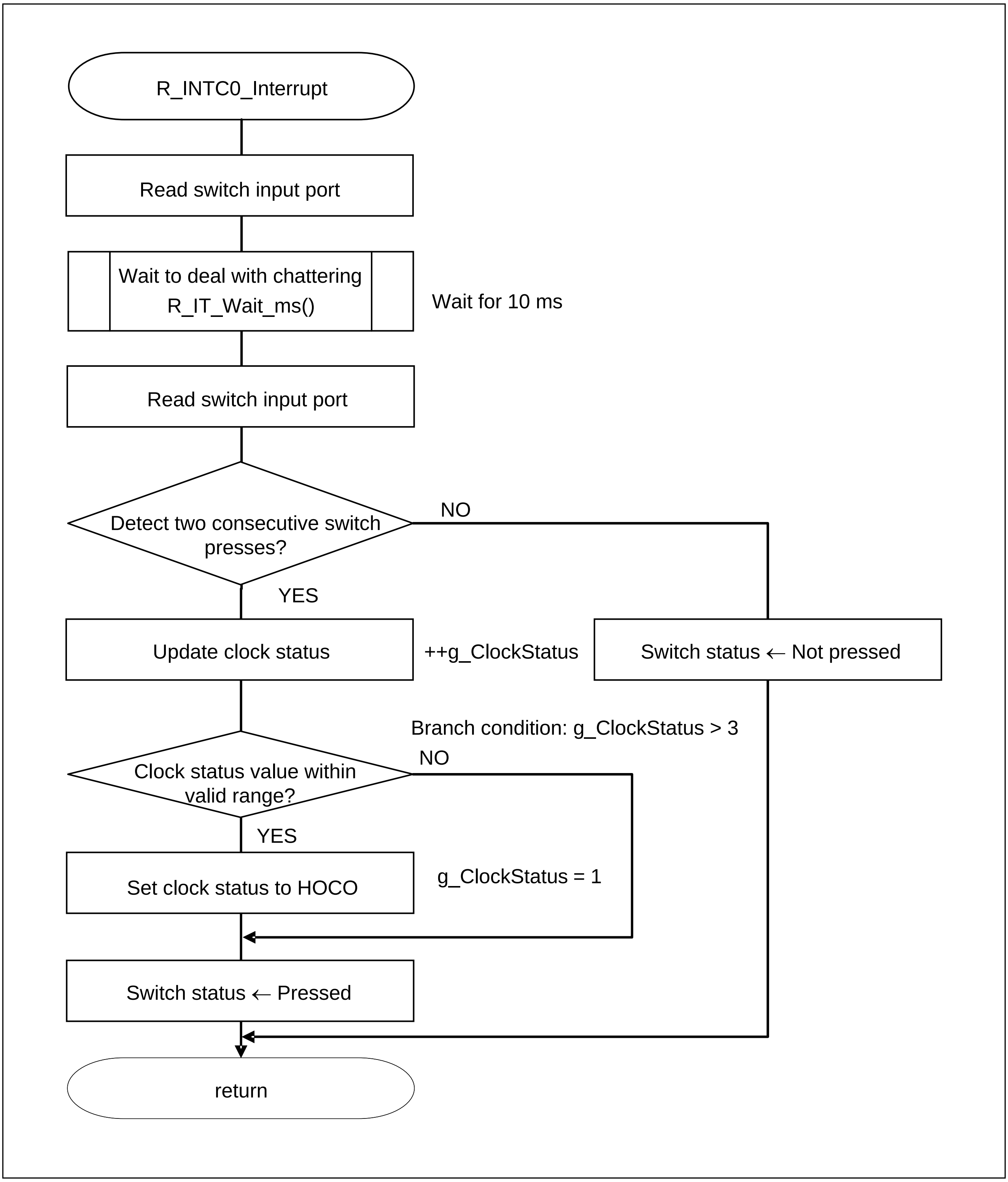

Figure 5.26 Processing of INTP0 External Interrupt

5.8.26 Waiting in Units of 1 ms

Figure 5.27 shows the flowchart for waiting in units of 1 ms.

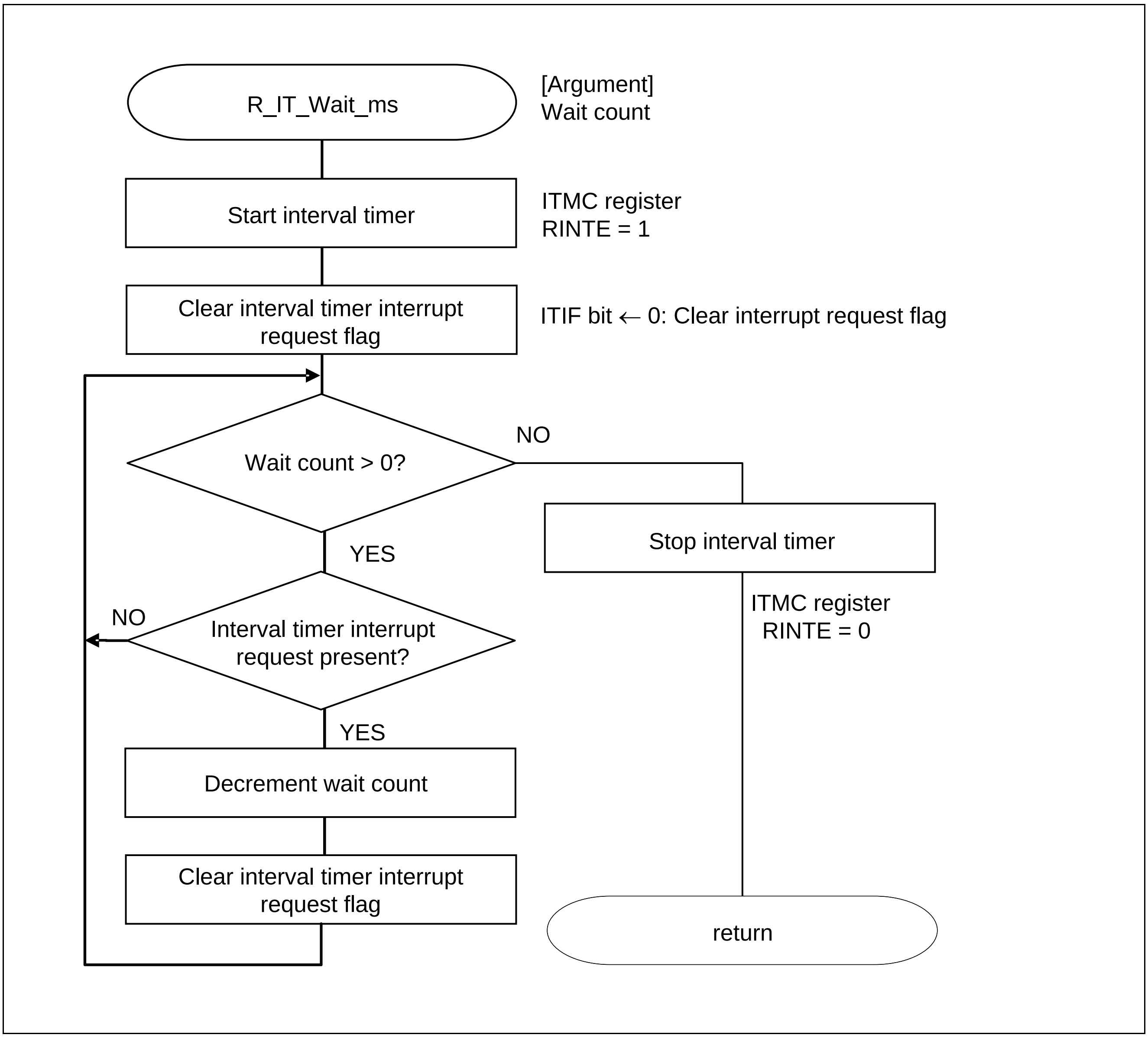

Figure 5.27 Waiting in Units of 1 ms

6. Sample Code

The sample code is available on the Renesas Electronics Website.

7. Documents for Reference

RL78/G13 User's Manual: Hardware (R01UH0146E)

RL78 Family User's Manual: Software (R01US0015E)

(The latest versions of the documents are available on the Renesas Electronics Website.)

Technical Updates/Technical Brochures

(The latest versions of the documents are available on the Renesas Electronics Website.)

Website and Support

Renesas Electronics Website

- http://www.renesas.com/index.jsp

Inquiries

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

Revision Record	RL78/G13 Clock Generator (Clock Switching)

Rev.	Date	Description	
		Page	Summary
1.00	Feb. 08, 2012	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

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

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

3. Prohibition of Access to Reserved Addresses
- Access to reserved addresses is prohibited.
  - The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals
- After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
  - When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products
- Before changing from one product to another, i.e. to one with a different part number, confirm that the change will not lead to problems.
  - The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

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

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

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

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

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

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

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

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

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

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

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

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

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