RL78/G12

Flash Data Library Type04

R01AN1363EJ0110
Rev. 1.10
June 01, 2016

Introduction

This application note explains how to writes and reads data to and from data flash memory using the Flash Data Library Type 04 (flash data library).

This application note is intended for the use of the R5F10266 and R5F10366, which have a smaller RAM capacity than any other microcontrollers belonging to the RL78/G12 group. The RAM capacity is limited to 256 bytes.

Target Device

RL78/G12

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 note explains how to use the flash data library.

The sample program covered in this document reads the time for keeping a switch pressed from the data flash memory and flashes LED2 at the same interval as the time. When the switch is kept pressed, this sample program starts counting the time for keeping it pressed. While the switch is kept pressed, LED1 is turned on to indicate the time is being counted. The time is limited to 10 seconds or less. If the switch is kept pressed for over 10 seconds, the sample program determines it is kept pressed for 10 seconds and performs processing. After the switch exits the pressed status, the sample program determines the time for keeping it pressed. The sample program turns off LED1 and writes the time to the data flash memory.

Table 1.1 lists the peripheral functions to be used and their uses.

Table 1.1 Peripheral Functions to be Used and their Uses

Peripheral Function	Use
Port I/O	Turns on and off LED1 and LED2.
Channel 0 of timer array unit 0	Switches LED2 flash.
Channel 1 of timer array unit 0	Counts the time for keeping a switch pressed. Generates the wait time for avoiding chatters.
External interrupt input (INTP0)	Starts counting the time for keeping a switch pressed.

1.1 Outline of the Flash Data Library

The flash data library is a software library that is used to manipulate the data flash memory using the firmware installed on the RL78 microcontroller.

The flash data library carries out the reprogramming and reading of the data flash memory by being called by the user program.

1.2 Hardware Environment of the Flash Data Library

The Flash Data Library Type04 for the RL78 microcontrollers controls the reprogramming of the data flash memory using a sequencer. Since the control of data flash memory is carried out by the sequencer, it is possible to run user programs while the data flash memory is being controlled. This is referred to as BGO (background operation).

Although the data flash memory cannot be referenced while it is being subjected to reprogramming, the code flash memory can be referenced during that period. Consequently, interrupt processing routines, user programs, and the Flash Data Library Type04 can be allocated to ROM as usual for execution.

Figure 1.1 shows a reprogramming state of data flash memory. Figure 1.2 shows a reprogramming control example of data flash memory.

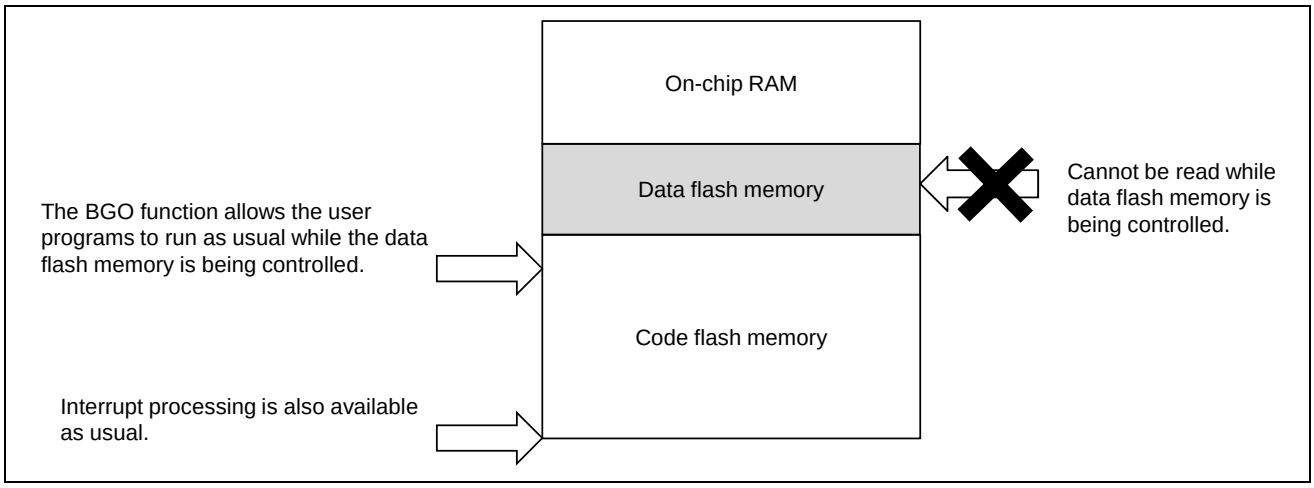

Figure 1.1 Reprogramming State of Data Flash Memory

Control is returned to the user program immediately after a call for executing the required processing is made to the sequencer of the RL78 microcontroller. For the result of controlling the data flash memory, the user program needs to check the data flash memory control state by calling a status check function (PFDL_Handler function).

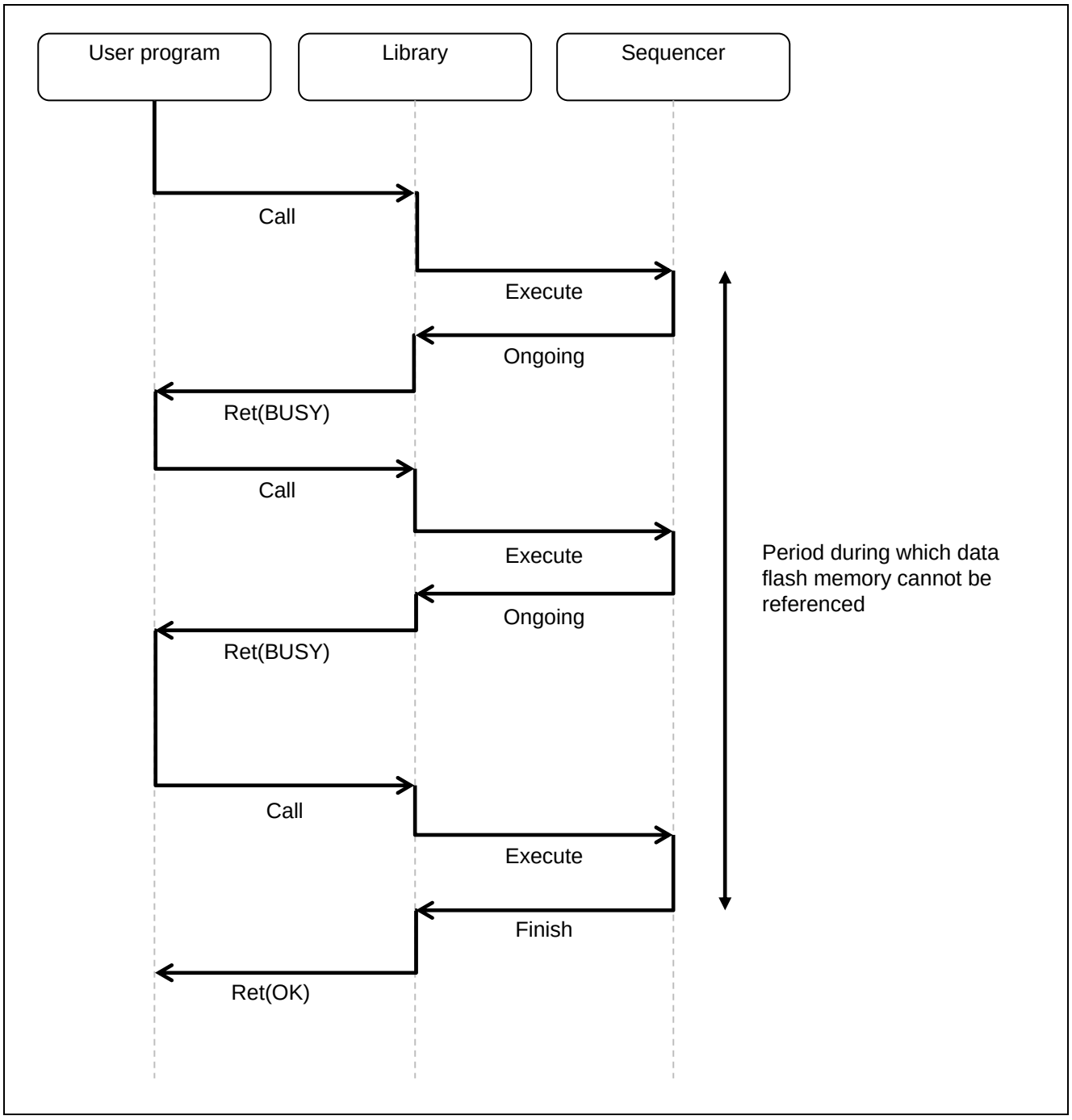

Figure 1.2 Reprogramming Control Example of Data Flash Memory

1.2.1 Data Flash Memory

The configuration of the data flash memory for the RL78/G12 (R5F1026A) is shown below.

The flash memory of the RL78 microcontrollers is divided into 1-Kbyte blocks. The flash data library performs erase processing on the data flash memory on a block basis. It specifies the start address and the execution size when performing read, write, blank check, and internal verify processing.

Figure 1.3 shows the placement of blocks in data flash memory and their block numbers.

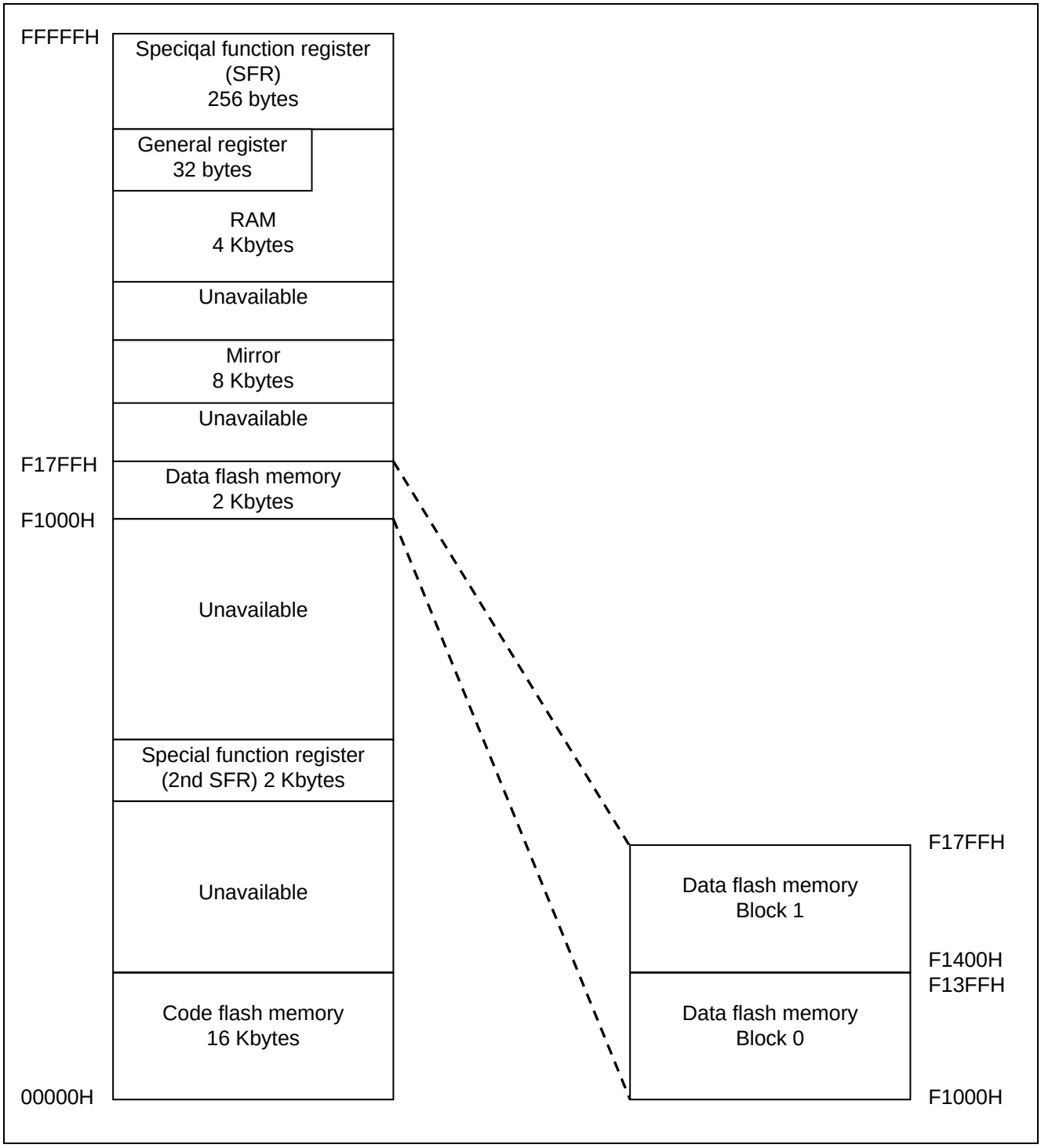

Figure 1.3 Placement of Blocks in Data Flash Memory and their Block Numbers

1.3 Software Environment of the Flash Data Library

The Flash Data Library Type04 consumes the volume of program area equal to the size of the library to be used to allocate the correspondent program to the user area. The Flash Data Library Type04 uses the CPU, stack, and data buffer.

1.3.1 Self-RAM

The Flash Data Library Type04 sometimes uses 1 Kbyte of RAM as a work area. This area is called the self-RAM when it is used as a work area. Since it is defined within the library, the user needs to make no setting for this area.

Data in the self-RAM area is rewritten by calling a flash data library function.

1.3.2 Register

The Flash Data Library Type04 uses the general registers, ES/CS registers, SP, and PSW on the register bank that is selected by the user.

1.3.3 Stack Data Buffer

The Flash Data Library Type04 uses a sequencer to perform programming into the data flash memory. It uses the CPU for preliminary configuration and control. Accordingly, the stack that is designated by the user program is required to use the Flash Data Library Type04.

Caution: Link directives are used to allocate the stack and data buffer to the user-specified addresses.

- Stack
  It is necessary to reserve in advance the size of stack area necessary for the flash data library functions in addition to the size of the stack to be used by the user programs and allocate it in such a manner that the RAM that is being used by the user is not destroyed during the stack processing that is executed for the Flash Data Library Type04. The areas for the stack that can be specified are the self-RAM and the internal RAM area excluding addresses FFE20H to FFEFFH.
- Data buffer
  The uses of the data buffer are listed below.
  — As the work area for internal processing of the Flash Data Library Type04
  — As the area for storing the programming data in write mode
  As the area for storing the read data in read mode
  The start address of the data buffer must fall within the self-RAM or the internal RAM area excluding addresses FFE20H to FFEFFH as for the stack.

1.4 How to Get the Flash Data Library

Before compiling the sample program, please download the latest flash data library and copy the library files to the following folders below “Workspace”.

incrl78 folder : pfdl.h, pfdl.inc, pfdl_types.h

librl78 folder : pfdl.lib

The Flash Data library is available on the Renesas Electronics Website.

Please contact your local Renesas Electronics sales office or distributor for more information.

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/G12 (R5F1026A)
Operating frequency	• High-speed on-chip oscillator (HOCO) clock: 24 MHz • CPU/peripheral hardware clock: 24 MHz
Operating voltage	5.0 V (Operation is possible over a voltage range of 2.9 V to 5.5 V.) LVD operation (V_{LVD}): Reset mode which uses 2.81 V (2.76 V to 2.87 V)
Integrated development environment	CS+ V3.02.00 from Renesas Electronics Corp.
C compiler	CA78K0R V1.72 from Renesas Electronics Corp.
Board to be used	RL78/G12 target board (QB-R5F1026A-TB)
Flash memory self-programming library (Type, Ver)	FDLRL78 Type04, Ver1.04 Note

Note: Use and evaluate the latest version.

3. Related Application Notes

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

- RL78/G12 Initialization (R01AN1030E) Application Note
- RL78 Microcontroller Flash Data Library Type04 (R01AN0608E) 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.

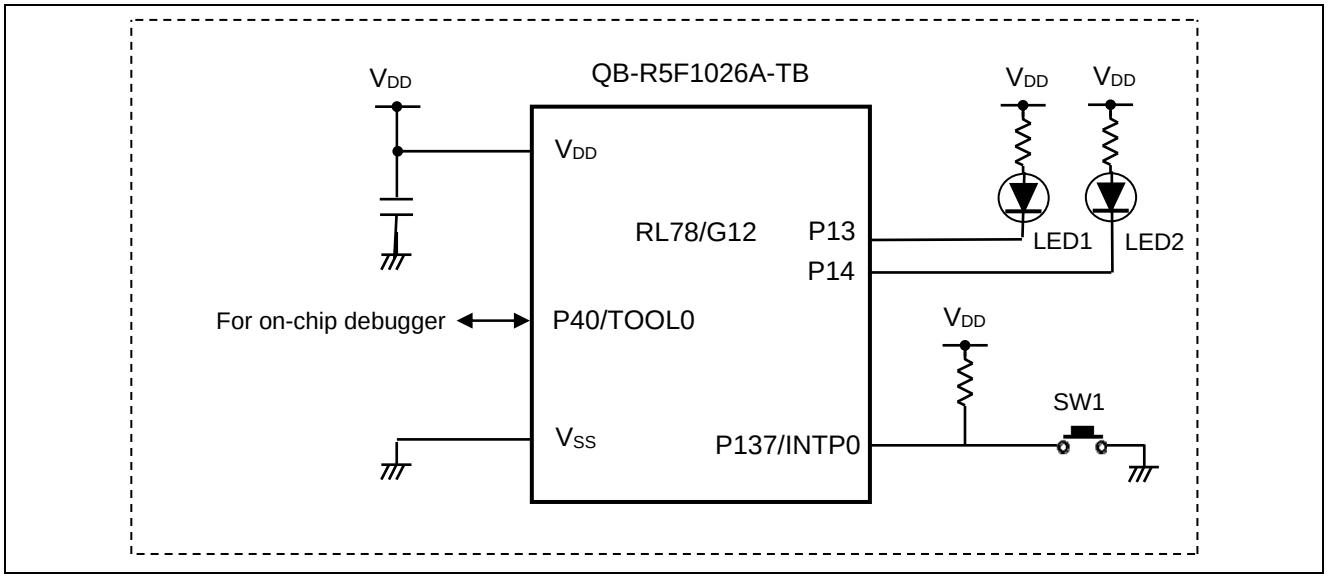

Figure 4.1 Hardware Configuration

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. V_{DD} must be held at not lower than the reset release voltage (V_{LVD}) that is specified as LVD.

4.2 List of Pins to be Used

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

Table 4.1 Pins to be Used and their Functions

Pin Name	I/O	Description
P13	Output	LED1 on/off control
P14	Output	LED2 on/off control
P137/INTP0	Input	Switch press

5. Description of the Software

5.1 Operation Outline

This application note explains how to use the flash data library.

The sample program covered in this document reads the time for keeping a switch pressed from the data flash memory and flashes LED2 at the same interval as the time. When the switch is kept pressed, this sample program starts counting the time for keeping it pressed. While the switch is kept pressed, LED1 is turned on to indicate the time is being counted. The time is limited to 10 seconds or less. If the switch is kept pressed for over 10 seconds, the sample program determines it is kept pressed for 10 seconds and performs processing. After the switch exits the pressed status, the sample program determines the time for keeping it pressed. The sample program turns off LED1 and writes the time to the data flash memory.

(1) Sets up the I/O port.

<Setting conditions>

- LED on/off control ports (LED1 and LED2): Sets P13 and P14 for output.

(2) Sets up the TAU0.

<Setting conditions>

- Uses the TAU0 channels 0 and 1.
- Uses the 5.86-kHz operation clock in channel 0 and 6-MHz operation clock in channel 1.
- Enables only the software start trigger as the start trigger.
- Sets the enable edge to the falling edge.
- Sets the operation mode to interval timer mode.
- Sets the count start and interrupt settings to “Generates no timer interrupt at the beginning of counting.”
- Uses the timer interrupts (INTTM00 and INTTM01).
- Sets the interrupt timing to 10 ms in channels 0 and 1.
- Sets the interrupt priority level to 1 in channels 0 and 1.

(3) Sets up the external interrupt input.

<Setting conditions>

- Sets the interrupt priority level to 1.
- Sets the enable edge to the falling edge.

(4) Enables interrupts.

(5) Starts the INTP.

- Enables edge detection interrupt processing of the INTP0 pin.
- Enables interrupt of the INTP0 pin.

(6) Reads the contents of the data flash memory.

- Reads 2-byte data from addresses 0xF1000 to 0xF1001 of the data flash memory.
- Changes the read value to 1000 if it is greater than 1000.

(7) Updates the timing to flashing LED2.

- Sets the TDR00 register value so that the interval for the interval timer of the TAU0 channel 0 is set to “a value read from the data flash memory * 10 ms.”

(8) Starts the TAU0 channel 0.

(9) Switches into HALT mode and waits for a press of the switch.

- Inverts an LED2 state and enters the HALT mode again if the sample program returns from the HALT mode upon a TAU0 channel 0 interrupt request.

(10) When a switch-triggered external interrupt occurs, exits the HALT mode and takes the following actions to avoid chatters:

- Starts the TAU0 channel 1.
- Waits until a TAU0 channel 1 interrupt occurs.
- Has the TAU0 channel 1 interrupt handler test the switch status. More specifically, the interrupt handler checks the level of the input at P137.
- If the level of P137 is 0, sets the switch-pressed flag and sets the number of interrupts to 1, determining that a switch has been pressed.
- If the level of P137 is 1, clears the switch-pressed flag and returns to step (9), determining that none of the switches have been pressed.

(11) Stops the TAU0 channel 0, turns off LED2, and turns on LED1.

(12) Switches into HALT mode and waits until the switch exits pressed status.

- Returns from the HALT mode upon a TAU0 channel 1 interrupt which is input every 10 ms during the switch-pressed status.
- Has the TAU0 channel 1 interrupt handler test the switch status.
- If the level of P137 is 0, counts interrupts and returns to step (12), determining that a switch is being pressed.
  * If the switch is kept pressed for over 10 ms, the sample program does not count interrupts.
- If the level of P137 is 1, clears the switch-pressed flag, determining that the switch exits pressed status.

(13) Stops the TAU0 channel 1 if the switch exits pressed status.

(14) Writes the time for keeping the switch pressed (the number of TAU0 channel 1 interrupts) to addresses 0xF1000 to 0xF1001 of the data flash memory.

- If the block initialization fails, the sample program turns on LED1 and LED2 and suppresses the execution of the subsequent operations.
- If the programming fails, the sample program turns on LED1 and LED2 and suppresses the execution of the subsequent operations.
- After programming the write value, the sample program reads it and compares the read value with the write value.
- If the write value and read value do not match, the sample program turns on LED1 and LED2 and suppresses the execution of the subsequent operations.

(15) Returns to step (7).

5.2 File Configuration

Table 5.1 shows the file configuration. It lists the files added to the project, excluding the files attached to the flash data library (FDLRL78 Type04, Ver1.04).

Table 5.1 File Configuration

File Name	Outline	Remarks
r_main.asm	Main processing	Subroutines: main IINTTM00 IINTTM01 IINTP0
r_init.asm	Initial setting	Subroutines: SSTARTINTP0 SSTOPINTP0 SSTARTINTV0 SSTOPINTV0 SSTARTINTV1 SSTOPINTV1
r_pfdl.asm	Flash data library execution	Subroutines: SFDLINIT SFDLBLANKCHECK SFDLERASE SFDLVERIFY SFDLWRITE SFDLREAD SFDLFIRSTREAD SFDLEXECUTEWRITE
r_lk_dr	Link directive file	-

5.3 List of Option Byte Settings

Table 5.2 summarizes the settings of the option bytes.

Table 5.2 Option Byte Settings

Address	Setting	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	11100000B	HS mode, HOCO: 24 MHz
000C3H/010C3H	10000100B	Enables the on-chip debugger Erases the data in the flash memory when on-chip debug security ID authentication fails.

The option bytes of the RL78/G12 comprise the user option bytes (000C0H to 000C2H) and on-chip debug option byte (000C3H).

The option bytes are automatically referenced and the specified settings are configured at power-on time or the reset is released.

The option bytes must be specified from "User Option Byte Values" on the "Device" panel in the "Link Options" of CS+. Set "Set up user option bytes" to "Yes (-gb)."

5.4 Link Directive File

The link directive file is used to reserve the RAM area to be used by the flash self-programming library so that it is not available as the standard RAM area.

The outline of the link directive file that this sample program uses is shown below.

```
;************************************************************
; Redefined RAM area
;************************************************************
; ------------------------------------------------------------
; Redefined default data segment RAM
; ------------------------------------------------------------
MEMORY RAM          : ( 0FFC8AH, 0196H )
; ------------------------------------------------------------
; Define new memory entry for saddr area
; ------------------------------------------------------------
MEMORY RAM_SADDR : ( 0FFE20H, 01E0H )
```

Set up a library-dedicated area so that it is not to be used as the standard RAM area.

5.5 List of Constants

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

Table 5.3 Constants for the Sample Program

Constant	Setting	Description
PFDL_NG	01H	Failure of FDL function
WRITE_SIZE	02H	Number of bytes written to data flash memory
WRITE_ADDRESS Note	0000H	Target write address of data flash memory

Note: Change the target address in the range of 0000H to 03FEH.

5.6 List of Variables

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

Table 5.4 Global Variables for the Sample Program

Type	Variable Name	Contents	Subroutine Used
8 bytes	RARG	Argument storage variable	SFDLINIT SFDLBLANKCHECK SFDLERASE SFDLVERIFY SFDLWRITE SFDLREAD
2 bytes	RWRITEVALUE	Read value	IINTTM01 SFDLWRITE SFDLEXECUTEWRITE
2 bytes	RREADVALUE	Write value	main SFDLREAD SFDLFIRSTREAD SFDLEXECUTEWRITE
1 byte	RPUSHFLAG	Switch press start flag	main IINTTM01

5.7 List of Functions (Subroutines)

Table 5.5 summarizes the functions (subroutines) that are used in this sample program.

Table 5.5 List of Functions (Subroutines)

Function Name	Outline
SSTARTINTP0	Starts INTP0.
SFDLFIRSTREAD	Processes first read of data from flash data library
SFDLINIT	Starts flash data library.
SFDLREAD	Processes data read command.
SSTARTINTV0	Starts TAU0 channel 0.
IINTTM00	TAU0 channel 0 interrupt
SSTOPINTV0	Stops TAU0 channel 0.
IINTP0	INTP0 external interrupt
SSTARTINTV1	Starts TAU0 channel 1.
IINTTM01	TAU0 channel 1 interrupt
SSTOPINTV1	Stops TAU0 channel 1.
SFDLEXECUTEWRITE	Executes writing.
SFDLBLANKCHECK	Processes blank check command.
SFDLERASE	Processes block erase command.
SFDLWRITE	Processes data write command.
SFDLVERIFY	Processes verify command.

5.8 Function (Subroutine) Specifications

This section describes the specifications for the functions (subroutines) that are used in the sample program.

[Function Name] SSTARTINTP0

Synopsis	Start INTP1.
Explanation	This function starts the INTP1.
Arguments	None
Return value	None
Remarks	None

[Function Name] SFDLFIRSTREAD

Synopsis	Process first read of data from data flash memory.
Explanation	This function processes the first read of data from the data flash memory. The function changes the read value to 1000 if the value is greater than 1000.
Arguments	None
Return value	None
Remarks	None

[Function Name] SFDLINIT

Synopsis	Start flash data library.
Explanation	This function initializes and starts the RAM to be used by the Flash Data Library Type04.
Arguments	None
Return value	None
Remarks	None

[Function Name] SFDLREAD

Synopsis	Process data write command.
Explanation	This function executes the data write command.
Arguments	None
Return value	None
Remarks	None

[Function Name] SSTARTINTV0

Synopsis	Start TAU0 channel 0.
Explanation	This function starts the TAU0 channel 0.
Arguments	None
Return value	None
Remarks	None

[Function Name] IINTTM00

Synopsis	Process TAU0 channel 0 interrupt.
Explanation	This function turns on and off LED2.
Arguments	None
Return value	None
Remarks	None

[Function Name] SSTOPINTV0

Synopsis	Stop TAU0 channel 0.
Explanation	This function stops the TAU0 channel 0.
Arguments	None
Return value	None
Remarks	None

[Function Name] IINTP0

Synopsis	Process INTP0 external interrupt.
Explanation	This function starts the TAU0 channel 1.
Arguments	None
Return value	None
Remarks	None

[Function Name] SSTARTINTV1

Synopsis	Start TAU0 channel 1.
Explanation	This function starts the TAU0 channel 1.
Arguments	None
Return value	None
Remarks	None

[Function Name] IINTTM01

Synopsis	Process TAU0 channel 1 interrupt.
Explanation	This function sets RWRITEVALUE to 1 if the interrupt is the first one which occurs after a press of a switch. The function increments RWRITEVALUE if it is below 1000.
Arguments	None
Return value	None
Remarks	None

[Function Name] SSTOPINTV1

Synopsis	Stop TAU0 channel 1.
Explanation	This function stops the TAU0 channel 1.
Arguments	None
Return value	None
Remarks	None

[Function Name] SFDLEXECUTEWRITE

Synopsis	Execute writing.
Explanation	This function write the write value to the data flash memory.
Arguments	None
Return value	C register • Normal termination: PFDL_OK • Abnormal termination: PFDL_NG
Remarks	None

[Function Name] SFDLBLANKCHECK

Synopsis	Process blank check command.
Explanation	This function checks the target address to determine if it is blank.
Arguments	None
Return value	C register • Normal termination: PFDL_OK • Idle state: PFDL_IDLE • Blank check error: PFDL_ERR_MARGIN
Remarks	None

[Function Name] SFDLERASE

Synopsis	Process block erase command.
Explanation	This function erases the entire block.
Arguments	None
Return value	C register • Normal termination: PFDL_OK • Idle state: PFDL_IDLE • Block erase error: PFDL_ERR_ERASE
Remarks	None

[Function Name] SFDLWRITE

Synopsis	Process data write command.
Explanation	This function writes data into the data flash memory.
Arguments	None
Return value	• Normal termination: PFDL_OK • Idle state: PFDL_NG • Write error: PFDL_ERR_WRITE
Remarks	None

[Function Name] SFDLVERIFY

Synopsis	Process verify command.
Explanation	This function checks whether the written data is correct.
Arguments	None
Return value	C register • Normal termination: PFDL_OK • Idle state: PFDL_IDLE • Verify error: PFDL_ERR_MARGIN
Remarks	None

5.9 Flowcharts

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

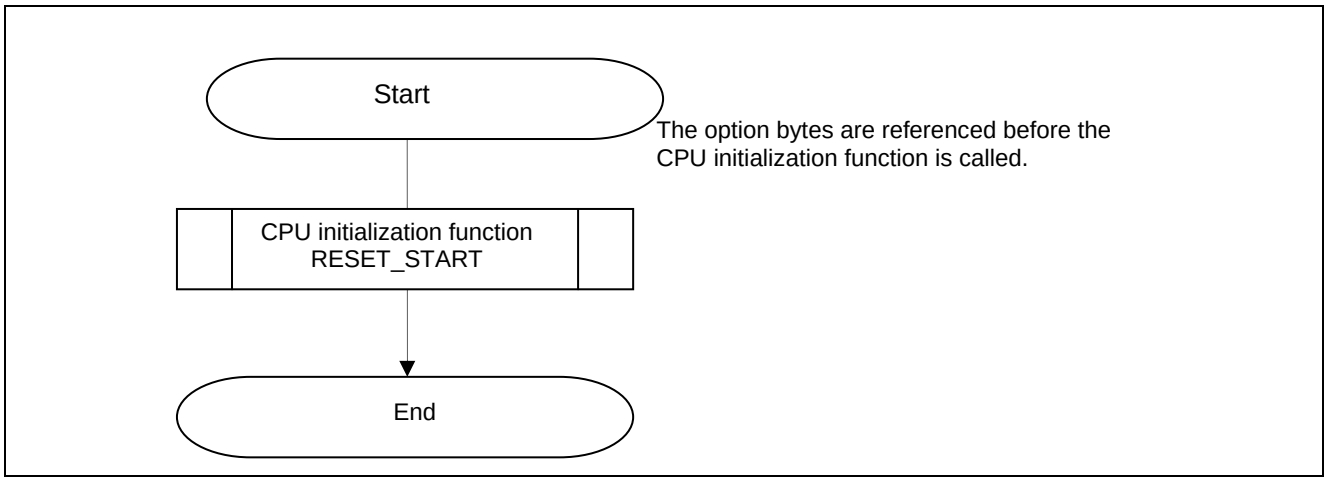

Figure 5.1 Overall Flow

5.9.1 CPU Initialization Function

Figure 5.2 shows the flowchart for the CPU initialization function.

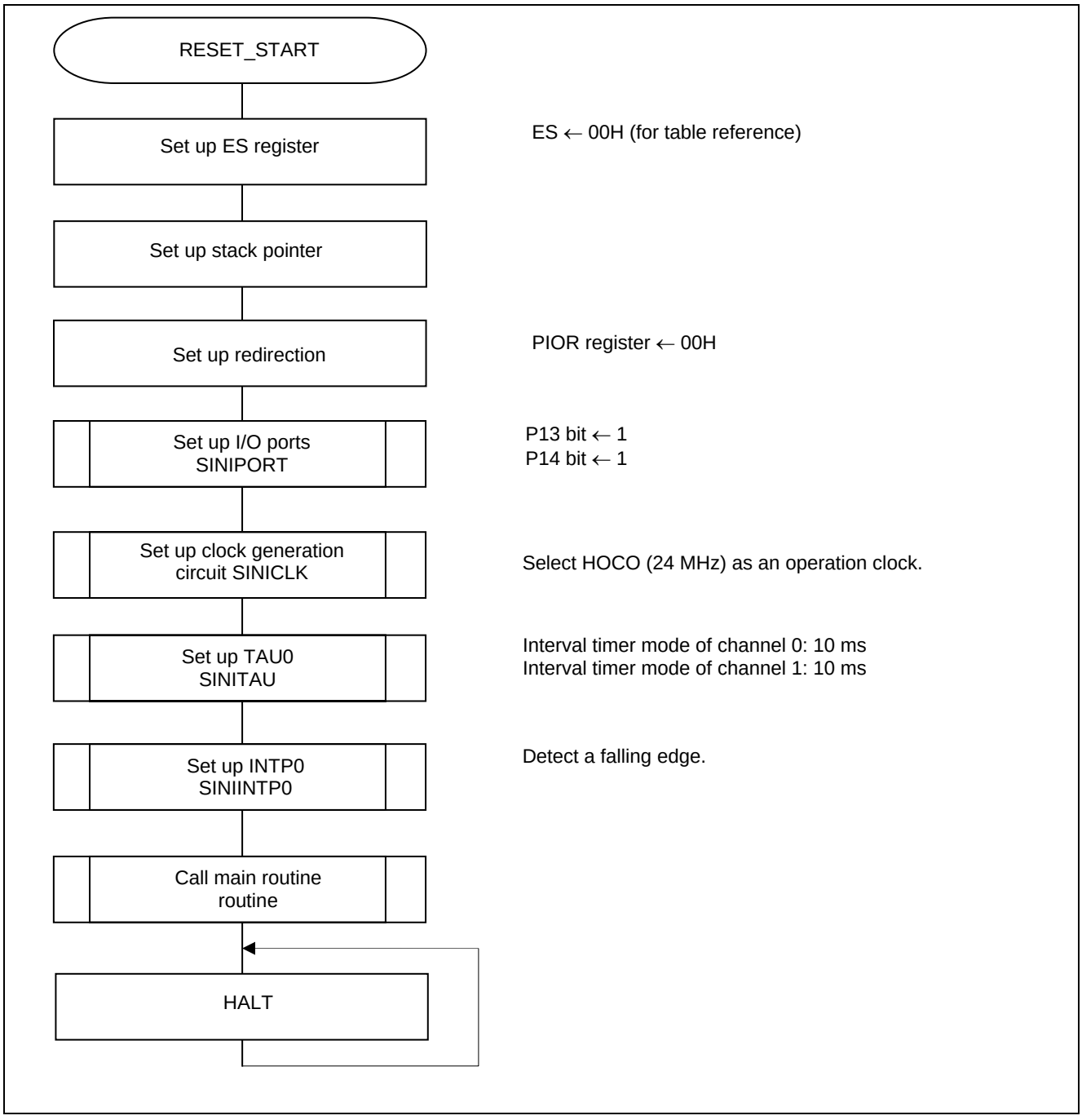

Figure 5.2 CPU Initialization Function

5.9.2 I/O Port Setup

Figure 5.3 shows the flowchart for setting up the I/O ports.

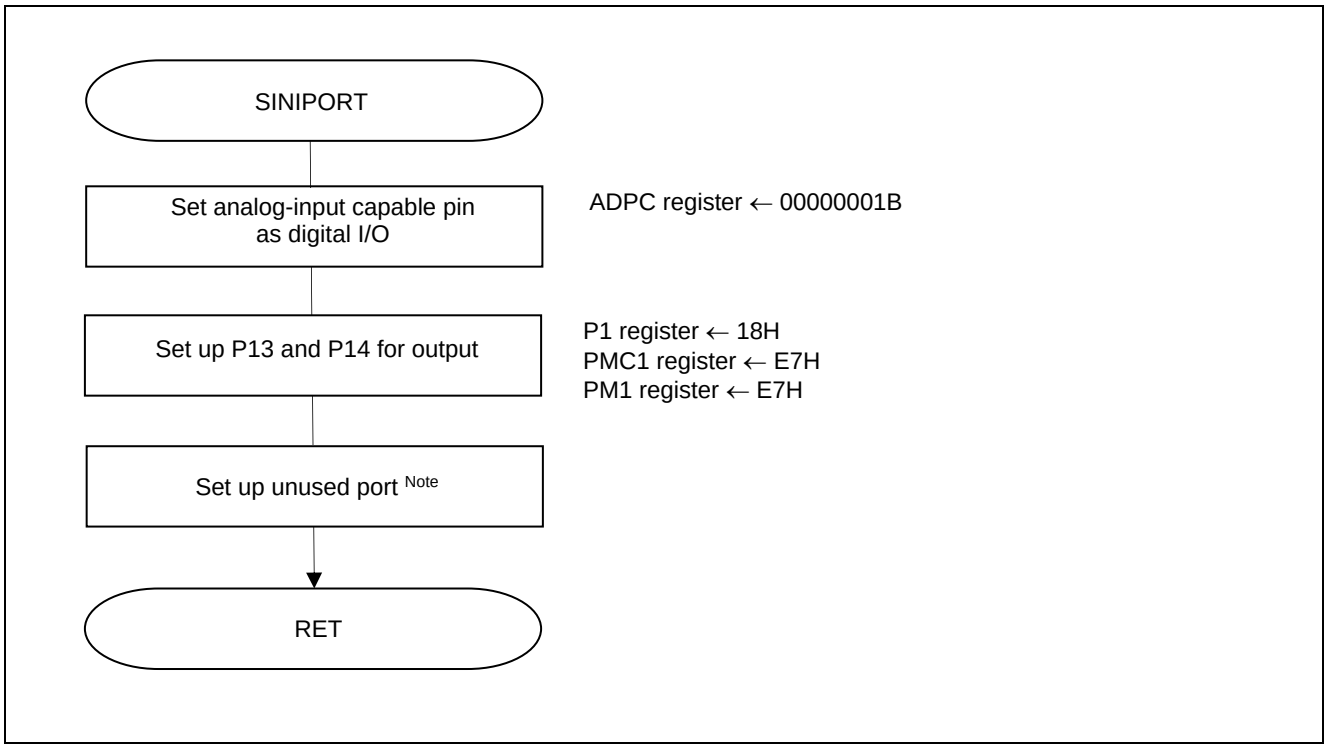

Figure 5.3 I/O Port Setup

Note: Refer to the section entitled "Flowcharts" in RL78/G12 Initialization Application Note (R01AN1030E) for the configuration of the unused ports.

Caution: Provide proper treatment for unused pins so that their electrical specifications are observed. Connect each of any unused input-only ports to V_{DD} or V_{SS} via separate resistors.

5.9.3 Setting up the Clock Generation Circuit

Figure 5.4 shows the flowchart for setting up the clock generation circuit.

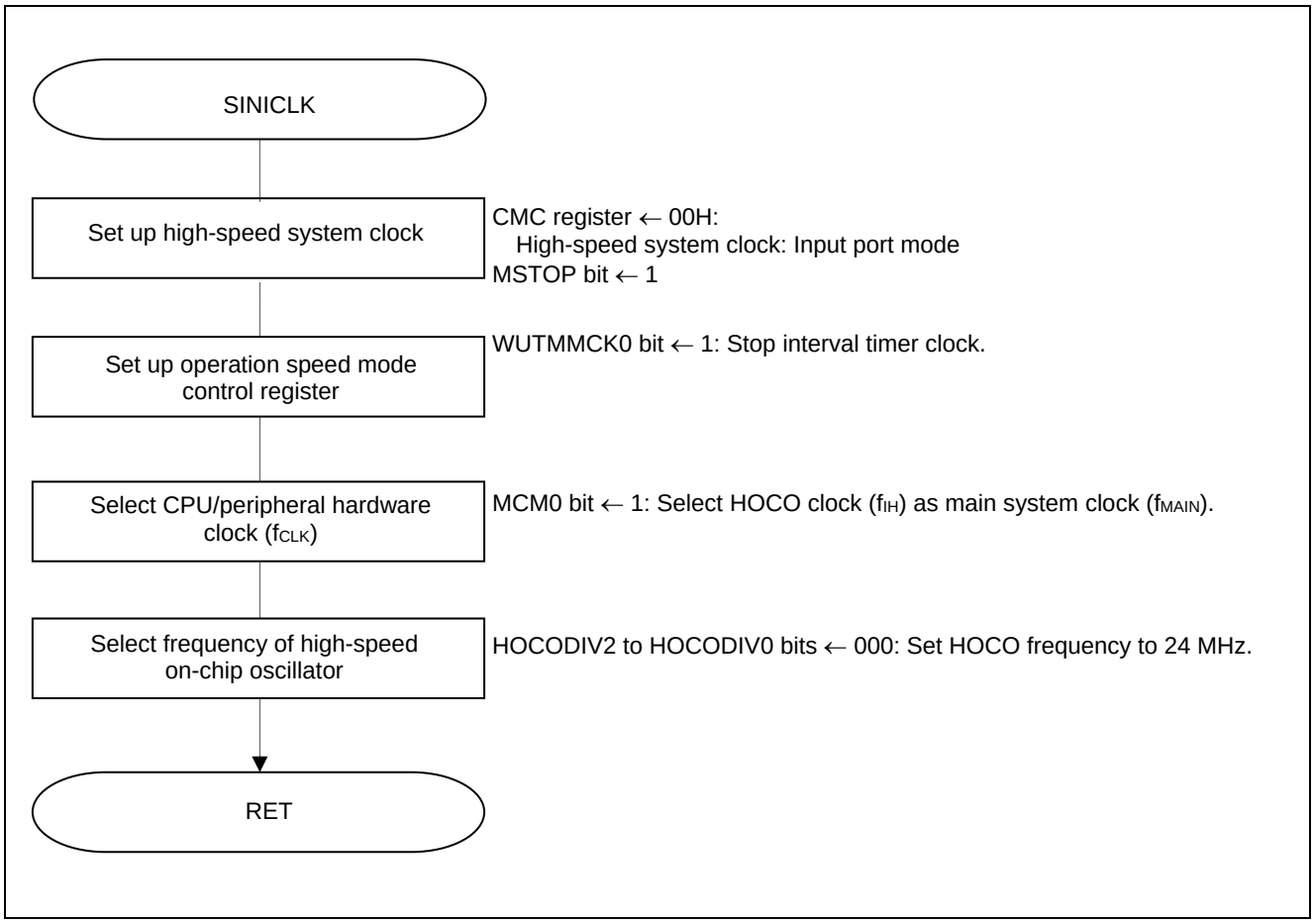

Figure 5.4 Clock Generator Circuit Setup

Caution: For details on the procedure for setting up the CPU clock (SINICLK), refer to the section entitled "Flowcharts" in RL78/G12 Initialization Application Note (R01AN1030E).

5.9.4 Setting up the Timer Array Unit

Figure 5.5 shows the flowchart for setting up the timer array unit.

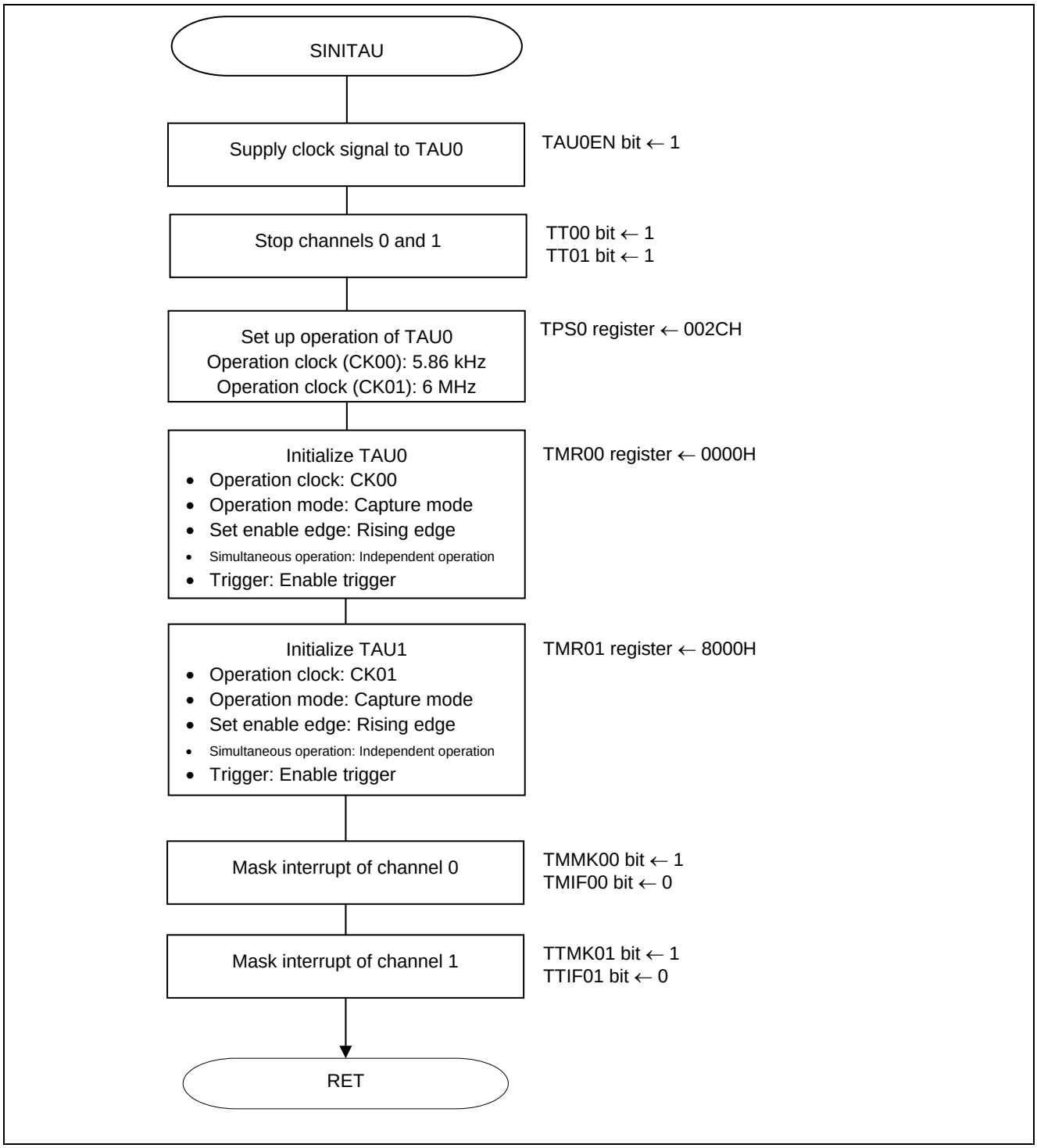

Figure 5.5 Setting up the Timer Array Unit

5.9.5 INTP0 Initialization

Figure 5.6 shows the flowchart for INTP0 Initialization.

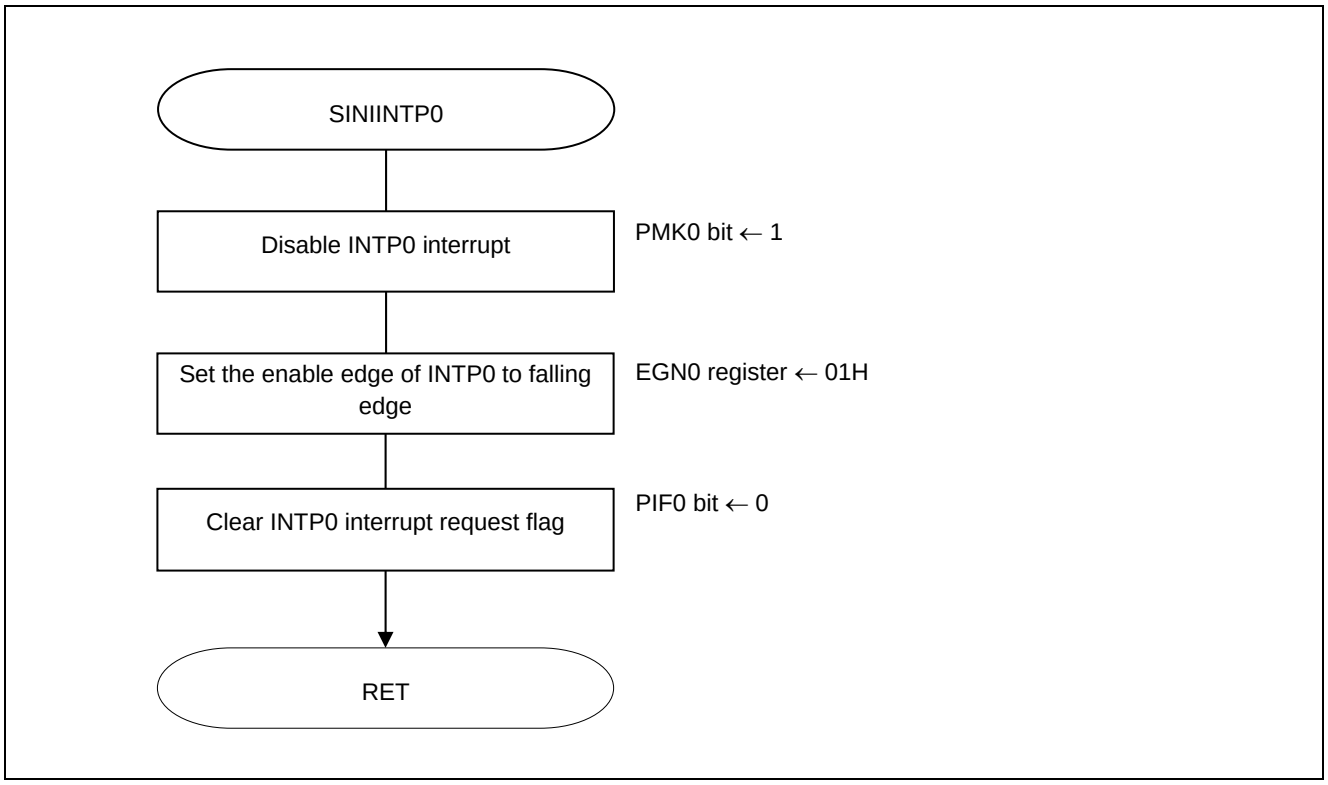

Figure 5.6 INTP0 Initialization

5.9.6 Main Processing

Figures 5.7 and 5.8 show the flowcharts for main processing.

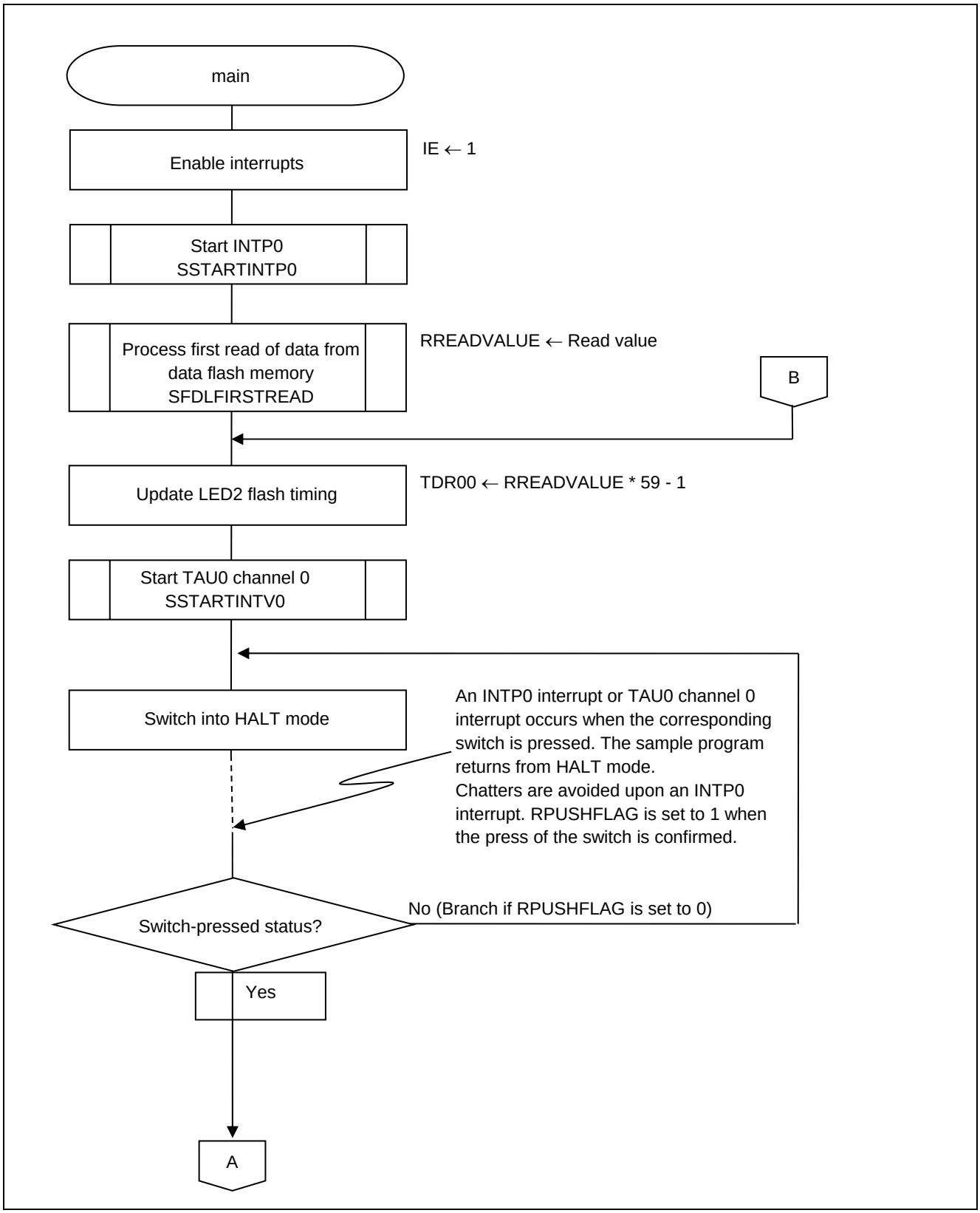

Figure 5.7 Main Processing (1/2)

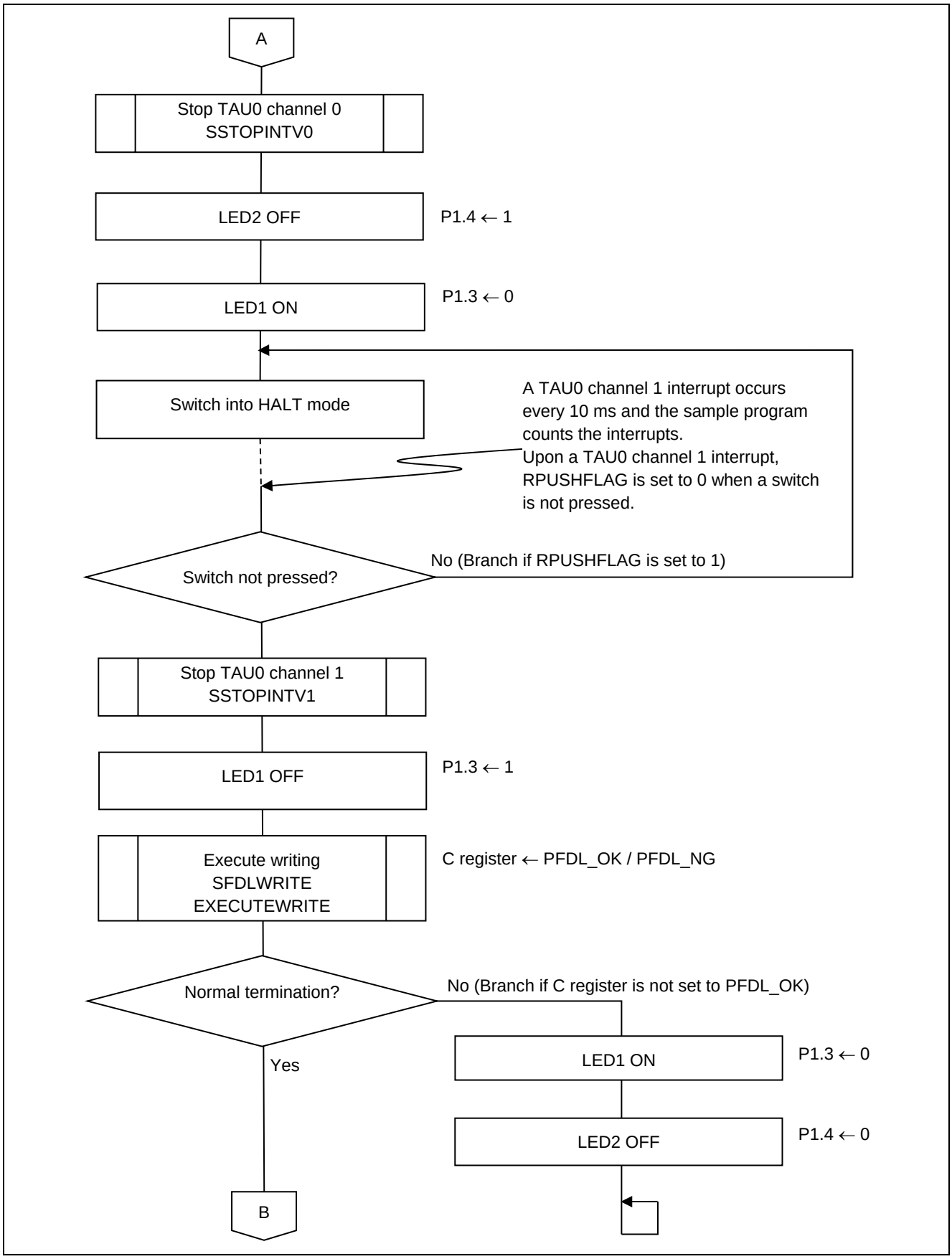

Figure 5.8 Main Processing (2/2)

5.9.7 Starting the INTP0

Figure 5.9 shows the flowchart for starting the INTP0.

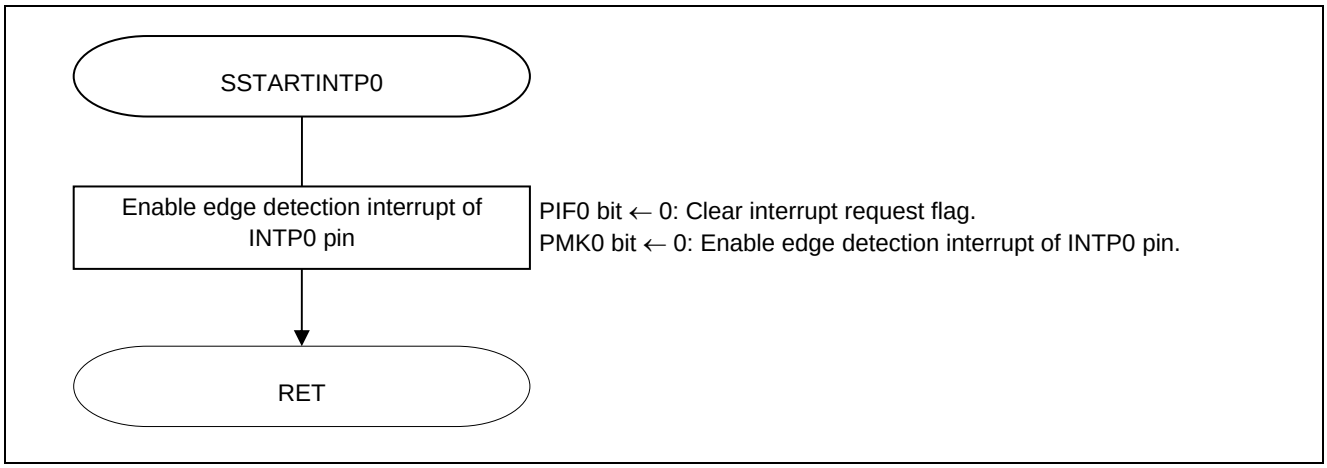

Figure 5.9 Starting the INTP0

5.9.8 Processing the First Read of Data from Data Flash Memory

Figure 5.10 shows the flowchart for processing the first read of data from the data flash memory.

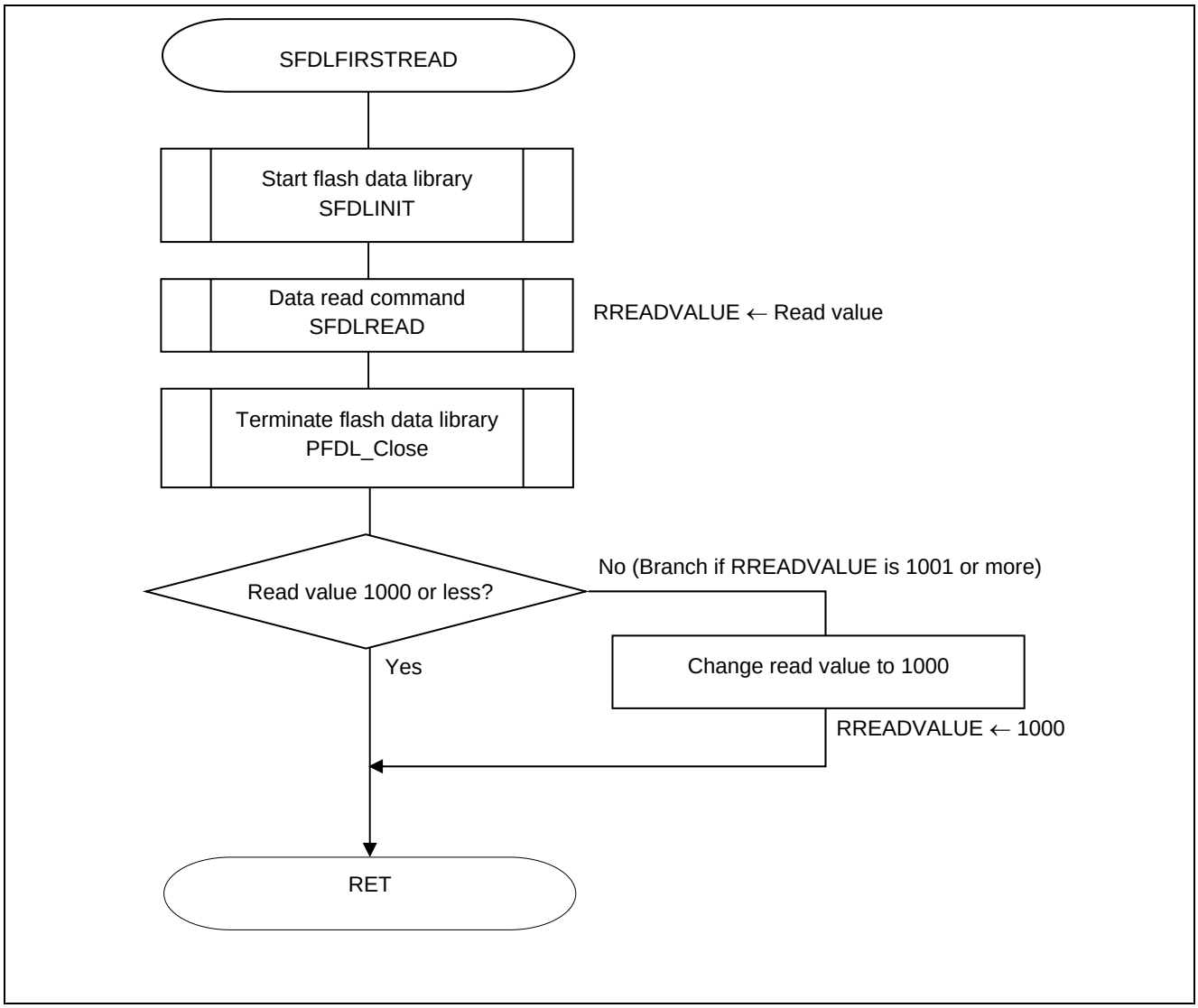

Figure 5.10 Processing the First Read of Data from Data Flash Memory

5.9.9 Starting the Flash Data Library

Figure 5.11 shows the flowchart for starting the flash data library.

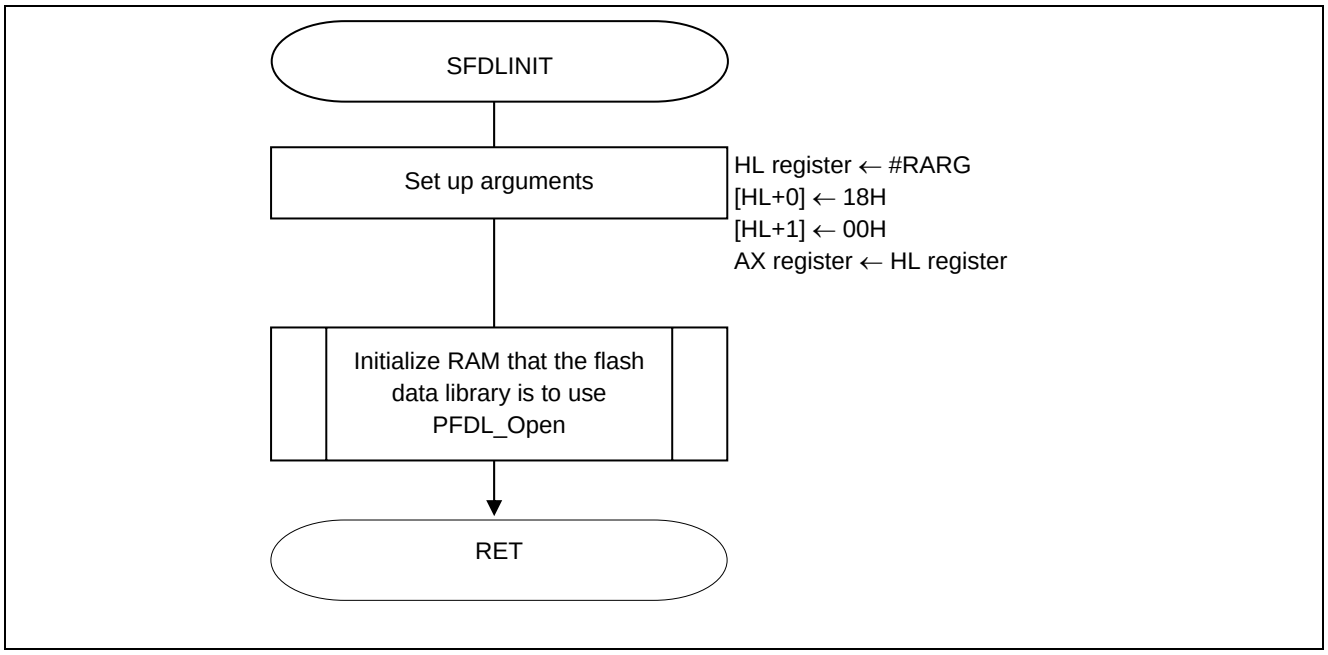

Figure 5.11 Starting the Flash Data Library

5.9.10 Processing the Data Read Command

Figure 5.12 shows the flowchart for processing the data read command.

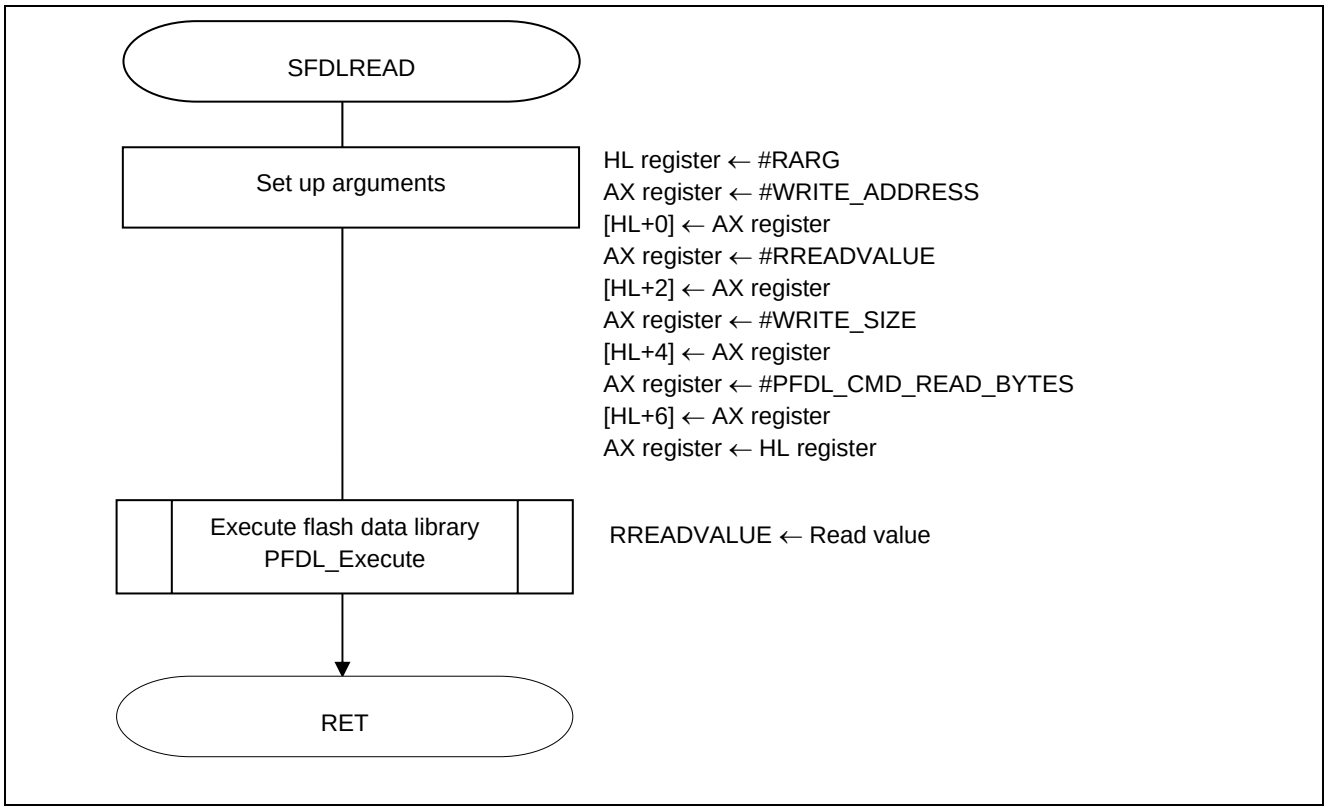

Figure 5.12 Processing the Data Read Command

5.9.11 Starting the TAU0 Channel 0

Figure 5.13 shows the flowchart for starting the TAU0 channel 0.

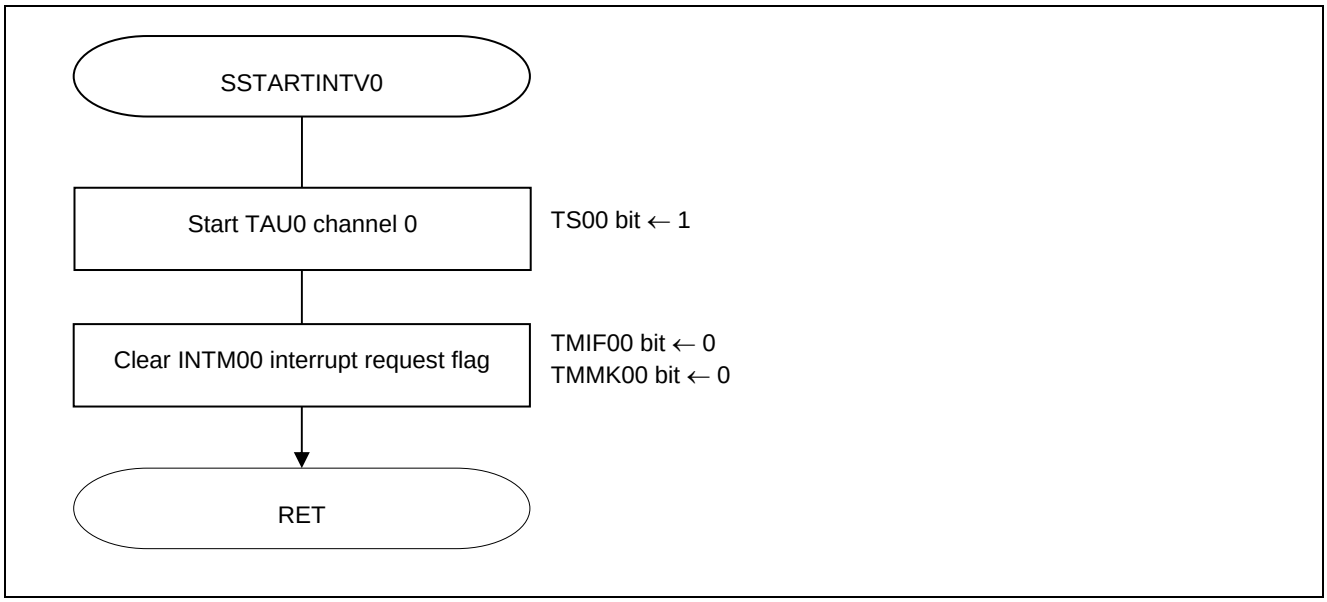

Figure 5.13 Starting the TAU0 Channel 0

5.9.12 Processing the TAU0 Channel 0 Interrupt

Figure 5.14 shows the flowchart for processing the TAU0 channel 0 interrupt.

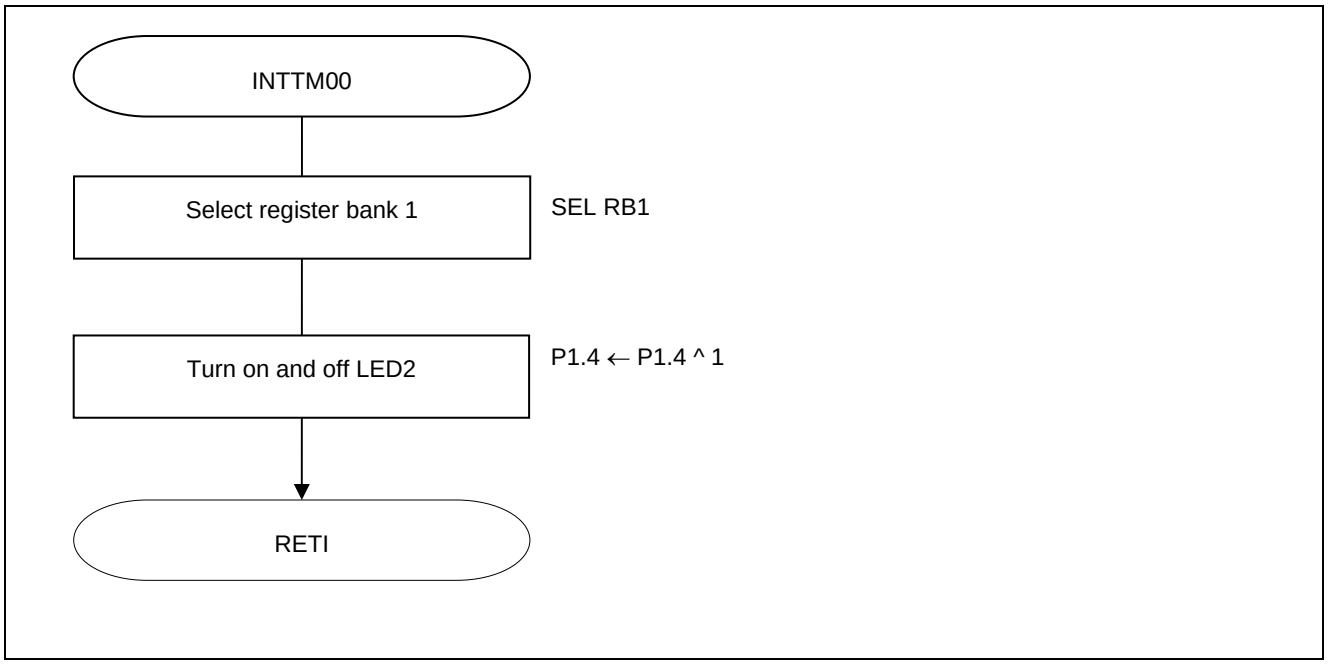

Figure 5.14 Processing the TAU0 Channel 0 Interrupt

5.9.13 Stopping the TAU0 Channel 0

Figure 5.15 shows the flowchart for stopping the TAU0 channel 0.

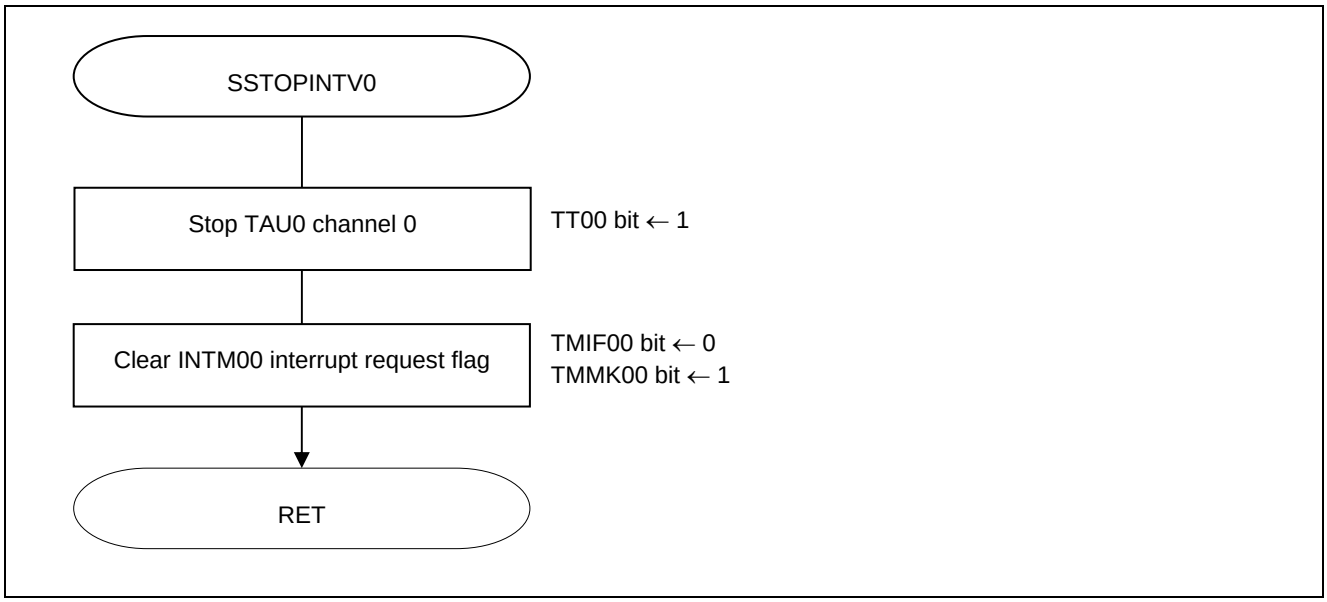

Figure 5.15 Stopping the TAU0 Channel 0

5.9.14 Processing the INTP0 External Interrupt

Figure 5.16 shows the flowchart for processing the INTP0 external interrupt.

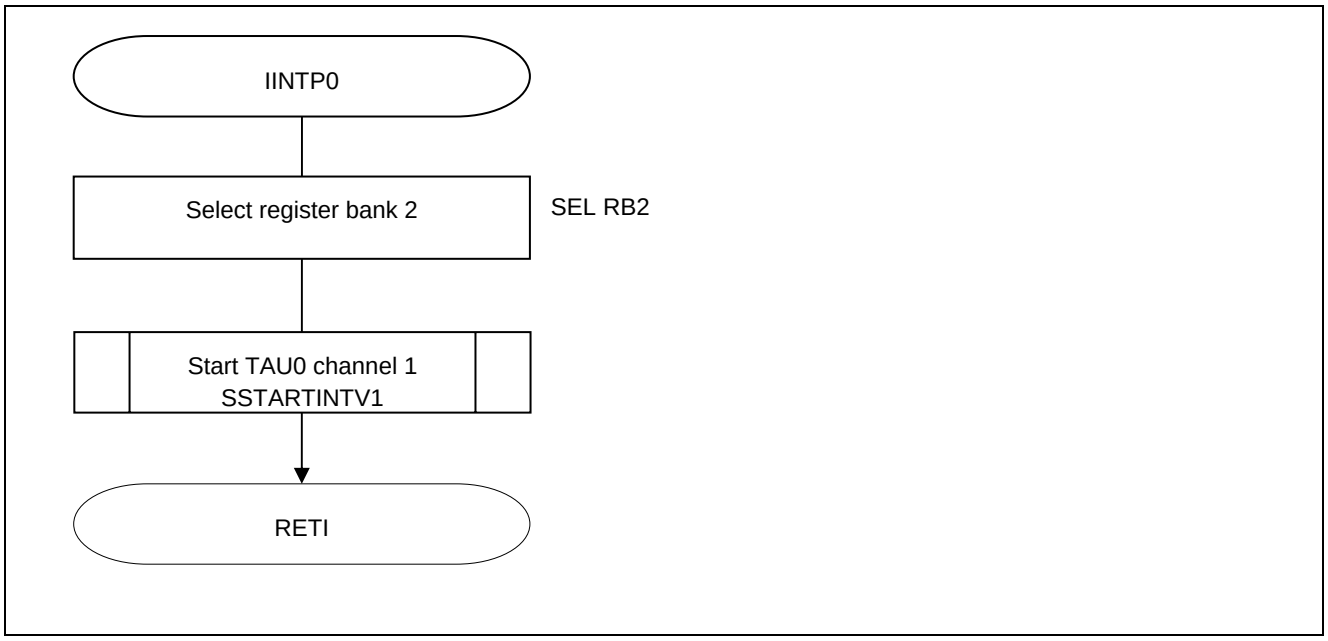

Figure 5.16 Processing the INTP0 External Interrupt

5.9.15 Starting the TAU0 Channel 1

Figure 5.17 shows the flowchart for starting the TAU0 channel 1.

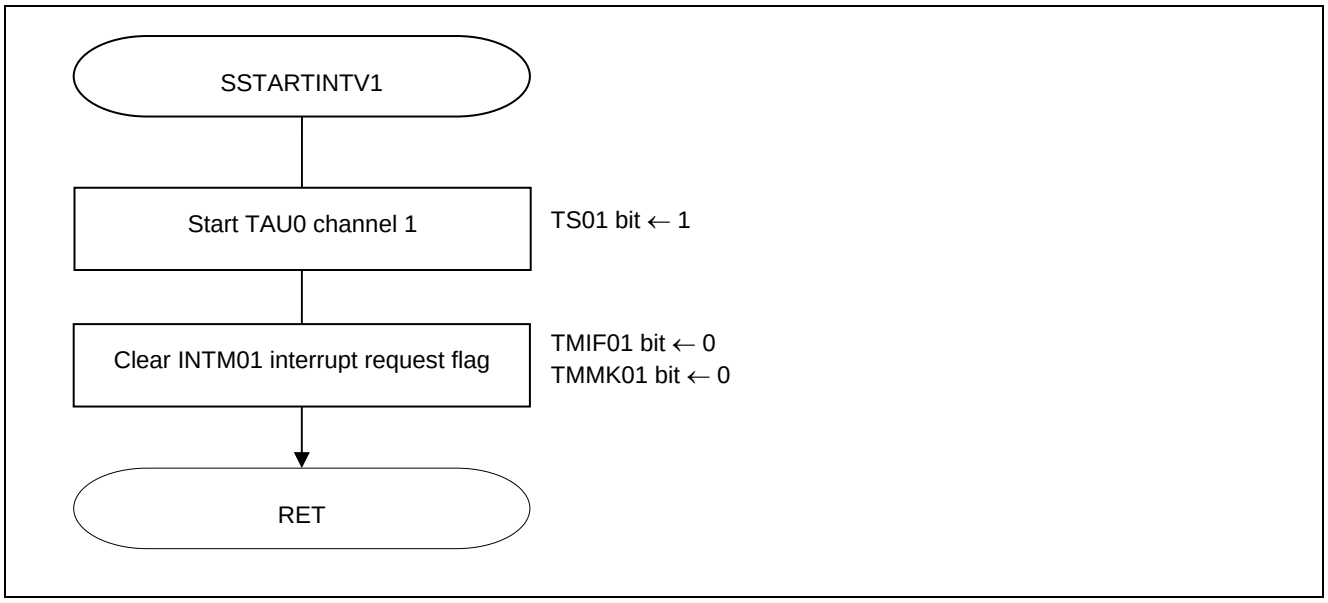

Figure 5.17 Starting the TAU0 Channel 1

5.9.16 Processing the TAU0 Channel 1 Interrupt

Figure 5.18 shows the flowchart for processing the TAU0 channel 1 interrupt.

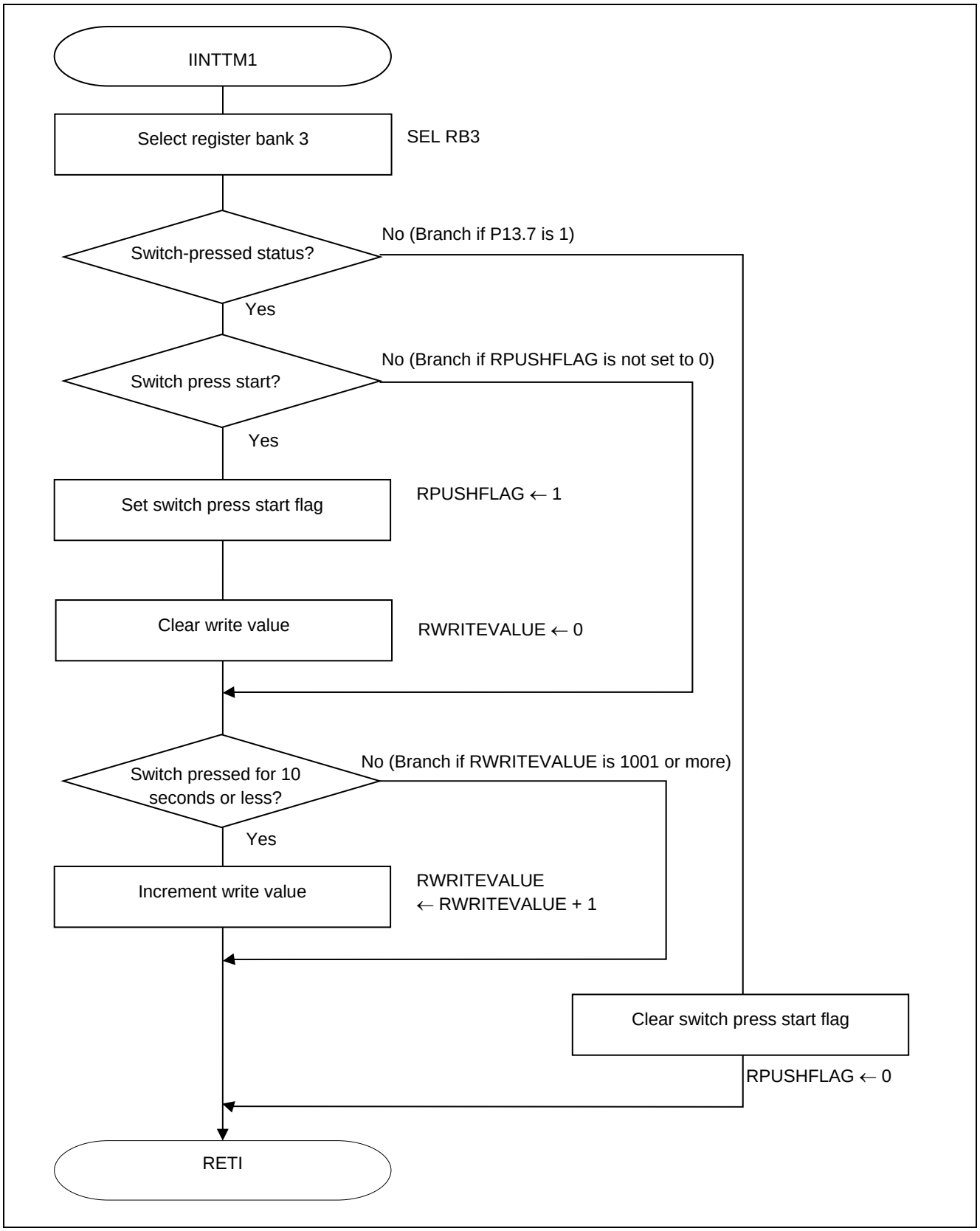

Figure 5.18 Processing the TAU0 Channel 1 Interrupt

5.9.17 Stopping the TAU0 Channel 1

Figure 5.19 shows the flowchart for stopping the TAU0 channel 1.

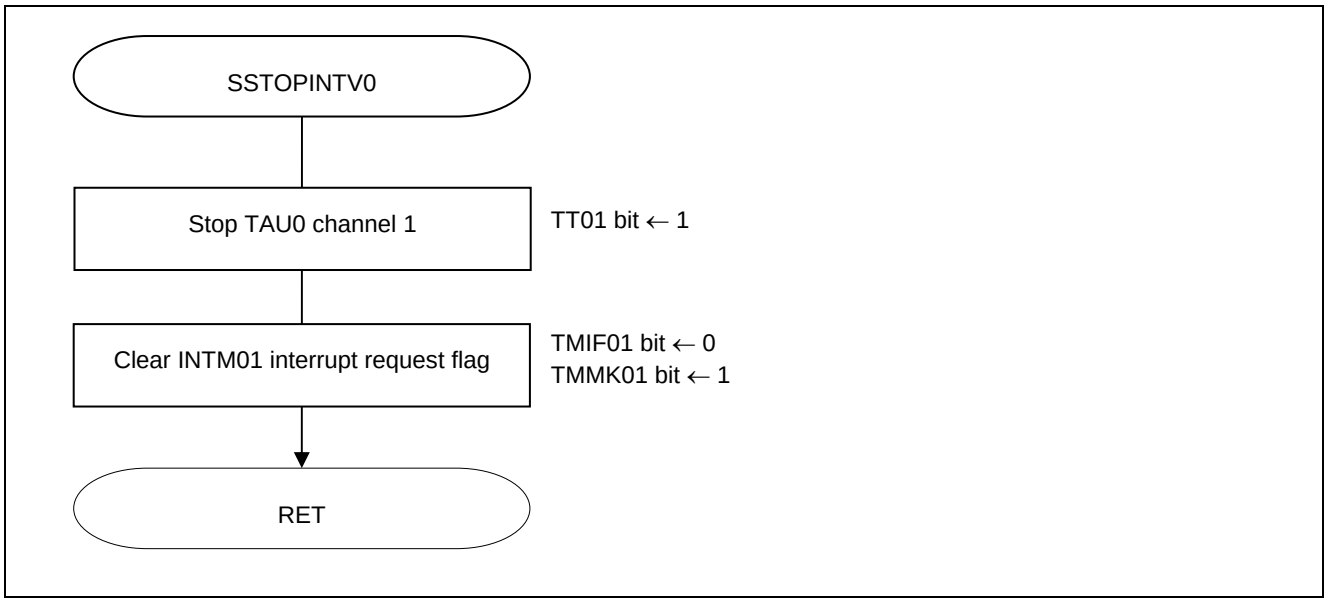

Figure 5.19 Stopping the TAU0 Channel 1

5.9.18 Writing Execution

Figures 5.20 and 5.21 show the flowcharts for writing execution.

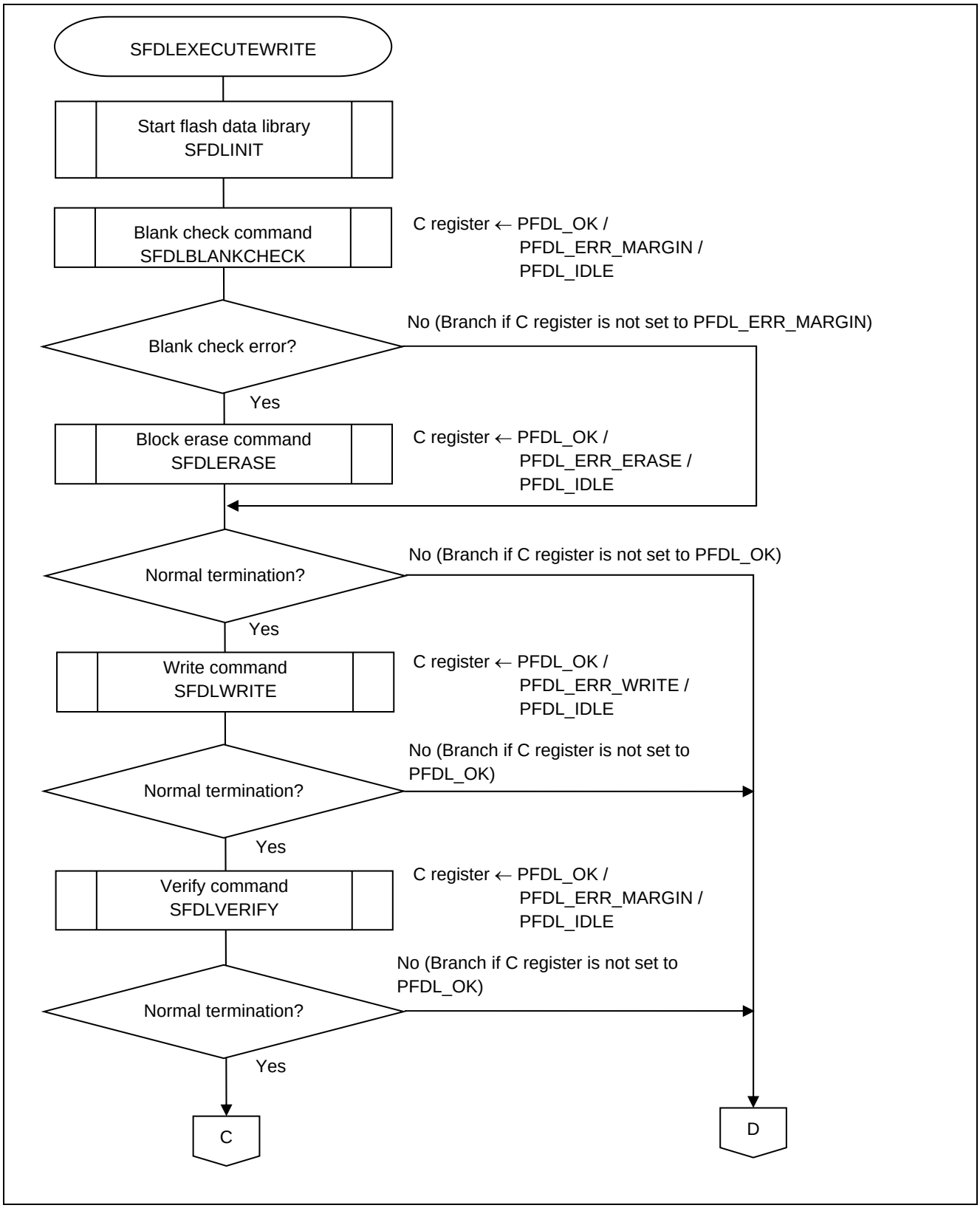

Figure 5.20 Writing Execution (1/2)

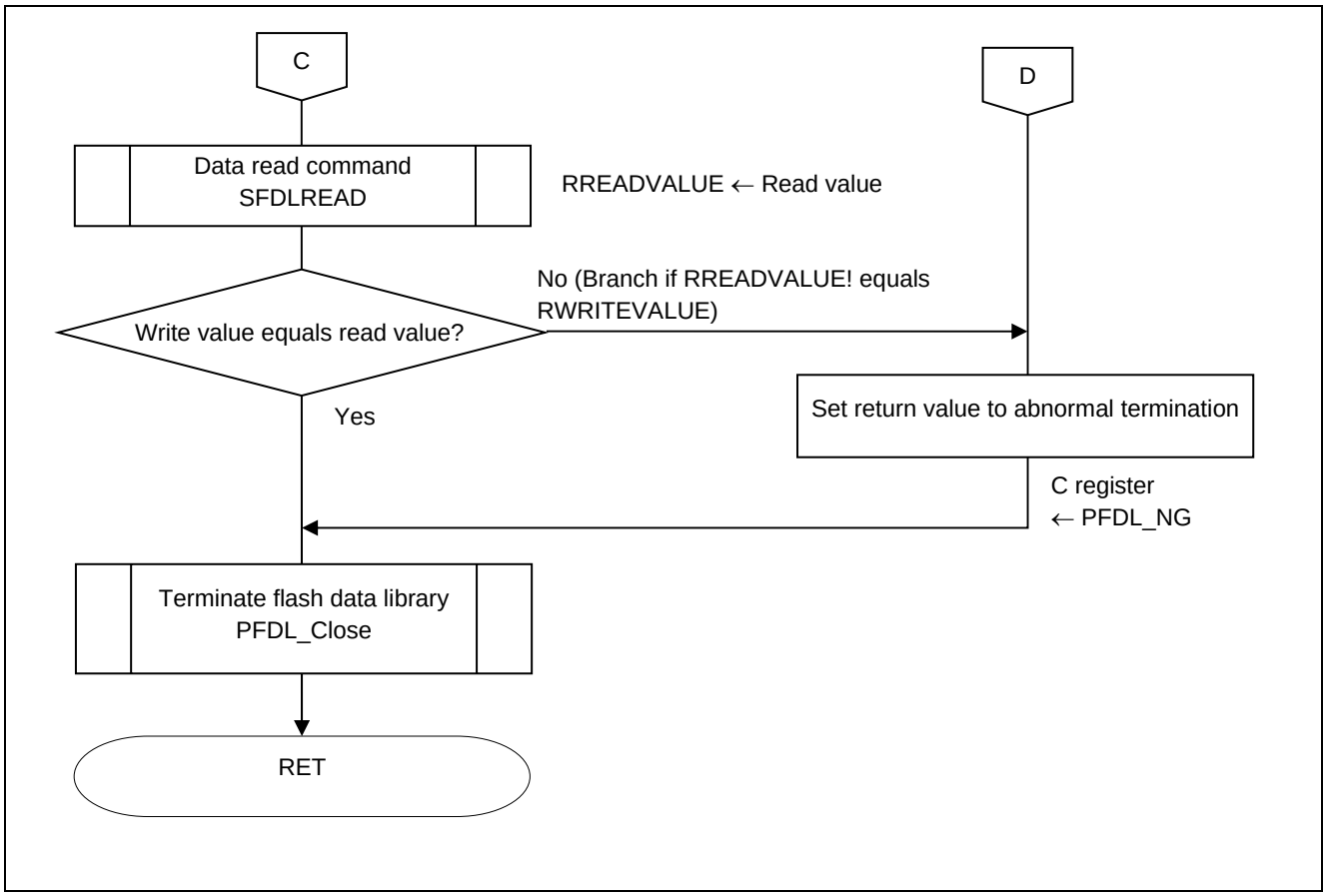

Figure 5.21 Writing Execution (2/2)

5.9.19 Processing the Blank Check Command

Figure 5.22 shows the flowchart for processing the blank check command.

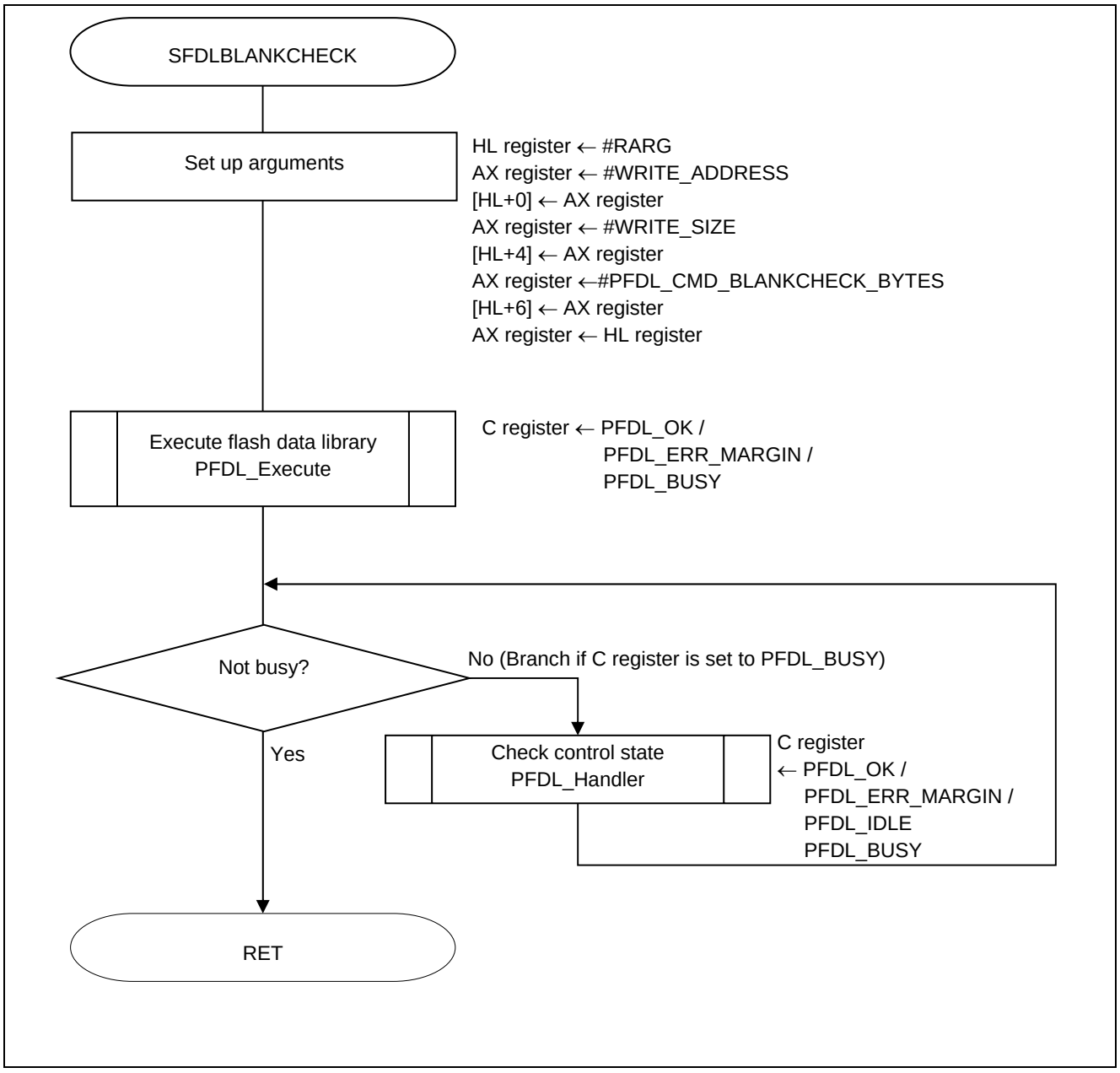

Figure 5.22 Processing the Blank Check Command

5.9.20 Processing the Block Erase Command

Figure 5.23 shows the flowchart for processing the block erase command.

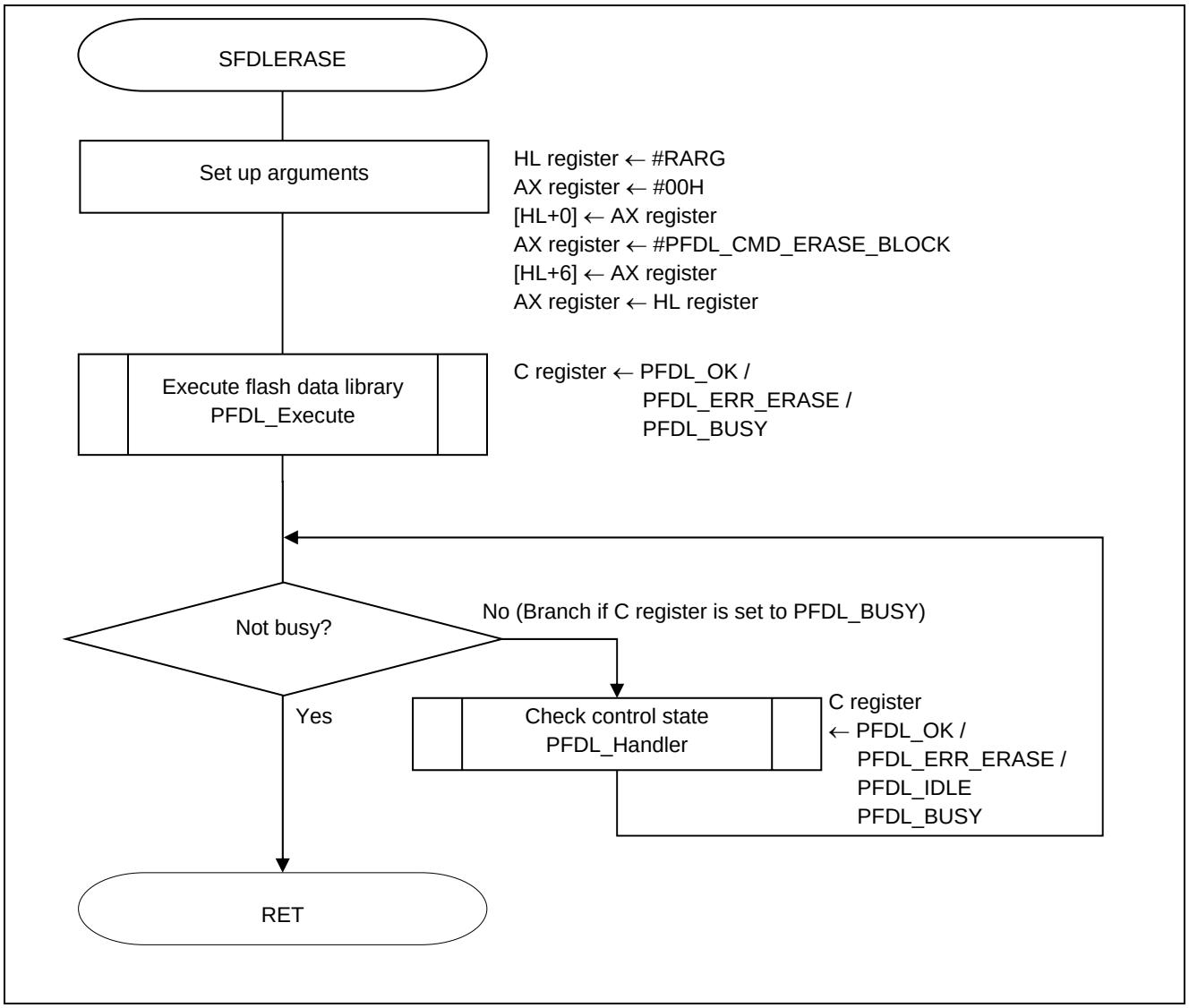

Figure 5.23 Processing the Block Erase Command

5.9.21 Processing the Data Write Command

Figure 5.24 shows the flowchart for processing the data write command.

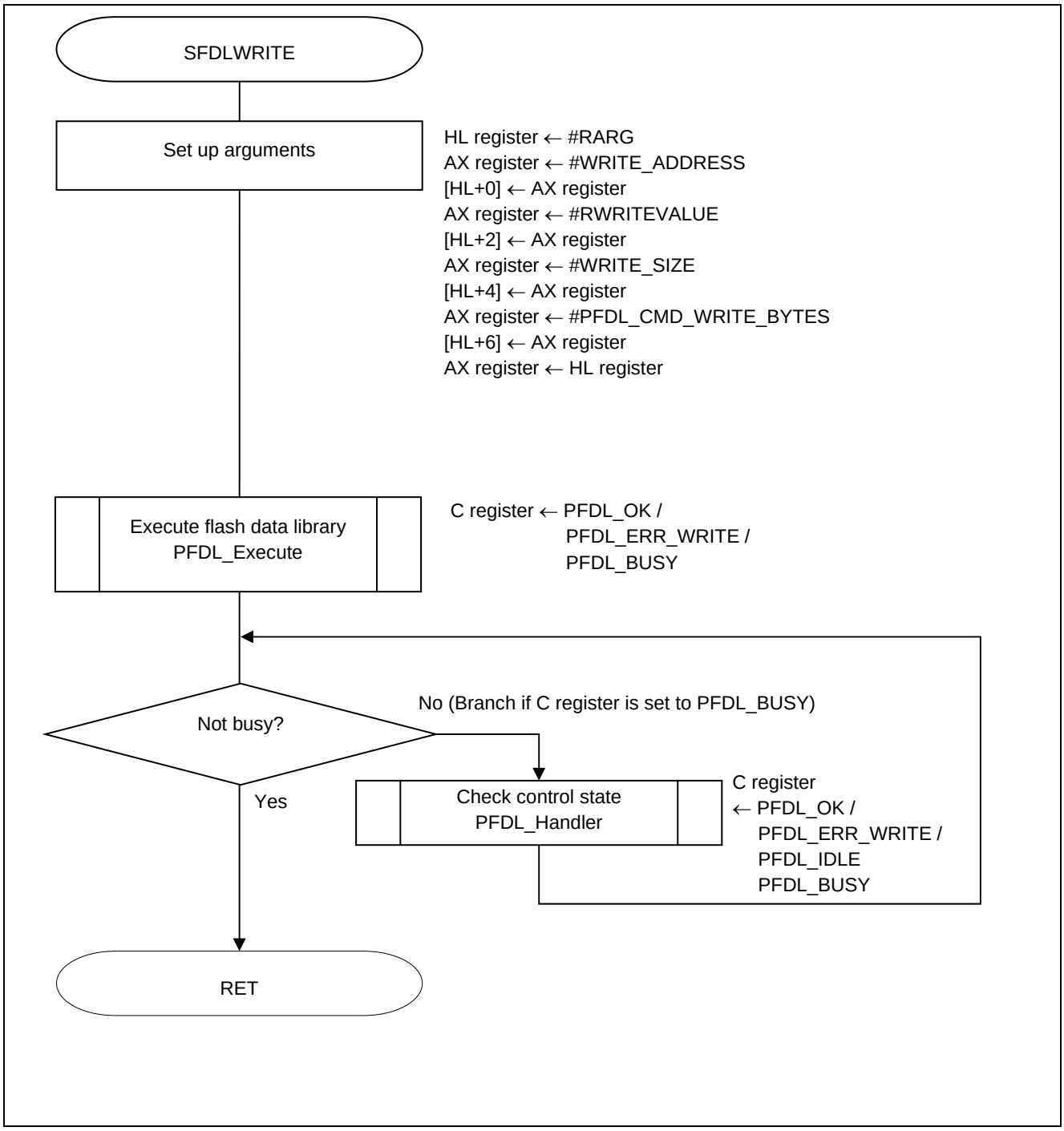

Figure 5.24 Processing the Data Write Command

5.9.22 Processing the Verify Command

Figure 5.25 shows the flowchart for processing the verify command.

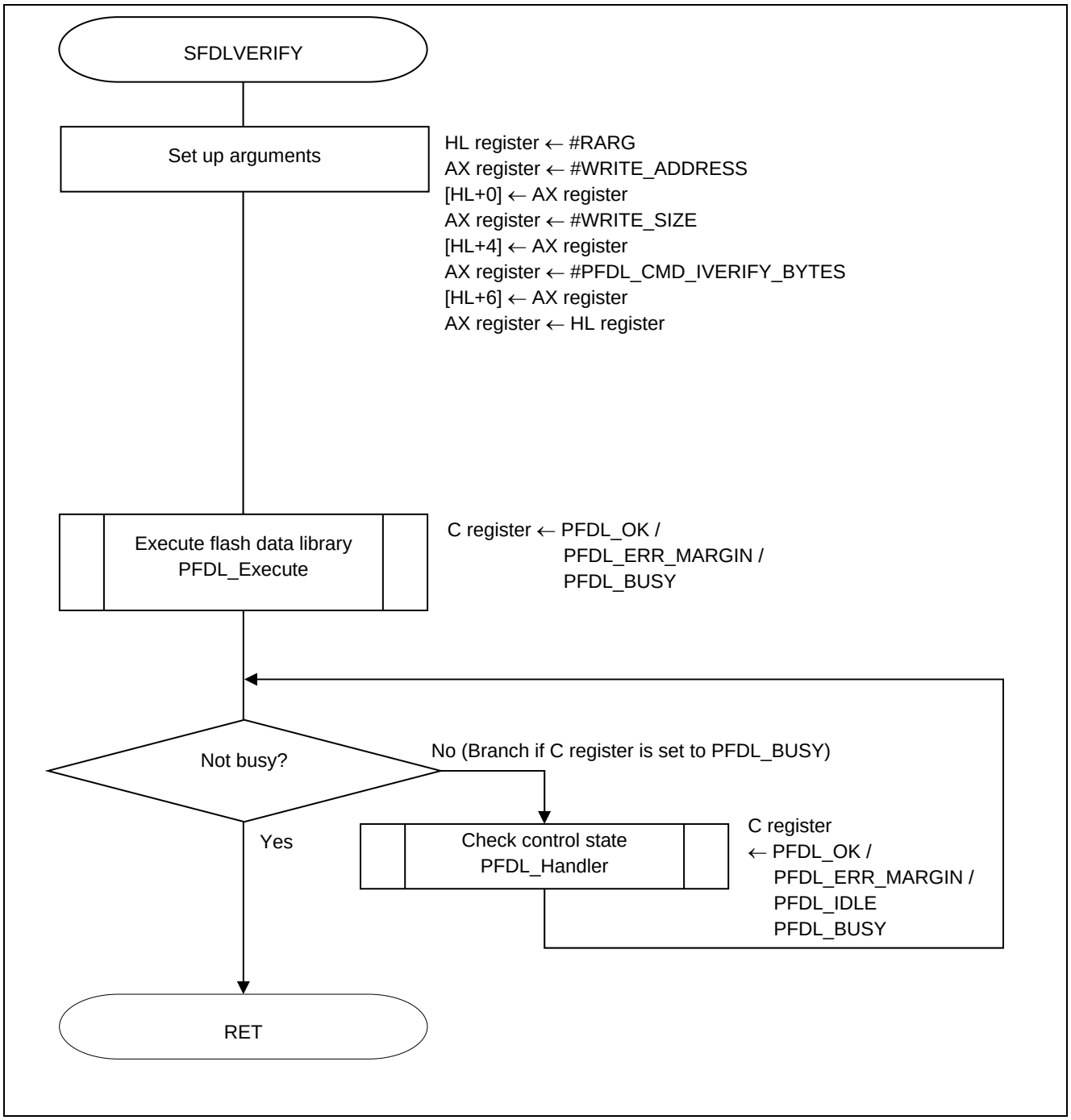

Figure 5.25 Processing the Verify Command

6. Sample Code

The sample code is available on the Renesas Electronics Website.

7. Documents for Reference

RL78/G12 User's Manual: Hardware (R01UH0200E)

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/G12 Flash Data Library Type04

Rev.	Date	Description	
		Page	Summary
1.00	Mar.01, 2013	—	First edition issued
1.10	June 01, 2016	8	Modification of 1.4 How to Get the Flash Data Library

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 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.

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

2. Processing at Power-on

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

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

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

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

4. Clock Signals

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

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

5. Differences between Products

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

   — The characteristics of an MPU or MCU 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.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

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