R32C/100 Series

Rewriting Internal Flash Memory Using EW0 Mode of CPU Rewrite Mode

REJ05B1225-0100
Rev.1.00
May 28, 2010

1. Abstract

This application note describes an example of rewriting the embedded flash memory using the EW0 mode in CPU rewrite mode.

2. Introduction

The application example described in this document applies to the following microcomputer (MCU):
MCU: R32C/118 Group

This program can be used with other R32C/100 Series MCUs which have the same special function registers (SFRs) as the R32C/118 Group. Check the user's manual for any additions or modifications to functions. Careful evaluation is recommended before using this application note.

3. Setting Procedure

3.1 CPU Rewrite Mode

In CPU rewrite mode, the CPU executes software commands to rewrite the flash memory. The CPU accesses the flash memory via the dedicated flash memory rewrite buses instead of the CPU buses.
Figure 3.1 shows the Flash Memory Access Path in CPU Rewrite Mode.

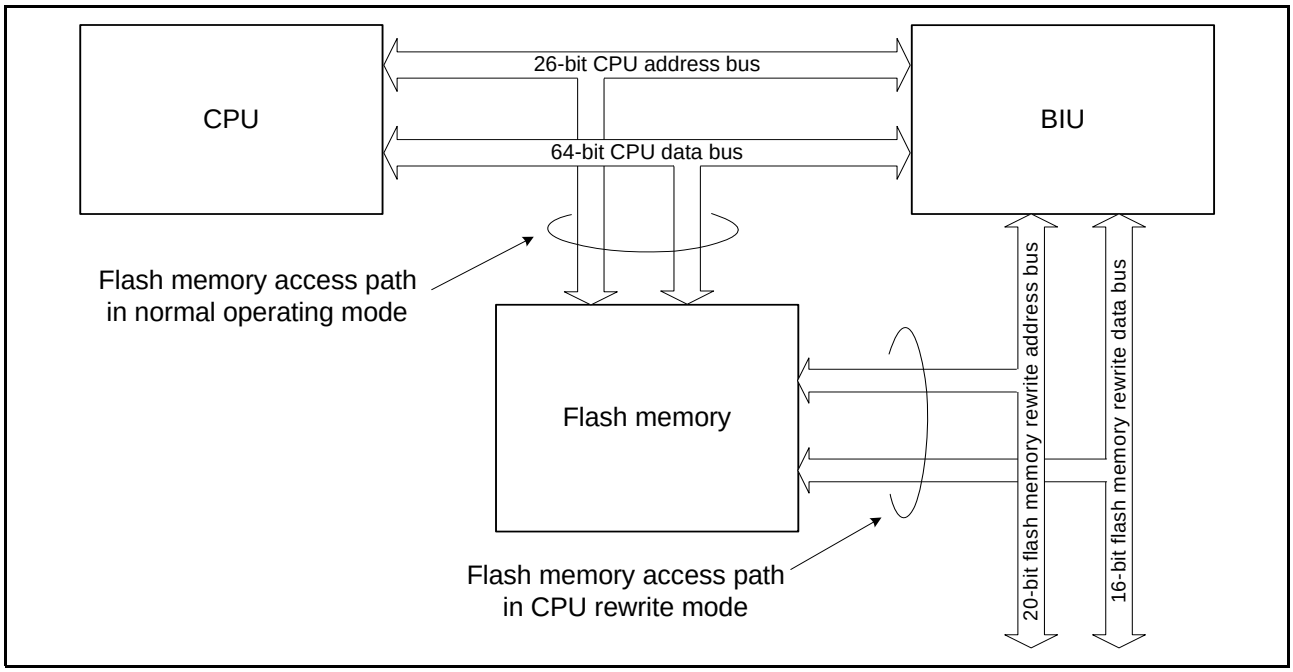


Figure 3.1 Flash Memory Access Path in CPU Rewrite Mode

3.2 EW0 Mode Features

EW0 mode allows the user to rewrite the user ROM and the data areas by issuing program and erase commands from the CPU rewrite program already transferred to the RAM.
In EW0 mode, the CPU operates during programming and erasing. Therefore, peripheral interrupts are accepted during programming and erasing by allocating the vector table and interrupt program on the RAM.
Table 3.1 shows the EW0 Mode Features.

Table 3.1 EW0 Mode Features

Item	Content
CPU operating modes	• Single-chip mode • Memory expansion mode
Rewrite program executable spaces	Spaces other than the embedded flash memory
Mode state after program/erase operation	Read status register mode
CPU state during program/erase operation	Operating
Flash memory state detection	• The FMSR0 register is read by a program. • The read status register command is executed to read data.

3.3 Flash Memory Rewrite Bus Timing

Bus setting for flash memory rewrite should be performed by the FEBC0 and/or FEBC3 registers. Refer to “Flash Memory Rewrite Bus Timing” and “Electrical Characteristics” for the appropriate bus setting.
Note that registers FEBC0 and FEBC3 in memory expansion mode share respective addresses with registers EBC0 and EBC3, That is, registers EBC0 and EBC3 should be set again after rewriting the FEBC0 and FEBC3 registers.

(1) Clock Conditions
MCU operates in single-chip mode and PLL mode before entering EW0 mode.
Table 3.2 lists the Clock Conditions after entering PLL mode. Table 3.3 lists the Clock-associated Registers Settings.

Table 3.2 Clock Conditions

Clock Name	Frequency
Main clock	16 MHz
PLL clock	100 MHz
Base clock	50 MHz
CPU clock	50 MHz
Peripheral bus clock	25 MHz
Peripheral clock source	25 MHz

Table 3.3 Clock-associated Registers Settings

Register Name	Setting Value	Comments
PLC0	04h	Refer to the user’s manual for details.
PLC1	03h	Same as above
PM3	40h	Bits PM36 and PM35 are 10b (peripheral clock source: divided by 4)
CCR	1Fh	Bits BCD1 and BCD0 are 11b (base clock: divided by 2) Bits CCD1 and CCD0 are 11b (CPU clock: no division) Bits PCD1 and PCD0 are 01b (peripheral bus clock: divided by 2) BCS bit is 0 (base clock source: PLL clock selected)
PBC	0504h	Refer to the user’s manual for details.
EBC0	0000h	Leave values as they are after reset due to operating in single-chip mode

(2) Flash Memory Conditions

Table 3.4 lists the flash memory standard values. These values are subject to change. Refer to "Electrical Characteristics" in the user's manual for details.

Table 3.4 Flash Memory CPU Rewrite Mode Timing

Symbol	Characteristics	Value		Unit
		Min.	Max.	
tcR	Read cycle time	200		ns
tsu(S-R)	Chip-select setup time for read	200		ns
th(R-S)	Chip-select hold time after read	0		ns
tsu(A-R)	Address setup time for read	200		ns
th(R-A)	Address hold time after read	0		ns
tw(R)	Read pulse width	100		ns
tcW	Write cycle time	200		ns
tsu(S-W)	Chip-select setup time for write	0		ns
th(W-S)	Chip-select hold time after write	30		ns
tsu(A-W)	Address setup time for write	0		ns
th(W-A)	Address hold time after write	30		ns
tw(W)	Write pulse width	50		ns

(3) Calculating the Necessary Number of Cycles

Calculate the number of cycles necessary to access the flash memory based on the clock and flash memory conditions for the setting value.

The reference clock is the base clock selected by setting bits BCD1 and BCD0 in the CCR register. The base clock is 50 MHz as listed in Table 3.2. Therefore, the value becomes 20 ns per cycle. The number of cycles in Table 3.5 are the values from Table 3.4 based on this cycle value.

Table 3.5 Required Number of Cycles

Symbol	Value		Unit	Cycles		Unit
	Min.	Max.		Min.	Max.	
tcR	200		ns	10		Cycles
tsu(S-R)	200		ns	10		Cycles
th(R-S)	0		ns	0		Cycles
tsu(A-R)	200		ns	10		Cycles
th(R-A)	0		ns	0		Cycles
tw(R)	100		ns	5		Cycles
tcW	200		ns	10		Cycles
tsu(S-W)	0		ns	0		Cycles
th(W-S)	30		ns	1.5		Cycles
tsu(A-W)	0		ns	0		Cycles
th(W-A)	30		ns	1.5		Cycles
tw(W)	50		ns	2.5		Cycles

(4) Bus Timing

The bus timing is determined based on the peripheral bus clock in Table 3.3 and Table 3.5.

The value of bits MPY1 and MPY0 that are determined by the peripheral bus clock, and the flash memory read and write cycles are required to determine the bus timing.

Table 3.6 shows the correlation between bits MPY1 and MPY0 and bits FWR4 to FWR0, and the number of read cycles when the peripheral bus clock is divided by 2.

Table 3.7 shows the correlation between bits MPY1 and MPY0, bits FSUW1 and FSUW0, and bits FWW1 and FWW0 and the number of write cycles when the peripheral bus clock is divided by 2. The shaded region indicates the setting value when both of the clock and cycle conditions are met.

Refer to the user's manual for details on the peripheral bus clock divided by 3 or divided by 4.

Table 3.6 Read Cycle and Bit Settings: MPY1 and MPY0, and FWR4 to FWR0 When Peripheral Bus Clock is Divided by 2 (unit: cycles)

FWR3 to FWR0 Bit Settings		FWR4 Bit Settings	MPY1 and MPY0 Bit Settings							
			10b				11b			
			mpy = 3				*mpy* = 4			
			tsu(S-R), tsu(A-R)	tw(R)	tcR	th(R-S), th(R-A)	tsu(S-R), tsu(A-R)	tw(R)	tcR	th(R-S), th(R-A)
0000b	*wr* = 1		4	3	4	0	6	5	6	0
		1	6	5	6	0	6	5	6	0
0001b	*wr* = 2	0	8	7	8	0	10	9	10	0
		1	8	7	8	0	10	9	10	0
0101b	*wr* = 3	0	10	9	10	0	14	13	14	0
		1	12	11	12	0	14	13	14	0
0110b	*wr* = 4	0	14	13	14	0	18	17	18	0
		1	14	13	14	0	18	17	18	0
1010b	*wr* = 5	0	16	15	16	0	22	21	22	0
		1	18	17	18	0	22	21	22	0
1011b	*wr* = 6	0	20	19	20	0	26	25	26	0
		1	20	19	20	0	26	25	26	0
1111b	*wr* = 7	0	22	21	22	0	30	29	30	0
		1	24	23	24	0	30	29	30	0

Table 3.7 Write Cycle and Bit Settings: MPY1 and MPY0, FSUW1 and FSUW0, and FWW1 and FWW0 When Peripheral Bus Clock is Divided by 2 (unit: cycles)

FSUW1 and FSUW0 Bit Settings		FWW1 and FWW0 Bit Settings		MPY1 and MPY0 Bit Settings							
				10b				11b			
				mpy = 3				*mpy* = 4			
				tsu(S-W), tsu(A-W)	tw(W)	tcW	th(W-S), th(W-A)	tsu(S-W), tsu(A-W)	tw(W)	tcW	th(W-S), th(W-A)
00b	*suw* = 0	00b	*ww* = 1	1	3	6	2	1	4	6	1
		01b	*ww* = 2	1	6	8	1	1	8	10	1
		10b	*ww* = 3	1	9	12	2	1	12	14	1
		11b	*ww* = 4	1	12	14	1	1	16	18	1
01b	*suw* = 1	00b	*ww* = 1	4	3	8	1	5	4	10	1
		01b	*ww* = 2	4	6	12	2	5	8	14	1
		10b	*ww* = 3	4	9	14	1	5	12	18	1
		11b	*ww* = 4	4	12	18	2	5	16	22	1
10b	*suw* = 2	00b	*ww* = 1	7	3	12	2	9	4	14	1
		01b	*ww* = 2	7	6	14	1	9	8	18	1
		10b	*ww* = 3	7	9	18	2	9	12	22	1
		11b	*ww* = 4	7	12	20	1	9	16	26	1
11b	*suw* = 3	00b	*ww* = 1	10	3	14	1	13	4	18	1
		01b	*ww* = 2	10	6	18	2	13	8	22	1
		10b	*ww* = 3	10	9	20	1	13	12	26	1
		11b	*ww* = 4	10	12	24	2	13	16	30	1

(5) Bus Timing Settings

Table 3.8 lists the optimal bus timing settings when the clock conditions and the flash memory conditions are met based on Table 3.6 and Table 3.7.

Table 3.8 Bus Timing Optimal Settings

Bit Symbol	Value	Function
FWR3 to FWR0	0101b	*wr* = 3
FWR4	0	No pulse width extension
FSUW1 and FSUW0	00b	*suw* = 0
FWW1 and FWW0	10b	*ww* = 3
MPY1 and MPY0	10b	*mpy* = 3 (peripheral bus clock divided by 2)

When the bus timing is set in each bit of the FEBC0 register, FEBC0 is 5885h.

Figure 3.2 shows the FEBC0 register setting.

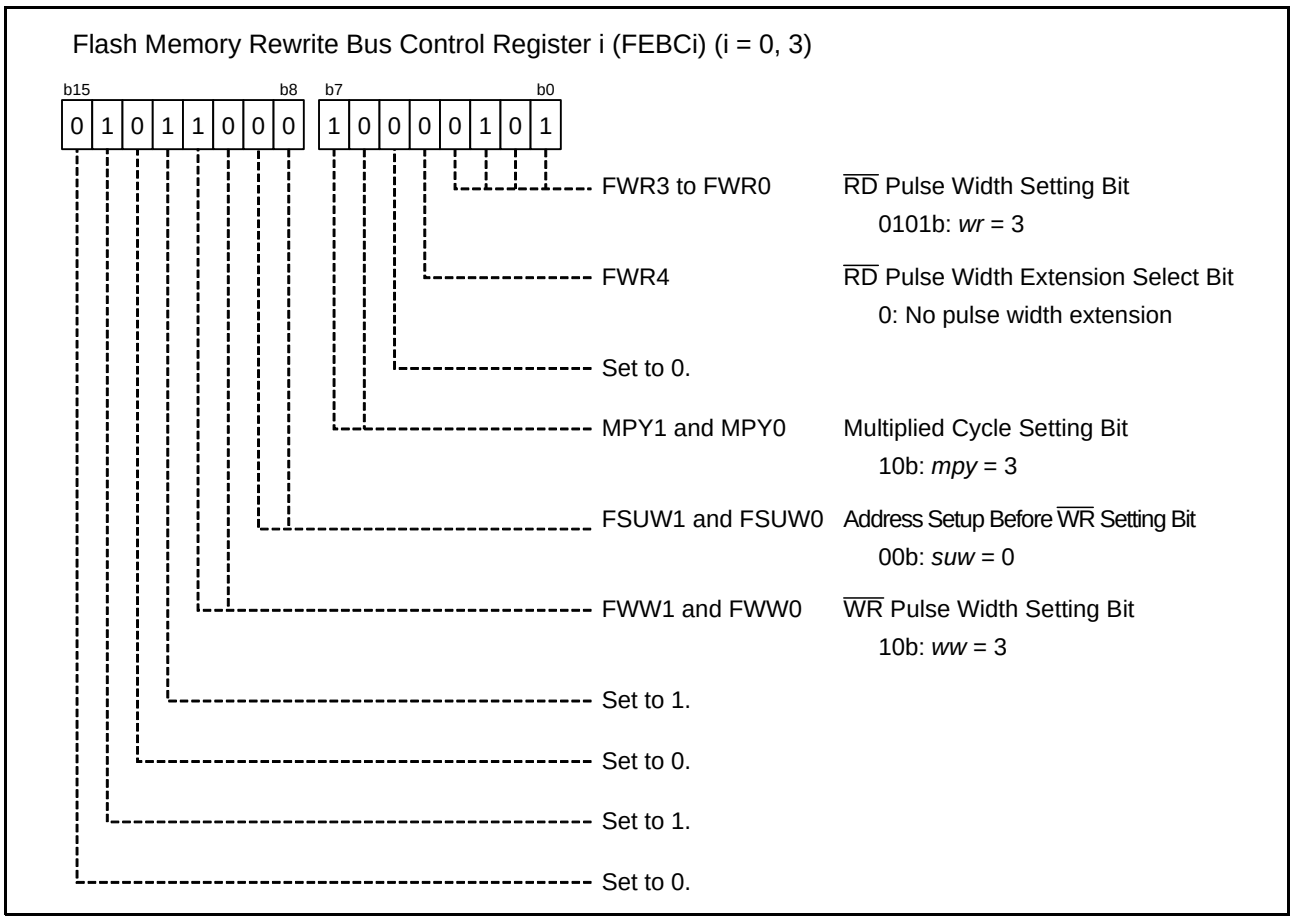


Figure 3.2 Flash Memory Rewrite Bus Control Register 0 (FEBC0)

(6) CPU Operating Mode and Flash Memory Rewrite

To rewrite the flash memory, the bus setting using by the FEBC0 and/or FEBC3 registers is required.

For exclusive use of single-chip mode, the FEBC3 register is not used. In this mode, do not change the reset value of registers CB01, CB12, and CB23. The bus setting for both the program area and data area can be performed using the FEBC0 register.

In cases other than the above, when the CPU operation is performed in memory expansion mode more than once, set registers CB01, CB12, and CB23 according to each setting range as shown in Table 3.9. The bus setting for program area and data area can be performed by the FEBC0 register and FEBC3 register, respectively.

Note that registers FEBC0 and FEBC3 in memory expansion mode share respective addresses with registers EBC0 and EBC3. That is, when the FEBCi register is set for the flash memory rewrite, the setting value for the EBCi register is accordingly changed (i = 0, 3). This may cause external devices allocated to the $\overline{CS0}$ space and/or $\overline{CS3}$ space in CPU rewrite mode to become inaccessible.

Table 3.9 lists the details of bus setting for the flash memory rewrite in each CPU operating mode.

Table 3.9 CPU Operating Mode and Flash Memory Rewrite

Item	CPU Operating Mode	
	Single-chip mode	Memory expansion mode
CB01 register	Hold the reset value 00h	Setting range: 04h to F8h Set value higher than that for the CB12 register
CB12 register	Hold the reset value 00h	Setting range: 03h to F7h Set value higher than that for the CB23 register and lower than that for the CB01 register
CB23 register	Hold the reset value 00h	Setting range: 02h to F6h Set value lower than that for the CB12 register
Bus setting for program area	FEBC0 register	FEBC0 register
Bus setting for data area	FEBC0 register	FEBC3 register
Status of $\overline{CS0}$ space and $\overline{CS3}$ space after the FEBCi register is set	N/A	• Separate bus format • 16-bit bus width • $\overline{RDY}$ ignored
Restrictions	None	• $\overline{HOLD}$ is ignored • In CPU rewrite mode, external devices become inaccessible to data with the bus format set for $\overline{CS0}$ space and/or $\overline{CS3}$ space as multiplexed bus • The change in bus timing may cause external devices in the $\overline{CS0}$ space and/or $\overline{CS3}$ space to become inaccessible

3.4 Setting Procedure

Figure 3.3 shows the CPU Rewrite Mode (EW0 Mode) Execution Flow.

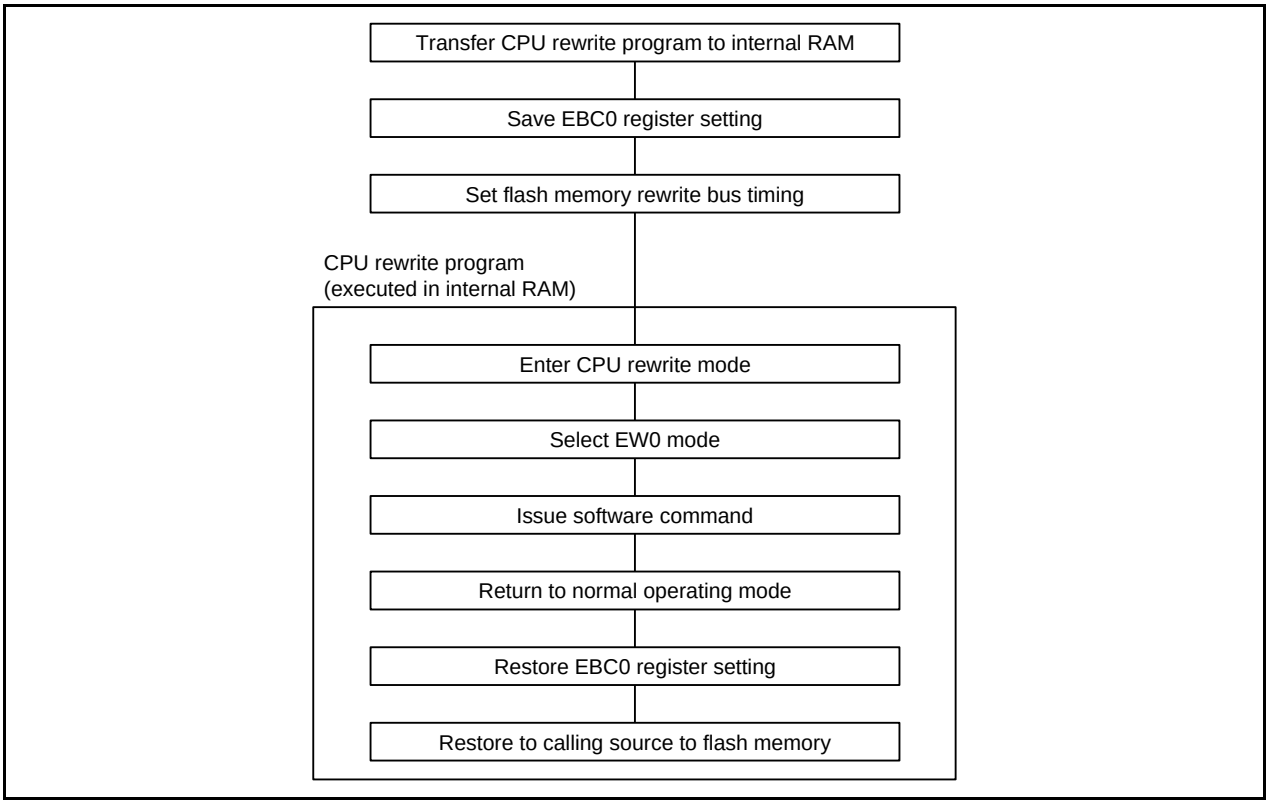
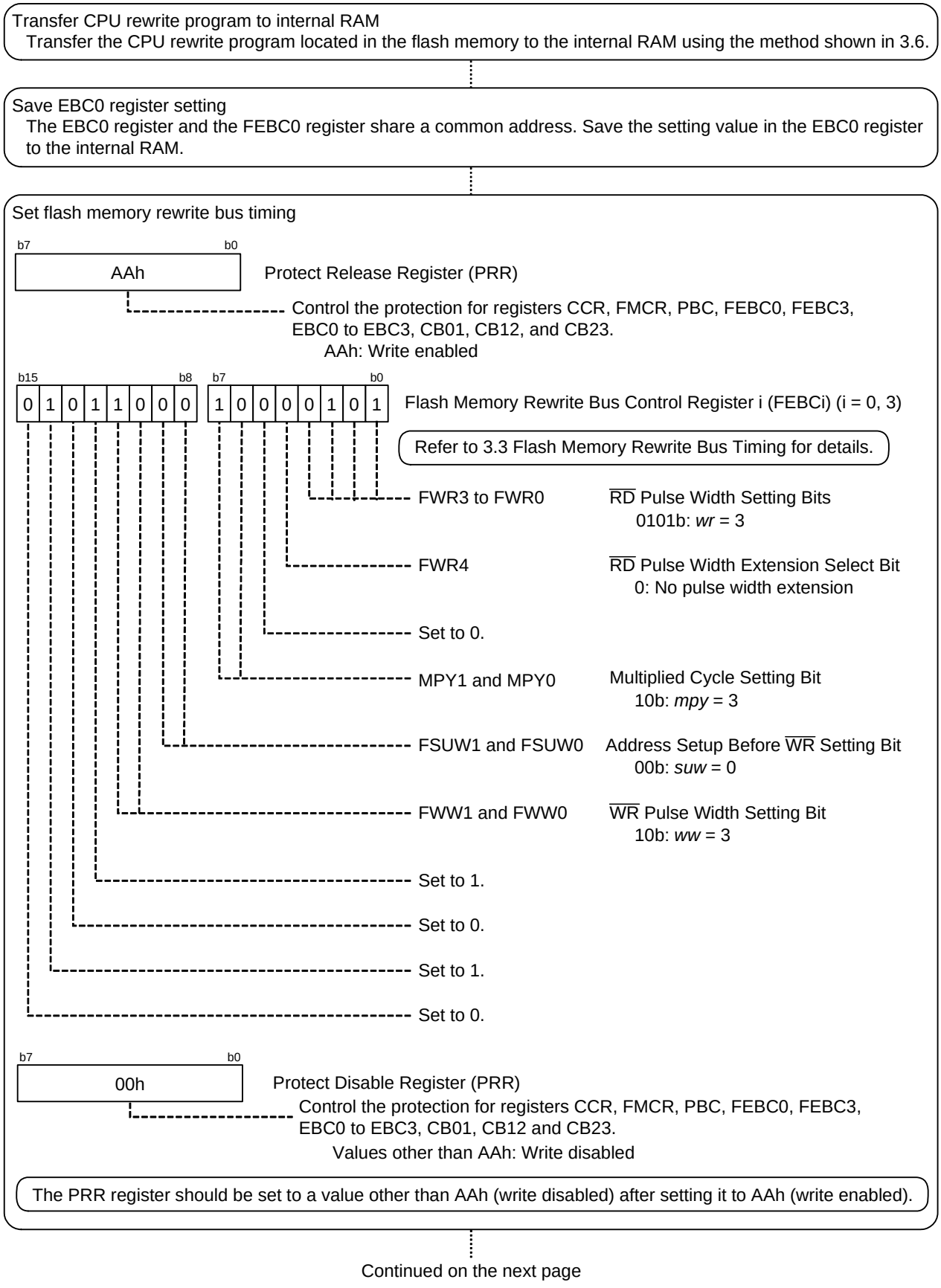


Figure 3.3 CPU Rewrite Mode (EW0 Mode) Execution Flow

3.5 Register Settings



Continued on the next page

Continued from the previous page

Call CPU rewrite program
Call the CPU rewrite program transferred to the internal RAM.

Enter CPU rewrite mode

b7 b0

AAh

Protect Release Register (PRR)

Control the protection for registers CCR, FMCR, PBC, FEBC0 to FEBC3, EBC0 to EBC3, CB01, CB12 and CB23.
AAh: Write enabled

b7 b0

1	0	0	0	0	0	0	1

Flash Memory Control Register (FMCR)

Set to 1.

Set to 0.

FEW CPU Rewrite Mode Setting Bit
1: CPU rewrite mode

b7 b0

00h

Protect Release Register (PRR)

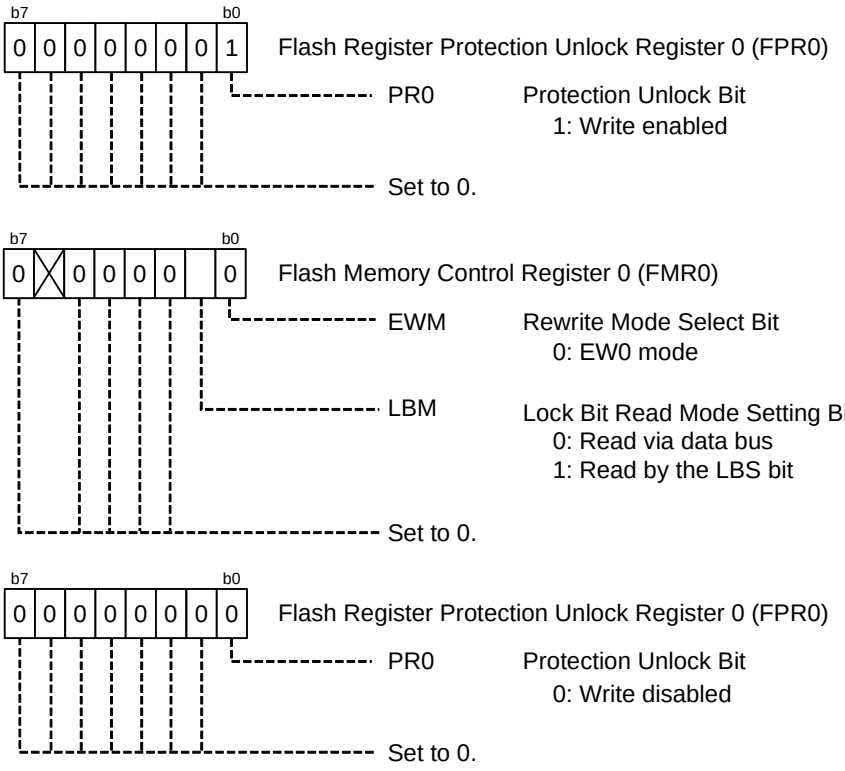
Control the protection for registers CCR, FMCR, PBC, FEBC0 to FEBC3, EBC0 to EBC3, CB01, CB12 and CB23.
Value other than AAh: Write disabled

The PRR register should be set to a value other than AAh (write disabled) after setting it to AAh (write enabled).

Continued on the next page

Continued from the previous page

Select EW0 mode



The PR0 bit in the FPR0 register should be set to 0 (write disabled) after setting it to 1 (write enabled).

Continued on the next page

Continued from the previous page

Issue software command (first command)
Issue the embedded flash memory software command (first command).

Issue software command (second command or later)

The following register setting is required to execute multiple flash memory commands. Before issuing the last command, wait until the final command accept busy flag in flash memory control register 0 becomes 0.

Flash Memory Control Register 0 (FMR0)

- EWM — Rewrite Mode Select Bit
  - 0: EW0 mode
- LBM — Lock Bit Read Mode Setting Bit
  - 0: Read via data bus
  - 1: Read by the LBS bit
- LBS — Lock Bit Status Flag
  - 0: Locked
  - 1: Unlocked
- RRDY — Read Ready Flag
  - 0: Busy
  - 1: Ready
- FCA — Final Command Accept Busy Flag
  - 0: Final command accept ready
  - 1: Final command accept busy
- Set to 0.

Next, issue the embedded flash memory software command (last command).

Continued on the next page

Continued from the previous page

Return to normal operating mode and restore to EBC0 register setting

b7 b0

AAh

Protect Release Register (PRR)

Control the protection for registers CCR, FMCR, PBC, FEBC0, FEBC3, EBC0 to EBC3, CB01, CB12 and CB23.
AAh: Write enabled

b7 b0

0	0	0	0	0	0	0	1

Flash Memory Control Register (FMCR)

Set to 1.

Set to 0.

FEW CPU Rewrite Mode Setting Bit
0: Normal operating mode

b15 b8 b7 b0

External Bus Control Register (EBC0, EBC3)

Reset the value saved in the internal RAM.

b7 b0

00h

Protection Release Register (PRR)

Control the protection for registers CCR, FMCR, PBC, FEBC0, FEBC3, EBC0 to EBC3, CB01, CB12 and CB23.
Value other than AAh: Write disabled

The PRR register should be set to any value other than AAh (write disabled) after setting it to AAh (write enabled).

Restored to the calling source on the flash memory

3.6 Transferring the CPU Rewrite Control Program to RAM

The CPU rewrite control program should be operated in RAM. The example below explains how to transfer the CPU rewrite control program stored in address FFFD0000h to RAM. Figure 3.4 shows the program transfer.

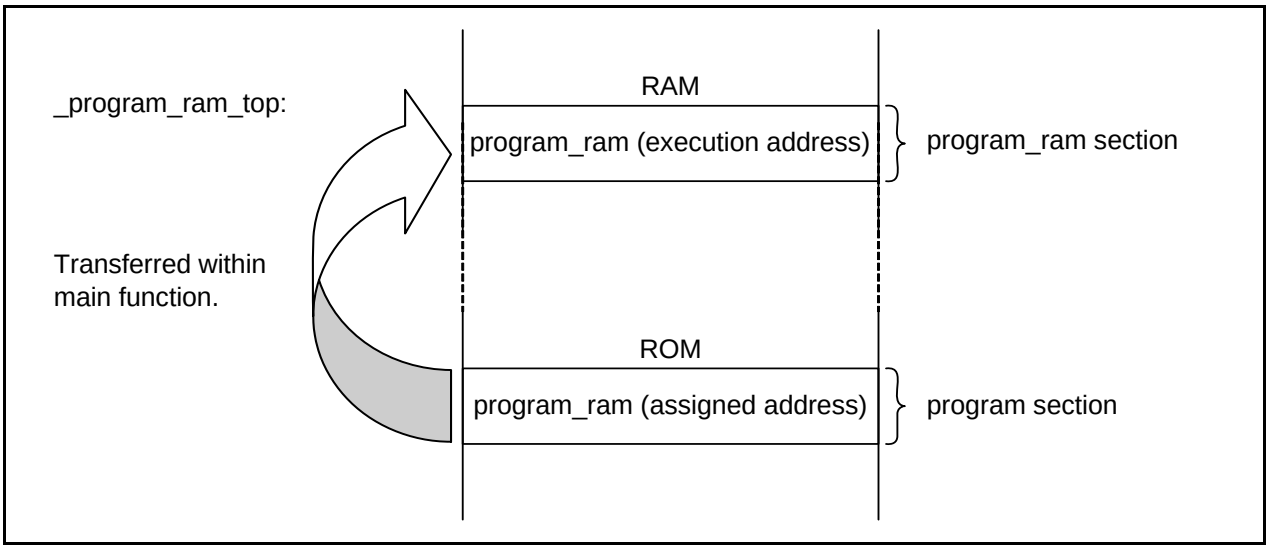


Figure 3.4 Program Transfer Image

(1) Changing the Section Name

Add program_ram as a section name. In this section, allocate the CPU rewrite control program to be operated in the RAM. To relocate the CPU rewrite control program from the program section to the program_ram section, write the following in C language.

```
void main(void)
{
/* This program is located in the program section */
}
/* All the programs following the #pragma SECTION declaration are located in the program_ram
section */
#pragma SECTION program program_ram
void ew0_mode_program(void)
{
 /* This program is located in the program_ram section */
}
```

The #pragma SECTION is the #pragma extension function that changes the section name generated by the compiler. If this declaration is executed in the program section, the names of the function sections described after the #pragma SECTION declaration will be changed.

(2) Transferring the CPU Rewrite Control Program

Add the process (macro definition) that transfers the CPU rewrite control program to the RAM.

```
/*****************************
*   #pragma declaration   *
*****************************/

#define pcopy(X,Y) _asm(" .initsct "X",code,align\n"\
                "       .initsct "X"_INIT,rom"Y"\n"\
                "       mov.l       #(topof "X"_INIT),A0\n"\
                "       mov.l       #(topof "X"),A1\n"\
                "       mov.l       #sizeof "X",R7R5\n"\
                "       smovf.b")
```

Added macro definition

As shown below, write the above process in the beginning of the program.

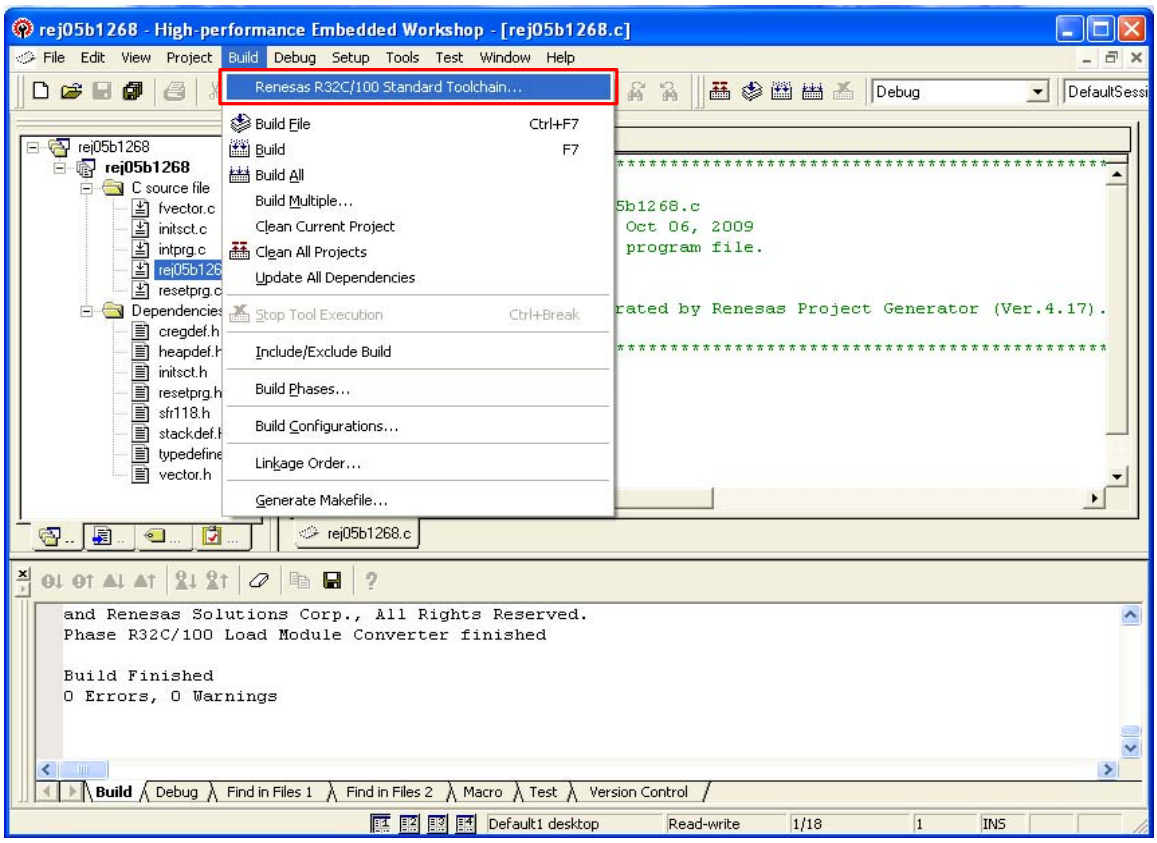
```
/*""FUNC COMMENT""*****************************************************
* Outline               : Flash Memory Version CPU Rewrite Mode
*                       :  (EW0 Mode) Sample
* Declaration           : void main(void)
* Description           : This program is the execution sample in CPU rewriting mode.
* Argument              : none
* Return Value          : none
* Calling Functions     : SetPLLClock()       : Configuring PLL Mode
*                       : ew0_mode_control()  : CPU rewriting program execution
*""FUNC COMMENT END""*************************************************/
void main(void)
{
   unsigned char tmp;

   pcopy("program_ram","data,align");          // (1) Transfer rewrite program to RAM

   asm("FCLR I") ;                             // (2) Interruption inhibiton
```

Transfer process

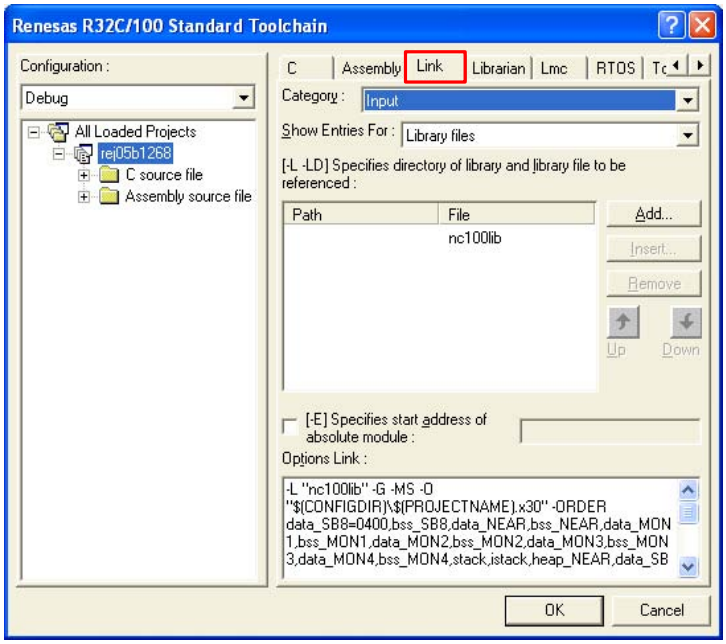
(3) Specifying the Program Storage Location

To execute the program transferred to the RAM, it is necessary to specify in the linker (ln100) that the program assigned address (in the ROM) and execution address (in the RAM) are to be located separately, as shown below.

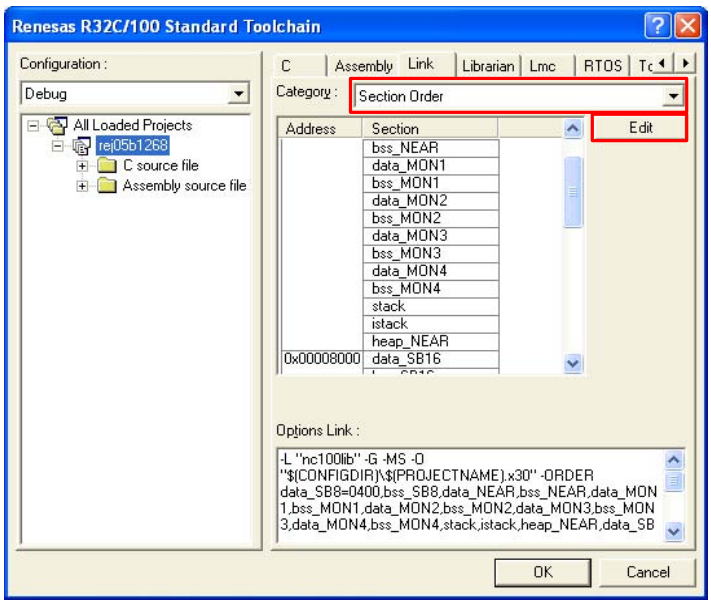
Select Renesas R32C/100 Standard Toolchain from the Build menu.



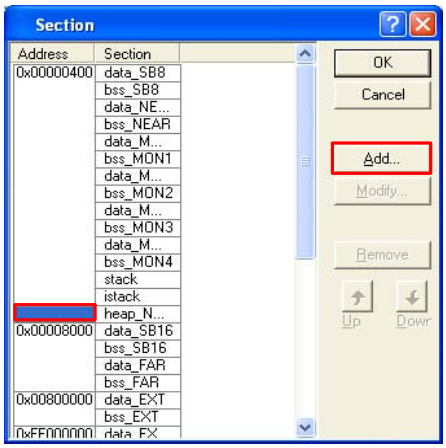
Select the Link tab in the Renesas R32C/100 Standard Toolchain window.

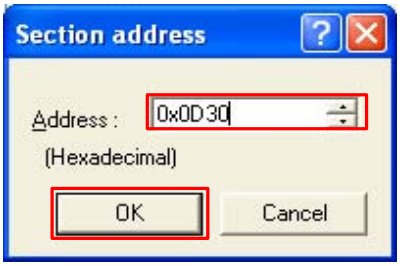


After selecting Section Order from the Category pull-down menu, the section memory map will be displayed. Click Edit to display the Section window.

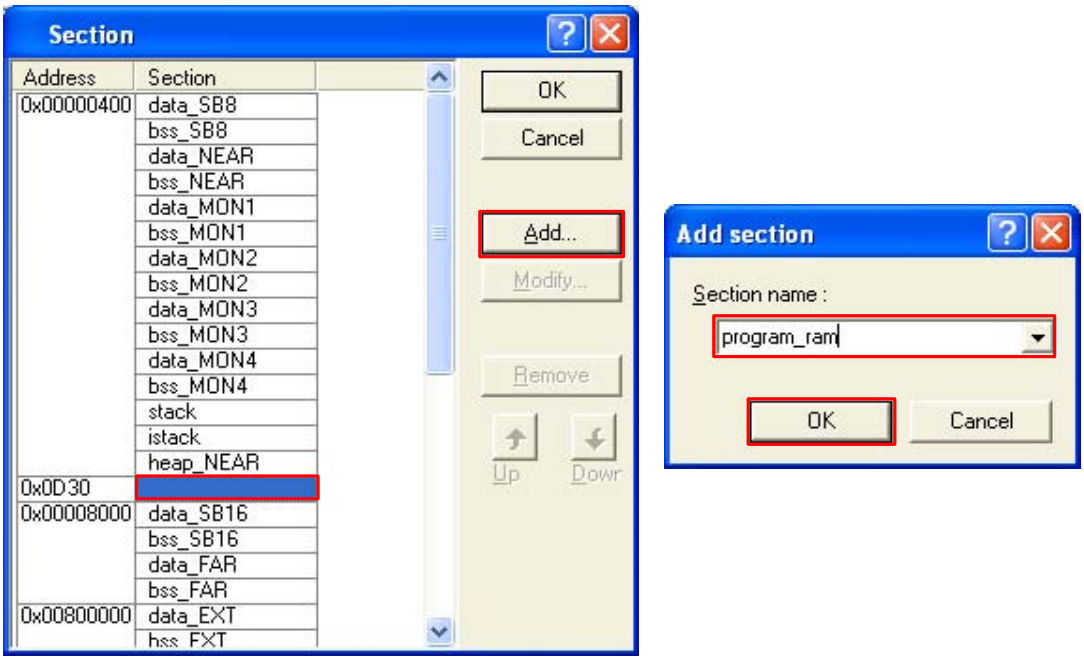


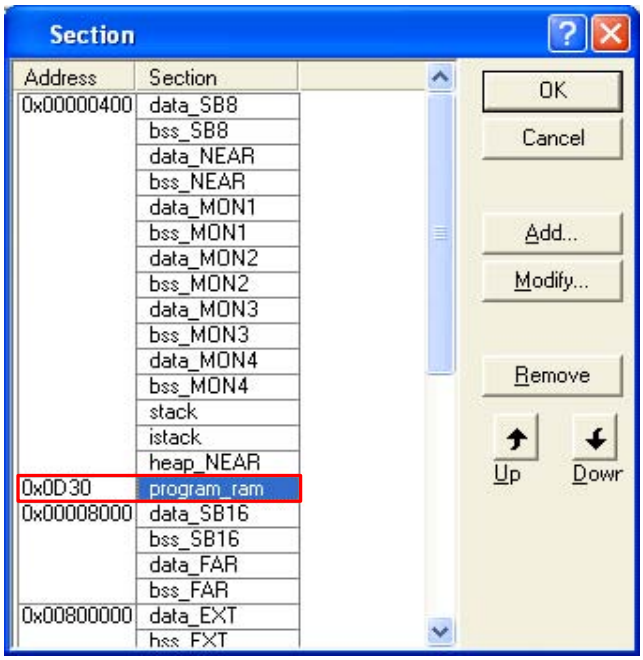
Then, select the field above 0x00008000 in the Address column and click Add. Set 0x0d30 in the Section address window and click OK.



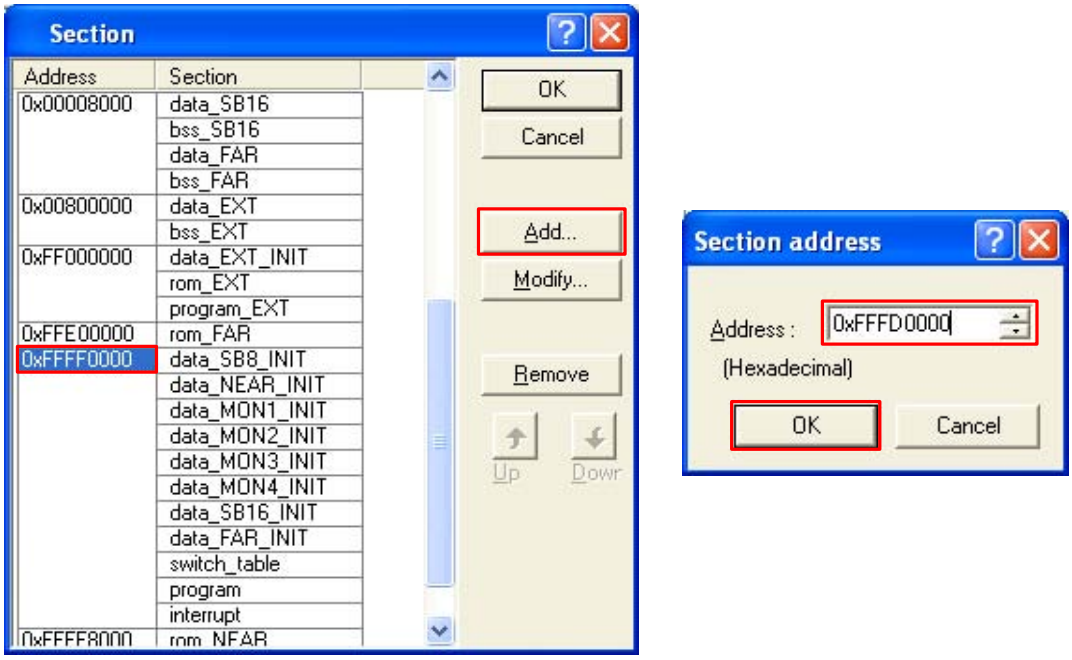


Select the field next to the added 0x0d30 in the Section column and click Add. When the Add section window is displayed, type program_ram in the Section name field and click OK.

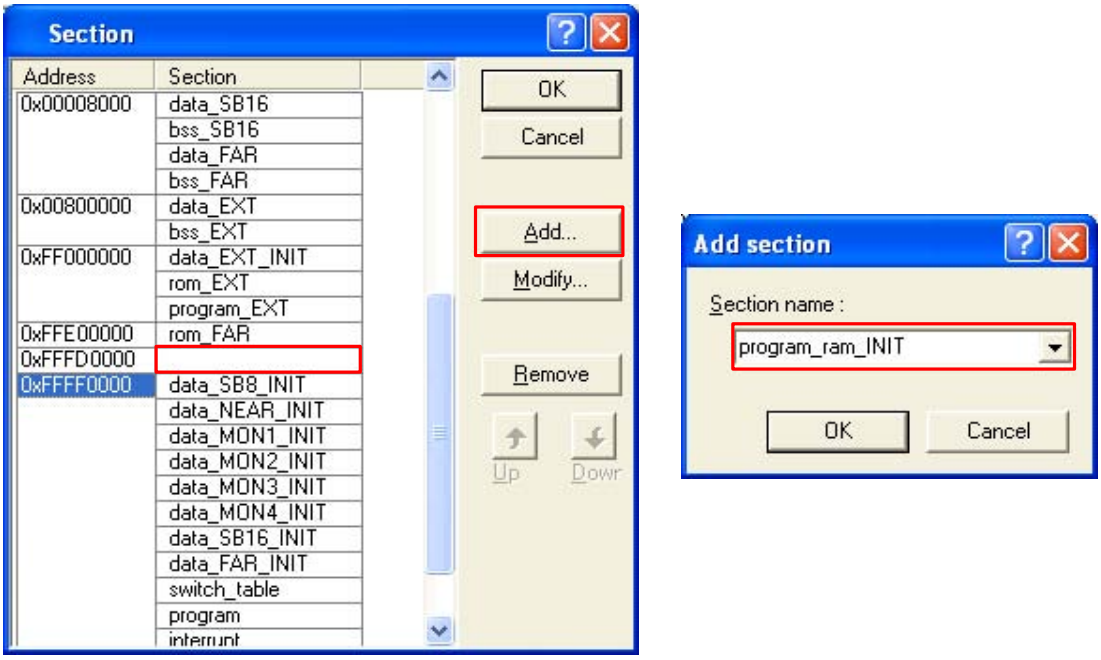


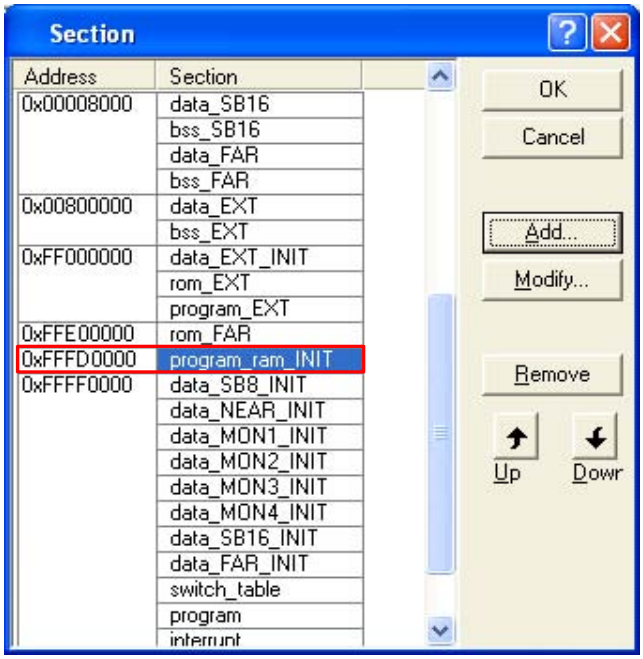


Select 0xFFFF0000 in the Address column and click Add. Set 0xFFFD0000 in the Section address window and click OK.

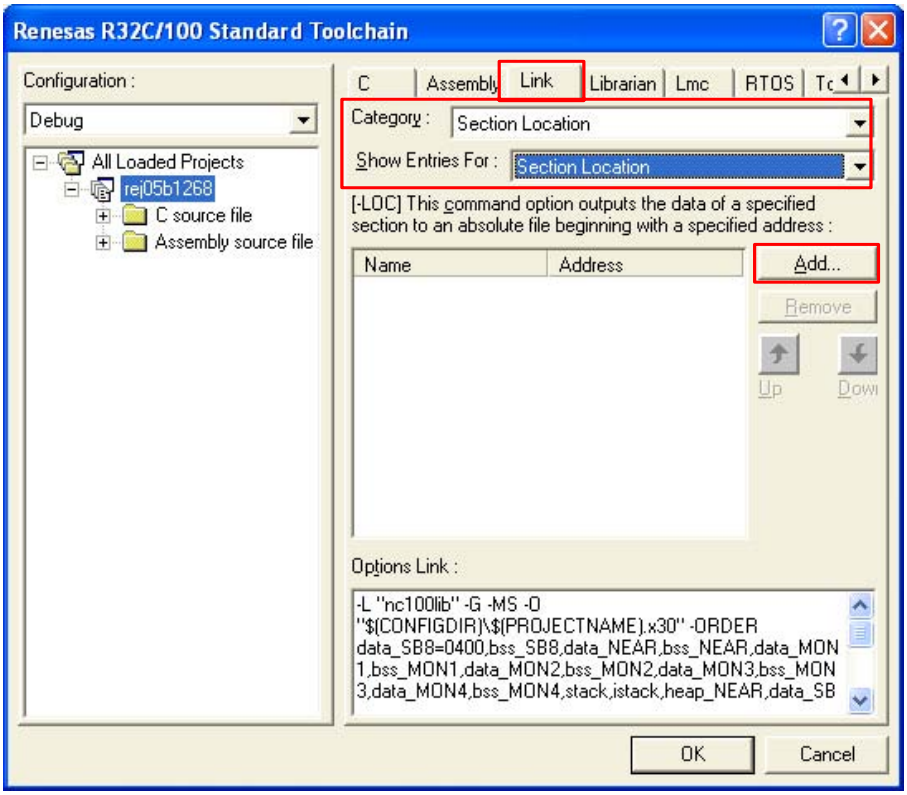


Select the field next to the added 0xFFFD0000 in the Section column and click Add. When the Add section window is displayed, type program_ram_INIT in the Section name field and click OK.



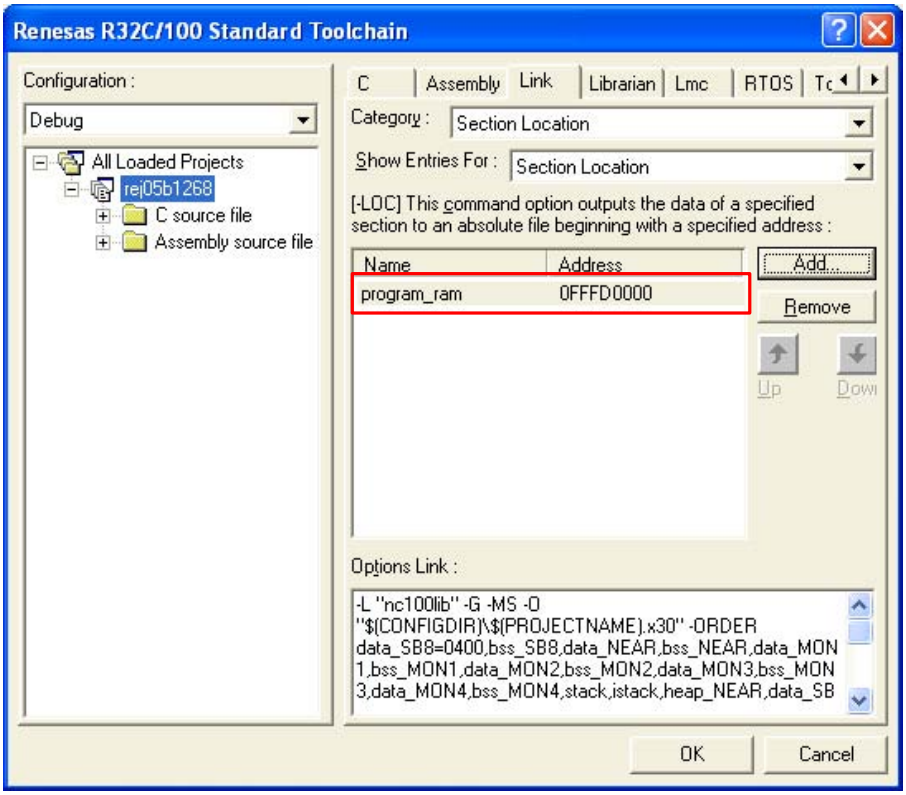


Next, select Section Location from the Category pull-down menu and Section Location from the Show Entries For pull-down menu. Click Add to display the Add Location window.



Type program_ram in the Section name field and 0xFFFD0000 in the Address field, and click OK.

The program_ram section is stored from address 0FFFD0000 (in the ROM) using the above option settings.



3.7 Notes on Flash Memory Rewriting

(1) Note on Power Supply

- Keep the supply voltage constant within the range specified in the electrical characteristics while a rewrite operation on the flash memory is in progress. If the supply voltage goes beyond the guaranteed value, the device cannot be guaranteed.

(2) Note on Hardware Reset

- Do not perform a hardware reset while a rewrite operation on the flash memory is in progress.

(3) Note on Flash Memory Protection

- If an ID code written in an assigned address has an error, any read/write operation on the flash memory in standard serial I/O mode is disabled.

(4) Notes on Programming

- . Do not set the FEW bit in the FMCR register to 1 (CPU rewrite mode) in low speed mode or low power mode.
- The program, block erase, lock bit program, and protect bit program software commands are interrupted by an NMI, a watchdog timer interrupt, an oscillator stop detection interrupt, or a low voltage detection interrupt. If any of the software commands above are interrupted, erase the corresponding block and then execute the same command again. If the block erase command is interrupted, the lock bit and protect bit values become undefined. Therefore, disable the lock bit, and then execute the block erase command again.

(5) Notes on Interrupts

- To use interrupts assigned to the relocatable vector table, the vector table should be addressed in RAM space.
- If either of an NMI, a watchdog timer interrupt, an oscillator stop detection interrupt, or a low voltage detection interrupt is generated, the flash memory module automatically enters read array mode. Therefore these interrupts are enabled even during a rewrite operation. However, the rewrite operation in progress is aborted by an interrupt and registers FMR0 and FRSR0 are reset. When the interrupt handler has ended, set the LBD bit in the FMR1 register to 1 (lock bit protection disabled) to re-execute the rewrite operation.
- Instructions BRK, INTO, and UND, which refer to data on the flash memory, cannot be used in this mode.

(6) Notes on Rewrite Control Program

- If the supply voltage decreases during the rewrite operation of blocks having the rewrite control program, the rewrite control program may not be successfully rewritten, which might lead to the rewrite operation itself may not being performed. In this case, perform the rewrite operation by serial programmer or parallel programmer.

(7) Note on Number of Programming/Erase Operations and Software Command Execution Time

- The time to execute software commands (program, block erase, lock bit program, and protect bit program) increases as the number of program and erase operations increases. If the number of program and erase operations exceed the minimum endurance value specified in the electrical characteristics, it may take an unpredictable amount of time to execute the software commands. The wait time for executing software commands should be set much longer than the execution time specified in the electrical characteristics.

(8) Other Notes

- The required time to perform program/erase operations specified in the electrical characteristics found in the user's manual can be guaranteed within the minimum values of programming/erasure endurance specified in the same table. Even if the number of programming/erasure exceeds the minimum endurance value, the program/erase operation may be performed.
- Chips repeatedly programmed and erased for debugging should not be used for commercial products.

4. Sample Program

A sample program can be downloaded from the Renesas Electronics website.

4.1 Description of the Sample Program

In the sample program, the MCU starts up in PLL mode, enters CPU rewrite mode (EW0 mode) triggered by an INT0 interrupt request, and backs up the 16-byte data stored in the internal RAM to block 7 (addresses FFFA0000h to FFFAFFFFh). Block 7 is block-erased and then programmed (to save data in the RAM area) in the sample program. Figure 4.1 shows the Sample Program Memory Map.

The sample program execution status can be confirmed by the status of port P4. Table 4.1 shows the Port P4 Status.

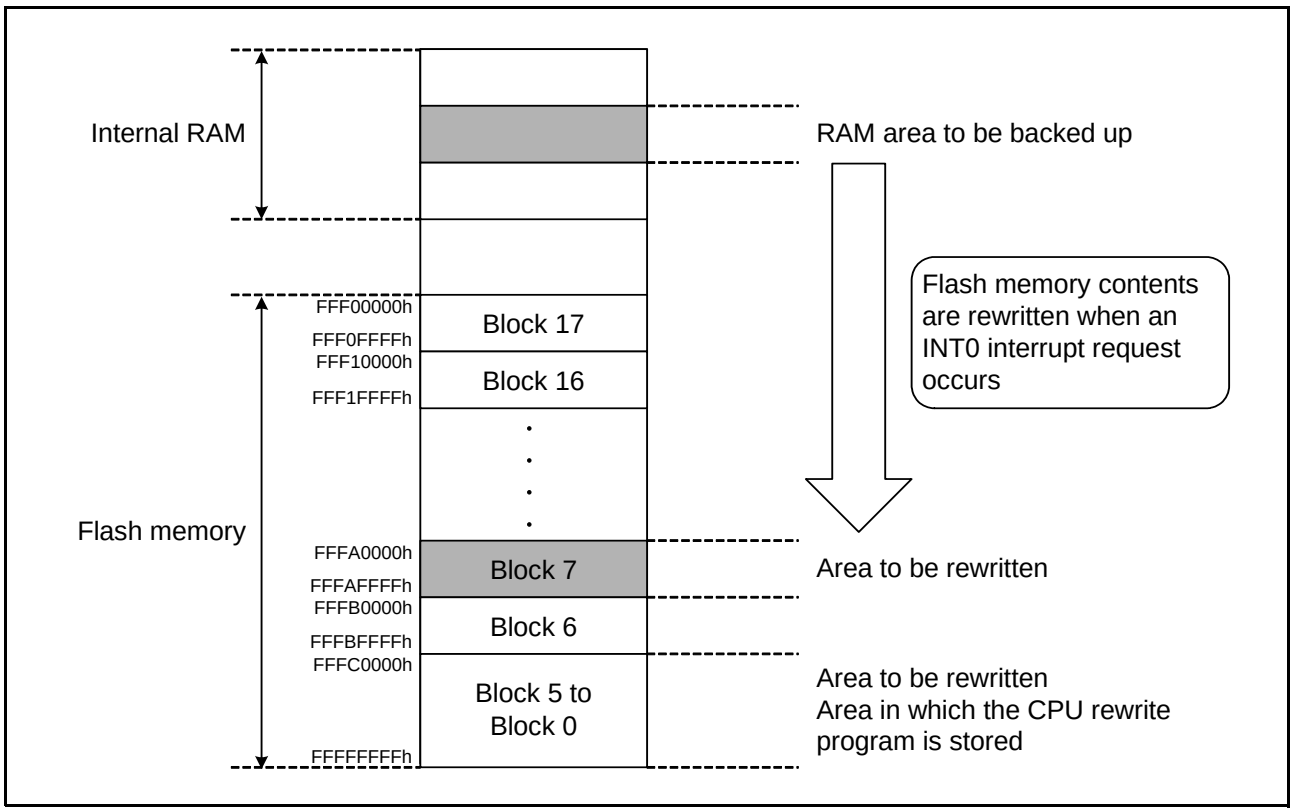


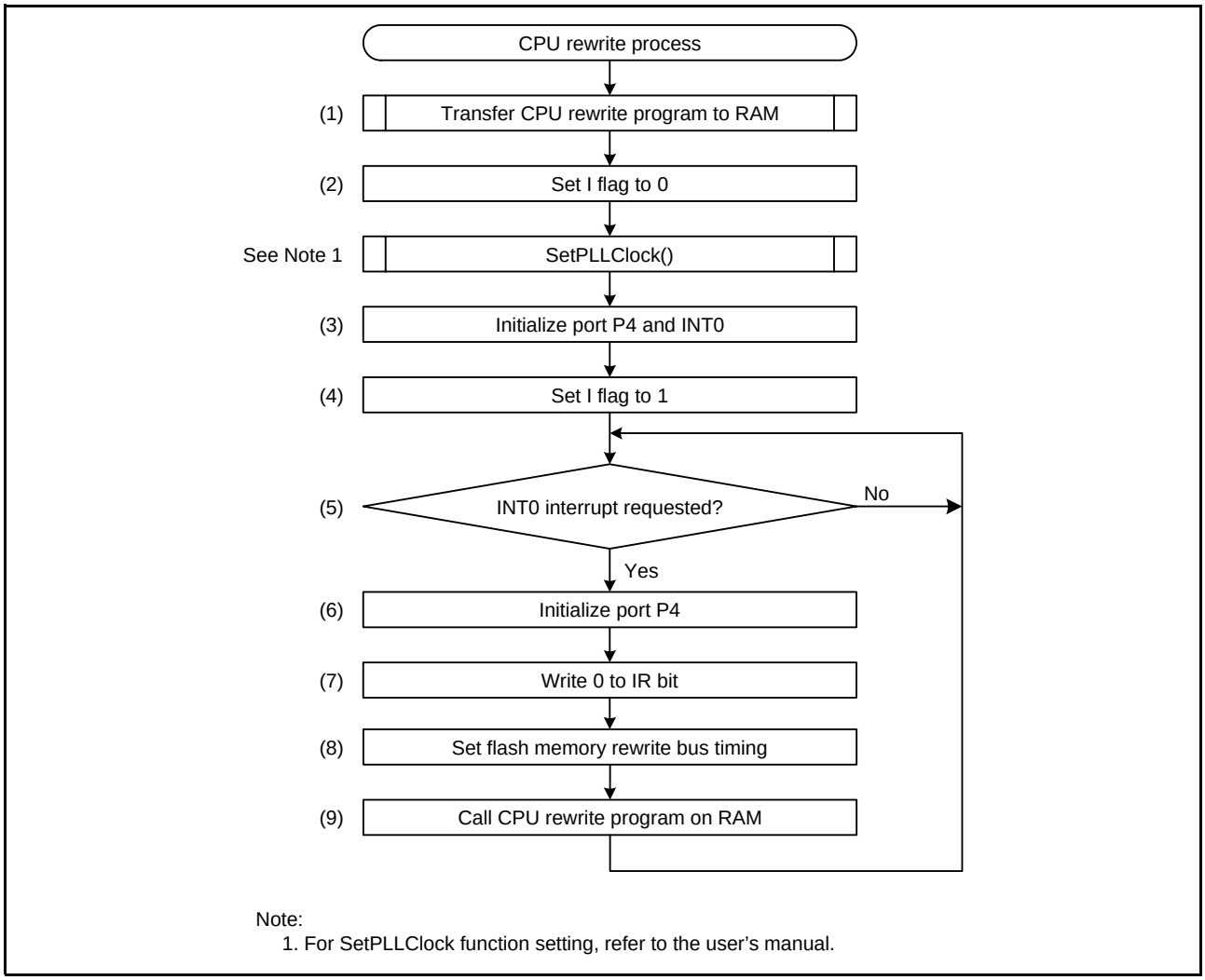
Figure 4.1 Sample Program Memory Map

Table 4.1 Port P4 Status

P4	Output Status 0	
	1	0
P4_0	Stopped	Rewriting CPU
P4_1	Program erase successfully completed	Erase error occurred
P4_2	Program successfully completed	Program error occurred

4.2 Program Flowchart

Figure 4.2 shows the entire process of the sample program.



Note:
1. For SetPLLClock function setting, refer to the user's manual.

Figure 4.2 CPU Rewrite Process

Figure 4.3 shows the process flowchart of the CPU rewrite program.

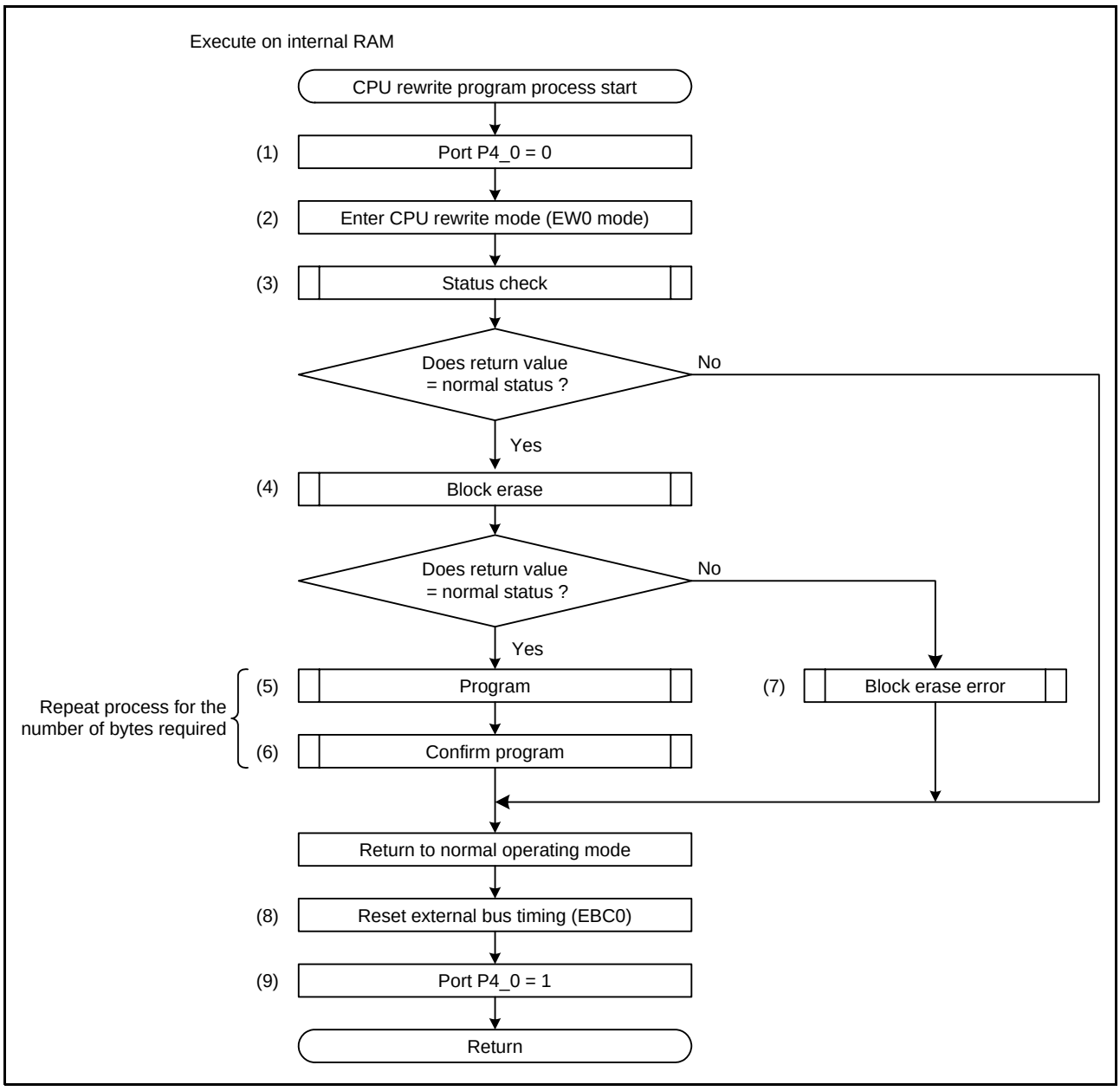


Figure 4.3 CPU Rewrite Program on RAM

(1) Block Erase

Figure 4.4 shows the flowchart of Block Erase Function. Write commands to the flash memory in 16-bit units.

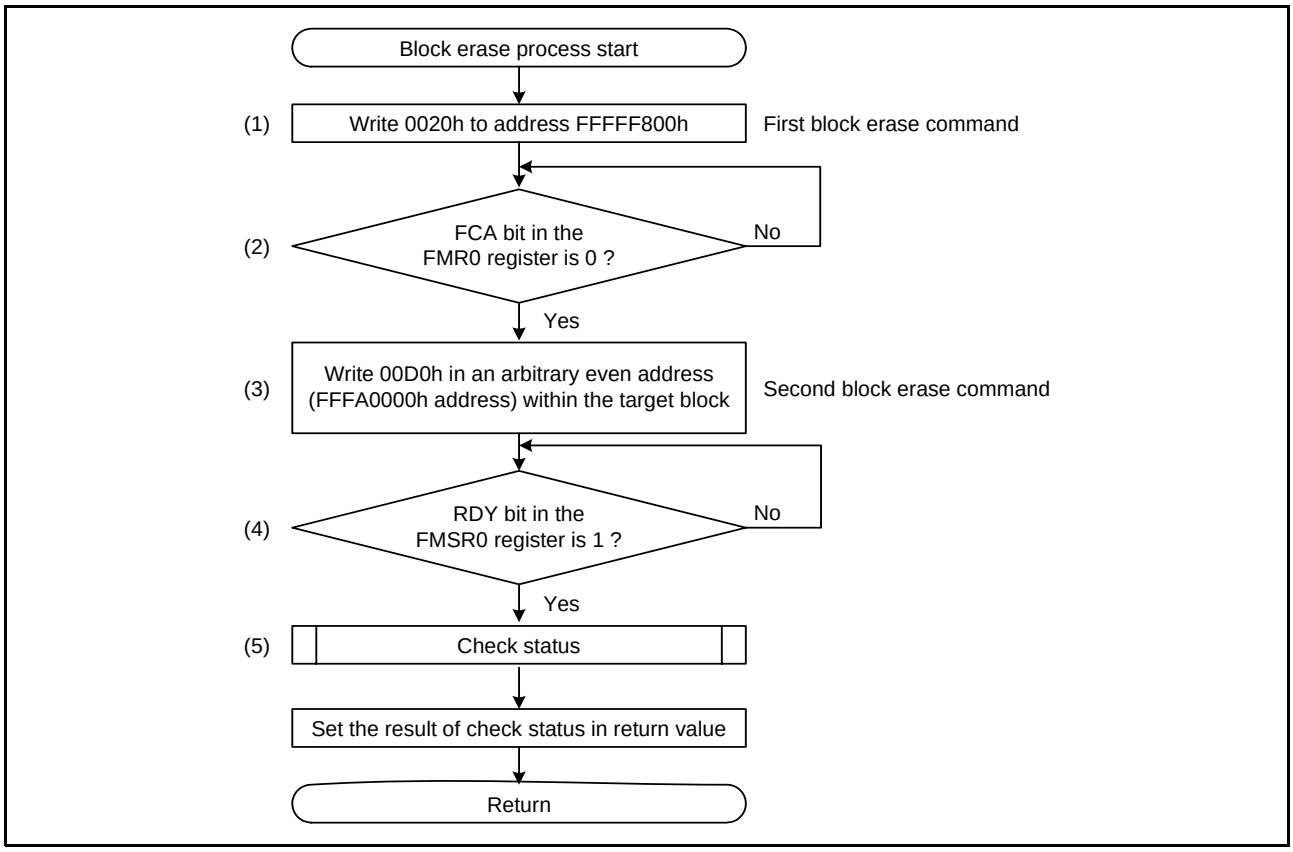


Figure 4.4 Block Erase Function

(2) Block Erase Error

Figure 4.5 shows the flowchart of Block Erase Error Function. Write commands to the flash memory in 16-bit units. When an erase error occurs in the block erase command, reexecute the block erase command after executing the clear status register command. Repeat the block erase at least three times after the erase error stops occurring.

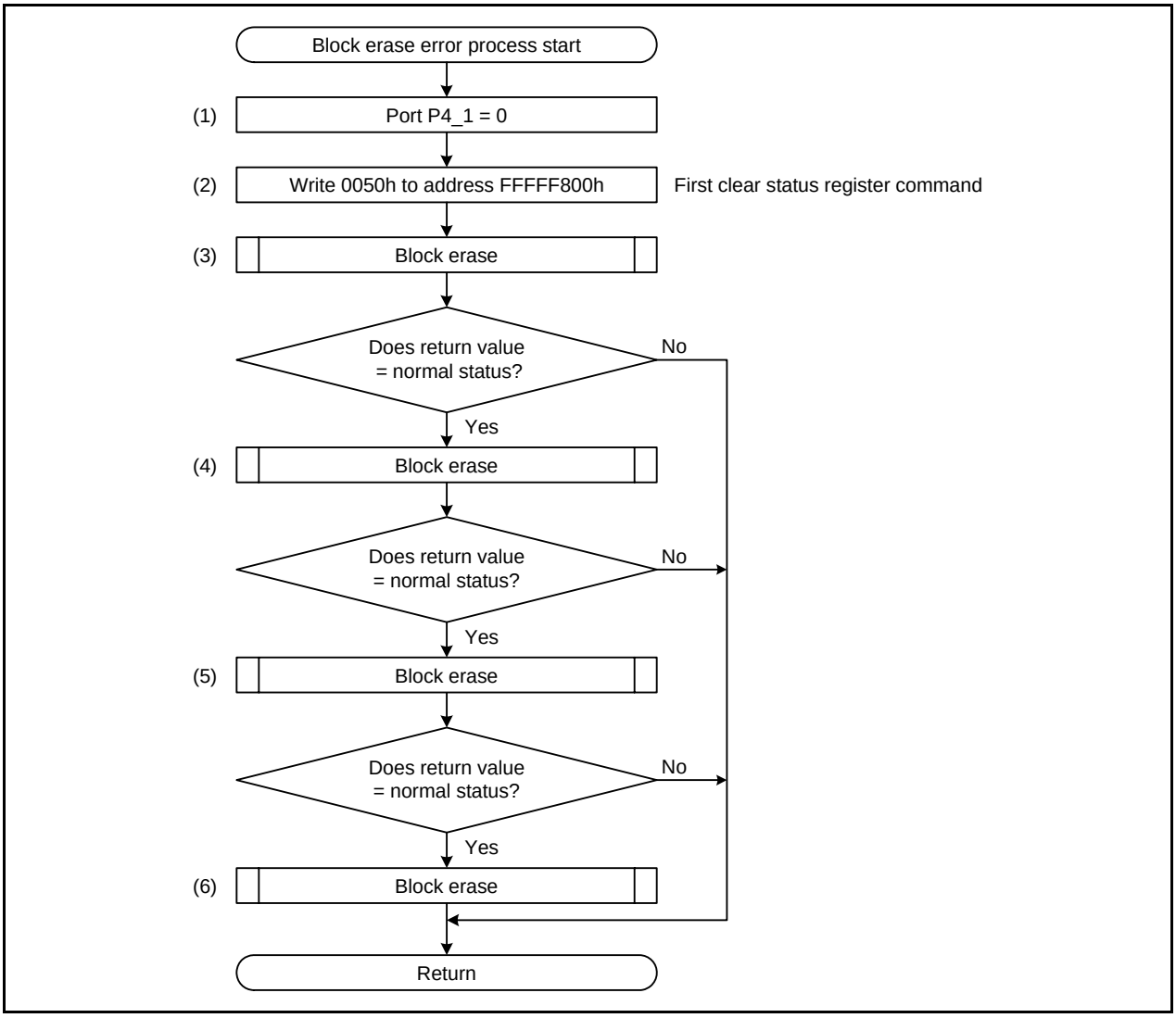


Figure 4.5 Block Erase Error Function

(3) Program

Figure 4.6 shows the flowchart of Program Function. Write commands to the flash memory in 16-bit units.

This program is executed in 64-bit units (4 words). The second through the fifth commands are a series of commands. Fix the 29 upper bits of the write address. In the 3 lower bits, set the following values in the order shown starting from the second command: 000b-010b-100b-110b (0h-2h-4h-6h or 8h-Ah-Ch-Eh).

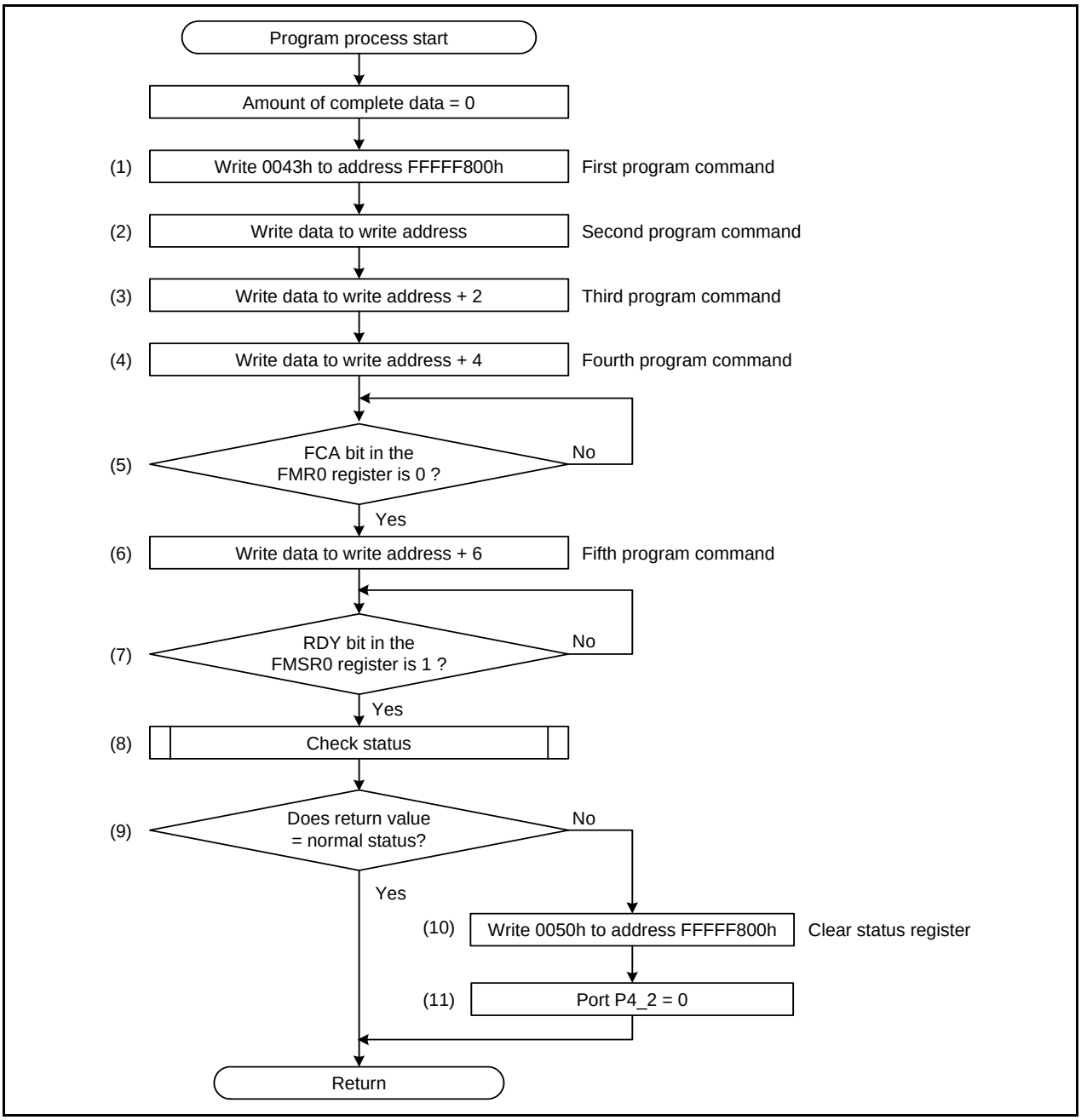


Figure 4.6 Program Function

(4) Program confirmation

Figure 4.7 shows the flowchart of Program Confirmation Function. Confirm whether the value written in the program function is the value expected.

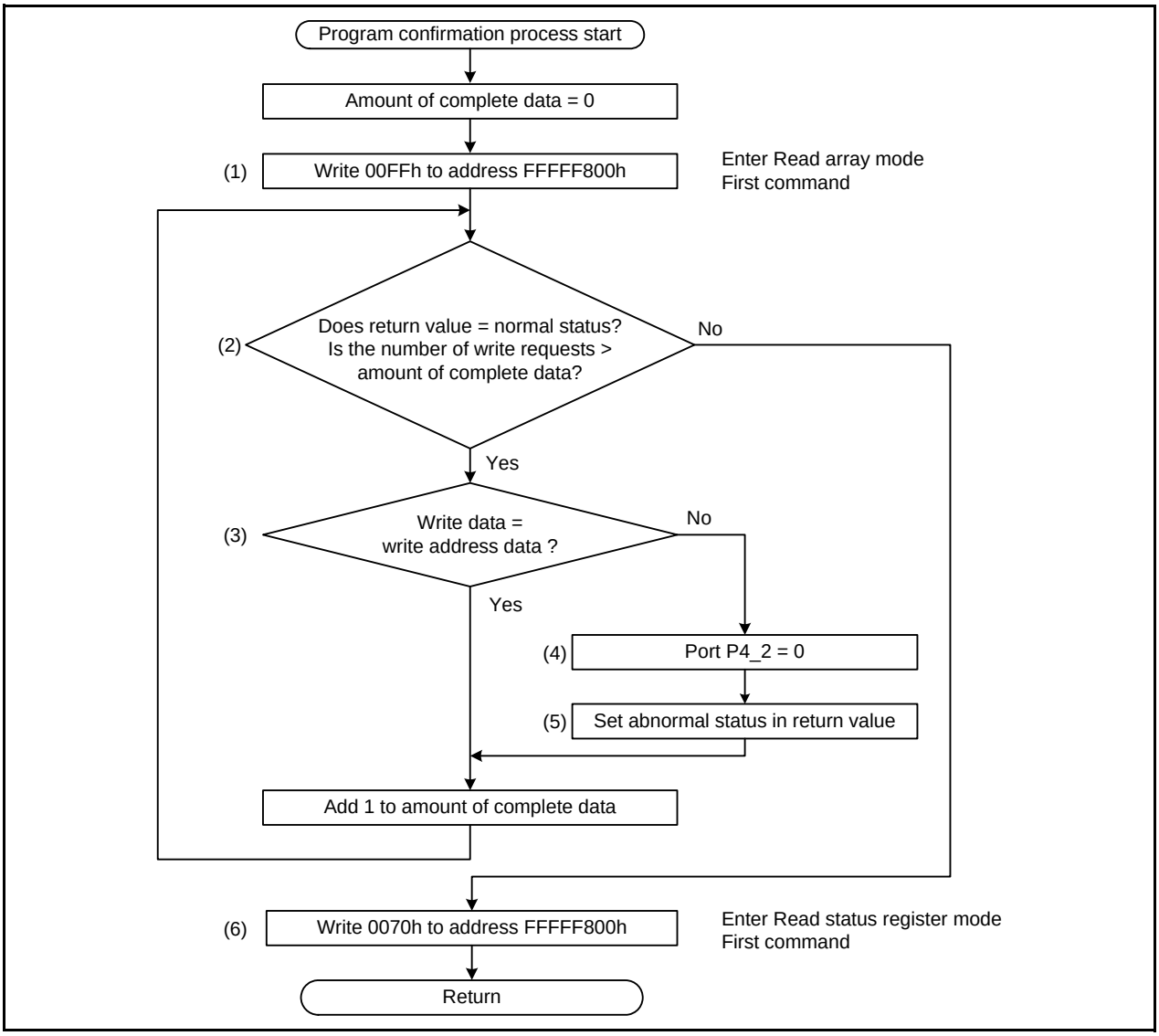


Figure 4.7 Program Confirmation Function

(5) Status Check

Figure 4.8 shows the flowchart of Status Check Function. When an error occurs, execute the clear status register command, then handle the error. If erase errors or program errors occur frequently even though the program is correct, the corresponding block may be disabled.

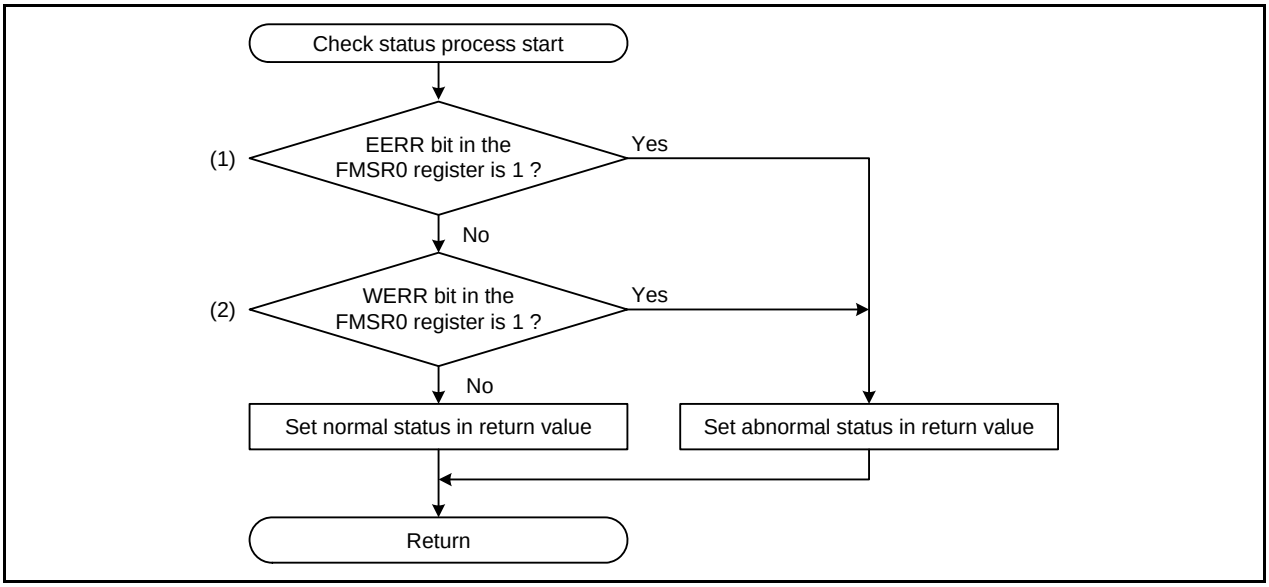


Figure 4.8 Status Check Function

5. Reference Documents

User's Manual
R32C/118 Group User's Manual Rev.1.00
The latest version can be downloaded from the Renesas Electronics website.

Technical Update/Technical News
The latest information can be downloaded from the Renesas Electronics website.

C compiler manual
R32C/100 Family C compiler package V.1.02 C compiler user manual Rev.2.00
The latest version can be downloaded from the Renesas Electronics website.

Website and Support

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

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

REVISION HISTORY	R32C/100 Series Rewriting ROM Area Using EW1 Mode of CPU Rewrite Mode

Rev.	Date	Description	
		Page	Summary
1.00	May 28, 2010	-	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 an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

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

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.

   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.

   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

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

4. Clock Signals

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

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

5. Differences between Products

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

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

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
   - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
   - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
   - "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

RENESAS

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

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

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

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

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

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

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

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

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

Renesas Electronics Taiwan Co., Ltd.
7F, 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

Colophon 1.0