To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

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

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

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.

M16C/62P Group

Flash Memory Version CPU Rewrite Mode (EW0 Mode) Sample

1. Abstract

This application note presents an example method for using CPU rewrite mode (EW0 mode) in the flash memory version of microcomputers.

2. Introduction

The explanation of this issue is applied to the following condition:
Applicable MCU: M16C/62P Group

This program can also be used when operating other microcomputers within the M16C family, provided they have the same SFR (Special Function Registers) as the M16C/62P microcomputers. However, some functions may have been modified.
Refer to the User's Manual for details. Use functions covered in this Application Note only after careful evaluation.

3. Explanation of Example Usage

Features of EW0 mode:
In EW0 mode, the CPU rewrite program is transferred into the RAM, and by issuing programming and erasing commands from the CPU rewrite program in the RAM, the user ROM and the data areas can be rewritten. Since while in EW0 mode the CPU continues operating even during a programming or erasing operation, peripheral function interrupts can be accepted during a programming or erasing operation providing that the vectors for those interrupts and the interrupt service routines are located in the RAM.

3.1 CPU Rewrite Mode (EW0 Mode) Execution Flow

Figure 1 shows CPU Rewrite Mode (EW0 Mode) Execution Flow.

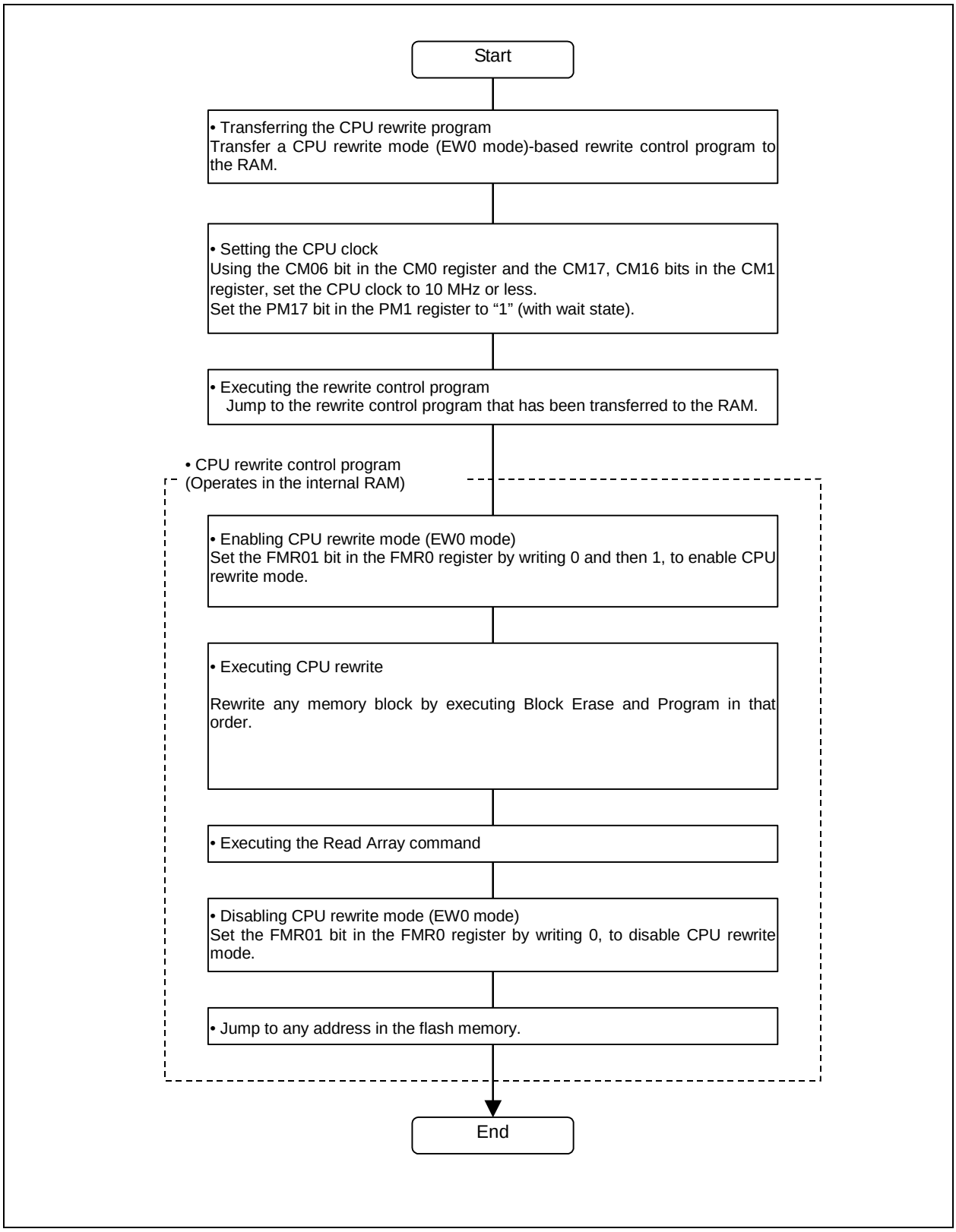
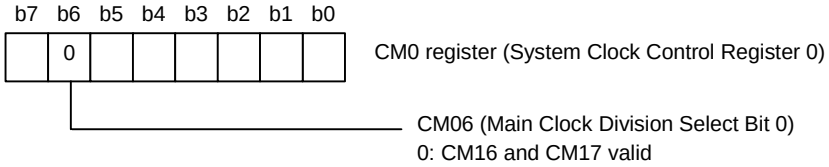


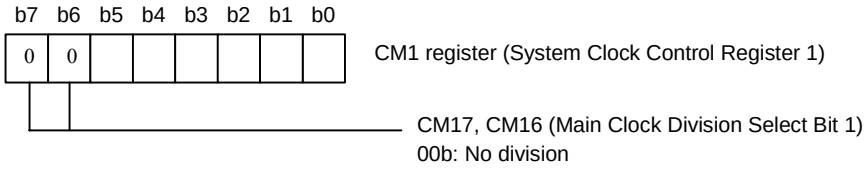
Figure 1 CPU Rewrite Mode (EW0 Mode) Execution Flow

3.2 Set Up Procedure

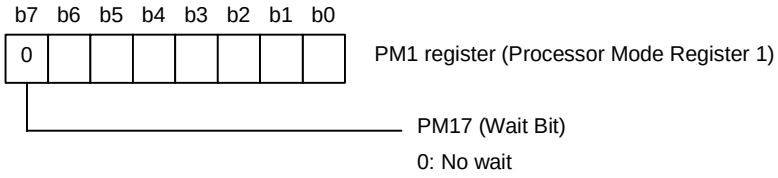
3.2.1 Setting CPU Clock

(1) Setting the main clock divide-by-N value





(2) Setting wait states



3.2.2 Transferring the CPU Rewrite Control Program into RAM

The CPU rewrite control program needs to be run in RAM. Here, the following explains an example for transferring the CPU rewrite control program from the ROM area in which it is stored beginning with the address 0FA000h to an area in RAM.

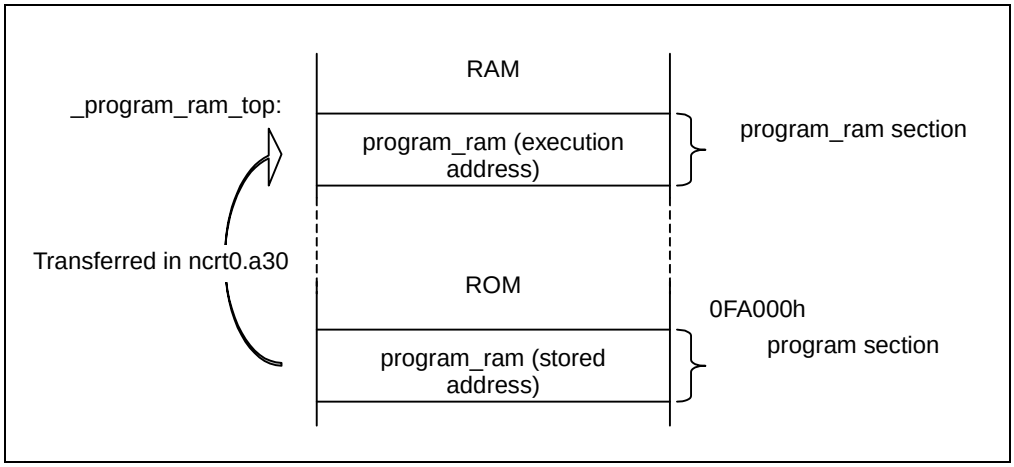


Figure 2 Program Location

(1) Change the Section Name.

Add a section name “program_ram,” and locate the program to be run in RAM in that section. To relocate the program from the program section to the program_ram section, write a process as shown below.

```
void main(void)
{
    /* This program part is located in the program section */
}
```

```
/* The program part following the #pragma SECTION declaration is located in the program_ram section */
#pragma SECTION program program_ram
void low_power(void)
{
    /* This program part is located in the program_ram section */
}
```

(2) Changing sect30.inc

Add the program_ram section to sect30.inc. In the example here, it is located after the heap section. Note also that the program_ram_top label is used when transferring the program.

```
;------------------------------------------------------------
;       heap section
;------------------------------------------------------------
        .section   heap,DATA
heap_top:
        .blkb      HEAPSIZE
```

```
;------------------------------------------------------------
; RAM program area
;------------------------------------------------------------
        .section   program_ram,ALIGN
_program_ram_top:
        .glb       _program_ram_top
```

Add here.

(3) Transferring the Program

Add a process for transferring the program into RAM in the startup routine (ncrt0.a30).

```
;======================================================================
; Initialize standard I/O
;----------------------------------------------------------------------
.if __STANDARD_IO__ != 1
        .glb       _init
        .call      _init,G
        jsr.a      _init
.endif
```

```
;======================================================
; Program Ram initialize
;   _from_addr is defined by as30 option "-D_from_addr=0fa000h"
;------------------------------------------------------------
      N_BCOPY _from_addr,_program_ram_top,program_ram
;
```

Add here

```
;=====================================================
; Call main() function
;---------------------------------------------------------------------
        ldc     #0h,fb    ; for debuger

        .glb    _main
        jsr.a   _main
```

(4) Specifying the Program Storage Location

To run the program transferred into RAM, it is necessary to specify in the linker (ln30) that the program storage address (in ROM) and execution address (in RAM) be located separately.

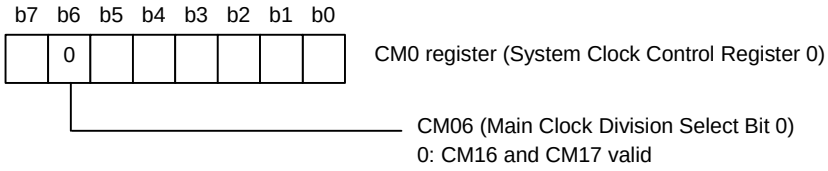
```
ln30 –LOC program_ram=0FA000
```

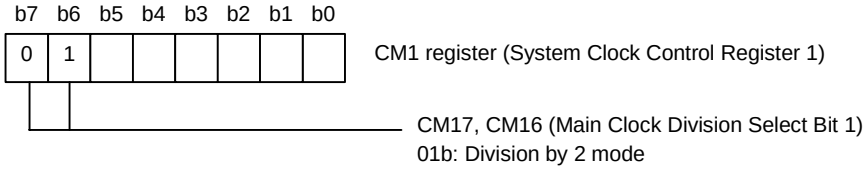
In the above option, the program_ram section is stored beginning with the address 0FA000h.

3.2.3 Processing in the CPU Rewrite Control Program

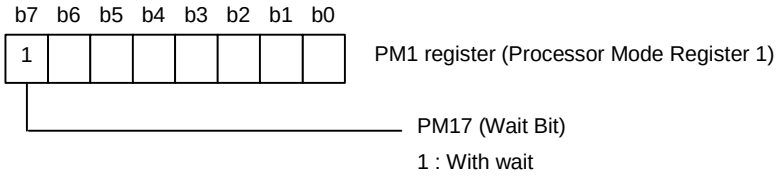
(1) Set the CPU clock to 10MHz or less.

Setting the main clock divide-by-N value

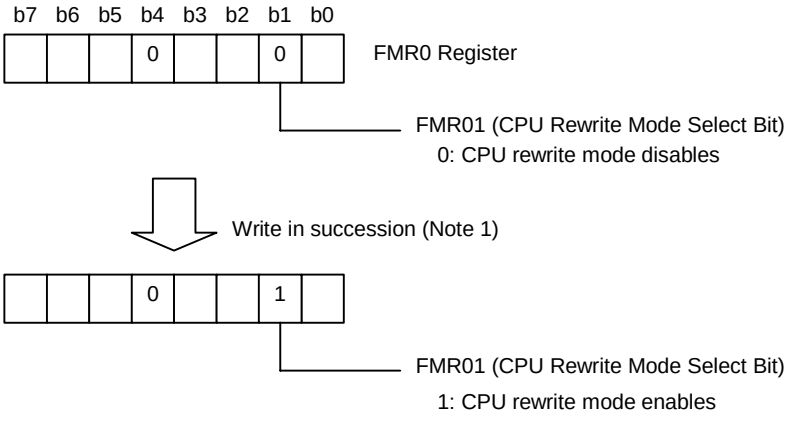




Setting wait states



(2) Enable the CPU rewrite mode.



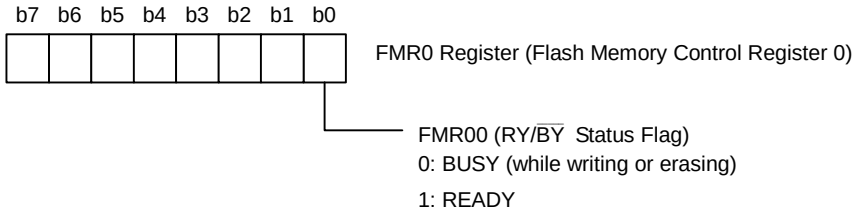
Note: To set the FMR01 bit to “1”, write “0” and then “1” to the FMR01 bit in succession. Make sure no interrupts or DMA transfers occur before the CPU writes “1” after writing “0”.
Make sure writes to the FMR01 bit is performed in other than the internal flash memory.
Also make sure this write operation is performed while the $\overline{\text{NMI}}$ pin is in the high state.
Set this bit to “0” after read array command.

(3) Block erase processing

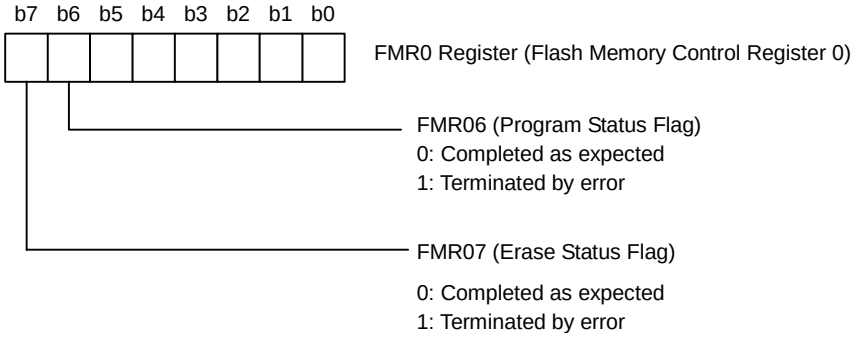
- Executing the Block Erase command
  Write “0020h” and then “00d0h” in succession to the most significant address (Note 1) of the memory block to be block-erased.

  Note 1: If block 5 is block-erased, write to the address F7FFEh.

- Waiting for Block Erase to complete
  Wait until the FMR00 bit in the FMR0 register is set to “1” (ready).



- Status check
  Check the FMR06 and FMR07 bits in the FMR0 register to see if an erase error has occurred. If an error is found to have occurred during the erase operation, write “0050h” (Clear Status command) to the address to which the Block Erase command was written, to stop CPU rewrite processing.



(4) Programming process
Program the entire area of the relevant memory block one word at a time, by following the procedure described below.

- Executing the Program command
  Write “0040h” (Program command) and then the program data to the address to be programmed.

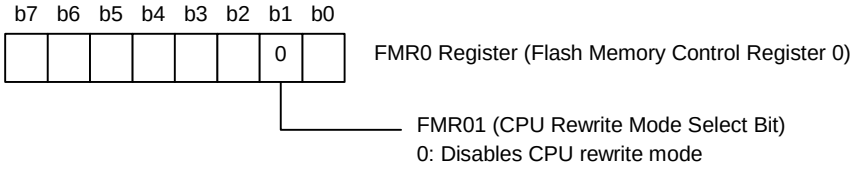
- Waiting for Program to complete
  Wait until the FMR00 bit in the FMR0 register is set to “1” (ready).

- Status check
  Check the FMR06 and FMR07 bits in the FMR0 register to see if a programming error has occurred. If an error is found to have occurred during the programming operation, write “0050h” (Clear Status command) to the address to which the Program command was written, to stop CPU rewrite processing.

(5) Disable CPU rewrite mode.

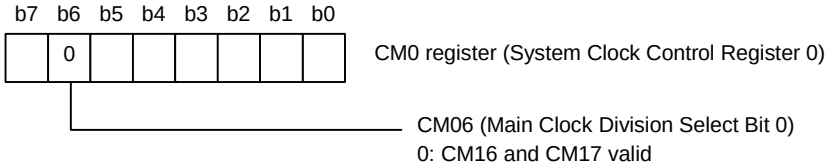
- Executing the Read Array command
  Write “00FFh” (Read Array command) to the most significant address of the relevant memory block.

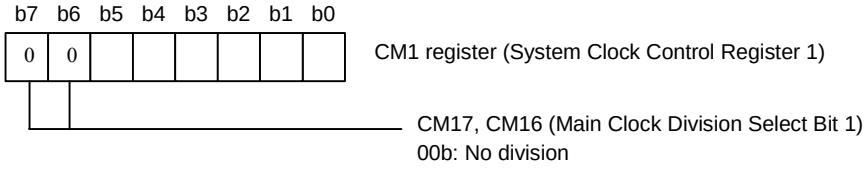
- Disable CPU rewrite mode



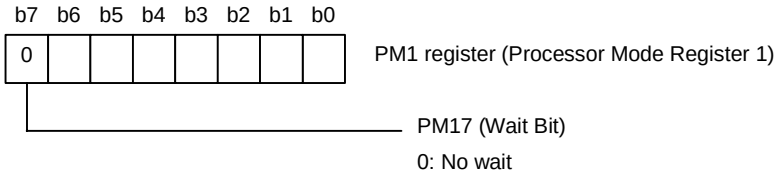
(6) Restore the CPU clock to the original one.

- Setting the main clock divide-by-N value





- Setting wait states



(7) Return to the program in the flash memory.

3.3 Precautions in CPU Rewrite Mode (EW0 Mode)

The following describes the precautions to be observed when using CPU rewrite mode (EW0 mode). (Please consult the manual to get the latest information.)

(1) Operating Speed

Before entering CPU rewrite mode (EW0 or EW1 mode), set the CM11 bit in the CM1 register to "0" (main clock), select 10 MHz or less for CPU clock using the CM06 bit in the CM0 register and CM17 to CM16 bits in the CM1 register. Also, set the PM17 bit in the PM1 register to "1" (with wait state).

(2) Instructions inhibited against use

The following instructions cannot be used in EW0 mode because the flash memory's internal data is referenced: UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction.

(3) Interrupts (EW0 Mode)

- Any interrupt which has a vector in the relocatable vector table can be used providing that its vector is transferred into the RAM area.
- The $\overline{\text{NMI}}$ and watchdog timer interrupts can be used because the FMR0 register and FMR1 register are initialized when one of those interrupts occurs. The jump addresses for those interrupt service routines should be set in the fixed vector table.
  Because the rewrite operation is halted when a $\overline{\text{NMI}}$ or watchdog timer interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.
- The address match interrupt cannot be used because the flash memory's internal data is referenced.

(4) Interrupts (EW1 Mode)

- Make sure that any interrupt which has a vector in the variable vector table or address match interrupt will not be accepted during the auto program or auto erase period.
- Avoid using watchdog timer interrupts.
- The $\overline{\text{NMI}}$ interrupt can be used because the FMR0 register and FMR1 register are initialized when this interrupt occurs. The jump address for the interrupt service routine should be set in the fixed vector table.
  Because the rewrite operation is halted when a $\overline{\text{NMI}}$ interrupt occurs, the rewrite program must be executed again after exiting the interrupt service routine.

(5) How to Access

To set the FMR01, FMR02, or FMR11 bit to “1”, write “0” and then “1” in succession. This is necessary to ensure that no interrupts or DMA transfers will occur before writing "1" after writing “0”. Also only when $\overline{\text{NMI}}$ pin is “H” level.

(6) Writing in the User ROM Area (EW0 Mode)

If the power supply voltage drops while rewriting any block in which the rewrite control program is stored, a problem may occur that the rewrite control program is not correctly rewritten and, consequently, the flash memory becomes unable to be rewritten thereafter. In this case, standard serial I/O or parallel I/O mode should be used.

(7) Writing in the User ROM Area (EW1 Mode)

Avoid rewriting any block in which the rewrite control program is stored.

(8) DMA Transfer

In EW1 mode, make sure that no DMA transfers will occur while the FMR00 bit in the FMR0 register = 0 (during the auto program or auto erase period).

(9) Writing Command and Data

Write the command code and data at even addresses.

(10) Wait Mode

When shifting to wait mode, set the FMR01 bit to “0” (CPU rewrite mode disabled) before executing the WAIT instruction.

(11) Stop Mode

When the microcomputer enters stop mode, execute the instruction which sets the CM10 bit to "1" (stop mode) after setting the FMR01 bit to "0" (CPU rewrite mode disabled) and disabling the DMA transfer.

(12) Low power dissipation mode, on-chip oscillator low power dissipation mode

If the CM05 bit is set to “1” (main clock stop), the following commands must not be executed.

- Program
- Block erase
- Erase all unlocked blocks
- Lock bit program

4. Sample Programming Code

The following shows an example program for using CPU rewrite mode (EW0 mode) to back up the internal RAM (addresses 1800h–2BFFh) to the block 5 (addresses F0000h–F7FFFh) as triggered by an $\overline{\text{INT0}}$ interrupt request generated. In this example program, block 5 is block-erased and then programmed (to save the RAM area).

Memory map:

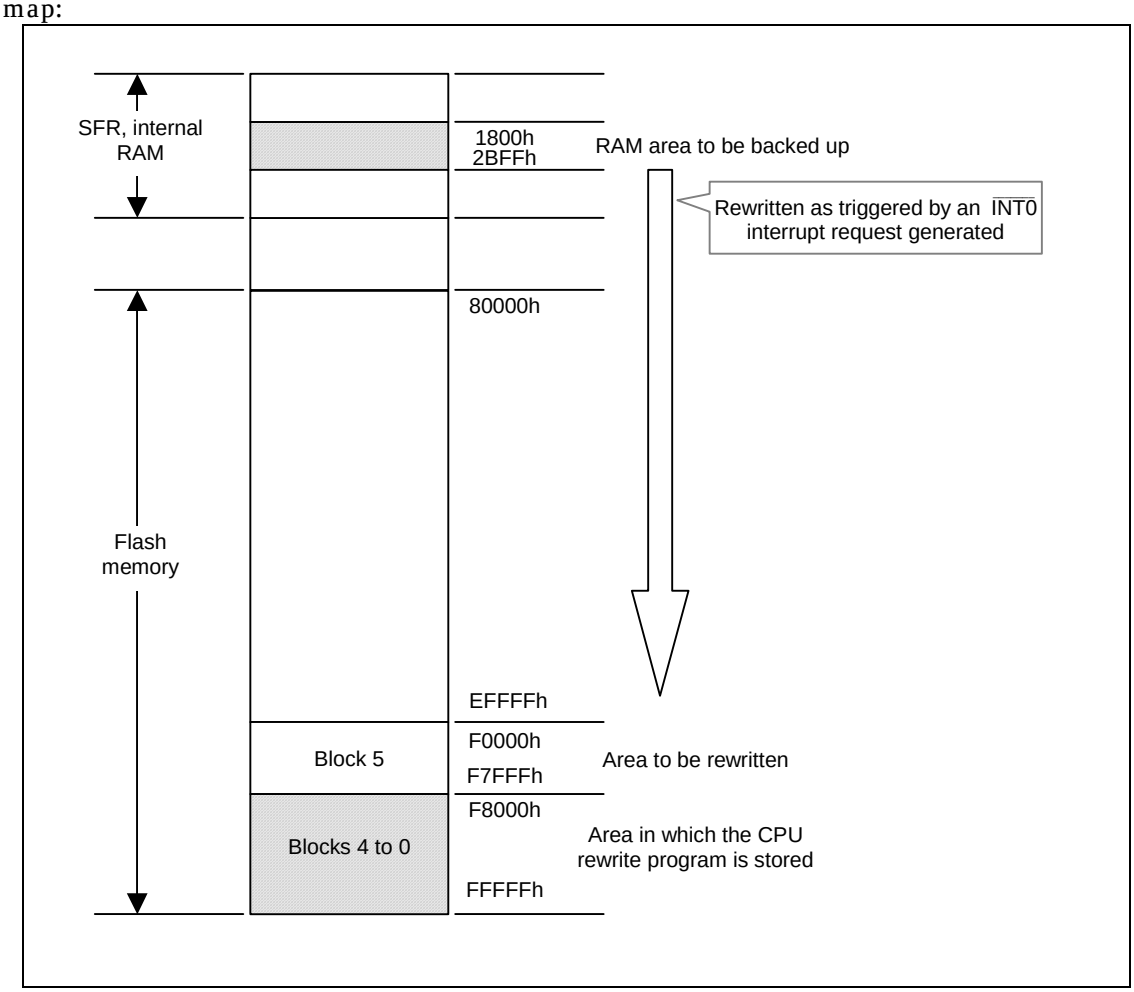


Figure 3 Memory Map

Operating Conditions:
(1) VCC1=VCC2=5V
(2) XIN=16MHz
(During CPU rewrite mode, the device must be operated with 4 MHz by setting the CM06 bit in the CM0 register and the CM16, CM17 bits in the CM1 register, and PM17 bit in the PM1 register.)

Amount of memory used:
The CPU rewrite program transferred into RAM requires 312 bytes of memory when it is compiled without optimization options.

4.1 Processing Flow

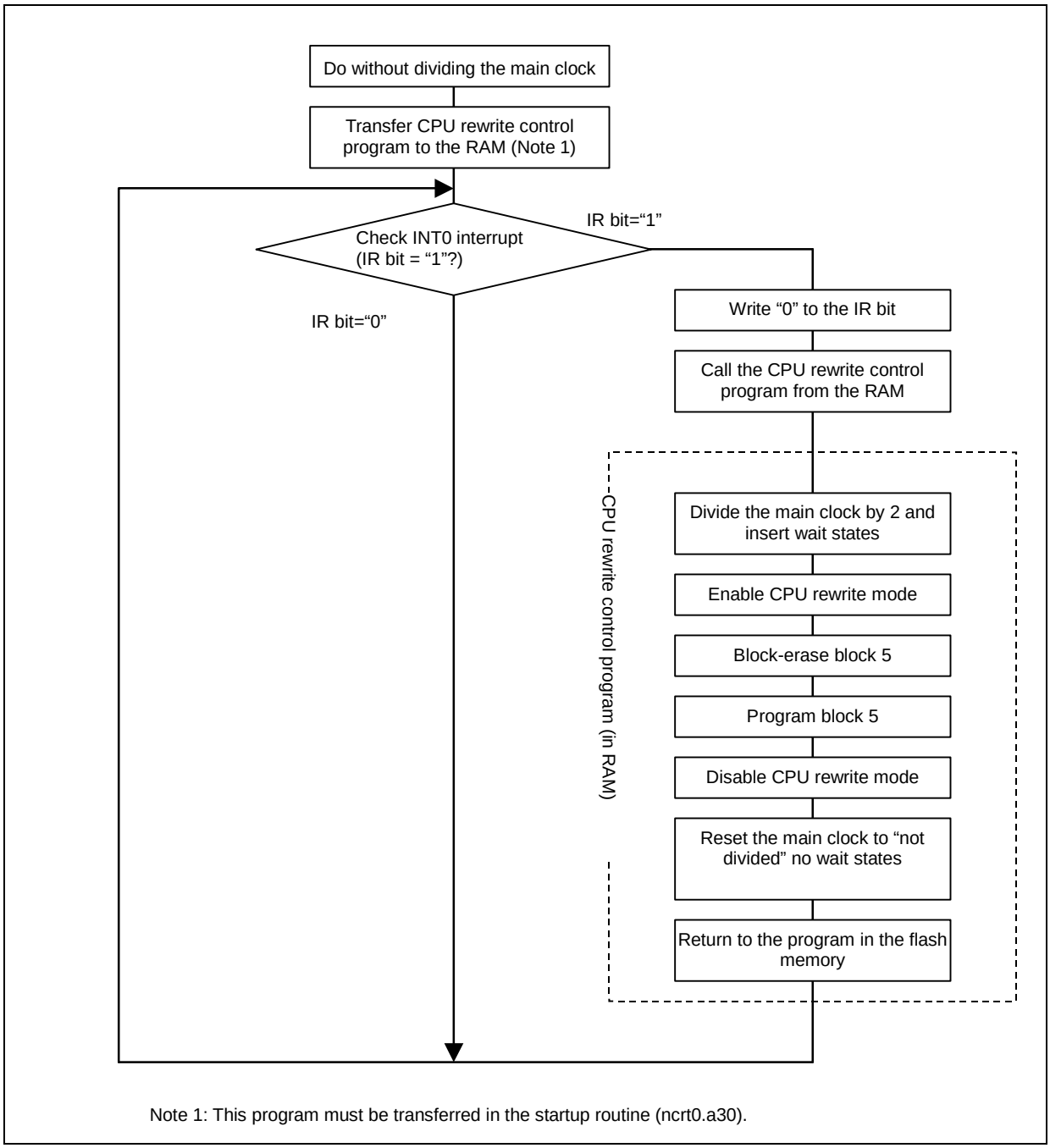


Figure 4 Sample Program Processing Flow

5. Reference

Renesas Technology Corporation Home Page
http://www.renesas.com/

E-mail Support
E-mail: csc@renesas.com

Hardware Manual
M16C/62P Group Hardware Manual
(Use the latest version on the home page: http://www.renesas.com)

REVISION HISTORY

Rev.	Date	Description	
		Page	Summary
1.00	2005.04.25	-	First edition issued

Keep safety first in your circuit designs!

1. Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
2. Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.