RX Family

Setting Example of DTC Block Transfer Mode and Chain Transfer

R01AN2130EJ0100
Rev. 1.00
Aug. 1, 2014

Abstract

This document describes block transfer and chain transfer of the data transfer controller (DTC) in the RX Family.

Products

RX Family *

* Except RX610, RX62N, RX621, RX62T, and RX62G Groups.

This document describes block transfer and chain transfer using the RX63N Group. When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Contents

1. Specifications

DTC block transfer and chain transfer are performed using the IRQ as the DTC activation source.

When a falling edge of the external interrupt request pin (IRQ10) is detected, block transfer is performed. When a falling edge of the external interrupt request pin (IRQ8) is detected, chain transfer is performed.

Table 1.1 lists the Peripheral Functions and Their Applications and Figure 1.1 shows the Block Diagram.

Table 1.1 Peripheral Functions and Their Applications

Peripheral Function	Application
DTCa	ROM to RAM transfer (block transfer and chain transfer)
IRQ10	DTC activation source for block transfer
IRQ8	DTC activation source for chain transfer

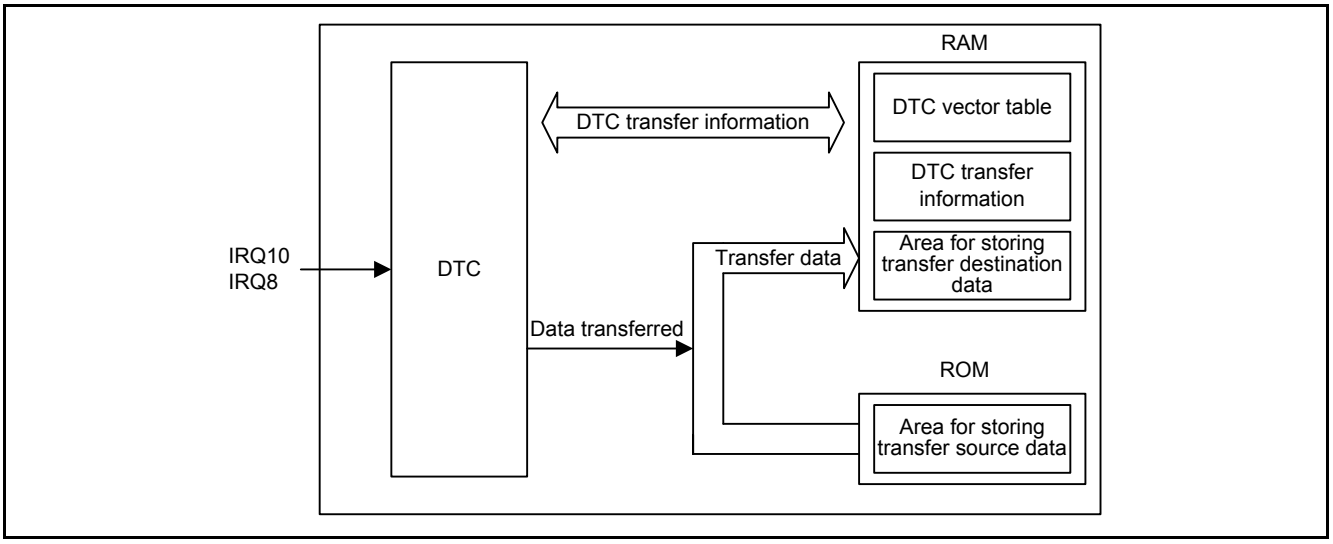


Figure 1.1 Block Diagram

2. Operation Confirmation Conditions

The sample code accompanying this application note has been run and confirmed under the conditions below.

Table 2.1 Operation Confirmation Conditions

Item	Contents
MCU used	R5F563NBDDFC (RX63N Group)
Operating frequencies	- Main clock: 12 MHz - PLL: 192 MHz (main clock divided by 1 and multiplied by 16) - System clock (ICLK): 96 MHz (PLL divided by 2) - Peripheral module clock B (PCLKB): 48 MHz (PLL divided by 4)
Operating voltage	3.3 V
Integrated development environment	Renesas Electronics Corporation High-performance Embedded Workshop Version 4.09.01
C compiler	Renesas Electronics Corporation C/C++ Compiler Package for RX Family V.1.02 Release 01
	Compile options -cpu=rx600 -output=obj="$(CONFIGDIR)\$(FILELEAF).obj" -debug -nologo (The default setting is used in the integrated development environment.)
iodefine.h version	Version 1.6A
Endian	Little endian
Operating mode	Single-chip mode
Processor mode	Supervisor mode
Sample code version	Version 1.00
Board used	Renesas Starter Kit+ for RX63N (product part no.: R0K50563NC000BE)

3. Reference Application Note

For additional information associated with this document, refer to the following application note.

- RX63N Group, RX631 Group Initial Setting Rev. 1.10 (R01AN1245EJ)

The initial setting functions in the reference application note are used in the sample code in this application note. The revision number of the reference application note is current as of when this application note was made. However the latest version is always recommended. Visit the Renesas Electronics Corporation website to check and download the latest version.

4. Block Transfer Mode and Chain Transfer

This section explains block transfer mode and chain transfer.

4.1 Block Transfer Mode

In block transfer mode, a single block (transfer size × block size) of data can be transferred with a single activation source. The transfer source or transfer destination is specified for the block area.

The following describes registers associated with block transfer mode.

MRA.SZ bit and CRAH register

The MRA.SZ bit and the CRAH register are used to set the transfer size and block size. Any of 8, 16, or 32 bits can be set to the MRA.SZ bit as the transfer size. Any of a size from 1 to 256 can be set to the CRAH register as the block size.

Figure 4.1 shows the Relation Between the MRA.SZ bit and the CRAH Register in Block Transfer Mode.

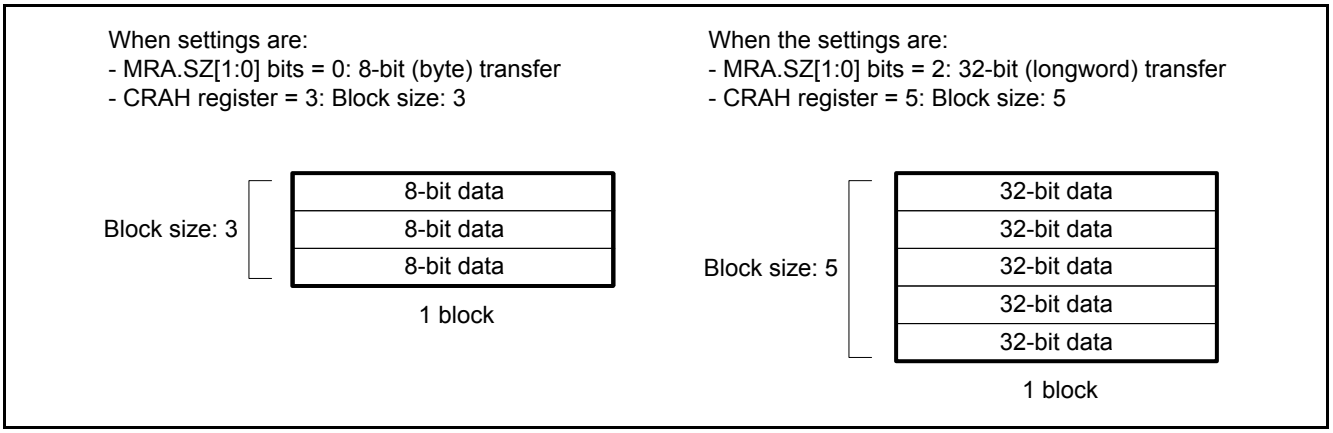


Figure 4.1 Relation Between the MRA.SZ bit and the CRAH Register in Block Transfer Mode

MRB.DTS bit

The MRB.DTS bit specifies either the transfer source or transfer destination for the block area. On completion of a single block transfer, when the transfer source address is specified for the block area, the SAR register is initialized, and when the transfer destination address is specified for the block area, the DAR register is initialized.

Figure 4.2 shows the Single Block Transfer.

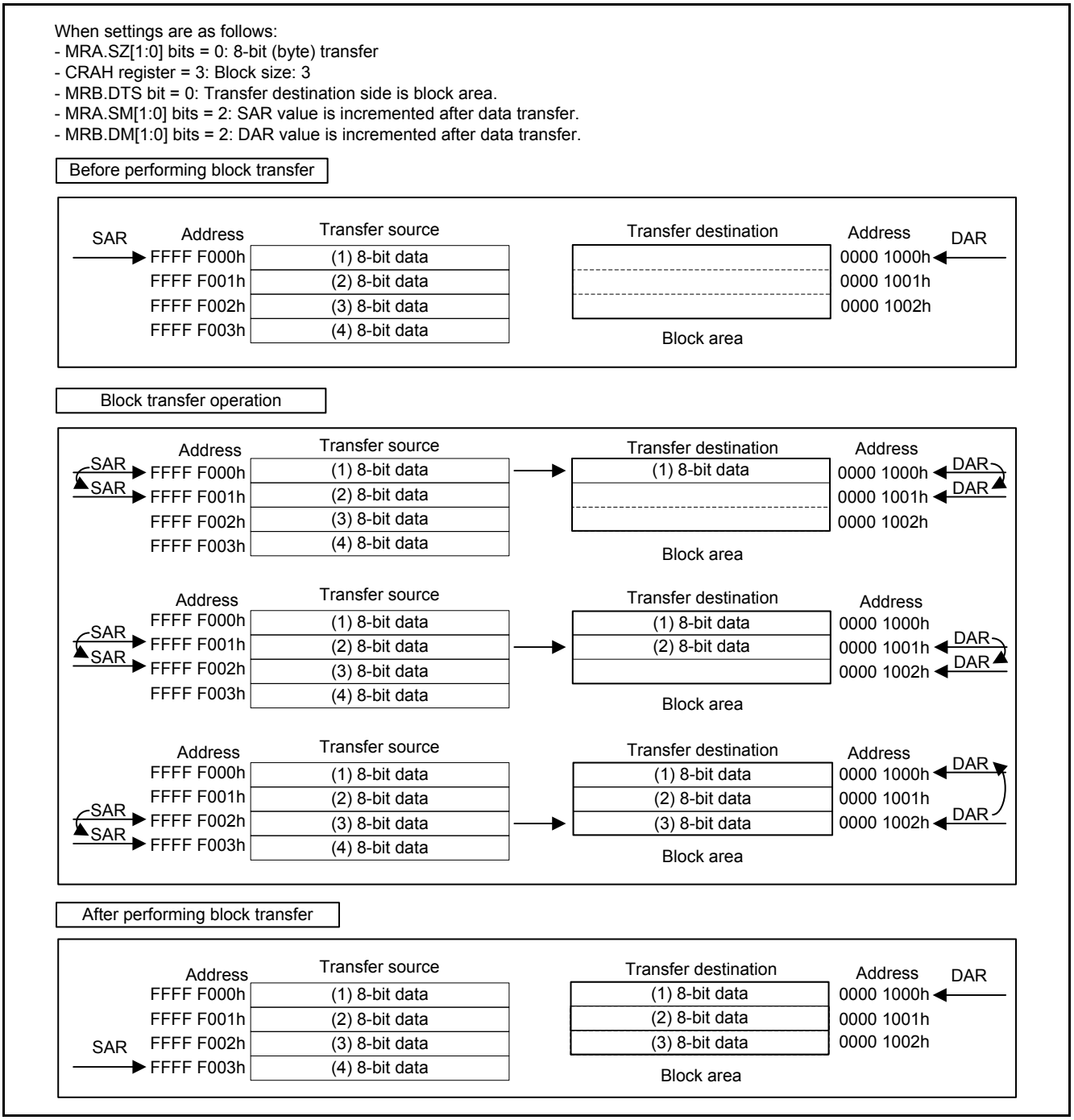


Figure 4.2 Single Block Transfer

4.2 Chain Transfer

Chain transfer is the function to transfer data based on the multiple transfer information with single activation source. Chain transfer is available all in normal transfer mode, block transfer mode, and repeat transfer mode.

Registers associated with chain transfer are described on the next page.

MRB.CHNS bit

The MRB.CHNS bit determines the condition for chain transfer execution when the MRB.CHNE bit is set to 1 (chain transfer is enabled). When the MRB.CHNS bit is 0, chain transfer is executed without any conditions. When the MRB.CHNS bit is 1, the condition for chain transfer execution differs depending on transfer modes.

Figure 4.3 shows the Chain Transfer in Normal Transfer Mode.

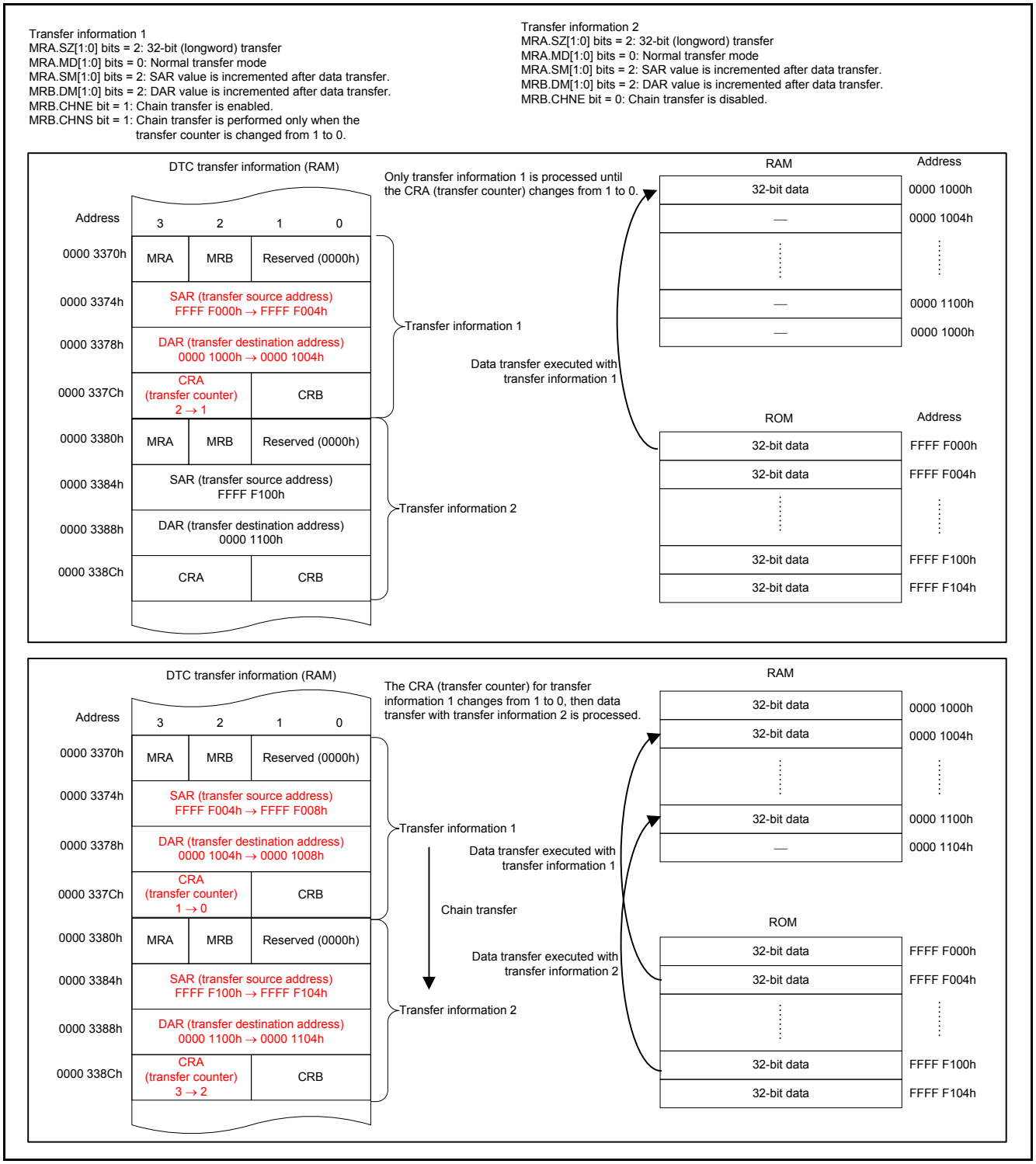


Figure 4.3 Chain Transfer in Normal Transfer Mode

5. Hardware

5.1 Hardware Configuration

Figure 5.1 shows the Connection Example.

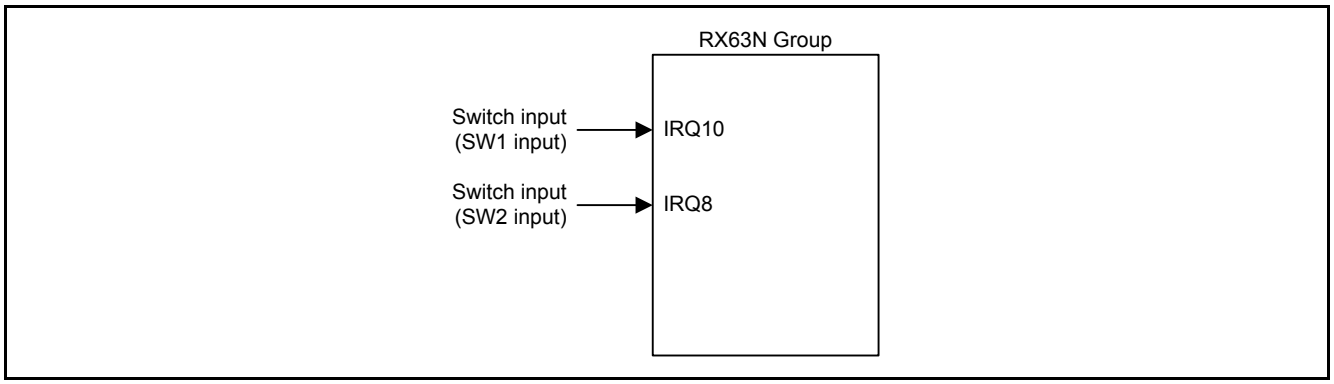


Figure 5.1 Connection Example

5.2 Pins Used

Table 5.1 lists the Pins Used and Their Functions.

Table 5.1 Pins Used and Their Functions

Pin Name	I/O	Function
P02/IRQ10	Input	Switch input as the DTC activation source for block transfer
P00/IRQ8	Input	Switch input as the DTC activation source for chain transfer

6. Software

When the switch (SW) is pressed, DTC transfer starts. When DTC transfer is completed for the specified number of times, the IRQ pin interrupt request is generated.

In the interrupt handling, DTC transfer information is reset to perform DTC transfer again.

When pushing SW1, DTC transfer is performed in block transfer mode. When data transfer is performed five times, the IRQ10 pin interrupt handling is processed.

When pushing SW2, DTC transfer is performed in normal transfer mode with two different settings using chain transfer. When data transfer is completed for the specified number of times, the IRQ8 pin interrupt handling is processed.

Peripheral function settings are shown below.

DTC

DTC setting for block transfer mode:

- Activation source: IRQ10 pin interrupt request
- DTC addressing mode: Full-address mode
- Transfer mode: Block transfer mode
- Single block size: 32 bits (longword) × 3
- Transfer source addressing mode: SAR value is incremented after data transfer
- Transfer source address: Start address of dtc_block_sar (ROM)
- Transfer destination addressing mode: DAR value is incremented after data transfer
- Transfer destination address: Start address of dtc_block_dar (RAM)
- Data transfer: Transferred in 32 bits (longword)
- Number of transfers: 5 times
- Chain transfer: Disabled
- Interrupt: An interrupt request to the CPU is generated when specified data transfer is completed.

DTC setting for the first transfer when using chain transfer:

- Activation source: IRQ8 pin interrupt request
- DTC addressing mode: Full-address mode
- Transfer mode: Normal transfer mode
- Transfer source addressing mode: SAR value is incremented after data transfer
- Transfer source address: Start address of dtc_chain0_sar (ROM)
- Transfer destination addressing mode: DAR value is incremented after data transfer
- Transfer destination address: Start address of dtc_chain0_dar (RAM)
- Data transfer: Transferred in 32 bits (longword)
- Number of transfers: 3 times
- Chain transfer: Enabled
- Chain transfer select: Chain transfer is performed continuously.

DTC setting for the second transfer when using chain transfer:

- DTC addressing mode: Full-address mode
- Transfer mode: Normal transfer mode
- Transfer source addressing mode: SAR value is incremented after data transfer
- Transfer source address: Start address of dtc_chain1_sar (ROM)
- Transfer destination addressing mode: DAR value is incremented after data transfer
- Transfer destination address: Start address of dtc_chain1_dar (RAM)
- Data transfer: Transferred in 32 bits (longword)
- Number of transfers: 3 times
- Chain transfer: Disabled
- Interrupt: An interrupt request to the CPU is generated when specified data transfer is completed.

IRQ10

- Detection method: Falling edge
- Digital filter: Enabled (PCLK/64)
- Interrupt: Used (Activation source for the DTC)

IRQ8

- Detection method: Falling edge
- Digital filter: Enabled (PCLK/64)
- Interrupt: Used (Activation source for the DTC)

6.1 Setting the DTC

6.1.1 DTC Initialization

(1) Allocating the transfer information

Registers MRA, MRB, SAR, DAR, and CRB are the DTC internal registers and they cannot be accessed from the CPU directly. Setting values in these internal registers are allocated in the RAM area as transfer information. Figure 6.1 shows the allocation of transfer information with little endian.

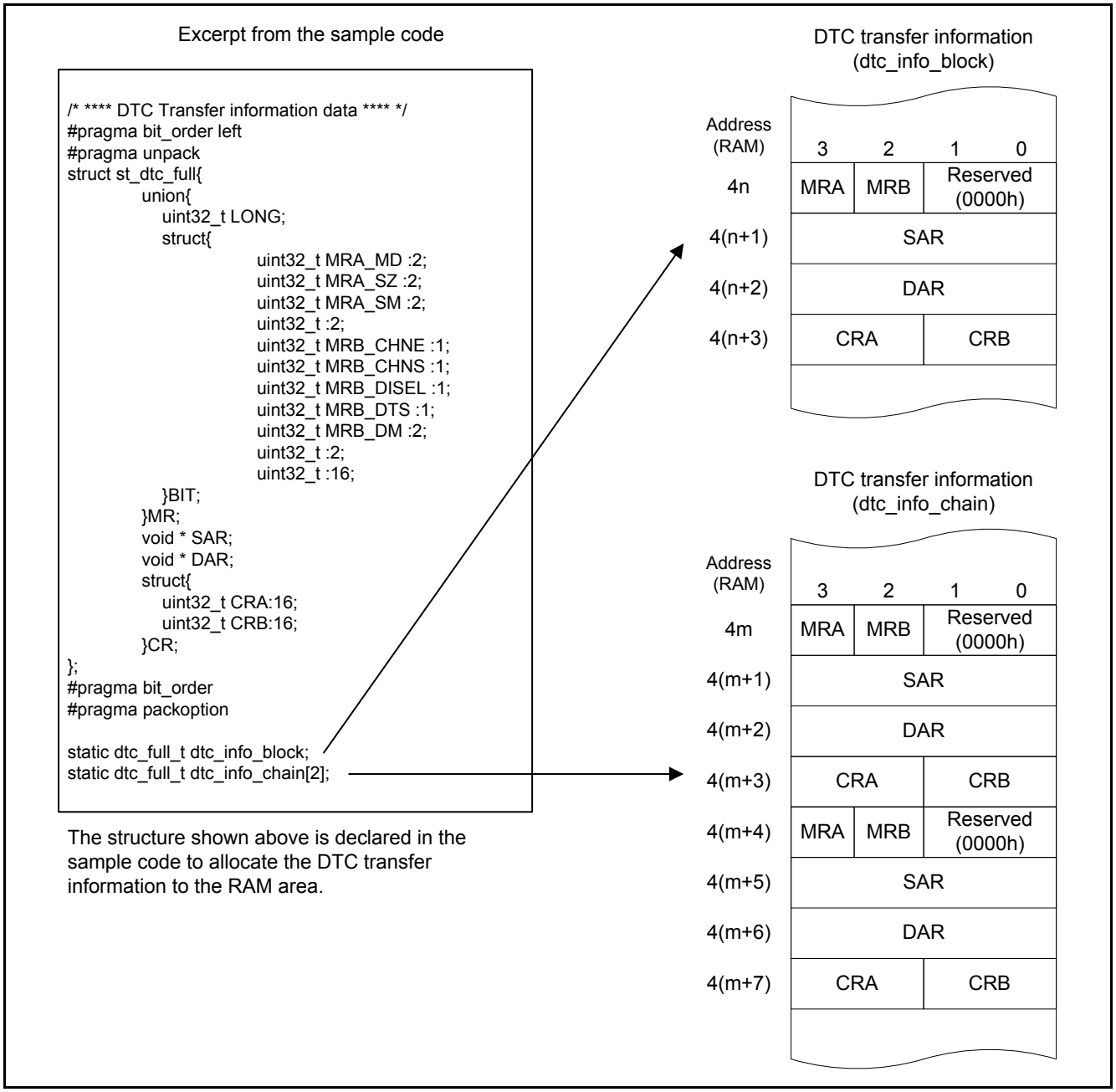


Figure 6.1 Allocating the Transfer Information

(2) Adding a section

The DTC reads the specified transfer information from the DTC vector table following the activation source. The start address of the transfer information is set to the DTC vector table.

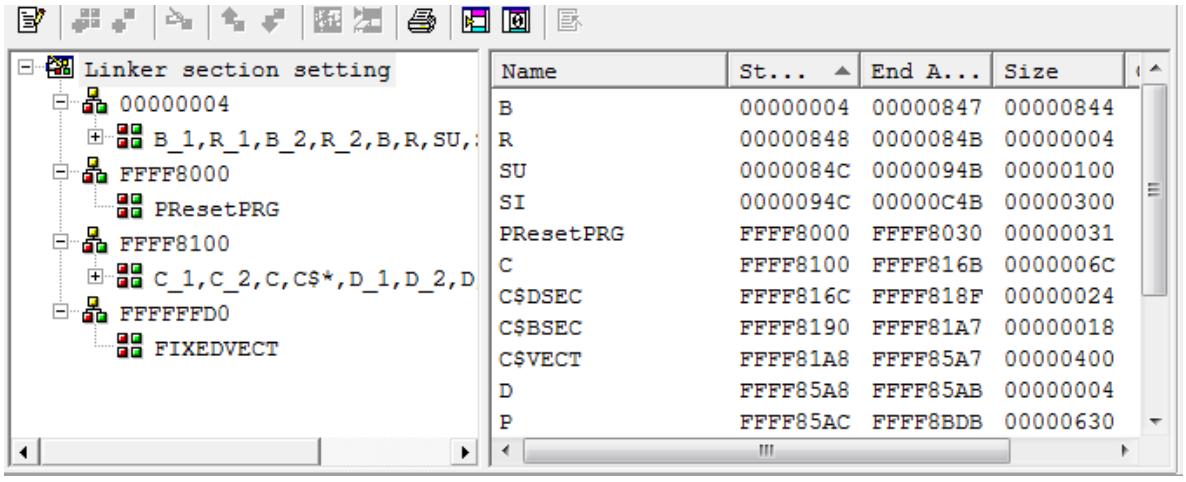
In the sample code, a section (DTC_SECTION) is added to allocate the information in the start address (3000h) of the DTC vector table (RAM). A section can be added by setting in the Map Section Information window of the integrated development environment; High-performance Embedded Workshop (HEW). Refer to the latest version of the High-performance Embedded Workshop User's Manual for details on adding, editing, and deleting a section.

Adding a section in the HEW

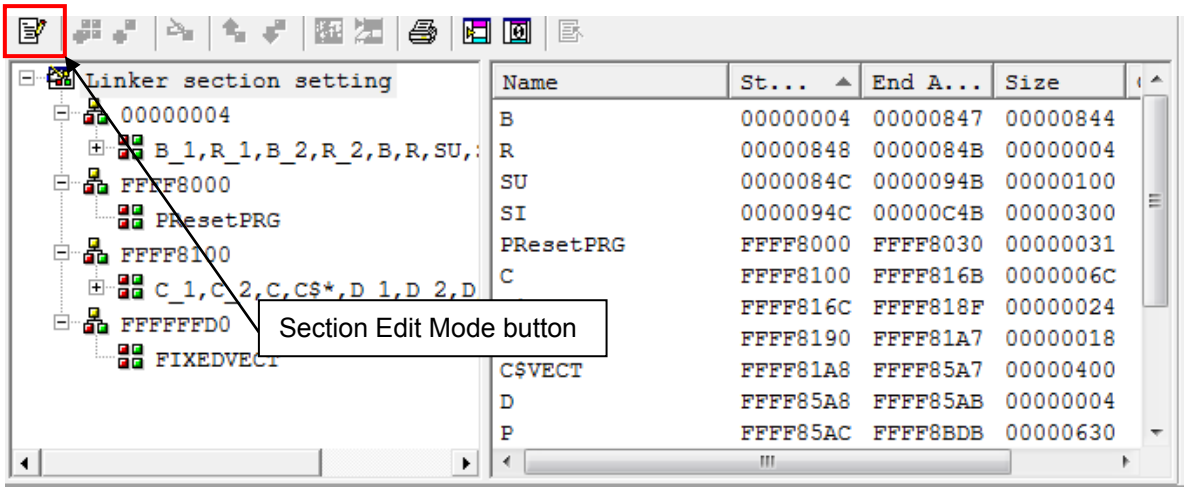
Open the HEW Map Section Information window following the procedure below.

1. Select [View -> Map] from the menu bar.
2. Select "Map Section Information" in the Map drop-down list.
3. Click OK.

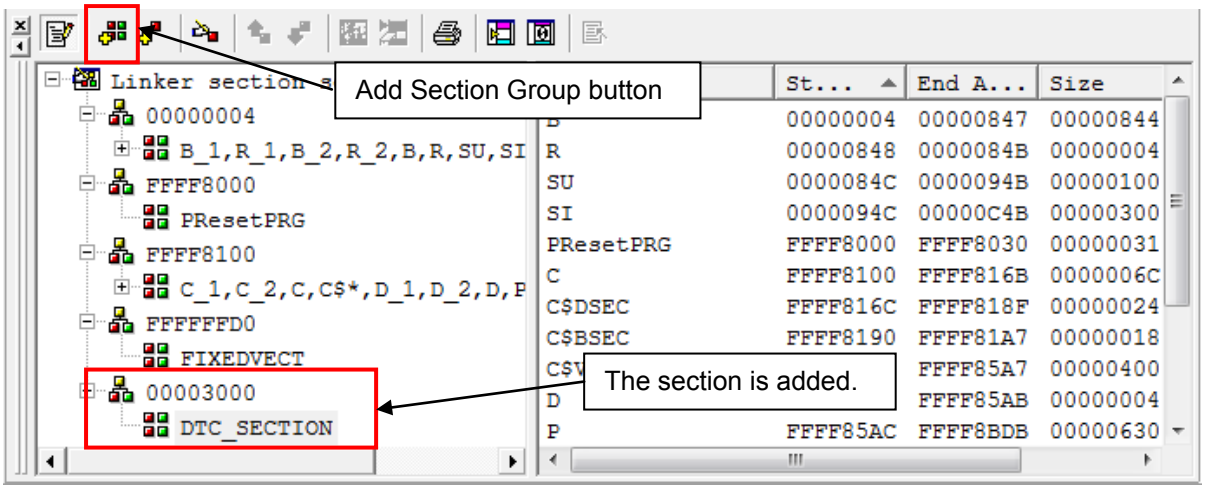
- Map Section Information window



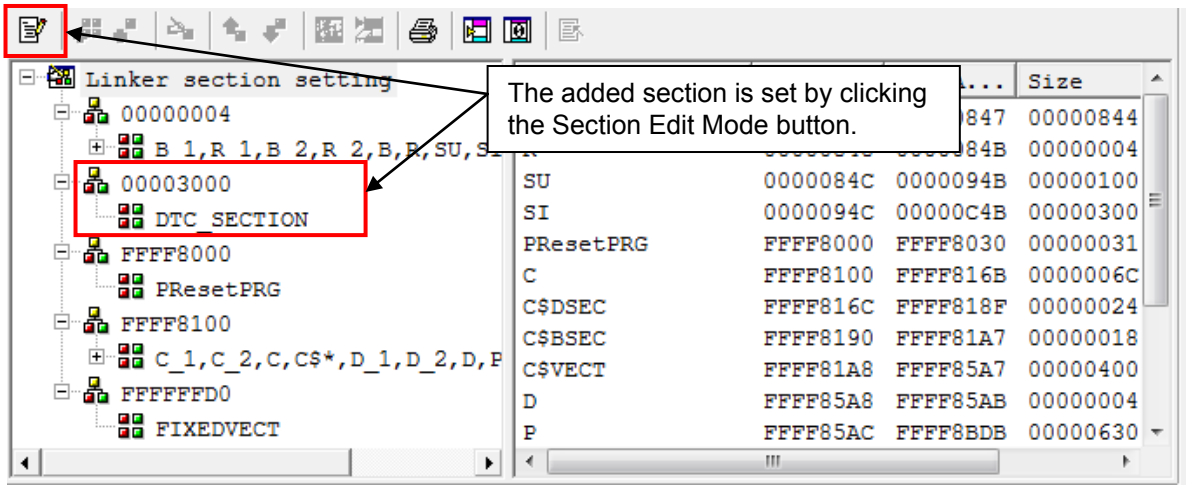
- Click the Section Edit Mode button from the tool bar.



- Click the Add Section Group button from the tool bar and set the start address and group name for the section added. In the sample code, the start address is set to 3000h and the group name is set to DTC_SECTION.



- Click the Section Edit Mode button from the tool bar to fix the edited information of the linker section.



(3) Allocating the DTC vector table

The DTC vector table is allocated to the added section “DTC_SECTION”. Figure 6.2 shows the sample code to allocate the DTC vector table and the range of the DTC vector table. Refer to the latest version of the RX Family C/C++ Compiler Package User’s Manual for details on sections.

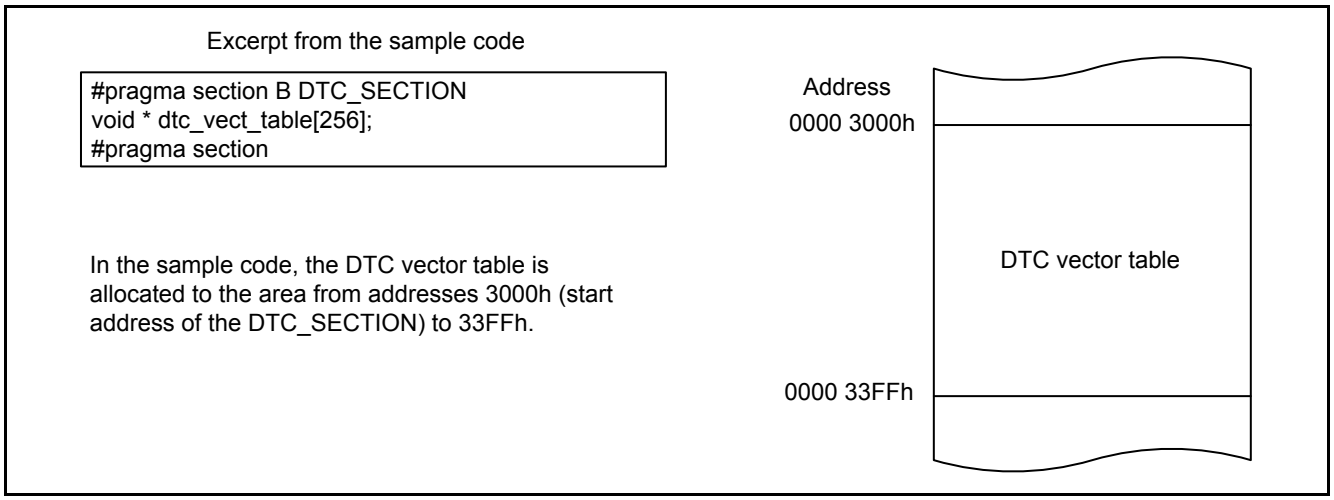
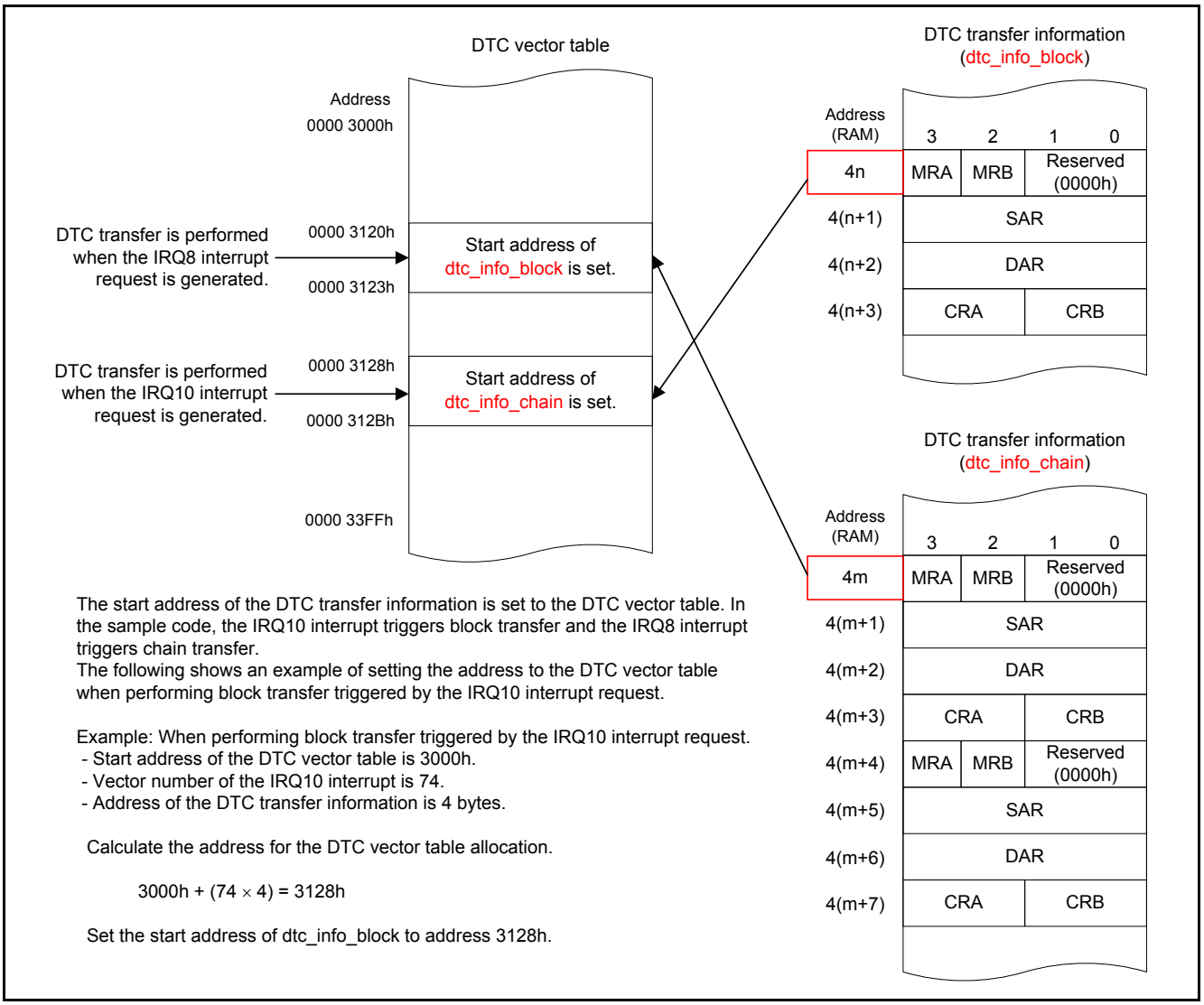


Figure 6.2 Allocating the DTC Vector Table

(4) Setting the start address for the DTC transfer information to the DTC vector table

The start address of the transfer information is set to the DTC vector table. In the sample code, the DTC transfer information is specified to perform DTC transfer when the IRQ10 and IRQ8 interrupt requests are generated. Figure 6.3 shows the Setting the DTC Transfer Information to the DTC Vector Table.



The start address of the DTC transfer information is set to the DTC vector table. In the sample code, the IRQ10 interrupt triggers block transfer and the IRQ8 interrupt triggers chain transfer.
The following shows an example of setting the address to the DTC vector table when performing block transfer triggered by the IRQ10 interrupt request.

Example: When performing block transfer triggered by the IRQ10 interrupt request.
- Start address of the DTC vector table is 3000h.
- Vector number of the IRQ10 interrupt is 74.
- Address of the DTC transfer information is 4 bytes.

Calculate the address for the DTC vector table allocation.

3000h + (74 × 4) = 3128h

Set the start address of dtc_info_block to address 3128h.

Figure 6.3 Setting the DTC Transfer Information to the DTC Vector Table

6.1.2 Resetting the DTC Related Registers

When performing DTC transfer again, some registers need to be reset after transfer is completed for the specified number of times. Table 6.1 lists the Reset Requirement of the DTC Related Registers and Table 6.2 lists the Values in SAR and DAR After Data Transfer is Performed for Specified Number of Times.

Table 6.1 Reset Requirement of the DTC Related Registers

Register Name	Reset	Description	
DTCERn (n = interrupt vector number)	Mandatory	When data transfer is completed for the specified number of times (in the last chain transfer with chain transfer), the DTCE bit becomes 0 (DTC activation is disabled). Thus reset is required.	
MRA	Optional	Reset is not required when data is transferred with the same setting as the previous one.	
MRB	Optional	Reset is not required when data is transferred with the same setting as the previous one.	
SAR, DAR	Optional	Reset requirement depends on the transfer mode and setting values in bits MRA.SM[1:0] and MRB.DM[1:0]. Refer to Table 6.2 for details.	
CRA	Depends on the transfer mode	Normal transfer mode	Reset is required when data transfer is completed for the specified number of times.
		Repeat transfer mode	Reset is not required except when the number of transfers is changed.
		Block transfer mode	Reset is not required except when the block size is changed.
CRB	Depends on the transfer mode	This register is only used in block transfer mode. Reset is required when data transfer is completed for the specified number of times.	
DTCVBR	Not required	—	
DTCST	Not required	—	
DTCADMOD	Not required	—	

Table 6.2 Values in SAR and DAR After Data Transfer is Performed for Specified Number of Times

Setting in MRA.MD[1:0]	Setting in MRA.SM[1:0]	Setting in MRB.DM[1:0]	Setting in MRB.DTS	Value in SAR	Value in DAR
Normal transfer mode	Address fixed	Address fixed	—	Initial value	Initial value
	Address incremented or decremented	Address incremented or decremented	—	Incremented or decremented value	Incremented or decremented value
Repeat transfer mode	Address incremented or decremented	Address incremented or decremented	SAR specified for repeat area	Initial value	Incremented or decremented value
	Address incremented or decremented	Address incremented or decremented	DAR specified for repeat area	Incremented or decremented value	Initial value
Block transfer mode	Address incremented or decremented	Address incremented or decremented	SAR specified for block area	Initial value	Incremented or decremented value
	Address incremented or decremented	Address incremented or decremented	DAR specified for block area	Incremented or decremented value	Initial value

6.2 File Composition

Table 6.3 lists the Files Used in the Sample Code, Table 6.4 lists the Standard Include Files and Table 6.5 lists the Functions and Setting Values in the Reference Application Note. Files generated by the integrated development environment are not included in this table.

Table 6.3 Files Used in the Sample Code

File Name	Outline
main.c	Main processing
r_init_stop_module.c	Stop processing for active peripheral functions after a reset
r_init_stop_module.h	Header file for r_init_stop_module.c
r_init_non_existent_port.c	Nonexistent port initialization
r_init_non_existent_port.h	Header file for r_init_non_existent_port.c
r_init_clock.c	Clock initialization
r_init_clock.h	Header file for r_init_clock.c
dtc.c	Block transfer and chain transfer
dtc.h	Header file for dtc.c

Table 6.4 Standard Include Files

File Name	Outline
stdint.h	Defines macros declaring the integer type with the specified width.
stdbool.h	Defines macros associated with Boolean and its value.
machine.h	Defines types of intrinsic function for the RX Family.

Table 6.5 Functions and Setting Values in the Reference Application Note
(RX63N Group, RX631 Group Initial Setting)

File Name	Function	Description
r_init_stop_module.c	R_INIT_StopModule()	Module-stop state is canceled for DMAC/DTC, EXDMAC, RAM0, and RAM1.
r_init_stop_module.h	—	
r_init_non_existent_port.c	R_INIT_NonExistentPort()	
r_init_non_existent_port.h	—	176-pin package is specified.
r_init_clock.c	R_INIT_Clock()	
r_init_clock.h	—	Sub-clock is not used.

6.3 Option-Setting Memory

Table 6.6 lists the Option-Setting Memory Configured in the Sample Code. When necessary, set a value suited to the user system.

Table 6.6 Option-Setting Memory Configured in the Sample Code

Symbol	Address	Setting Value	Contents
OFS0	FFFF FF8Fh to FFFF FF8Ch	FFFF FFFFh	The IWDT is stopped after a reset. The WDT is stopped after a reset.
OFS1	FFFF FF8Bh to FFFF FF88h	FFFF FFFFh	The voltage monitor 0 reset is disabled after a reset. HOCO oscillation is disabled after a reset.
MDES	FFFF FF83h to FFFF FF80h	FFFF FFFFh	Little endian

6.4 Structure/Union List

Figure 6.4 shows the Structure/Union Used in the Sample Code.

```
/* **** DTC Transfer information data **** */
#pragma bit_order     left          /* Bit field order: The bit field members are allocated from upper bits */
#pragma unpack                      /* The boundary alignment value for structure members: Alignment by member type */
typedef struct
{
    union
    {
        uint32_t LONG;
        struct
        {
            uint32_t MRA_MD         :2;
            uint32_t MRA_SZ         :2;
            uint32_t MRA_SM         :2;
            uint32_t                :2;
            uint32_t MRB_CHNE       :1;
            uint32_t MRB_CHNS       :1;
            uint32_t MRB_DISEL      :1;
            uint32_t MRB_DTS        :1;
            uint32_t MRB_DM         :2;
            uint32_t                :2;
            uint32_t                :16;
        } BIT;
    } MR;
    void * SAR;
    void * DAR;
    struct
    {
        uint32_t CRA:16;
        uint32_t CRB:16;
    } CR;
}dtc_full_t;
#pragma packoption                  /* End of specification for the boundary alignment value for structure members */
#pragma bit_order                   /* End of specification for the bit field order */
```

Figure 6.4 Structure/Union Used in the Sample Code

6.5 Variables

Table 6.7 lists the static Variables, and Table 6.8 lists the const Variables.

Table 6.7 static Variables

Type	Variable Name	Contents	Function
static dtc_full_t	dtc_info_block	DTC transfer information for block transfer	DTC_Init Excep_ICU_IRQ10
static dtc_full_t	dtc_info_chain[2]	DTC transfer information for chain transfer	DTC_Init Excep_ICU_IRQ8
static uint32_t	dtc_block_dar	Transfer destination area for dtc_info_block	DTC_Init
static uint32_t	dtc_chain0_dar	Transfer destination area for dtc_info_chain[0]	DTC_Init
static uint32_t	dtc_chain1_dar	Transfer destination area for dtc_info_chain[1]	DTC_Init
static void*	dtc_vect_table[256]	DTC vector table	DTC_Init

Table 6.8 const Variables

Type	Variable Name	Contents	Function
static const uint32_t	dtc_block_sar[15]	Transfer source area for dtc_info_block	DTC_Init Excep_ICU_IRQ10
static const uint32_t	dtc_chain0_sar[3]	Transfer source area for dtc_info_chain[0]	DTC_Init Excep_ICU_IRQ8
static const uint32_t	dtc_chain1_sar[3]	Transfer source area for dtc_info_chain[1]	DTC_Init Excep_ICU_IRQ8

6.6 Functions

Table 6.9 lists the Functions.

Table 6.9 Functions

Function Name	Outline	File
main	Main processing	main.c
port_init	Port initialization	main.c
R_INIT_StopModule	Stop processing for active peripheral functions after a reset	r_init_stop_module.c
R_INIT_NonExistentPort	Nonexistent port initialization	r_init_non_existent_port.c
R_INIT_Clock	Clock initialization	r_init_clock.c
peripheral_init	Peripheral function initialization	main.c
IRQ_Init	IRQ initialization	main.c
DTC_Init	DTC initialization	dtc.c
Excep_ICU_IRQ10	IRQ10 interrupt handling	dtc.c
Excep_ICU_IRQ8	IRQ8 interrupt handling	dtc.c

6.7 Function Specifications

The following tables list the sample code function specifications.

main	
Outline	Main processing
Header	None
Declaration	void main(void)
Description	Waits for SW1 (IRQ10) and SW2 (IRQ8) input.
Arguments	None
Return Value	None

port_init	
Outline	Port initialization
Header	None
Declaration	void port_init(void)
Description	Initializes ports.
Arguments	None
Return Value	None

R_INIT_StopModule	
Outline	Stop processing for active peripheral functions after a reset
Header	r_init_stop_module.h
Declaration	void R_INIT_StopModule(void)
Description	Configures the setting to enter the module-stop state.
Arguments	None
Return Value	None
Remarks	Transition to the module-stop state is not performed in the sample code. Refer to the RX63N Group, RX631 Group Initial Setting Rev. 1.10 application note for details on this function.

R_INIT_NonExistentPort	
Outline	Nonexistent port initialization
Header	r_init_non_existent_port.h
Declaration	void R_INIT_NonExistentPort(void)
Description	Initializes port direction registers for ports that do not exist in products with less than 176 pins.
Arguments	None
Return Value	None
Remarks	The number of pins in the sample code is set for the 176-pin package (PIN_SIZE=176). After this function is called, when writing in byte units to the PDR registers or PODR registers which have nonexistent ports, set the corresponding bits for nonexistent ports as follows: set the I/O select bits in the PDR registers to 1 and set the output data store bits in the PODR registers to 0. Refer to the RX63N Group, RX631 Group Initial Setting Rev. 1.10 application note for details on this function.

R_INIT_Clock	
Outline	Clock initialization
Header	r_init_clock.h
Declaration	void R_INIT_Clock(void)
Description	Initializes the clock.
Arguments	None
Return Value	None
Remarks	The sample code selects processing which uses PLL as the system clock without using the sub-clock. Refer to the RX63N Group, RX631 Group Initial Setting Rev. 1.10 application note for details on this function.

peripheral_init	
Outline	Peripheral function initialization
Header	None
Declaration	void peripheral_init(void)
Description	Initializes the peripheral functions used.
Arguments	None
Return Value	None

IRQ_Init	
Outline	IRQ initialization
Header	None
Declaration	static void IRQ_Init(void)
Description	Initializes IRQ10 and IRQ8.
Arguments	None
Return Value	None

dtc_init	
Outline	DTC initialization
Header	None
Declaration	static void dtc_init(void)
Description	Initializes the DTC.
Arguments	None
Return Value	None

Excep_ICU_IRQ10	
Outline	IRQ10 interrupt handling
Header	None
Declaration	static void Excep_ICU_IRQ10 (void)
Description	Resets the registers to perform DTC transfer again in block transfer mode.
Arguments	None
Return Value	None

Excep_ICU_IRQ8	
Outline	IRQ8 interrupt handling
Header	None
Declaration	static void Excep_ICU_IRQ8 (void)
Description	Resets the registers to perform DTC transfer again with chain transfer.
Arguments	None
Return Value	None

6.8 Flowcharts

6.8.1 Main Processing

Figure 6.5 shows the Main Processing.

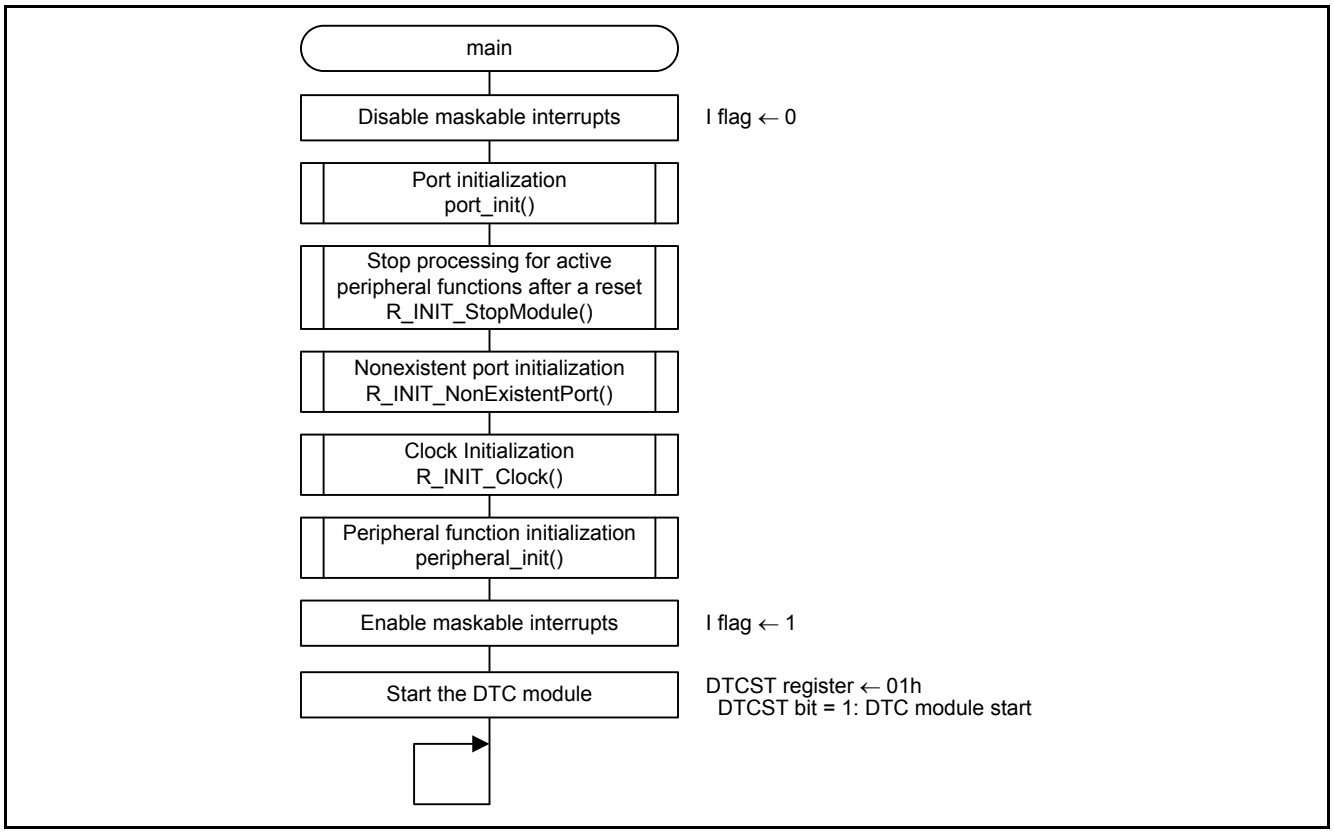


Figure 6.5 Main Processing

6.8.2 Port Initialization

Figure 6.6 shows the Port Initialization.

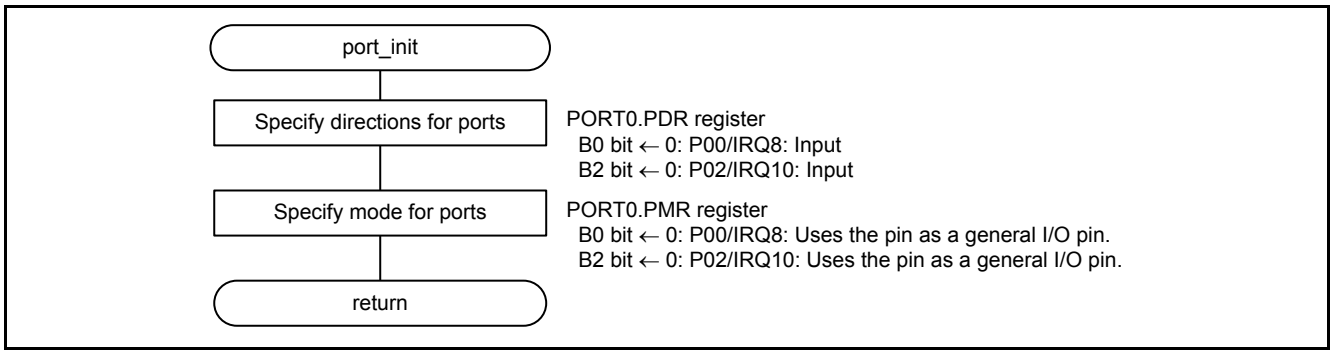


Figure 6.6 Port Initialization

6.8.3 Peripheral Function Initialization

Figure 6.7 shows the Peripheral Function Initialization.

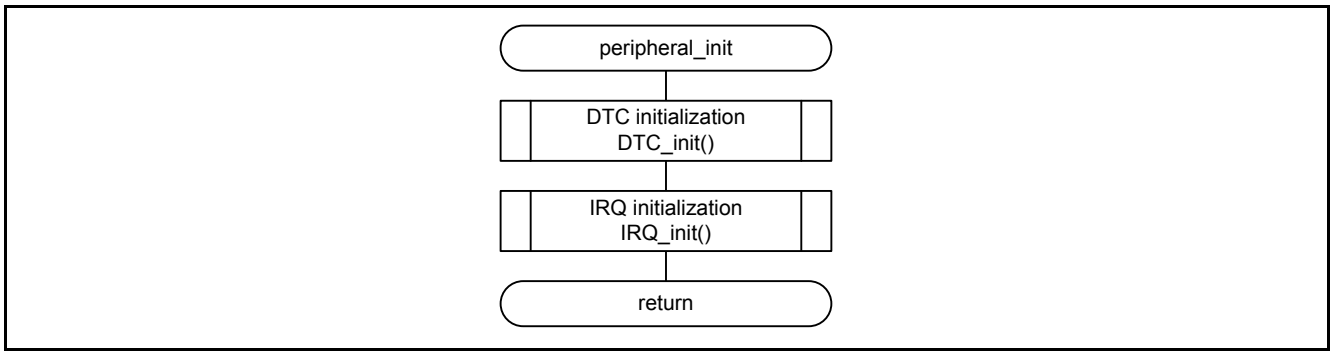


Figure 6.7 Peripheral Function Initialization

6.8.4 IRQ Initialization

Figure 6.8 and Figure 6.9 show the IRQ Initialization.

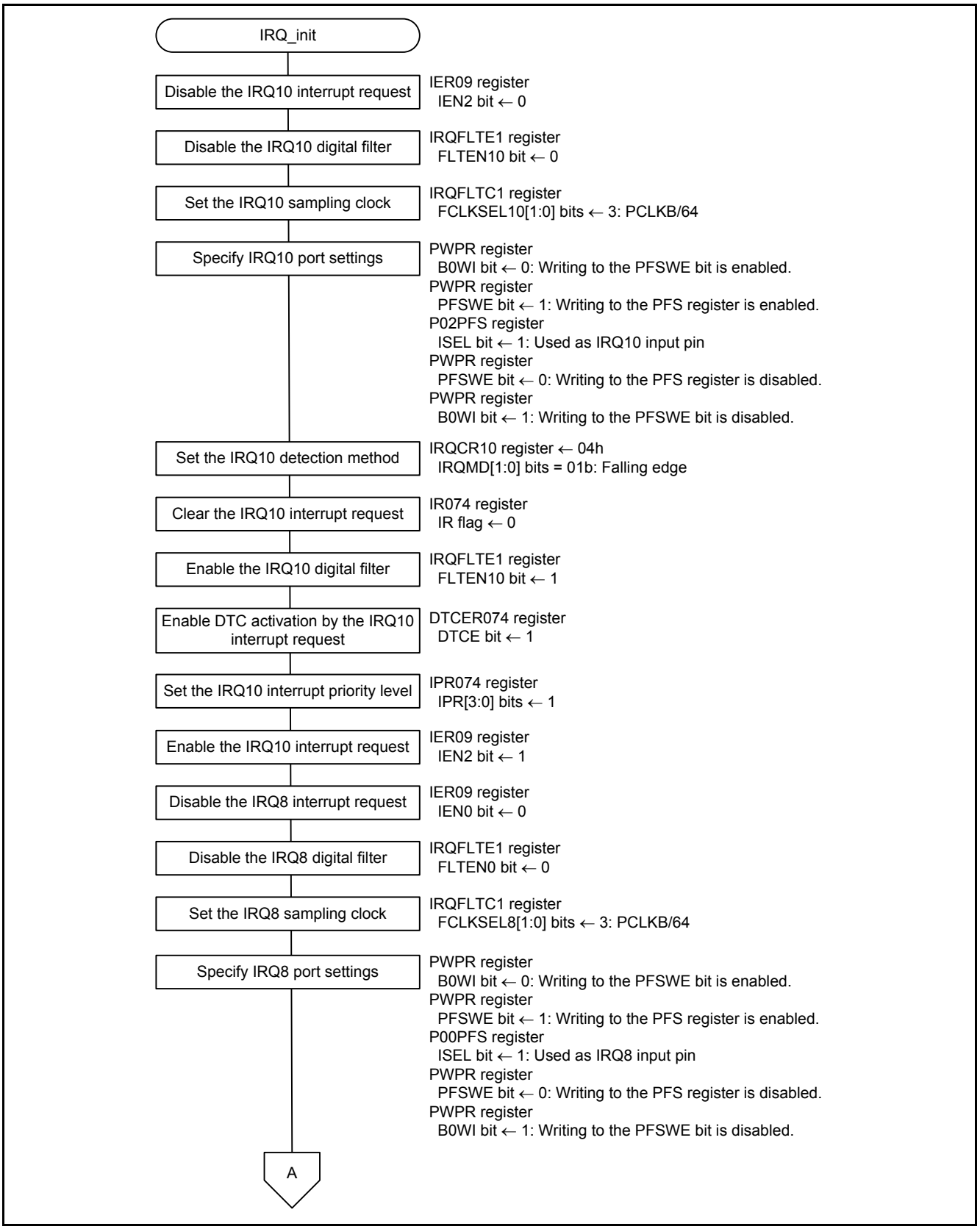


Figure 6.8 IRQ Initialization (1/2)

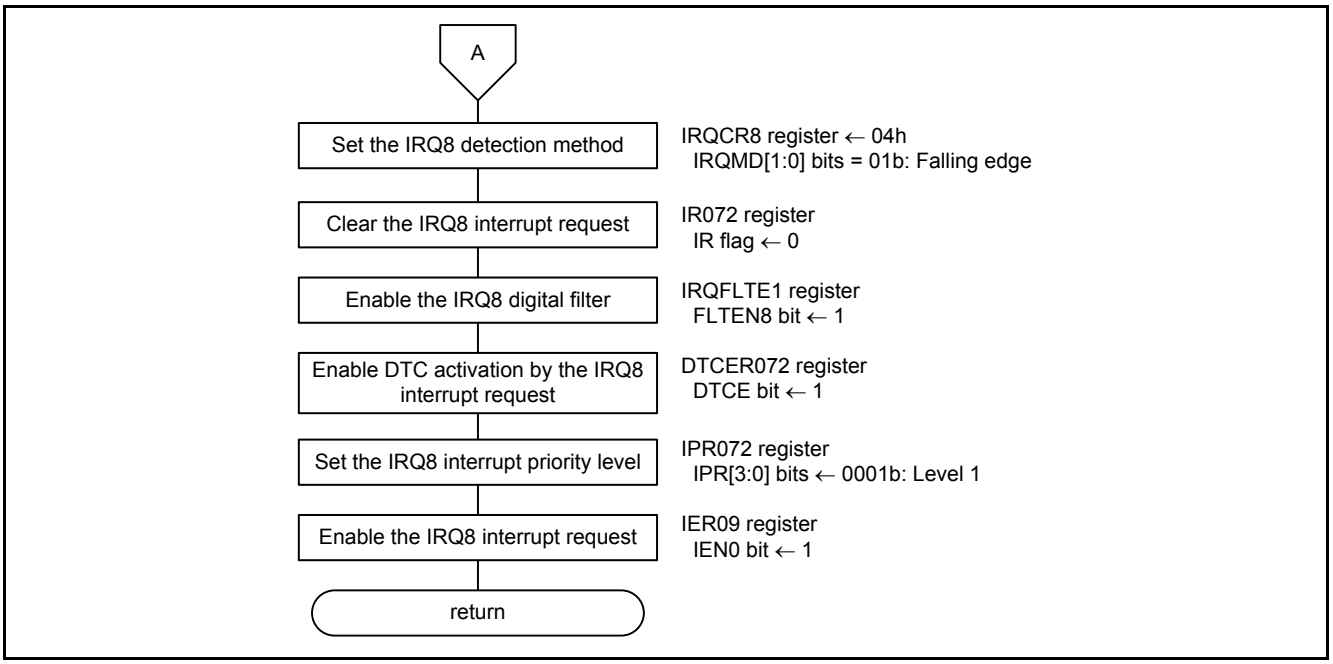


Figure 6.9 IRQ Initialization (2/2)

6.8.5 DTC Initialization

Figure 6.10 shows the DTC Initialization.

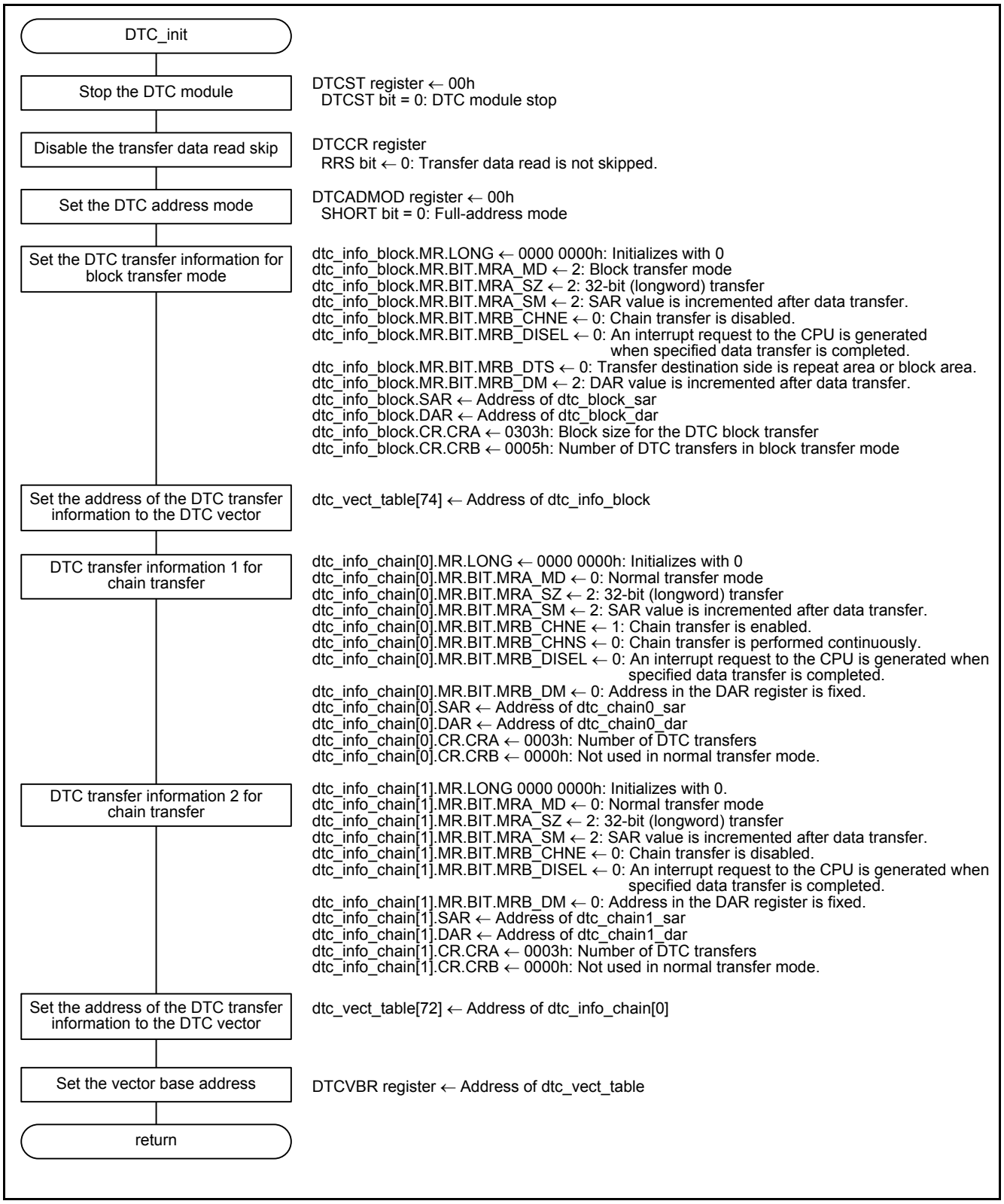


Figure 6.10 DTC Initialization

6.8.6 IRQ10 Interrupt Handling

Figure 6.11 shows the IRQ10 Interrupt Handling.

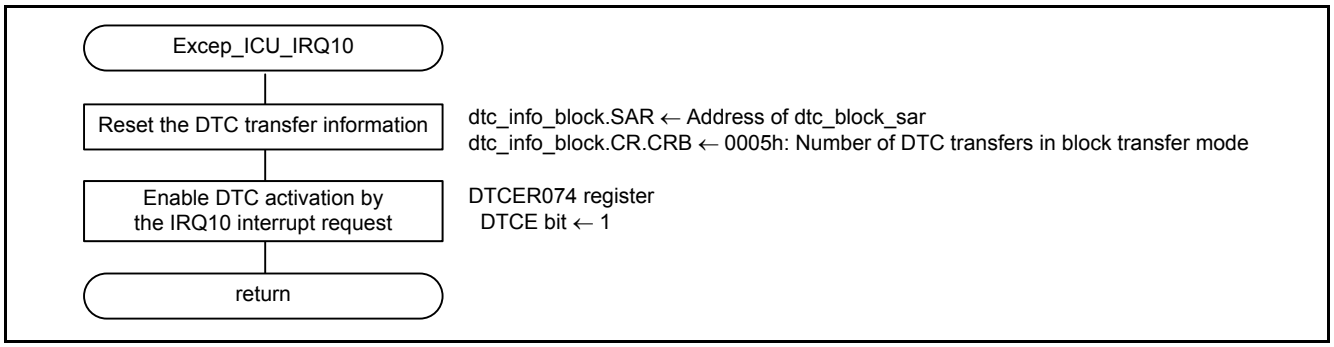


Figure 6.11 IRQ10 Interrupt Handling

6.8.7 IRQ8 Interrupt Handling

Figure 6.12 shows the IRQ8 Interrupt Handling.

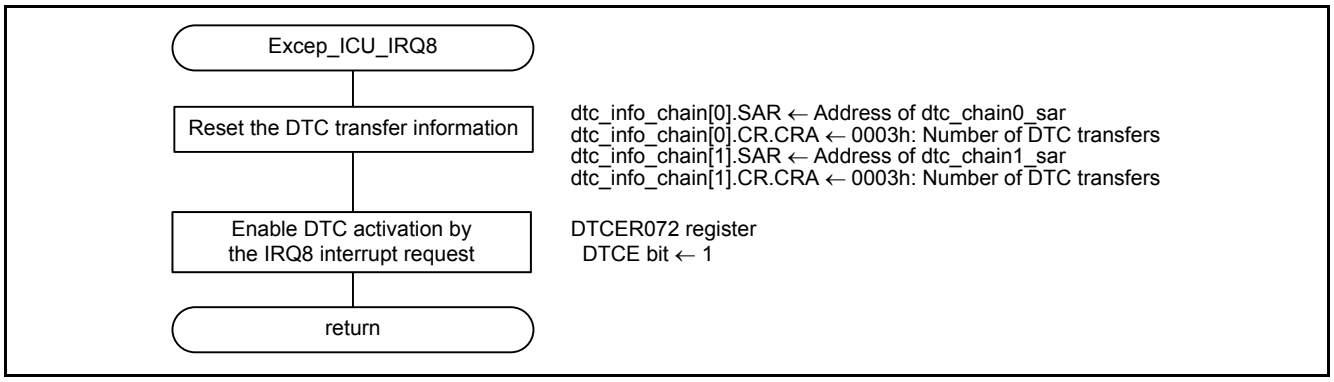


Figure 6.12 IRQ8 Interrupt Handling

7. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

8. Reference Documents

User's Manual: Hardware

RX Family User's Manual: Hardware Rev.1.80 (R01UH0041EJ)
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.

User's Manual: Development Tools

RX Family C/C++ Compiler Package V.1.01 User's Manual Rev.1.00 (R20UT0570EJ)
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/contact/

REVISION HISTORY	RX Family Application Note Setting Example of DTC Block Transfer Mode and Chain Transfer

Rev.	Date	Description	
		Page	Summary
1.00	Aug. 1, 2014	—	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 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
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: +49-211-6503-0, 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 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022/9044

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 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.
12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 4.0