1. Abstract
This application note describes the setting procedure to perform a multiple transfer when using DMAC II.

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. Operation Overview
Multiple memory-to-memory transfers (max. seven times) are performed from different source addresses to different destination addresses by one transfer request.
When the multiple transfer is selected, the following transfer functions are not available: the calculation transfer, burst transfer, chained transfer, and DMA II transfer complete interrupt.
Table 3.1 lists Specifications of DMAC II and Settings. Table 3.2 lists DMAC II Specifications.

### Table 3.1 Specifications of DMAC II and Settings

<table>
<thead>
<tr>
<th>Item</th>
<th>Selectable Function</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer sizes</td>
<td>8 bits or 16 bits</td>
<td>8 bits</td>
</tr>
<tr>
<td>Transfer types</td>
<td>Memory → Memory</td>
<td>Memory → Memory</td>
</tr>
<tr>
<td>Source addressing</td>
<td>Fixed Increment</td>
<td>Increment</td>
</tr>
<tr>
<td>Destination addressing</td>
<td>Fixed Increment</td>
<td>Increment</td>
</tr>
<tr>
<td>Calculation transfer</td>
<td>Not used</td>
<td>Not used</td>
</tr>
<tr>
<td>Burst transfer</td>
<td>Not used</td>
<td>Not used</td>
</tr>
<tr>
<td>Transfer complete interrupt</td>
<td>Not used</td>
<td>Not used</td>
</tr>
<tr>
<td>Chained transfer</td>
<td>Not used</td>
<td>Not used</td>
</tr>
</tbody>
</table>

### Table 3.2 DMAC II Specifications

<table>
<thead>
<tr>
<th>Item</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of transfers</td>
<td>2</td>
</tr>
<tr>
<td>Number of multiple transfers</td>
<td>2</td>
</tr>
<tr>
<td>Transfer data 1</td>
<td>01h, 03h</td>
</tr>
<tr>
<td>Transfer data 2</td>
<td>7Fh, FFh</td>
</tr>
<tr>
<td>Trigger for DMAC II</td>
<td>INT0 interrupt</td>
</tr>
</tbody>
</table>
### 3.1 Operation Example

This operation example explains DMA II transfer based on the settings in this application note. When an INT0 interrupt request is generated, multiple memory-to-memory transfers are performed in 8-bit data from different source addresses to different destination addresses with one transfer request. Figure 3.1 shows the DMAC II Multiple Transfer.

Numbers (1) and (2) in the parenthesis in the figure explain:

1. When an INT0 interrupt request is generated, data from different source addresses are transferred to their respective destination addresses. After the transfer is completed, the addresses in different source addresses and destination addresses are incremented by 1 and the counter value of the transfer counter is decremented by 1.

2. When the next INT0 interrupt is generated, the same process as described above in (1) is carried out. Then, the counter value of the transfer counter becomes 0, indicating the multiple transfer is completed.

![Figure 3.1 DMAC II Multiple Transfer](image)

<table>
<thead>
<tr>
<th>(1)</th>
<th>(2)</th>
<th>DMA II multiple transfer is completed.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer mode</td>
<td>802Eh</td>
<td>Transfer mode</td>
</tr>
<tr>
<td>Transfer counter</td>
<td>0002h</td>
<td>Transfer counter</td>
</tr>
<tr>
<td>Source address 1</td>
<td>00000500h</td>
<td>Source address 1</td>
</tr>
<tr>
<td>Destination address 1</td>
<td>00000510h</td>
<td>Destination address 1</td>
</tr>
<tr>
<td>Source address 2</td>
<td>00000600h</td>
<td>Source address 2</td>
</tr>
<tr>
<td>Destination address 2</td>
<td>00000610h</td>
<td>Destination address 2</td>
</tr>
</tbody>
</table>

| 802Eh | 0001h | 0000h |
| 01h | 0000500h | 01h |
| 03h | 0000501h | 03h |
| 01h | 0000510h | 01h |
| 03h | 0000511h | 03h |
| 7Fh | 0000600h | 7Fh |
| FFh | 0000601h | FFh |
| 7Fh | 0000610h | 7Fh |
| FFh | 0000611h | FFh |

DMA II multiple transfer is completed.
4. Application Example

This application note explains settings for multiple memory-to-memory transfers using DMAC II.

4.1 Setting for DMAC II

When using DMAC II, set the following:
- Registers RIPL1 and RIPL2
- DMAC II index
- The interrupt control register of the peripheral function triggering DMAC II
- The relocatable vector of the peripheral function triggering DMAC II
- IRLT bit in the IIOiIE register if the intelligent I/O interrupt is used.

Refer to the user’s manual for details on the IIOiIE register (i = 0 to 11).

4.1.1 Registers RIPL1 and RIPL2

When the DMAII bits in registers RIPL1 and RIPL2 are set to 1 (DMA II transfer selected) and the FSIT bits are set to 0 (normal interrupt selected), DMAC II is triggered by an interrupt of any peripheral function with bits ILVL2 to ILVL0 in the corresponding interrupt control register set to 111b (level 7). Set the same value to registers RIPL1 and RIPL2.

Table 4.1 lists Setting Values in Registers RIPL1 and RIPL2.

<table>
<thead>
<tr>
<th>Register Name</th>
<th>Setting Value</th>
<th>Remarks</th>
</tr>
</thead>
</table>
| RIPL1 and RIPL2 | 20h | Bits RLVL2 to RLVL0 are 000b (level 0)  
FSIT bit is 0 (use interrupt request level 7 for normal interrupt)  
b4 is 0  
DMAII bit is 1 (use interrupt request level 7 for fast interrupt)  
b7, b6 are 0 |
4.1.2 DMAC II Index

The DMAC II index is a data table of 12 to 60 bytes. It stores parameters for transfer mode, transfer counter, source address, and destination address.

The DMAC II index should be allocated on the RAM.

Figure 4.1 shows the Configuration of the DMAC II Index when the multiple transfer is selected.

![Figure 4.1 Configuration of the DMAC II Index](image)

The following are the details on the DMAC II index.

- Transfer mode (MOD)
  2-byte data is required to set transfer mode.
- Transfer counter (COUNT)
  2-byte data is required to set the transfers to be performed.
- Source address (SADR) (i = 1 to 7)
  4-byte data is required to set a source address in a memory or an immediate data.
- Destination address (DADRi)
  4-byte data is required to set a destination address in a memory.

The numbers of multiple transfers in the source addresses and destination addresses should be set.
4.1.3 Interrupt Control Register of the Peripheral Function
Set bits ILVL2 to ILVL0 in the interrupt control register for the peripheral interrupt triggering DMAC II to 111b (level 7).

4.1.4 Relocatable Vector Table of the Peripheral Function
Set the starting address of the DMAC II index to the interrupt vector for the peripheral interrupt triggering DMAC II.

In this application note, the INT0 interrupt is used for the interrupt triggering DMAC II. Figure 4.2 shows the setting by asm function in the C language program. In this setting example, the DMAC II index (dm_index) is set as the relocatable vector table.

```
asm(".rvector 31, dm_index"); //Definite DMACII Index (Software Interrupt Number 31)
```

Figure 4.2 Setting Example for the Starting Address of the DMAC II Index to the Interrupt Vector
4.2 Setting Overview

Figure 4.3 shows the Setting Procedure for DMAC II Multiple Transfer. Refer to section 4.3 "Register Settings" for detailed settings of the items below.

- Set registers RIPL1 and RIPL2
- Set transfer mode (MOD)
- Set transfer counter (COUNT)
- Set source address (SADR)
- Set destination address (DADR)
- Set interrupt control register of peripheral function triggering DMAC II
- Set relocatable vector table of peripheral function
- Generate interrupt request of the peripheral function
- Start multiple transfer

Figure 4.3 Setting Procedure for DMAC II Multiple Transfer
4.3 Register Settings

Set registers RIPL1 and RIPL2.

Wake-up IPL Setting Register i (RIPLi) (i = 1, 2)

<table>
<thead>
<tr>
<th>b7</th>
<th>1</th>
<th>b0</th>
</tr>
</thead>
</table>

- RLVL2 to RLVL0: Interrupt Priority Level for Wake-up Select Bit
  - 000b: Level 0

- FSIT: Fast Interrupt Select Bit
  - 0: Use interrupt request level 7 for normal interrupt

Set to 0

- DMAII: DMA II Select Bit
  - 1: Use interrupt request level 7 for DMA II transfer

Set to 0.

Registers RIPL1 and RIPL2 should be identically set

Bits ILVL2 to ILVL0 in the interrupt control register should be set after the DMAII bit is set
The DMA II transfer is not affected by the I flag or the IPL

Continued on next page
Set the transfer mode (MOD).

- **Transfer Mode (MOD)**
  - **SIZE**
  - **IMM**
  - **UPDS**
  - **UPDD**
  - **CNT2 to CNT0**
  - **CHAIN**
  - **MULT**

**The MOD should be allocated on the RAM**

- **Transfer Size Select Bit**
  - 8 bits
  - Reserved

- **Source Addressing Select Bit**
  - Incrementing addressing

- **Destination Addressing Select Bit**
  - Incrementing addressing

- **Transfer Number Set Bit**
  - Once
  - Twice
  - Six times
  - Seven times
  - Reserved

As shown above, when the MULT bit is 0, bits b4 to b6 are OPER, BRST, and INTE, respectively. When the MULT bit is 1, bits b4 to b6 become CNT0, CNT1, and CNT2, respectively.

Set bits CNT2 to CNT0 to select the number of multiple transfers

Set the transfer counter (COUNT).

- **Transfer Counter (COUNT)**
  - When transfer counter is set to 0000h, data transfer is not performed.
  - Set the number of transfers to be performed.

Continued on next page
continued from previous page

Set the source address (SADR).

\[ \begin{array}{c}
\text{b31} \\
\text{b0}
\end{array} \]

Source Address \( i \) (SADR) \((i = 1 \text{ to } 7)\)

Set the source address

continued on next page

Set the destination address (DADR).

\[ \begin{array}{c}
\text{b31} \\
\text{b0}
\end{array} \]

Destination Address \( i \) (DADR) \((i = 1 \text{ to } 7)\)

Set the destination address

continued on next page
Notes:
1. Set the IR bit to 0.
2. This bit should be set to 0 (the falling edge or low level) to set the corresponding bit in registers IFSR0 and IFSR1 to 1 (both edges).
3. To select the level sensitive, the corresponding bit in registers IFSR0 and IFSR1 should be set to 0 (one edge).
Generate a peripheral interrupt request.
An interrupt request of the peripheral interrupt triggering DMAC II should be generated

Start multiple memory-to-memory transfers
A DMAC II multiple memory-to-memory transfer starts by receiving an interrupt request from any peripheral function.
5. Sample Program

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

5.1 Program Flowchart

The sample program is comprised of the main function.
Figure 5.1 shows the Main Function Flowchart.

```
main function (main)
  ↓
1) Disable interrupts
  ↓
2) Set port P8 input
  ↓
3) Set registers RIPL1 and RIPL2
  ↓
4) Set DMAC II index
  ↓
5) Set INT0 interrupt control register
```

Figure 5.1 Main Function Flowchart
6. 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 Series C Compiler Package V.1.02 C Compiler User’s 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
DMA II Setting Example (Multiple Transfer)

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Description</th>
<th>Page</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>May 28, 2010</td>
<td>First Edition issued</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

All trademarks and registered trademarks are the property of their respective owners.
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 solely to illustrate the operation of semiconductor products and application example. 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 product 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 manufacturers, 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 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.

8. “Standard” - Computer; office equipment; communication equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots.

9. “High Quality” - Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment.

10. “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.

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.

(Notes)

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

(2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.