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.

RENESAS

Application Note

Multimedia Processor for Mobile Applications

# DMA Controller (DMAC)

---

# EMMA Mobile1

Document No. S19891EJ1V0AN00
Date Published Aug. 2009

# PREFACE

**Purpose** The purpose of this document is to specify the usage of EMMA Mobile1 DMA controller.

**Organization** This document includes the following:

- Introduction
- Usage of DMAC
- Example of DMAC Operation
- DMAC Driver Function

**Notation** Here explains the meaning of following words in text:

| | |
|---|---|
| **Note** | Explanation of item indicated in the text |
| **Caution** | Information to which user should afford special attention |
| **Remark** | Supplementary information |

**Related document** The following tables list related documents.

**Reference Document**

| Document Name | Version/date | Author | Description |
|---|---|---|---|
| S19255EJ1V0UM00_DMA.pdf | 1st Edition | NECEL | DMAC user‘s manual |
| S19265EJ1V0UM00_ASMUGIO.pdf | 1st Edition | NECEL | SMU&GPIO interface user’s manual |
| S19268EJ1V0UM00_1chip.pdf | 1st Edition | NECEL | 1chip user’s manual |
| S19907EJ1V0AN00_GD.pdf | 1st Edition | NECEL | GD Spec |
| | | | |

**Disclaimers**

- **The information contained in this document is subject to change without prior notice in the future. Refer to the latest applicable data sheet(s) and user manual when designing a product for mass production.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC 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 NEC Electronics products listed in this documents or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customers' equipment shall be done under the full responsibility of the customer. NEC Electronics assume no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.

Note)

1. "NEC Electronics" as used in this document means NEC Electronics Corporation and also includes its majority-owned subsidiaries.
2. "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above)
3. All trademarks or registered trademarks are the property of their respective owners. Registered trademarks ® and trademarks™ are not noted in this document.

# CONTENTS

## LIST OF TABLES



## LIST OF FIGURES

# Chapter 1 Introduction

## 1.1 Outline

This document introduces how to use DMA controller (DMAC) of EMMA Mobile1.
More details about DMAC feature please refer to **EMMA Mobile 1 DMAC user's manual**.

## 1.2 Development Environment

- Hardware environment of this project is listed as below.

**Table 1-1 Hardware Environment**

| Name | Version | Maker |
|---|---|---|
| EMMA Mobile 1 evaluation board (PSKCH2Y-S-0016-01) | - | NEC Electronics |
| PARTNER-Jet ICE ARM | M20 | Kyoto Microcomputer Co. Ltd |

- Software used in this project is listed as below.

**Table 1-2 Software Environment**

| Name | Version | Maker |
|---|---|---|
| GNUARM Toolchain | V4.3.2 | GNU |
| WJETSET-ARM | V5.10a | Kyoto Microcomputer Co. Ltd |

# Chapter 2 Usage of DMAC

## 2.1 Outline of DMA Transfer

EMMA Mobile 1 DMAC has three physical channels (PCH0, PCH2, PCH3) and 30 logical channels (LCH).

- PCH0 (LCH0 to LCH3): Memory-to-memory transfer
- PCH2 (LCH0 to LCH14): Memory-to-peripheral transfer. (LCH6 and LCH8 are reserved.)
- PCH3 (LCH0 to LCH14): Peripheral-to-memory transfer. (LCH6 and LCH8 are reserved.)

Normal DMA data transfer procedure flow chart is shown as below.

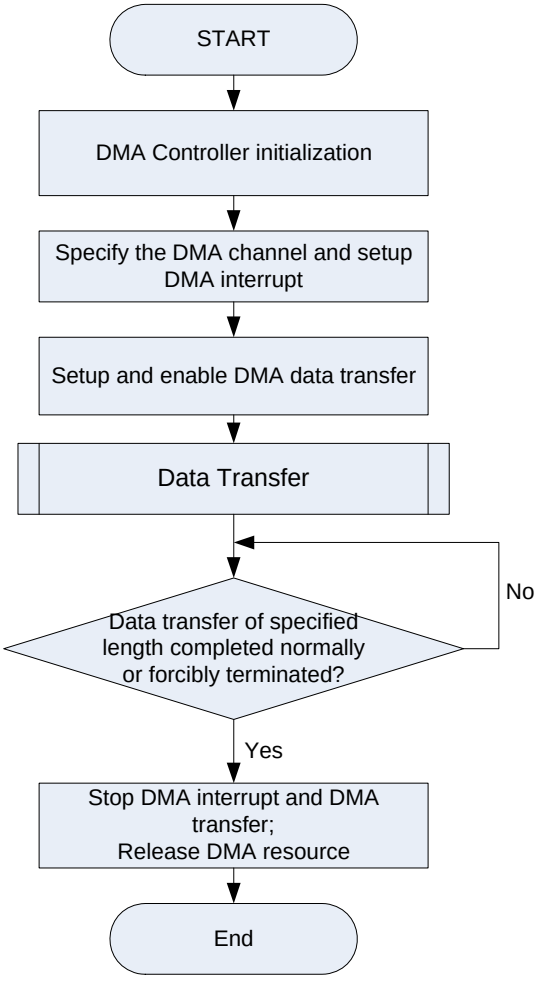



Figure 2-1 Normal DMA Data Transfer Flow

**Caution:**

About the explanation of all the DMA registers mentioned in this document, please refer to **"EMMA Mobile 1 DMA Controller User's Manual"**.

## 2.2 Detail of Normal DMA Data Transfer Procedure

### 2.2.1 DMA Controller Initialization

EMMA Mobile 1 DMA controller initialization including following sequences:

- Close and open DMA clock
  Register list:
  ASMU_GCLKCTRL0

- Reset and release reset DMA
  Register list:
  ASMU_RESETREQ0ENA
  ASMU_RESETREQ0

- Set DMA Interrupt.
  This step will clear interrupt.
  Register list:
  1) M2M
     DMA_ARM_PE0_LCH0LCH3_INT_REQ_CL
  2) M2P
     DMA_M2P_PE0_LCH0LCH3_INT_REQ_CL
     DMA_M2P_PE0_LCH4LCH7_INT_REQ_CL
     DMA_M2P_PE0_LCH8LCH11_INT_REQ_CL
     DMA_M2P_PE0_LCH12LCH14_INT_REQ_CL
  3) P2M
     DMA_P2M_PE0_LCH0LCH3_INT_REQ_CL
     DMA_P2M_PE0_LCH4LCH7_INT_REQ_CL
     DMA_P2M_PE0_LCH8LCH11_INT_REQ_CL
     DMA_P2M_PE0_LCH12LCH14_INT_REQ_CL

More about clock/reset control please refer to **“EMMA Mobile 1 System Control/General-Purpose I/O Interface User’s Manual”**.

### 2.2.2 Setup DMA

This step will set source data start address, source address offset, source block size count, destination start address, destination address offset, data total length, block size, and transfer mode .

Register List:

1) M2M

DMA_ARM_LCHx_AADD
DMA_ARM_LCHx_AOFF
DMA_ARM_LCHx_ASIZE_COUNT
DMA_ARM_LCHx_BADD
DMA_ARM_LCHx_BOFF
DMA_ARM_LCHx_BSIZE_COUNT
DMA_ARM_LCHx_LENG
DMA_ARM_LCHx_SIZE
DMA_ARM_LCHx_MODE

**Note:**

x = 0 to 3

2) M2P

DMA_M2P_LCHx_AADD
DMA_M2P_LCHx_AOFF
DMA_M2P_LCHx_ASIZE
DMA_M2P_LCHx_ASIZE_COUNT
DMA_M2P_LCHx_BADD
DMA_M2P_LCHx_LENG
DMA_M2P_LCHx_MODE
DMA_M2P_LCHn_TIME (n = 0/1/2)

**Note:**

x = 0 to 14

3) P2M

DMA_P2M_LCHx_AADD
DMA_P2M_LCHx_BADD
DMA_P2M_LCHx_BOFF
DMA_P2M_LCHx_BSIZE
DMA_P2M_LCHx_BSIZE_COUNT
DMA_P2M_LCHx_LENG
DMA_P2M_LCHx_MODE
DMA_P2M_LCHn_TIME (n = 0/1/2)

**Note:**

x = 0 to 14

### 2.2.3 Start DMA Transfer

This step will start DMA data transfer via set DMA control register.

Register List:

1) M2M
   DMA_ARM_CONT: for M2M LCHx (x = 0 to 3)
2) M2P
   DMA_M2P_CONT: for M2P LCHx (x = 0 to 14)
3) P2M
   DMA_P2M_CONT: for P2M LCHx (x = 0 to 14)

### 2.2.4 DMA Transfer

During DMA transfer procedure, the source data will be read from the source address, stored in internal buffer, and then written to the destination address.

When the specified block size of data has been transferred, a block interrupt will be outputted. Upon the completion of transfer, a length interrupt is generated.

### 2.2.5 DMA Stop

DMA transfer will stop under the following cases.

1) The transfer is completed normally. Can disable related DMA interrupt and release the DMA logical channel.
2) An error occurs during DMA transfer. DMA controller will suspend the transfer. And an interrupt will be generated.
3) End via set DMA end control register
   Register List:
   a) M2M
      DMA_ARM_END
   b) M2P
      DMA_M2P_END
   c) P2M
      DMA_P2M_END

4) End due to timeout (for UART only) under M2P or P2M mode.

Can get DMA transfer status according to DMA status register,

Register List:

a) M2M
   DMA_ARM_CONTSTATUS
b) M2P
   DMA_M2P_CONTSTATUS
c) P2M
   DMA_P2M_CONTSTATUS

# Chapter 3 Example of DMAC Operation

This chapter will show the following example of DMAC operation

- How to use DMA M2M data transfer (6 samples)
- **How to use DMA P2M&M2P data transfer (TBD)**
- Byte Lane Conversion Function

## 3.1 DMA M2M Data Transfer

### 3.1.1 Registers Configuration of DMA M2M Data Transfer

Table 3-1 shows the items list of DMA M2M transfer mode registers configuration.

**Table 3-1 Items list of DMA M2M Register Configuration**

| Items | Detail | Related Register |
|---|---|---|
| Set interrupt signals output destination | LCH0 to LCH3 | DMA_ARM_LCH0LCH3_INT_SEL |
| Mask interrupt source | Mask error, block, length interrupt of LCHx (1: mask) | DMA_ARM_LCH0LCH3_INT_ENABLE_CL |
| Clear interrupt source | Clear error, block, length interrupt of LCHx (1: clear) | DMA_ARM_LCH0LCH3_INT_REQ_CL |
| Enable interrupt source | Enable error, block, length interrupt of LCHx (1: enable) | DMA_ARM_LCH0LCH3_INT_ENABLE |
| Source address | 32 bit | DMA_ARM_LCHx_AADD (x = 0 to 3) |
| Source address offset | 0 – 65535 bytes | DMA_ARM_LCHx_AOFF (x = 0 to 3) |
| Address order | Ascending or reverse | |
| Block number in a loop on source side | 1 – 64 blocks | DMA_ARM_LCHx_ASIZE_COUNT (x = 0 to 3) |
| Block number in a loop on destination side | 1 – 64 blocks | DMA_ARM_LCHx_BSIZE_COUNT (x = 0 to 3) |
| Destination address | 32 bit | DMA_ARM_LCHx_BADD (x = 0 to 3) |
| Destination address offset | 0 – 65535 bytes | DMA_ARM_LCHx_BOFF (x = 0 to 3) |
| Address order | Ascending or reverse | |
| Length of transferred data | 0 – 16777215 bytes | DMA_ARM_LCHx_LENG (x = 0 to 3) |
| Block size | 1 – 65535 bytes | DMA_ARM_LCHx_SIZE (x = 0 to 3) |
| Select Logical channel and start DMA | Start DMA transfer | DMA_ARM_CONT |
| Stop DMA | Forces end DMA transfer | DMA_ARM_END |
| Byte lane conversion[Note 1] | Byte Lane on R/W side | DMA_ARM_LCHx_MODE (x = 0 to 3) |
| Repeat control | Repeat/No Repeat | |

**Note**

1. For byte lane conversion, please refer to "**Chapter 3.3 Byte Lane Conversion Function**".

### 3.1.2 Sample1 (No Repeat, Without Offset, address-ascending order)

Table 3-2 shows conditions and registers setting of Sample1.

**Table 3-2 Conditions and Registers Setting of Sample1**

| Items | Detail | Related Register |
|---|---|---|
| Logical channel used | LCH0 | DMA_ARM_CONT = 0x00000001 |
| Source address | 0x30100000[Note 1] | DMA_ARM_LCH0_AADD = 0x30100000 |
| Destination address | 0x30200000[Note 1] | DMA_ARM_LCH0_BADD = 0x30200000 |
| Block size | 4096 bytes | DMA_ARM_LCH0_SIZE = 0x00001000 |
| Offset between blocks (source side) | 0 bytes | DMA_ARM_LCH0_AOFF = 0x00000000 |
| Address order (source side) | Ascending | |
| Offset between blocks (destination side) | 0 bytes | DMA_ARM_LCH0_BOFF = 0x00000000 |
| Address order (destination side) | Ascending | |
| Length of transferred data | 20480 bytes | DMA_ARM_LCH0_LENG = 0x00005000 |
| Byte lane conversion[Note 2] | [D3][D2][D1][D0] → [D2][D3][D0][D1] | DMA_ARM_LCH0_MODE = B1E4_0000H |
| Repeat control | No repeat | |

**Note**

1. If the DMA controller attempts to access an address to which no device is assigned, an error response is made and DMA transfer ends. Refer to "**Chapter 3.5 (1) Forced end due to an error response of EMMA Mobile 1 DMA Controller User's Manual**".

2. When performing byte lane conversion, parameters must be set in word units. However, parameter setting in other than word units is enabled if a condition described in "**Chapter 3.3 Byte Lane Conversion Function**".

Figure 3-1 shows the register setting flow of DMA M2M transfer Sample1.

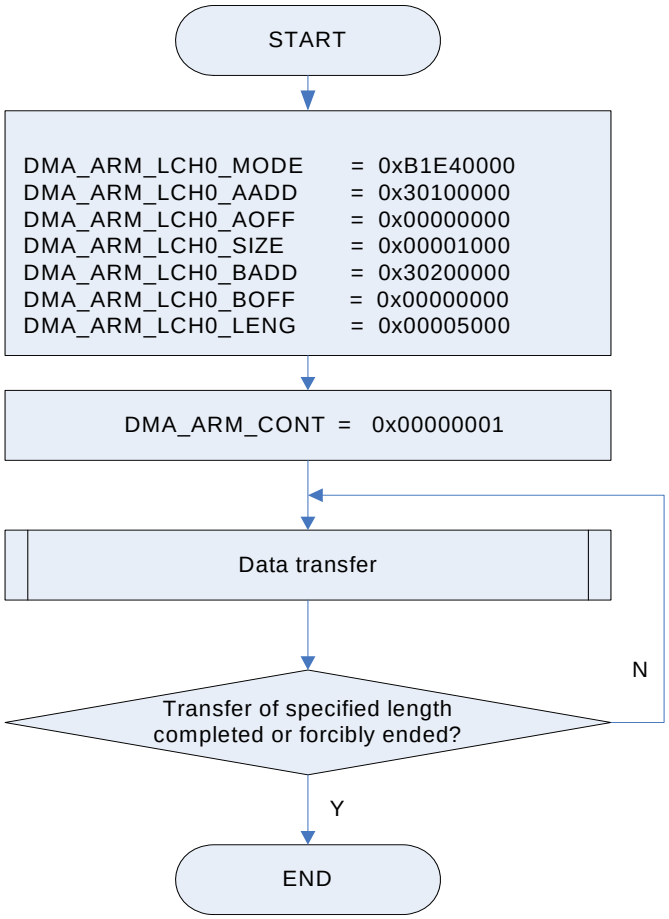



Figure 3-1 Flow of M2M transfer sample1

In this sample, because of without offset on both source side and destination side, the transfer is continued by incrementing the address value on these two sides, beginning with the start address. After the specified block size or length size of data has been transferred, an interrupt request signal will be outputted.

Figure 3-2 shows the data transfer flow of DMA M2M transfer Sample1.

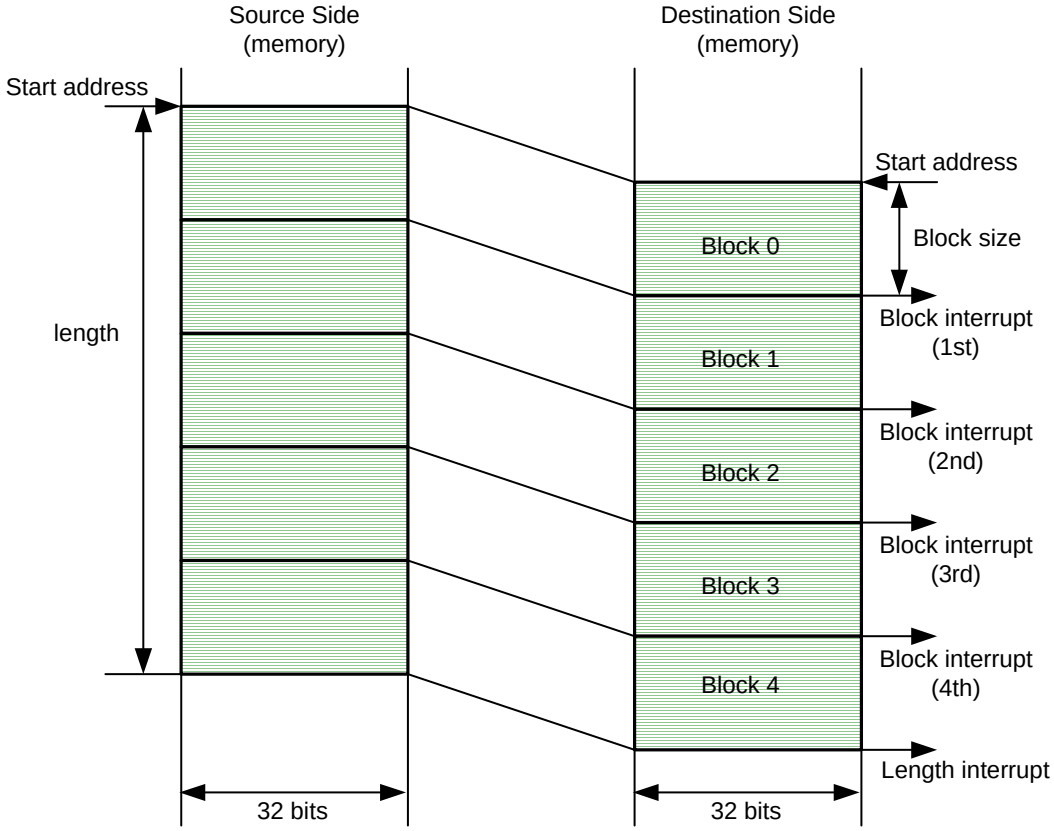



Figure 3-2 Data Flow of M2M transfer Sample1

### 3.1.3 Sample2 (No Repeat, With Offset, address-ascending order)

Table 3-3 shows conditions and registers setting of Sample2.

**Table 3-3 Conditions and Registers Setting of Sample2**

| Items | Detail | Related Register |
|---|---|---|
| Logical channel used | LCH1 | DMA_ARM_CONT = 0x00000002 |
| Source address | 0x30300000H[Note 1] | DMA_ARM_LCH1_AADD = 0x30300000 |
| Destination address | 0x30400000H[Note 1] | DMA_ARM_LCH1_BADD = 0x30400000 |
| Block size | 4096 bytes | DMA_ARM_LCH1_SIZE = 0x00001000 |
| Offset between blocks (source side) | 1024 bytes | DMA_ARM_LCH1_AOFF = 0x00000400 |
| Address order (source side) | Ascending | |
| Offset between blocks (destination side) | 0 bytes | DMA_ARM_LCH1_BOFF = 0x00000000 |
| Address order (destination side) | Ascending | |
| Length of transferred data | 12288 bytes | DMA_ARM_LCH1_LENG = 0x00003000 |
| Byte lane conversion[Note 2] | [D3][D2][D1][D0] → [D2][D3][D0][D1] | DMA_ARM_LCH1_MODE = 0xB1E40000 |
| Repeat control | No repeat | |

**Note**

1. If the DMA controller attempts to access an address to which no device is assigned, an error response is made and DMA transfer ends. Refer to "**Chapter 3.5 (1) Forced end due to an error response of EMMA Mobile 1 DMA Controller User's Manual**".
2. When performing byte lane conversion, parameters must be set in word units. However, parameter setting in other than word units is enabled if a condition described in "**Chapter 3.3 Byte Lane Conversion Function**".

Figure 3-3 shows the register setting flow of DMA M2M transfer Sample2.

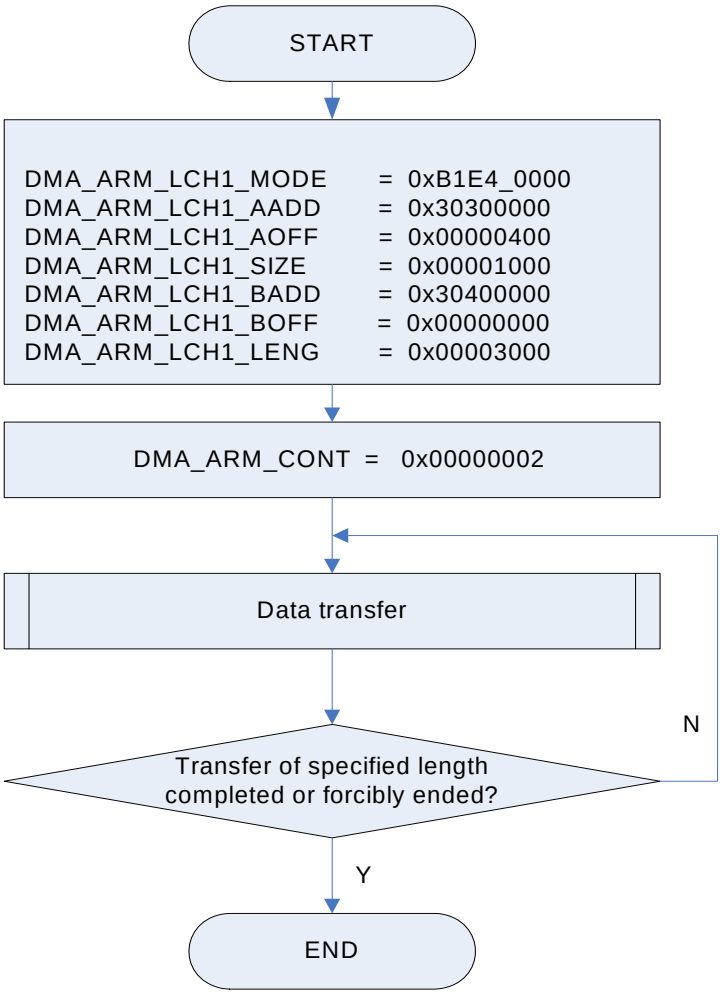



Figure 3-3 Flow of M2M transfer sample2

In this sample, the transfer (send) is performed on the source side as follows:
<1>. The start address is incremented, and the block size data is transferred.
<2>. After the increment, the specified offset is added to the address value.
<3>. Another block of data is transferred.

The transfer (receive) is performed on the destination side as follows:
The transfer is continued by incrementing the address by block size, beginning with the start address.
After the specified block size or length size of data has been transferred, an interrupt request signal will be outputted.

As for the source side, this offset function also can be supported on the destination side.

Figure 3-4 shows the data transfer flow of DMA M2M transfer Sample2.

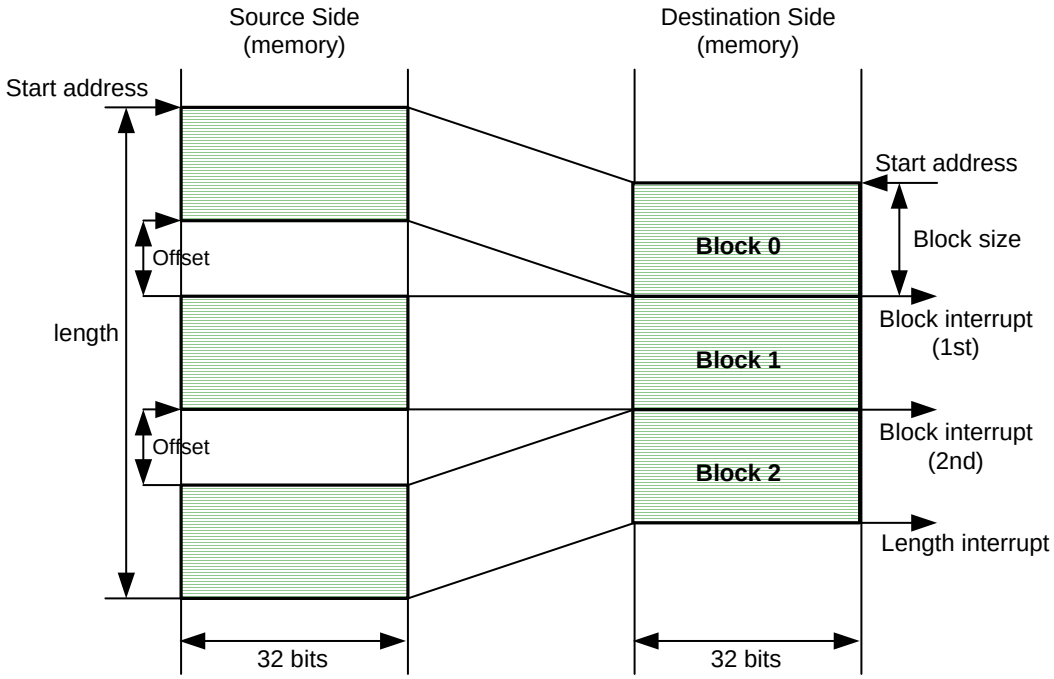



Figure 3-4 Data Flow of M2M transfer Sample2

### 3.1.4 Sample3 (Repeat on , Without Offset, address-ascending order)

Table 3-4 shows conditions and registers setting of Sample3.

**Table 3-4 Conditions and Registers Setting of Sample3**

| Items | Detail | Related Register |
|---|---|---|
| Logical channel used | LCH3 | DMA_ARM_CONT_SEC = 0x00000008 |
| Source address | 0x30500000[Note 1] | DMA_ARM_LCH3_AADD = 0x30500000 |
| Destination address | 0x30600000[Note 1] | DMA_ARM_LCH3_BADD = 0x30600000 |
| Block size | 4096 bytes | DMA_ARM_LCH3_SIZE = 0x00001000 |
| Offset between blocks (source side) | 0 bytes | DMA_ARM_LCH3_AOFF = 0x00000000 |
| Address order (source side) | Ascending | |
| Offset between blocks (destination side) | 0 bytes | DMA_ARM_LCH3_BOFF = 0x00000000 |
| Address order (destination side) | Ascending | |
| Length of transferred data | 20480 bytes | DMA_ARM_LCH3_LENG = 0x00005000 |
| Byte lane conversion[Note 2] | [D3][D2][D1][D0] → [D2][D3][D0][D1] | DMA_ARM_LCH3_MODE = 0xB1E40100 |
| Repeat control | Repeat on destination side | DMA_ARM_LCH3_BSIZE_COUNT = 0x00000000 |
| Repeat size | 1 block | |

**Note**

1. If the DMA controller attempts to access an address to which no device is assigned, an error response is made and DMA transfer ends. Refer to "**Chapter 3.5 (1) Forced end due to an error response of EMMA Mobile 1 DMA Controller User's Manual**".
2. When performing byte lane conversion, parameters must be set in word units. However, parameter setting in other than word units is enabled if a condition described in "**Chapter 3.3 Byte Lane Conversion Function**".

Figure 3-5 shows the register setting flow of DMA M2M transfer Sample3.

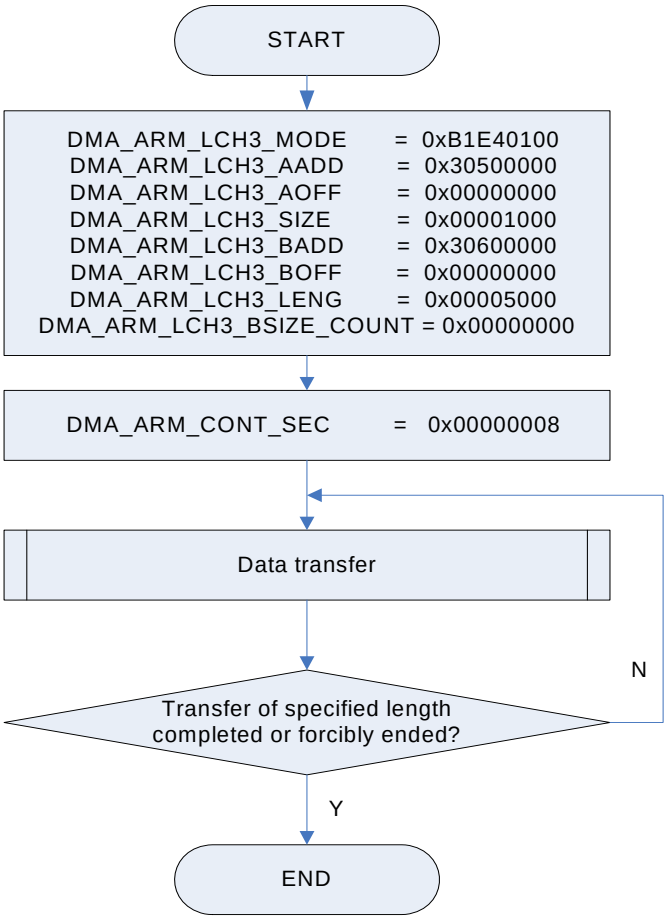



Figure 3-5 Flow of M2M transfer sample3

In this sample, the repeat transfer mode only is operated on the destination side.
So on the source side, the transfer (send) is continued in an incremental manner, beginning with the start address. Upon completion of transfer, a length interrupt is generated.

On the destination side, the transfer (receive) is performed as follows:
<1>. The start address is incremented, and the block size data is transferred.
<2>. After the increment, the address value is reset with the start address value.
<3>. Another block of data is transferred.
After the specified block size or length size of data has been transferred, an interrupt request signal will be outputted.

As for the destination side, this case of repeat function also can be supported on the source side.

Figure 3-6 shows the data transfer flow of DMA M2M transfer Sample3.

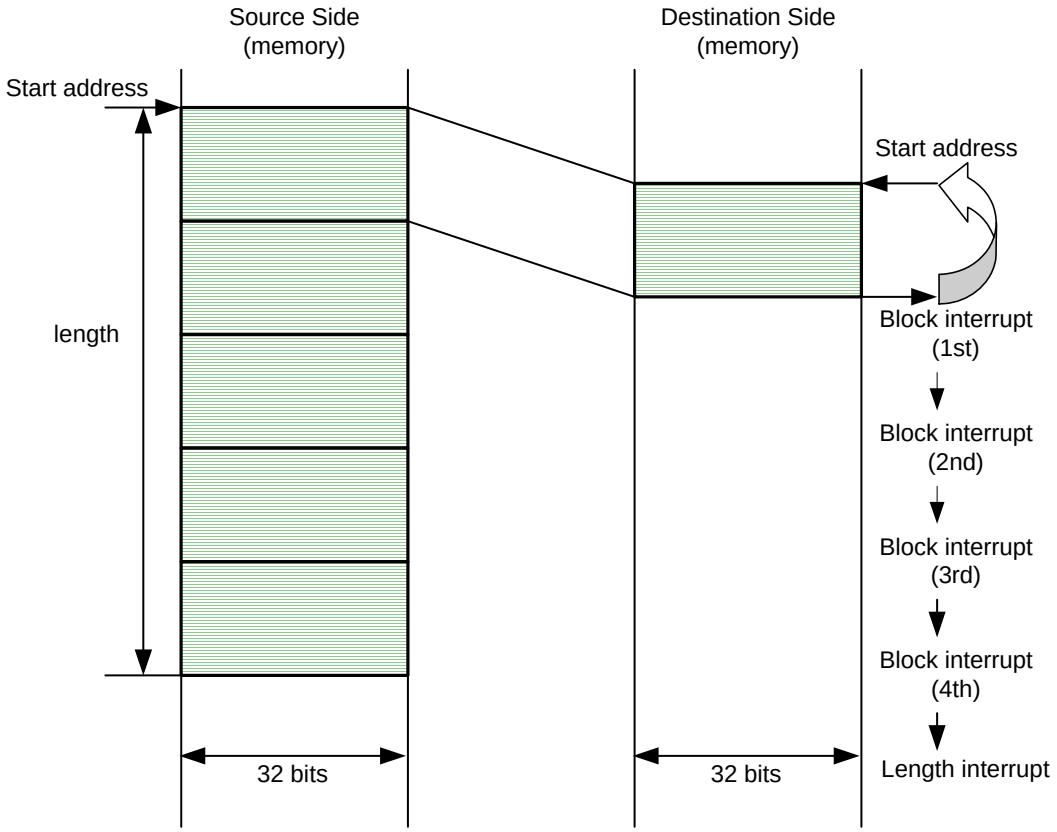



Figure 3-6 Data Flow of M2M transfer Sample3

### 3.1.5 Sample4 (Repeat, With Offset, address-ascending order)

Table 3-5 shows conditions and registers setting of Sample4.

**Table 3-5 Conditions and Registers Setting of Sample4**

| Items | Detail | Related Register |
|---|---|---|
| Logical channel used | LCH2 | DMA_ARM_CONT = 0x00000004 |
| Source address | 0x30700000[Note 1] | DMA_ARM_LCH2_AADD = 0x30700000 |
| Destination address | 0x30800000[Note 1] | DMA_ARM_LCH2_BADD = 0x30800000 |
| Block size | 4096 bytes | DMA_ARM_LCH2_SIZE = 0x00001000 |
| Offset between blocks (source side) | 1024 bytes | DMA_ARM_LCH2_AOFF = 0x00000400 |
| Address order (source side) | Asending | |
| Offset between blocks (destination side) | 0 bytes | DMA_ARM_LCH2_BOFF = 0x00000000 |
| Address order (destination side) | Asending | |
| Length of transferred data | 12288 bytes | DMA_ARM_LCH2_LENG = 0x00003000 |
| Byte lane conversion[Note 2] | Not performed | DMA_ARM_LCH2_MODE = 0xE4E40100<br>DMA_ARM_LCH2_BSIZE_COUNT = 0x00000000 |
| Repeat control | Repeat on destination side | |
| Repeat size | 1 block | |

**Note**

1. If the DMA controller attempts to access an address to which no device is assigned, an error response is made and DMA transfer ends. Refer to "**Chapter 3.5 (1) Forced end due to an error response of EMMA Mobile 1 DMA Controller User's Manual**".
2. When performing byte lane conversion, parameters must be set in word units. However, parameter setting in other than word units is enabled if a condition described in "**Chapter 3.3 Byte Lane Conversion Function**".

Figure 3-7 shows the register setting flow of DMA M2M transfer Sample4.

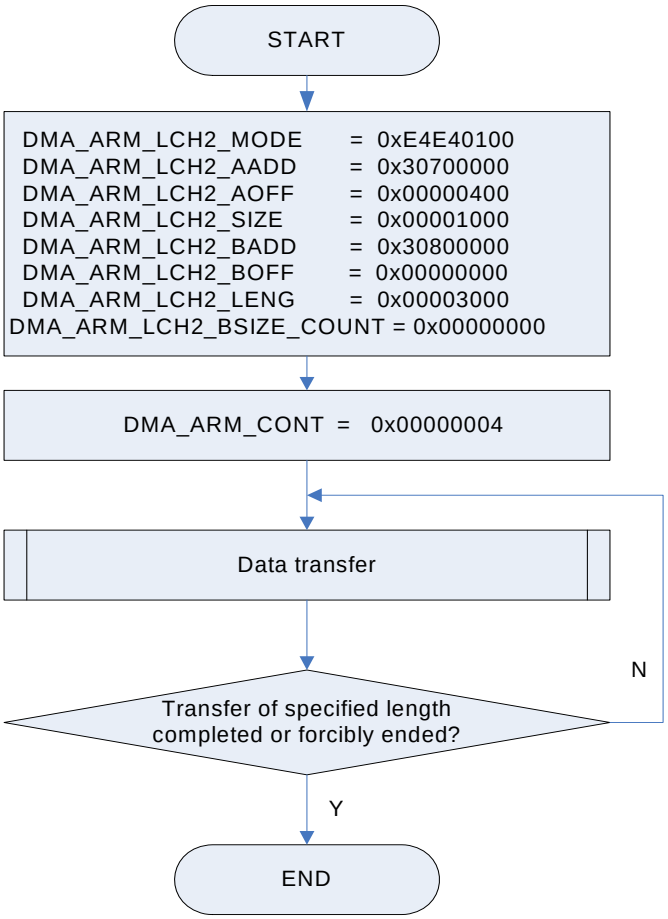



Figure 3-7 Flow of M2M transfer sample4

In this sample, the repeat transfer mode only is operated on the destination side.
So on the source side, the transfer (send) is performed as follows:
<1>. The start address is incremented, and the block size data is transferred.
<2>. After the increment, the specified offset is added to the address value.
<3>. Another block of data is transferred.

On the destination side, the transfer (receive) is performed as follows:
<1>. The start address is incremented, and the block size data is transferred.
<2>. After the increment, the address value is reset with the start address value.
<3>. Another block of data is transferred.
After the specified block size or length size of data has been transferred, an interrupt request signal will be outputted.

As for the destination side, this case of repeat function also can be supported on the source side.

Figure 3-8 shows the data transfer flow of DMA M2M transfer Sample4.

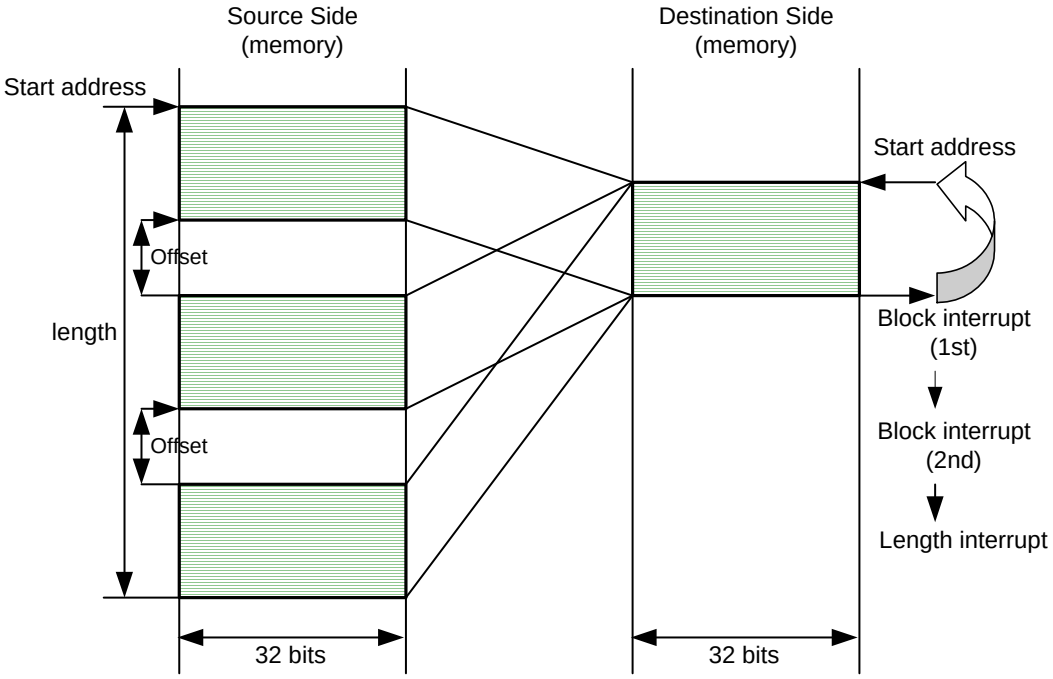



Figure 3-8 Data Flow of M2M transfer Sample4

### 3.1.6 Sample5 (Both Repeat, With Offset, address-ascending order)

Table 3-6 shows conditions and registers setting of Sample5.

**Table 3-6 Conditions and Registers Setting of Sample5**

| Items | Detail | Related Register |
|---|---|---|
| Logical channel used | LCH0 | DMA_ARM_CONT = 0x00000001 |
| Source address | 0x30900000[Note 1] | DMA_ARM_LCH0_AADD = 0x30900000 |
| Destination address | 0x30A00000[Note 1] | DMA_ARM_LCH0_BADD = 0x30A00000 |
| Block size | 2048 bytes | DMA_ARM_LCH0_SIZE = 0x00000800 |
| Offset between blocks (source side) | 1024 bytes | DMA_ARM_LCH0_AOFF = 0x00000400 |
| Address order (source side) | Ascending | |
| Offset between blocks (destination side) | 0 byte | DMA_ARM_LCH0_BOFF = 0x00000000 |
| Address order (destination side) | Ascending | |
| Length of transferred data | 40960 bytes | DMA_ARM_LCH0_LENG = 0x0000A000 |
| Byte lane conversion[Note 2] | [D3][D2][D1][D0] → [D2][D3][D0][D1] | DMA_ARM_LCH2_MODE = 0xB1E40100<br>DMA_ARM_LCH0_ASIZE_COUNT = 0x00000001<br>DMA_ARM_LCH0_BSIZE_COUNT = 0x00000001 |
| Repeat control | Repeat on both sides | |
| Repeat size | 2 blocks (source side)<br>2 blocks (destination side) | |

**Note**

1. If the DMA controller attempts to access an address to which no device is assigned, an error response is made and DMA transfer ends. Refer to **"Chapter 3.5 (1) Forced end due to an error response of EMMA Mobile 1 DMA Controller User's Manual"**.
2. When performing byte lane conversion, parameters must be set in word units. However, parameter setting in other than word units is enabled if a condition described in "**Chapter 3.3 Byte Lane Conversion Function**".

Figure 3-9 shows the register setting flow of DMA M2M transfer Sample5.

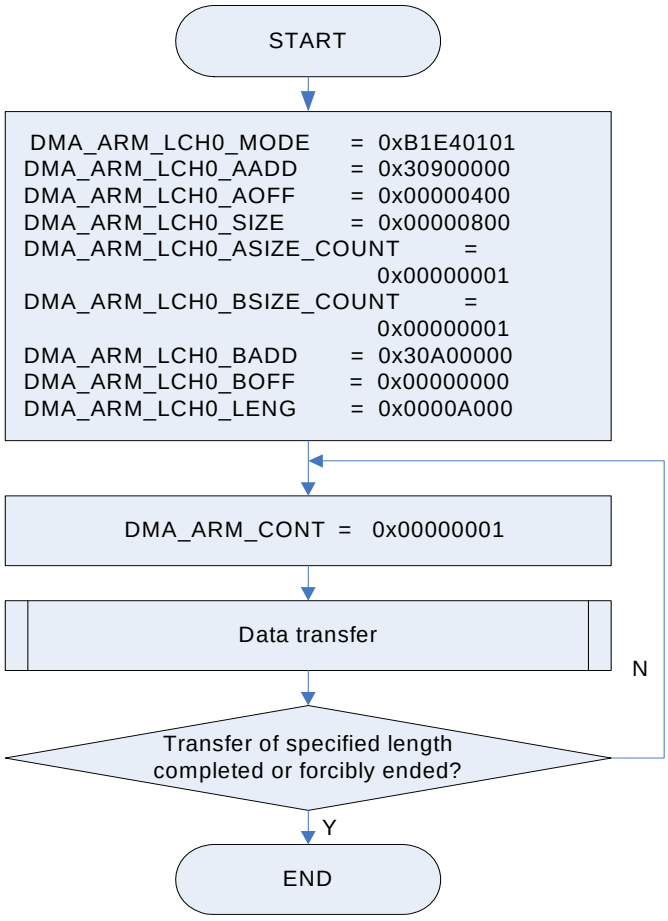



Figure 3-9 Flow of M2M transfer sample5

In this sample, the repeat transfer mode is operated on both the source and the destination sides.
On the source side, the transfer (send) is performed as follows:
<1>. The start address is incremented, and the block size data is transferred.
<2>. After the increment, the offset is added to the address value.
<3>. Another block of data is transferred.
<4>. After the increment, the address value is reset with the start address value.
<5>. Another block of data is transferred.
Upon completion of transfer, a length interrupt is generated.

On the destination side, the transfer (receive) is performed as follows:
<1>. The start address is incremented, and the block size data is transferred.
<2>. After the increment, the address value is reset with the start address value.
<3>. Another block of data is transferred.
After the specified block size or length size of data has been transferred, an interrupt request signal will be outputted.

Figure 3-10 shows the data transfer flow of DMA M2M transfer Sample5.

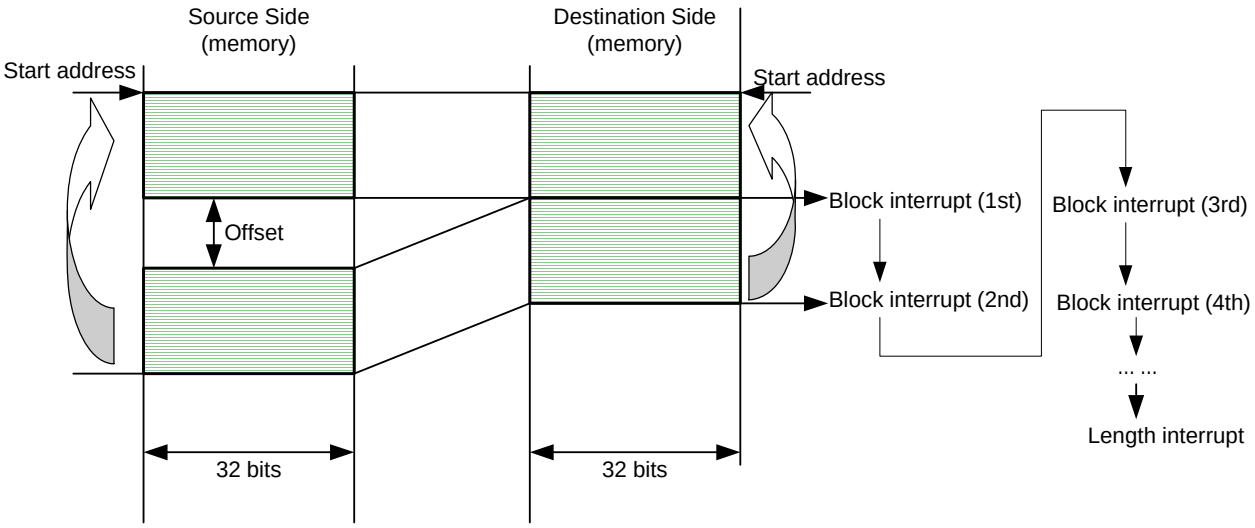



Figure 3-10 Data Flow of M2M transfer Sample5

### 3.1.7 Sample6 (No Repeat, With Offset, address-reverse order)

Table 3-7 shows conditions and registers setting of Sample6.

**Table 3-7 Conditions and Registers Setting of Sample6**

| Items | Detail | Related Register |
|---|---|---|
| Logical channel used | LCH0 | DMA_ARM_CONT = 0x00000001 |
| Source address | 0x30B00000[Note 1] | DMA_ARM_LCH0_AADD = 0x30B00000 |
| Destination address | 0x30C00000[Note 1] | DMA_ARM_LCH0_BADD = 0x30C00000 |
| Block size | 4096 bytes | DMA_ARM_LCH0_SIZE = 0x00001000 |
| Offset between blocks (source side) | 9216 bytes | DMA_ARM_LCH0_AOFF = 0x00012400 |
| Address order (source side) | Reverse | |
| Offset between blocks (destination side) | 1024 bytes | DMA_ARM_LCH0_BOFF = 0x00000400 |
| Address order (destination side) | Ascending | |
| Length of transferred data | 12288 bytes | DMA_ARM_LCH0_LENG = 0x00003000 |
| Byte lane conversion[Note 2] | Not preformed | DMA_ARM_LCH0_MODE = 0xE4E40000 |
| Repeat control | No repeat | |

**Note**

1. If the DMA controller attempts to access an address to which no device is assigned, an error response is made and DMA transfer ends. Refer to **"Chapter 3.5 (1) Forced end due to an error response of EMMA Mobile 1 DMA Controller User's Manual"**.
2. When performing byte lane conversion, parameters must be set in word units. However, parameter setting in other than word units is enabled if a condition described in "**Chapter 3.3 Byte Lane Conversion Function**".

Figure 3-11 shows the register setting flow of DMA M2M transfer Sample6.

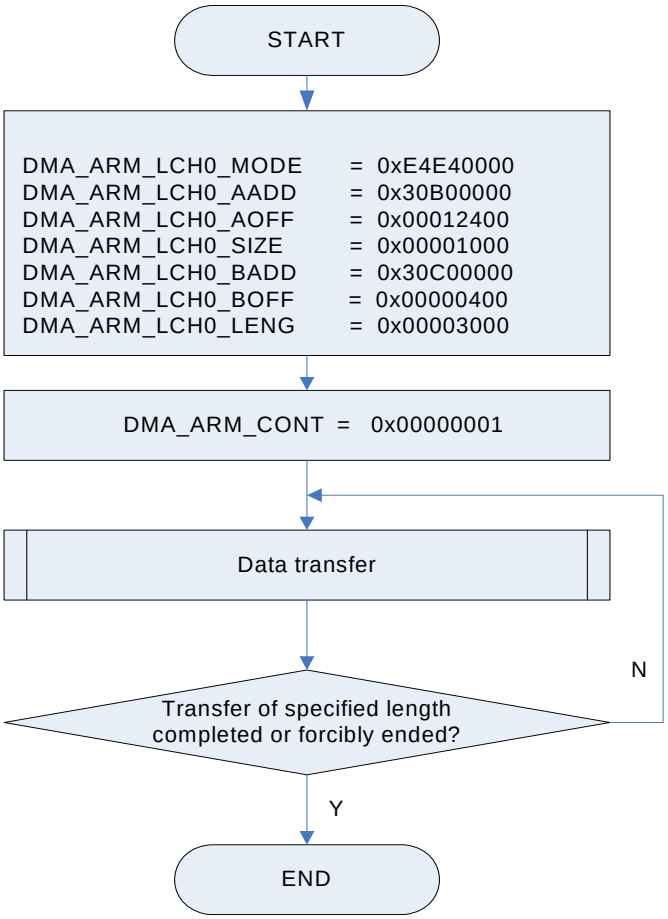



Figure 3-11 Flow of M2M transfer sample6

In this sample, the block data in reverse order on the source side are transferred.
On the source side, the transfer (send) is performed as follows:
<1>. The start address is incremented, and the block size data is transferred.
<2>. After the increment, the specified offset is subtracted from the address value.
<3>. Another block of data is transferred.

On the destination side, the transfer (receive) is performed as follows:
<1>. The start address is incremented, and the block size data is transferred.
<2>. After the increment, the specified offset is added to the address value.
<3>. Another block of data is transferred.
After the specified block size or length size of data has been transferred, an interrupt request signal will be outputted.

As for the source side, this function also can be supported on the destination side.

Figure 3-12 shows the data transfer flow of DMA M2M transfer Sample6.

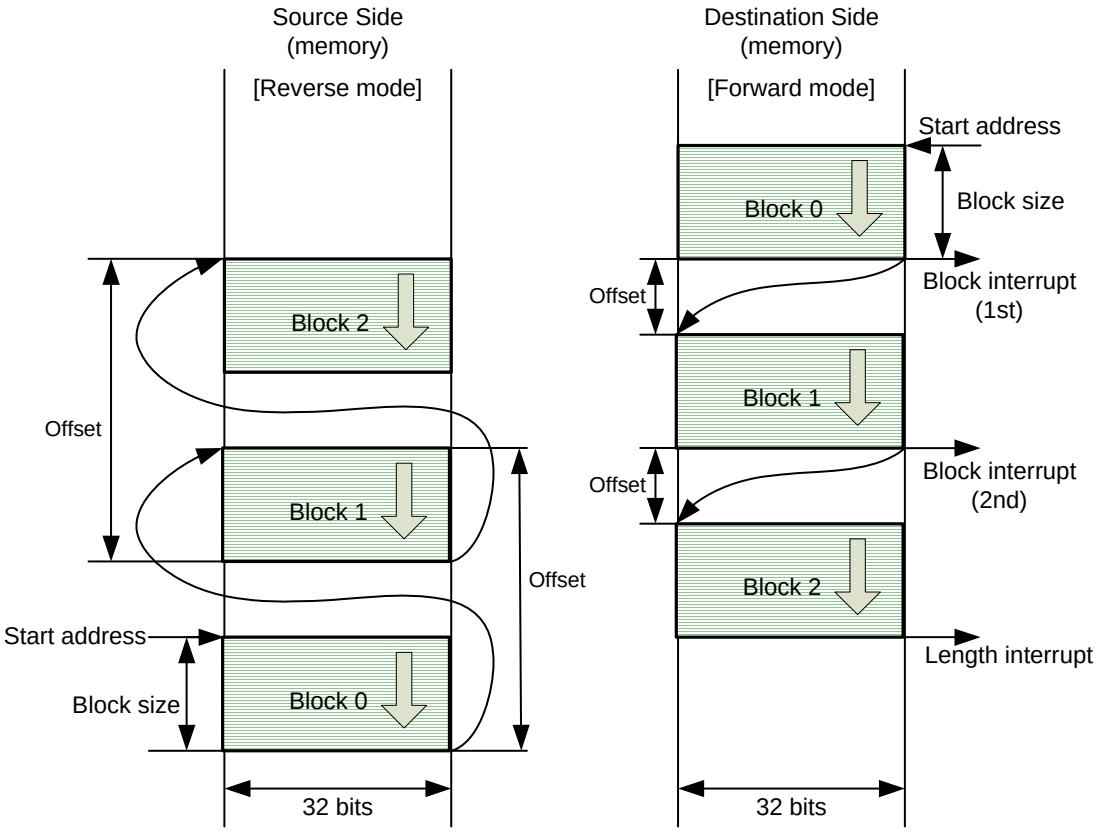



Figure 3-12 Data Flow of M2M transfer Sample6

## 3.2 DMA M2P & P2M Data Transfer

### 3.2.1 Registers Configuration of DMA M2P&P2M Data Transfer

Table 3-8 shows the items list of M2P DMA Configuration.

**Table 3-8 Items list of DMA M2P Register Configuration**

| Items | Detail | Related Register |
|---|---|---|
| Logical channel used | LCH0 to LCH14 | DMA_M2P_CONT |
| Set interrupt signals output destinations | LCH0 to LCH14 | DMA_M2P_LCH0LCH14_INT_SEL |
| Mask interrupt source | Mask error, block, length interrupt of LCHx (1: mask) (x = 0 to 14) | DMA_M2P_PE0_LCH0LCH3_INT_ENABLE_CL<br>DMA_M2P_PE0_LCH4LCH7_INT_ENABLE_CL<br>DMA_M2P_PE0_LCH8LCH11_INT_ENABLE_CL<br>DMA_M2P_PE0_LCH12LCH14_INT_ENABLE_CL |
| Clear interrupt source | Clear error, block, length interrupt of LCHx (1: clear) (x = 0 to 14) | DMA_M2P_PE0_LCH0LCH3_INT_REQ_CL<br>DMA_M2P_PE0_LCH4LCH7_INT_REQ_CL<br>DMA_M2P_PE0_LCH8LCH11_INT_REQ_CL<br>DMA_M2P_PE0_LCH12LCH14_INT_REQ_CL |
| Enable interrupt source | Enable error, block, length interrupt of LCHx (1: enable) (x = 0 to 14) | DMA_M2P_PE0_LCH0LCH3_INT_ENABLE<br>DMA_M2P_PE0_LCH4LCH7_INT_ENABLE<br>DMA_M2P_PE0_LCH8LCH11_INT_ENABLE<br>DMA_M2P_PE0_LCH12LCH14_INT_ENABLE |
| Source address | 32 bit | DMA_M2P_LCHx_AADD (x = 0 to 14) |
| Destination address | 32 bit | DMA_M2P_LCHx_BADD (x = 0 to 14) |
| Block size | 1 – 65535 bytes | DMA_M2P_LCHx_ASIZE (x = 0 to 14) |
| Offset between blocks | 0 – 65535 bytes | DMA_M2P_LCHx_AOFF (x = 0 to 14) |
| Transfer data length | 0 – 16777215 bytes | DMA_M2P_LCHx_LENG (x = 0 to 14) |
| Transfer bit width | 32/16/8 bits | DMA_M2P_LCHx_MODE (x = 0 to 14) |
| Byte lane conversion[Note] | None | DMA_M2P_LCHx_MODE (x = 0 to 14) |
| Source block size count | 1 – 16 blocks | DMA_M2P_LCHx_ASIZE_COUNT(x = 0 to 14) |
| Repeat control | Repeat/No Repeat | DMA_M2P_LCHx_MODE (x = 0 to 14) |
| Timer function | Only for UART (LCH0,LCH1,LCH2) | DMA_M2P_LCHx_MODE<br>DMA_M2P_LCHx_TIME (x = 0 to 2) |

**Note**

For byte lane conversion, please refer to **"Chapter 3.3 Byte Lane Conversion Function"**.

Table 3-9 shows the items list of DMA P2M registers configuration.

**Table 3-9 Items list of of DMA P2M Register Configuration**

| Items | Detail | Related Register |
|---|---|---|
| Logical channel used | LCH0 to LCH14 | DMA_P2M_CONT |
| Set interrupt signals output destinations | LCH0 to LCH14 | DMA_P2M_LCH0LCH14_INT_SEL |
| Mask interrupt source | Mask error, block, length interrupt of LCHx (1: mask) (x = 0 to 14) | DMA_P2M_PE0_LCH0LCH3_INT_ENABLE_CL<br>DMA_P2M_PE0_LCH4LCH7_INT_ENABLE_CL<br>DMA_P2M_PE0_LCH8LCH11_INT_ENABLE_CL<br>DMA_P2M_PE0_LCH12LCH14_INT_ENABLE_CL |
| Clear interrupt source | Clear error, block, length interrupt of LCHx (1: clear) (x = 0 to 14) | DMA_P2M_PE0_LCH0LCH3_INT_REQ_CL<br>DMA_P2M_PE0_LCH4LCH7_INT_REQ_CL<br>DMA_P2M_PE0_LCH8LCH11_INT_REQ_CL<br>DMA_P2M_PE0_LCH12LCH14_INT_REQ_CL |
| Enable interrupt source | Enable error, block, length interrupt of LCHx (1: enable) (x = 0 to 14) | DMA_P2M_PE0_LCH0LCH3_INT_ENABLE<br>DMA_P2M_PE0_LCH4LCH7_INT_ENABLE<br>DMA_P2M_PE0_LCH8LCH11_INT_ENABLE<br>DMA_P2M_PE0_LCH12LCH14_INT_ENABLE |
| Source address | 32 bit | DMA_P2M_LCHx_AADD (x = 0 to 14) |
| Destination address | 32 bit | DMA_P2M_LCHx_BADD (x = 0 to 14) |
| Block size | 1 – 65535 bytes | DMA_P2M_LCHx_ASIZE (x = 0 to 14) |
| Offset between blocks | 0 – 65535 bytes | DMA_P2M_LCHx_AOFF (x = 0 to 14) |
| Transfer data length | 0 – 16777215 bytes | DMA_P2M_LCHx_LENG (x = 0 to 14) |
| Transfer bit width | 32/16/8 bits | DMA_P2M_LCHx_MODE (x = 0 to 14) |
| Byte lane conversion[Note] | None | DMA_P2M_LCHx_MODE (x = 0 to 14) |
| Source block size count | 1 – 16 blocks | DMA_P2M_LCHx_ASIZE_COUNT(x = 0 to 14) |
| Repeat control | Repeat/No Repeat | DMA_P2M_LCHx_MODE (x = 0 to 14) |
| Timer function | Only for UART0/1/2 (LCH0,LCH1,LCH2) | DMA_P2M_LCHx_MODE<br>DMA_P2M_LCHx_TIME (x = 0 to 2) |

**Note**

For byte lane conversion, please refer to "**Chapter 3.3 Byte Lane Conversion Function"**.

### 3.2.2 Sample for M2P&P2M DMA Transfer

This sample will use M2P for writing EMMC and P2M for reading EMMC.

This chapter only explains DMA operation. About EMMC operation, please refer to **"EMMA Mobile 1 EMMC Application Note"**.

The sequence of M2P&P2M sample is as following,

1) Write the data of buffer 1 to EMMC by M2P DMA
2) Read EMMC data to buffer 2 by P2M DMA
3) Compare buffer 1 and buffer 2

If the buffer data is same, it means the sample of M2P&P2M DMA is correct. Otherwise, M2P&P2M DMA is incorrect.

Table 3-10 shows conditions and registers setting of M2P&P2M DMA Sample.

**Table 3-10 Conditions and Registers Setting of M2P&P2M DMA Sample**

| W/R | Items | Detail | Related Register |
|---|---|---|---|
| EMMC Write (M2P DMA) | Logical channel | LCH5 (EMMC interface) | DMA_M2P_CONT = 0x00000020 |
| | Source address | 0x31000000 | DMA_M2P_LCH5_AADD = 0x31000000 |
| | Destination address | SDIC_DMASD | DMA_M2P_LCH5_BADD = 0x50090300 |
| | Block size | 1024 bytes | DMA_M2P_LCH5_ASIZE = 0x00000400 |
| | Offset | 0 | DMA_M2P_LCH5_AOFF = 0x00000000 |
| | Length | 8M bytes | DMA_M2P_LCH5_LENG = 0x00800000 |
| | Transfer bit width | 16 bits | DMA_M2P_LCH5_MODE = 0xE4E40010 |
| | Byte lane conversion | default | |
| | Repeat control | No Repeat | |
| EMMC Read (P2M DMA) | Logical channel | LCH2 (EMMC interface) | DMA_P2M_CONT = 0x00000020 |
| | Source address | SDIC_DMASD | DMA_P2M_LCH5_AADD = 0x50090300 |
| | Destination address | 0x30000000 | DMA_P2M_LCH5_BADD = 0x30000000 |
| | Block size | 1024 bytes | DMA_P2M_LCH5_BSIZE = 0x00000400 |
| | Offset | 0 | DMA_P2M_LCH5_BOFF = 0x00000000 |
| | Length | 8M bytes | DMA_P2M_LCH5_LENG = 0x00800000 |
| | Transfer bit width | 16 bits | DMA_P2M_LCH5_MODE = 0xE4E40010 |
| | Byte lane conversion | default | |
| | Repeat control | No Repeat | |

Figure 3-13 shows DMA M2P&P2M register setting flow of this sample.


Figure 3-13 Flow of M2P&P2M transfer sample

## 3.3 Byte Lane Conversion Function

### 3.3.1 Registers Configuration

Table 3-11 shows the items list of Byte Lane Conversion Function Configuration. For details, please refer to **"Chapter 3.3 Byte Lane Conversion Function of EMMA Mobile 1 DMA Controller User's Manual"**.

**Table 3-11 Items list of DMA Configuration**

| Items | Detail | Related Register |
|---|---|---|
| ENDI_R | Byte lane to latch data on the read side | xxx_LCHx_MODE_ENDI_R_xx bit of mode register (DMA_xxx_LCHx_MODE) |
| ENDI_W | Byte lane to latch data on the write side | xxx_LCHx_MODE_ENDI_W_xx bit of mode register (DMA_xxx_LCHx_MODE) |
| AADD | Source address | DMA_xxx_LCHx_AADD |
| BADD | Destination address | DMA_xxx_LCHx_BADD |
| MODE_BIT | 32/16/8 bits | xxx_LCHx_MODE_BIT bit of mode register (DMA_xxx_LCHx_MODE) |

### 3.3.2 Setting Examples

Figure 3-14 to Figure 3-21 show examples of byte lane conversion settings and the output results.

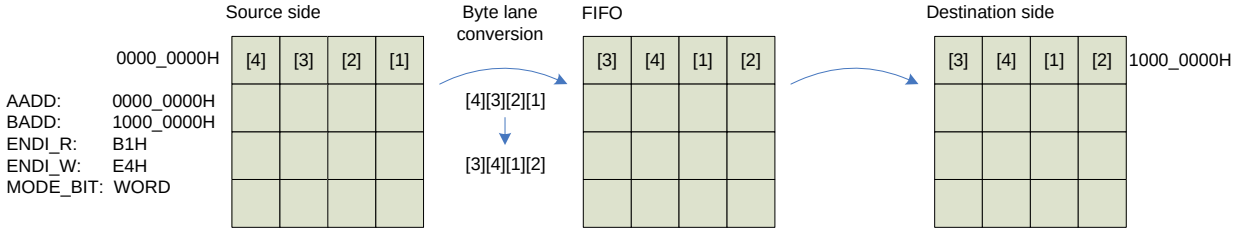



Figure 3-14 Byte Lane Conversion Example 1 (Read Side)

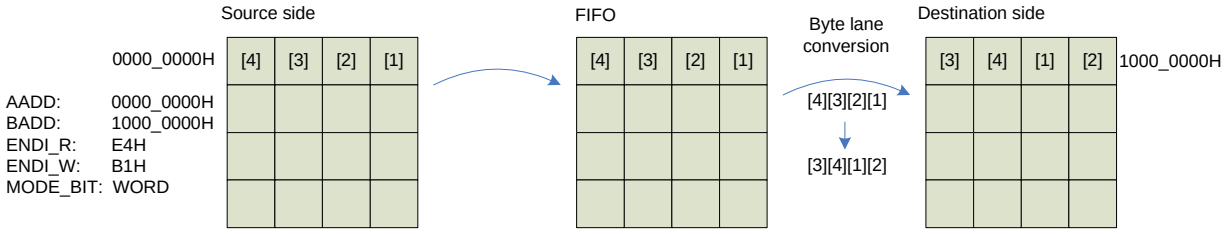



Figure 3-15 Byte Lane Conversion Example 2 (Write Side)

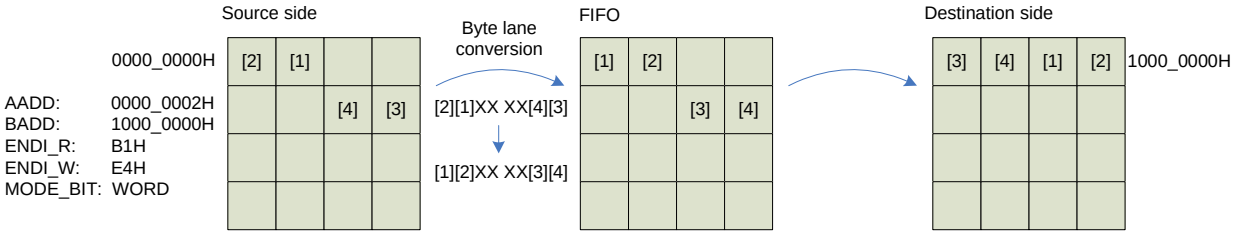



Figure 3-16 Byte Lane Conversion Example 3 (Source Address Is at Halfword Boundary, Read Side)

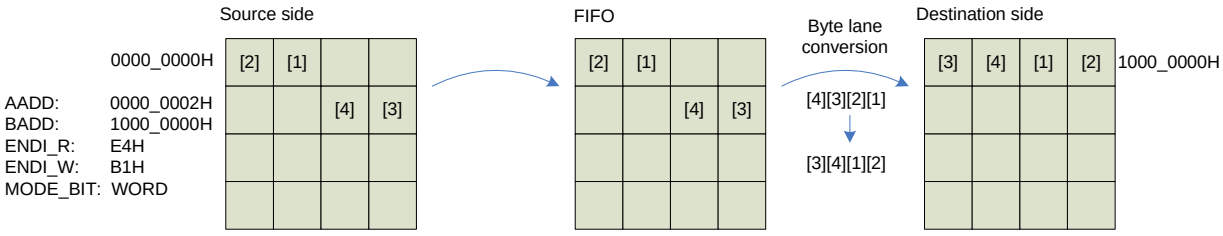



Figure 3-17 Byte Lane Conversion Example 4 (Source Address Is at Halfword Boundary, Write Side)

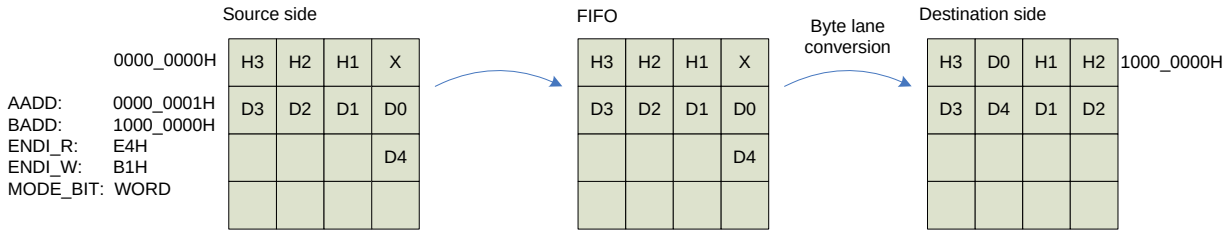



Figure 3-18 Byte Lane Conversion Example 5 (Exception 1, Write Side)

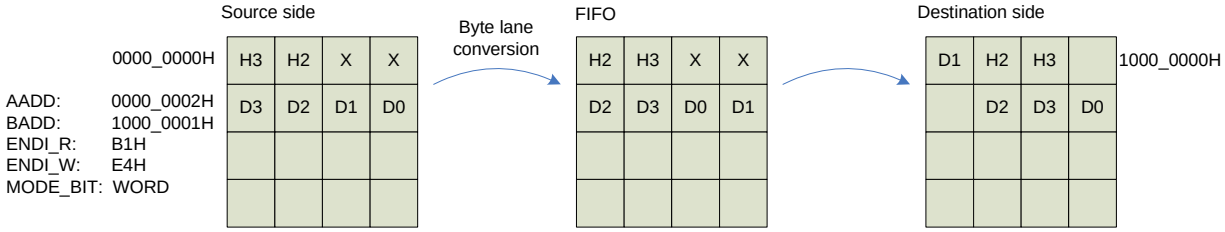



Figure 3-19 Byte Lane Conversion Example 6 (Exception 2, Read Side)

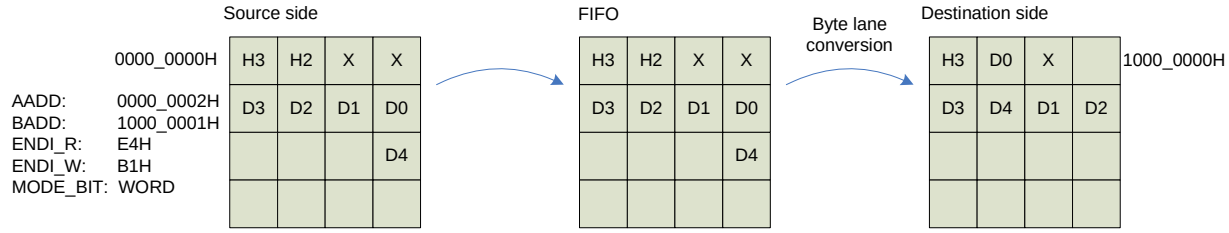



Figure 3-20 Byte Lane Conversion Example 7 (Prohibited Setting, Write Side)

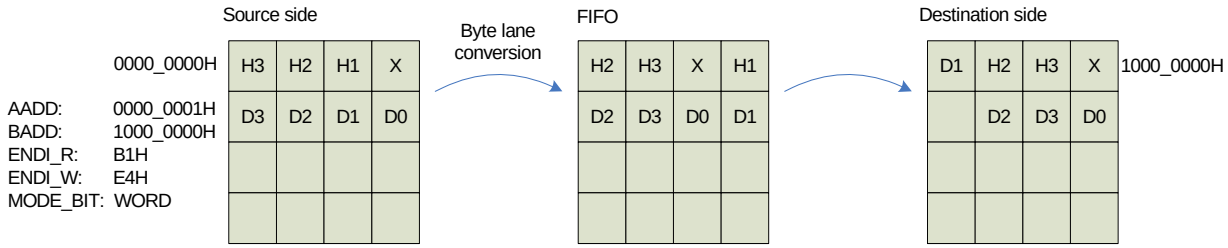



Figure 3-21 Byte Lane Conversion Example 8 (Prohibited Setting, Read Side)

**Remark**

AADD: Source address register (DMA_xxx_LCHx_AADD)

BADD: Destination address register (DMA_xxx_LCHx_BADD)

ENDI_R: xxx_LCHx_MODE_ENDI_R_xx bit of mode register (DMA_xxx_LCHx_MODE)

ENDI_W: xxx_LCHx_MODE_ENDI_W_xx bit of mode register (DMA_xxx_LCHx_MODE)

MODE_BIT: xxx_LCHx_MODE_BIT bit of mode register (DMA_xxx_LCHx_MODE)

# APPENDIX A DMAC Driver Function

## A.1 Function List

The following table shows the DMA driver interface functions:

**Table A-1 DMA Driver Function List**

| Class | Function Name | Function Detail |
|---|---|---|
| External function | em1_dma_init | Init dma device |
| | em1_dma_setup | Setup dma parameters |
| | em1_dma_start | Start dma transfer |
| | em1_dma_stop | Stop dma transfer |
| | em1_dma_get_status | Get dma current status |
| | em1_dma_clearInt | Clear dma interrupt |

## A.2 Global Variable Define

None

## A.3 Structure Define

**Table A-2 Structure Define**

| Structure Name | Detail |
|---|---|
| DMA_SETUP_ST | The structure of dma setup parameter |
| DMA_REG_ST | The structure of dma setting register |

### A.3.1 DMA_SETUP_ST

**Table A-3 Structure of DMA_SETUP_ST**

| Member | Detail |
|---|---|
| volatile uint channel | DMA channel |
| volatile uint src_add | Source address |
| volatile uint dst_add | Destination address |
| volatile uint block_size | Block size |
| volatile uint src_offset | Offset in source side |
| volatile uint dst_offset | Offset in destination side |
| volatile uint Asize_count | The block number of a loop when repeat mode on source side |
| volatile uint Bsize_count | The block number of a loop when repeat mode on destination side |
| volatile uint length | Transfer length |
| volatile uint time | Set the time allowed to elapse before times out |
| volatile uint endian | Set byte lane conversion |
| volatile ushort bit width | Specify the transfer bit width |
| volatile short swt_timer | Set whether to use the timer count |
| volatile short src_repeat | Set repeat mode on source side |
| volatile short dst_repeat | Set repeat mode on destination side |
| volatile short src_sign | Specify whether to add or subtract the offset on the source side |
| volatile short dst_sign | Specify whether to add or subtract the offset on the destination side |

### A.3.2 DMA_REG_ST

**Table A-4 Structure of DMA_REG_ST**

| Member | Detail |
|---|---|
| volatile uint AADD | Source address register |
| volatile uint BADD | Destination address register |
| volatile uint SIZE | Block size register |
| volatile uint AOFF | Offset register on source side |
| volatile uint BOFF | Offset register on destination side |
| volatile uint ASIZE_COUNT | Block count register on source side |
| volatile uint BSIZE_COUNT | Block count register on destination side |
| volatile uint LENG | Transfer length register |
| volatile uint MODE | Transfer mode register |
| volatile uint TIME | Register to set the time allowed to elapse before times out |

## A.4 Function Details

### A.4.1 DMA Init Function

**[Function Name]**

em1_dma_init

**[Format]**

void em1_dma_init(void);

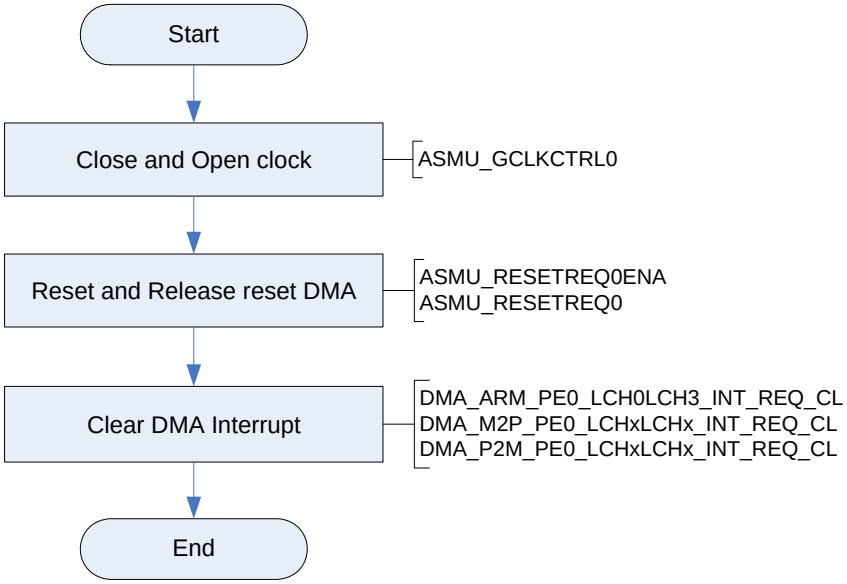
**[Argument]**

None

**[Function Return]**

None

**[Flow Chart]**




**[Note]**

None

### A.4.2 Setup DMA parameters

**[Function Name]**

em1_dma_setup

**[Format]**

DRV_RESULT em1_dma_setup(DMA_SETUP_ST * setup_st)

**[Argument]**

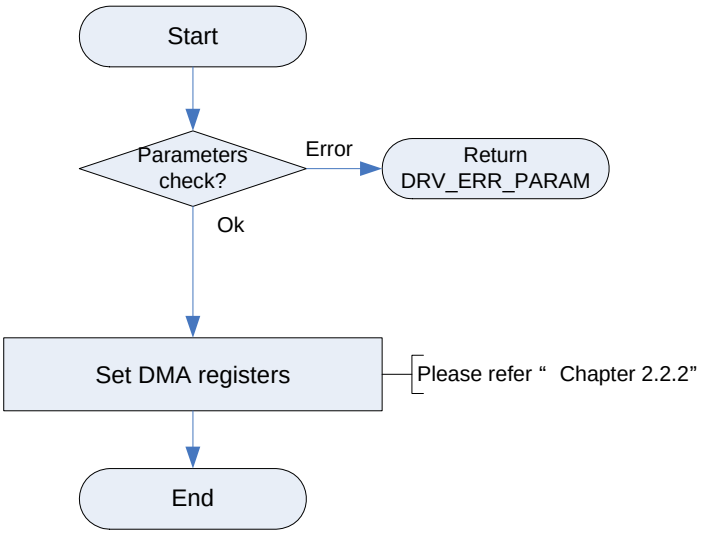
| Parameter | Type | I/O | Detail |
|---|---|---|---|
| setup_st | DMA_SETUP_ST * | I | The structure of dma setup parameters |

**[Function Return]**

DRV_OK

DRV_ERR_PARAM

**[Flow Chart]**




**[Note]**

None

### A.4.3 Start DMA transfer

**[Function Name]**

em1_dma_start

**[Format]**

DRV_RESULT em1_dma_start(uint channel);

**[Argument]**

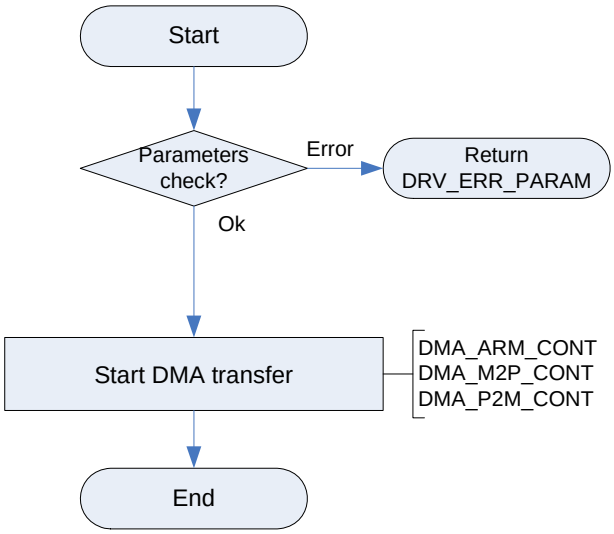
| Parameter | Type | I/O | Detail |
|---|---|---|---|
| channel | uint | I | The current used dma channel |

**[Function Return]**

DRV_OK

DRV_ERR_PARAM

**[Flow Chart]**




**[Note]**

None

### A.4.4 Stop DMA transfer

**[Function Name]**

em1_dma_stop

**[Format]**

DRV_RESULT em1_dma_stop(uint channel);

**[Argument]**

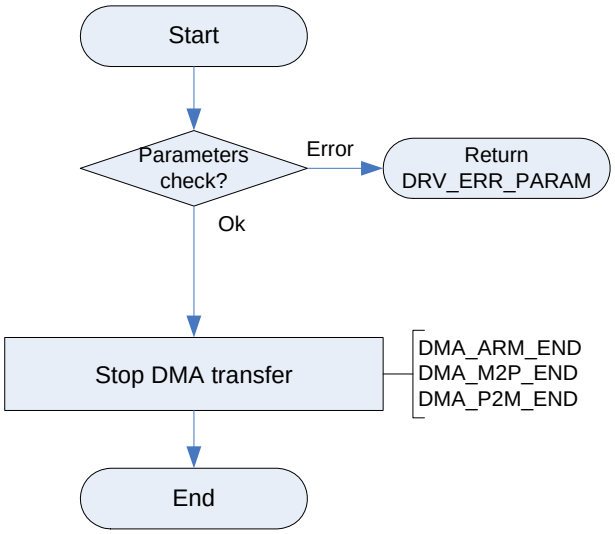
| Parameter | Type | I/O | Detail |
|---|---|---|---|
| channel | uint | I | The current used dma channel |

**[Function Return]**

DRV_OK

DRV_ERR_PARAM

**[Flow Chart]**




**[Note]**

None

### A.4.5 Get DMA status

**[Function Name]**

em1_dma_get_status

**[Format]**

DRV_RESULT em1_dma_get_status(uint channel);

**[Argument]**

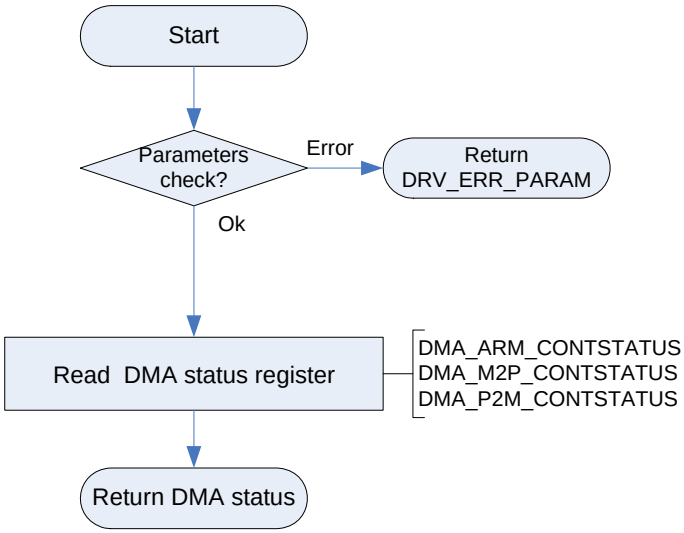
| Parameter | Type | I/O | Detail |
|---|---|---|---|
| channel | uint | I | The current used dma channel |

**[Function Return]**

DRV_ERR_PARAM

Others: the status of specified DMA channel

**[Flow Chart]**




**[Note]**

None

### A.4.6 Clear DMA interrupt

**[Function Name]**

em1_dma_clearInt

**[Format]**

DRV_RESULT em1_dma_clearInt(uint channel);

**[Argument]**

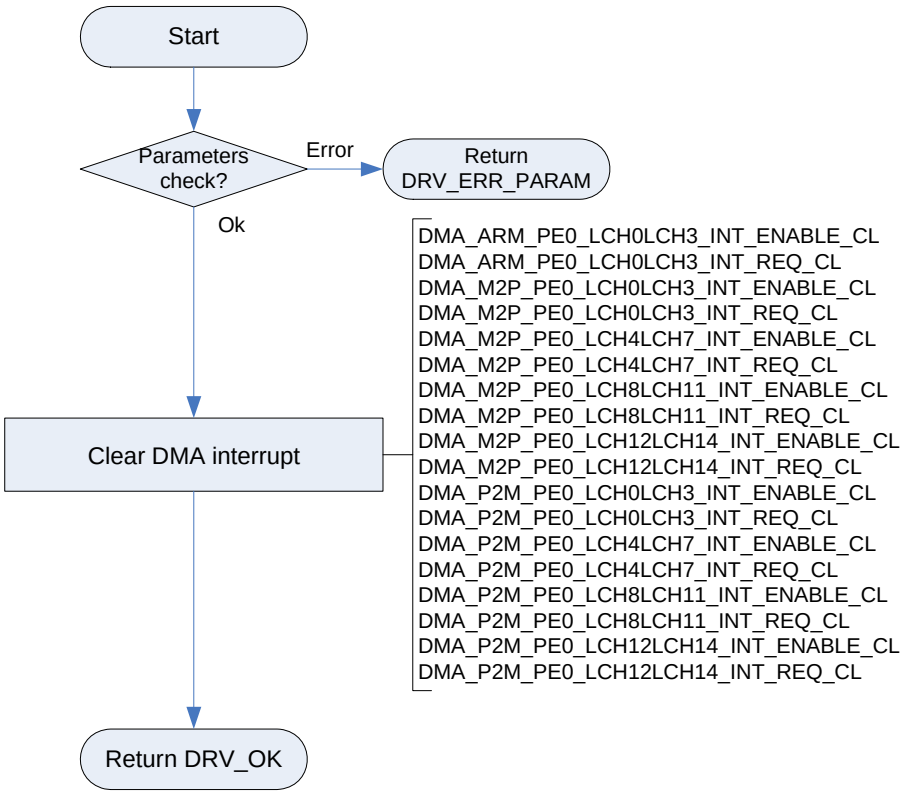
| Parameter | Type | I/O | Detail |
|---|---|---|---|
| channel | uint | I | The current used dma channel |

**[Function Return]**

DRV_OK

DRV_ERR_PARAM

**[Flow Chart]**




**[Note]**

None

# ANNEX Modification History

| Number | Modification Contents | Author | Date |
|---|---|---|---|
| Ver 1.00 | New version | | Aug, 4, 2009 |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |