32-bit RISC Microcontroller

TXZ Family

Reference Manual
High Speed DMA Controller
(HDMAC-A)

Revision 2.1

2019-02

TOSHIBA ELECTRONIC DEVICES & STORAGE CORPORATION

Contents

List of Figures

List of Tables

Preface

Related Documents

Document name
Clock Control and Operation Mode
Exception
Product Information

Conventions

- Numeric formats follow the rules as shown below:
  Hexadecimal: 0xABC
  Decimal: 123 or 0d123 – Only when it needs to be explicitly shown that they are decimal numbers.
  Binary: 0b111 – It is possible to omit the "0b" when the number of bit can be distinctly understood from a sentence.
- "_N" is added to the end of signal names to indicate low active signals.
- It is called "assert" that a signal moves to its active level, "deassert" to its inactive level.
- When two or more signal names are referred, they are described like as [m: n].
  Example: S[3: 0] shows four signal names S3, S2, S1 and S0 together.
- The characters surrounded by ***[ ]*** defines the register.
  Example: ***[ABCD]***
- "n" substitutes suffix number of two or more same kind of registers, fields, and bit names.
  Example: ***[XYZ1], [XYZ2], [XYZ3] → [XYZn]***
- "x" substitutes suffix number or character of units and channels in the Register List.
  In case of unit, "x" means A, B, and C …
  Example: ***[ADACR0], [ADBCR0], [ADCCR0] → [ADxCR0]***
  In case of channel, "x" means 0, 1, and 2…
  Example: ***[T32A0RUNA], [T32A1RUNA], [T32A2RUNA] →[T32AxRUNA]***
- The bit range of a register is written like as [m: n].
  Example: Bit[3: 0] expresses the range of bit 3 to 0.
- The configuration value of a register is expressed by either the hexadecimal number or the binary number.
  Example: ***[ABCD]***<EFG> = 0x01 (hexadecimal), ***[XYZn]***<VW> = 1 (binary)
- Word and Byte represent the following bit length.
  Byte: 8 bits
  Half word: 16 bits
  Word: 32 bits
  Double word: 64 bits
- Properties of each bit in a register are expressed as follows:
  R: Read only
  W: Write only
  R/W: Read and Write are possible
- Unless otherwise specified, register access supports only word access.
- The register defined as reserved must not be rewritten. Moreover, do not use the read value.
- The value read from the bit having default value of "-" is unknown.
- When a register containing both of writable bits and read-only bits is written, read-only bits should be written with their default value, In the cases that default is "-", follow the definition of each register.
- Reserved bits of the Write-only register should be written with their default value. In the cases that default is "-", follow the definition of each register.
- Do not use read-modified-write processing to the register of a definition which is different by writing and read out.

The Flash memory uses the Super Flash® technology under license from Silicon Storage Technology, Inc.
Super Flash® is registered trademark of Silicon Storage Technology, Inc.

All other company names, product names, and service names mentioned herein may be trademarks of their respective companies.

Terms and Abbreviations

Some of abbreviations used in this document are as follows:

DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
HDMAC	High speed Direct Memory Access Controller
LLI	Linked List Item
AHB	Advanced High-performance Bus

1. Outlines

The functions of HDMAC per unit are shown in the following table.

Table 1.1 HDMAC functions (per unit)

Function category	Function	Description
Transfer request	Peripheral function	Transfer request from a peripheral device (Single transfer request/Burst transfer request)
	Software	Transfer request by software (Single transfer request/Burst transfer request)
	Request channel	16 channels (Transfer request count by peripheral devices)
Transfer mode	Single transfer	Data transfer is executed once.
	Burst transfer	Data transfer is executed once or multiple times Burst size: 1, 4, 8, 16, 32, 64, 128, and 256 beats Continuous data transfer of the specified burst count is possible without releasing the bus using the lock transfer setting.
	Chain transfer	Continuous data transfer is possible using discontinuous addresses according to Linked List Item (LLI).
Transfer type	Transfer source -> Transfer destination	Peripheral device (Register) -> Memory Peripheral device (Register) -> Peripheral device (Register) Memory -> Peripheral device (Register) Memory -> Memory (start up by only software)
Transfer control	Transfer channel	2 channels (ch 0 and ch 1)
	Transfer address	Addresses of the transfer source and destination are set. Selection of the increment of the address or fixed address for either the transfer source or destination.
	Transfer data size	8 bits, 16 bits, or 32 bits The sizes of the transfer source and destination can be set independently.
	Priority	ch 0 > ch 1 (Fixed in the unit.) Unit A > Unit B (For the integrated units, refer to "Product Information".)
	FIFO	4 Words x 2 ch (1 Word = 32 bits)
Transfer count	Transfer count	4095 at maximum Infinite count transfers can be done using LLI.
Endian	Little endian	-
Interrupt	Transfer completion interrupt	Transfer completion interrupt (INTHDMACxTC) is generated when a transfer completes.
	Error interrupt	Error interrupt (INTHDMACxERR) is generated when a bus error or a memory protection error is detected during data transfer.

2. Configuration

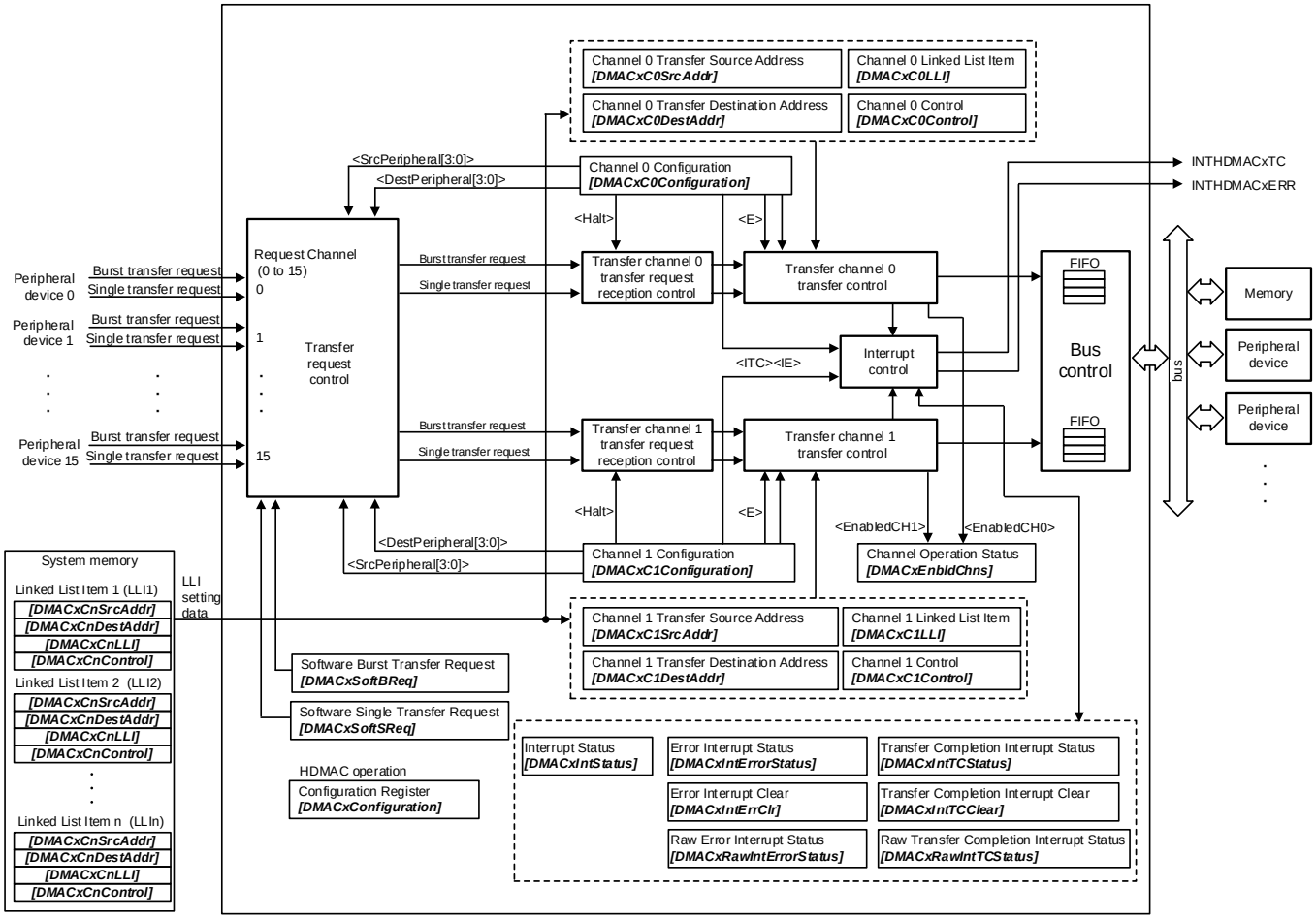

Figure 2.1 Block diagram of HDMAC (per unit)

Table 2.1 List of signals

No.	Symbol	Signal name	I/O	Reference manual
1	-	Burst transfer request (Request channel 0 to 15)	Input	Product Information
2	-	Single transfer request (Request channel 0 to 15)	Input	Product Information
3	INTHDMACxTC	Transfer completion interrupt	Output	Exception
4	INTHDMACxERR	Error interrupt	Output	Exception
5	-	LLI setting data	Input	This manual

3. Function and Operation

High speed DMA controller (HDMAC) controls the transfer operation using DMAC registers and Linked List Item (LLI). LLI is allocated in the memory. For the details of the data structure, refer to Section "3.8. Chain Transfer".

3.1. Clock Supply

When HDMAC is used, the corresponding clock enable bits should be set to "1" (Clock supply) in fsys supply stop register A (***[CGFSYSENA]***, ***[CGFSYSMENA]***), fsys supply stop register B (***[CGFSYSENB]***, ***[CGFSYSMENB]***), and fc supply stop register (***[CGFCEN]***). The corresponding registers and the bit locations depend on a product. Some products do not have all registers. For the details, refer to "Clock Control and Operation Mode" in Reference manual.

3.2. Operation setting of HDMAC

To set ***[DMACxConfiguration]***<E> to "1" enables HDMAC operation. Then, all registers except ***[DMACxConfiguration]*** can be accessed.

[DMACxConfiguration]<E> should be set to "0" to disable HDMAC operation, after it is checked that all transfer channels are invalid (no channels transfer any data) by reading ***[DMACxEnbldChns]***<EnabledCH1> <EnabledCH0>.

3.3. Transfer Type

HDMAC selects a transfer type from among 4 types, and transfers data.
The transfer type is set to ***[DMACxCnConfiguration]***<FlowCntrl[2:0]>.

Table 3.1 Transfer type

<table>
<tr><th>Transfer type</th><th>Transfer request source</th><th>Request</th><th>Description</th></tr>
<tr><td>Memory -> Memory (Note1)(Note2)</td><td>Software</td><td>-</td><td>The operation of the transfer channel is permitted and the transfer is started.</td></tr>
<tr><td>Memory -> Peripheral device (Note2)</td><td>Peripheral device (Transfer destination)</td><td>Burst transfer request</td><td>The burst size should be set to 1 for Word data transfer.</td></tr>
<tr><td>Peripheral device -> Memory</td><td>Peripheral device (Transfer source)</td><td>Burst transfer request
Single transfer request</td><td>When the transfer size is not an integral multiple of the burst size, both a burst transfer and a single transfer are used.
Transfer size ≥ Burst size: Burst transfer
Transfer size < Burst size: Single transfer</td></tr>
<tr><td rowspan="2">Peripheral device -> Peripheral device</td><td>Peripheral device (Transfer source)</td><td>Burst transfer request
Single transfer request</td><td rowspan="2"><table>
<tr><th rowspan="2">Transfer size</th><th colspan="2">Start-up request source</th></tr>
<tr><th>Transfer source</th><th>Transfer destination</th></tr>
<tr><td>Integral multiple of Burst size</td><td>Burst transfer request</td><td rowspan="3">Burst transfer request</td></tr>
<tr><td>No integral multiple of Burst size</td><td>Burst transfer request
Single transfer request</td></tr>
<tr><td>Single transfer</td><td>Single transfer request</td></tr>
</table></td></tr>
<tr><td>Peripheral device (Transfer destination)</td><td>Burst transfer request</td></tr>
</table>

Note1: When a lot of data is transferred with "Memory -> Memory" transfer type, it is recommended that a low priority transfer channel should be used. If a high priority transfer channel is used, another lower priority transfer channel cannot start until high priority transfer completes.
Note2: When accessing ROM data on the flash memory, access from the mirror area.

3.4. Transfer Channel

HDMAC can use 2 transfer channels (ch 0 and ch 1).

3.4.1. Enable of Transfer Channel

[DMACxCnConfiguration]<E> should be set to "1" in the corresponding channel register to enable the transfer operation of the channel. Once the transfer operation of the channel is enabled, HDMAC is in the transfer request wait state.
Before the operation is enabled, the registers of the channel (***[DMACxCnSrcAddr]***, ***[DMACxCnDestAddr]***, ***[DMACxCnLLI]***, ***[DMACxCnControl]***, and ***[DMACxCnConfiguration]***) should be set properly.

3.4.2. Disable of Transfer Channel

The operation of the transfer channel is disabled as follows.

1) When ***[DMACxCnConfiguration]***<E> is set to "0", the current transfer stops and the transfer channel operation is disabled. The data in FIFO is lost.

2) When the transfer completes, the transfer channel becomes automatically disabled.

3.4.3. Status of Transfer Channel

When the operation of a transfer channel is enabled, the corresponding bits are set to "1" in ***[DMACxEnbldChns]***<EnabledCH1><EnabledCH0>. When the operation completes, they are set to "0".
When the transfer completes, ***[DMACxCnConfiguration]***<E> becomes "0".

3.4.4. Priority Order of Transfer Channel

The priority order of the channels is ch 0 > ch 1. If the transfer requests of the two channels are issued simultaneously, ch 0 operation is done first.

3.5. Transfer Request

When a transfer request is received, the data transfer starts at the specified transfer channel. 16 transfer channels are available (the request channels are 0 to 15). Transfer request signals (a burst transfer request and a single transfer request) of each request channels are connected to a peripheral device. For the request channels and the connected peripheral device, refer to "Product Information" in Reference manual.

3.5.1. Generation of Transfer Request

1) Transfer request from a peripheral device
The transfer request is generated by a peripheral device. The connected line of the request channel defines the transfer request number.

2) Transfer request by software
When the corresponding bits of the request channels are set to "1" in ***[DMACxSoftBReq]*** and ***[DMACxSoftSReq]***, the transfer request is generated. The status of the generation of each request can be checked by reading the registers.

3.5.2. Transfer Request Mode

There are two modes in the transfer request.

1) Single transfer request
One data transfer is done. Then HDMAC waits for the next transfer request.

2) Burst transfer request
The transfer of a specified burst size is done. Then HDMAC waits for the next transfer request.

3.5.3. Transfer Request Reception Enable

The reception of the transfer request is enabled per transfer channel when ***[DMACxCnConfiguration]***<E> is set to "1" (the transfer channel is enabled) and ***[DMACxCnConfiguration]***<Halt> is set to "0".

3.5.4. Request Channel Setting for Each Transfer Channel

The request channels of the transfer source and the transfer destination for each transfer channel should be set to ***[DMACxCnConfiguration]***<SrcPeripheral[3:0]> and <DestPeripheral[3:0]>, respectively. When the transfer request of the specified request channel is received, the transfer operation starts. When both the transfer source and the transfer destination are memories, these settings are ignored.

Example:
[DMACxC0Configuration]<FlowCntrl[2:0]> = 010 (Peripheral device -> Memory)
When ***[DMACxC0Configuration]***<SrcPeripheral[3:0]> = 0011(request channel 3) and <DestPeripheral[3:0]> = 0100(request channel 4) are set, the transfer operation starts when the peripheral device connected to the line of the request channel 3 generates a transfer request.

3.6. Transfer Operation

When a transfer request is generated by software or a peripheral device, the data transfer for each transfer channel is executed according to the transfer request type (a single transfer request or a burst transfer request). The settings of the transfer operation should be done per transfer channel..

3.6.1. Transfer setting for each transfer channel

- Transfer source address
  The source peripheral device or memory address should be set to ***[DMACxCnSrcAddr]***<SrcAddr[31:0]>.
  [DMACxCnControl]<SI> can select a fixed transfer source address or an increment of the address per transfer.

  For the lowest address that can be set by the data bit width of the transfer source, see Table 3.2.

Table 3.2 Source bit width and address setting

Bit width of Transfer source *[DMACxCnControl]*<Swidth[2:0]>	Lowest address setting
000: Byte (8 bits)	0x0, 0x1, 0x2, 0x3, 0x4 ▪ ▪ ▪
001: Halfword (16 bits)	Multiple of 2 (0x0, 0x2, 0x4, 0x6, 0x8, 0xA, 0xC, 0xE ▪ ▪ ▪) should be set.
010: Word (32 bits)	Multiple of 4 (0x0, 0x4, 0x8, 0xC ▪ ▪ ▪) should be set.

- Transfer destination address
  The destination peripheral device or memory address should be set to ***[DMACxCnDestAddr]***
  <DestAddr[31:0]>.
  [DMACxCnControl]<DI> can select a fixed transfer destination address or an increment of the address per transfer.

  For the lowest address that can be set by the data bit width of the transfer destination, see Table 3.3.

Table 3.3 Transfer destination bit width and address setting

Bit width of Transfer destination *[DMACxCnControl]*<Dwidth[2:0]>	Lowest address setting
000: Byte (8 bits)	0x0, 0x1, 0x2, 0x3, 0x4 ▪ ▪ ▪
001: Halfword (16 bits)	Multiple of 2 (0x0, 0x2, 0x4, 0x6, 0x8, 0xA, 0xC, 0xE ▪ ▪ ▪) should be set.
010: Word (32 bits)	Multiple of 4 (0x0, 0x4, 0x8, 0xC ▪ ▪ ▪) should be set.

- Linked list item (LLI) address
  When the chain transfer function is used, LLI address should be set to ***[DMACxCnLLI]***<LLI[31:0]>. When the chain transfer function is not used, the register should be set to "0". For the details of the chain transfer function, refer to Section "3.8 Chain Transfer".

- Bit width
  The data bit widths of the transfer destination and the transfer source should be set to ***[DMACxCnControl]***
  <Dwidth[2:0]> and <Swidth[2:0]>, respectively. Byte (8 bits), Halfword (16 bits), or Word (32 bits) can be set.
  The destination width can be different from the source width. The following formula should be valid: (Data bit width of the transfer source) × (Transfer count) = (Data bit width of the transfer destination) × Integer.

- Burst size
  The burst sizes of the transfer destination and the transfer source should be set to ***[DMACxCnControl]*** <DBSize[2:0]> and <SBSize[2:0]>, respectively. The burst size is the amount of data which is transferred at the generation of a burst transfer request.

- Total transfer count
  The total transfer count should be set to ***[DMACxCnControl]***<TransferSize[11:0]> with the unit of the data bit width of the transfer source.
  The total transfer amount is unchanged regardless of the burst size as long as the data bit width of the transfer source or the total transfer count are not modified.

3.6.2. Transfer setting example

(Example 1)
Source bit width: 8 bits, destination bit width: 32 bits, total transfer count: 25 times
8 bits × 25 times = 200 bits (25 bytes)
N = 200 / 32 = 6.25 words
Since N is not an integer, this setting cannot be done.

(Example 2)
Transfer source bit width: 32 bits, transfer destination bit width: 16 bits, total transfer count: 13 times
32 bits × 13 times = 416 bits (13 words)
N = 416 / 16 = 26 Half word
Since N is an integer, this setting is possible.

3.6.3. Transfer Operation Flow

The outlines of the transfer operation flow of each transfer type are shown in this section. The interrupt clear and the transfer settings of each channel should be done before the operation in the following flows. No interrupt generation after the transfer completion can be also set.

Preparation before the flow execution:

- Select the channel considering the priority (ch 0 > ch 1)
- Clear the interrupt of the transfer channel to be used (***[DMACxIntTCClear]*** and ***[DMACxIntErrClr]***)
- Set the transfer settings of each transfer channel ("3.6.1. Transfer setting for each transfer channel").

1) Memory to Memory

When the transfer channel operation is enabled, the data transfer starts.

1. Set "1" to ***[DMACxCnConfiguration]***<E> to enable the transfer channel operation.
2. The data transfer starts. The transfers of the set transfer count are executed.
3. When the transfer count becomes "0", the transfer completes. Then, Transfer completion interrupt is generated.
4. Unless ***[DMACxCnLLI]*** is "0", the next LLI is reloaded and the data transfer continues.
5. When ***[DMACxCnLLI]*** is "0", HDMAC waits for the next transfer request.

2) Memory to Peripheral Device

When the transfer channel operation is enabled, the data transfer starts from a memory to FIFO in HDMAC. And when a transfer request is issued by the transfer destination, the data is transferred to the peripheral device.

1. Set "1" to ***[DMACxCnConfiguration]***<E> to enable the transfer channel operation.
2. Wait for a transfer request from the peripheral device (Transfer destination).
3. When the transfer request is issued, the transfer starts. The transfers of the set transfer count are executed.
4. When the transfer count becomes "0", the transfer completes. Then, Transfer completion interrupt is generated.
5. Unless ***[DMACxCnLLI]*** is "0", the next LLI is reloaded and the data transfer continues.
6. When ***[DMACxCnLLI]*** is "0", HDMAC waits for the next transfer request.

3) Peripheral Device to Memory

When the transfer channel operation is enabled and a transfer request is issued by the peripheral device of the transfer source, the data transfer starts.

1. Set "1" to ***[DMACxCnConfiguration]***<E> to enable the transfer channel operation.
2. Wait for a transfer request from the peripheral device (Transfer source).
3. When the transfer request is issued, the transfer starts. The transfers of the set transfer count are executed.
4. When the transfer count becomes "0", the transfer completes. Then, Transfer completion interrupt is generated.
5. Unless ***[DMACxCnLLI]*** is "0", the next LLI is reloaded and the data transfer continues.
6. When ***[DMACxCnLLI]*** is "0", HDMAC waits for the next transfer request.

4) Peripheral Device to Peripheral Device

When the transfer channel operation is enabled and both the transfer source and the transfer destination issue the transfer requests, the data transfer starts.

1. Set "1" to ***[DMACxCnConfiguration]***<E> to enable the transfer channel operation.
2. Wait for a transfer request of the transfer source.
3. When the transfer request is issued, the transfer starts. The transfers of the set transfer count are executed.
4. When the transfer request of the transfer destination is active and FIFO is not empty, the data is transferred to the transfer destination.
5. When the transfer count becomes "0", the transfer completes. Then, Transfer completion interrupt is generated.
6. Unless ***[DMACxCnLLI]*** is "0", the next LLI is reloaded and the data transfer continues.
7. When ***[DMACxCnLLI]*** is "0", HDMAC waits for the next transfer request.

3.6.4. Lock Control

When ***[DMACxConfiguration]***<Lock> is set to "1", the bus is not released during a burst transfer. The data transfer can continue until the data of the set burst count is transferred.

3.7. Interrupt

HDMAC has Transfer completion interrupt and Error interrupt.

3.7.1. Transfer Completion Interrupt

Transfer completion interrupt (INTHDMACxTC) is generated when the data transfer of the specified transfer count completes. The enable or disable of Transfer completion interrupt should be set to ***[DMACxCnControl]***<I>.
The mask or non-mask of the generated interrupt (to notify CPU of it, or not) should be set to ***[DMACxCnConfiguration]***<ITC>.

3.7.2. Error Interrupt

Error interrupt (INTHDMACxERR) is generated when a bus error or a memory protection error is detected during data transfer. The mask or non-mask of the generated interrupt (to notify CPU of it, or not) should be set to ***[DMACxCnConfiguration]***<IE>.

3.7.3. Interrupt Status

The interrupt status can be checked by the following registers per channel. Both before mask status and after mask status can be checked. (The mask is set to ***[DMACxCnConfiguration]***<ITC><IE>.)

Table 3.4 Interrupt status registers

Interrupt	Mask setting by *[DMACxCnConfiguration]*<ITC><IE>		
	Before mask	After mask	
Transfer completion interrupt	***[DMACxRawIntTCStatus]***	***[DMACxIntStatus]***	***[DMACxIntTCStatus]***
Error interrupt	***[DMACxRawIntErrorStatus]***		***[DMACxIntErrorStatus]***

3.7.4. Clear of Interrupt

The generated interrupt is not cleared automatically. The interrupt should be cleared in an interrupt routine program.

[DMACxIntTCClear]<IntTCClear1><IntTCClear0> should be set to "1" to clear Transfer completion interrupt. And each status of the interrupt is also cleared.

[DMACxIntErrClr]<IntErrClr1><IntErrClr0> should be set to "1" to clear Error interrupt. And each status of the interrupt is also cleared.

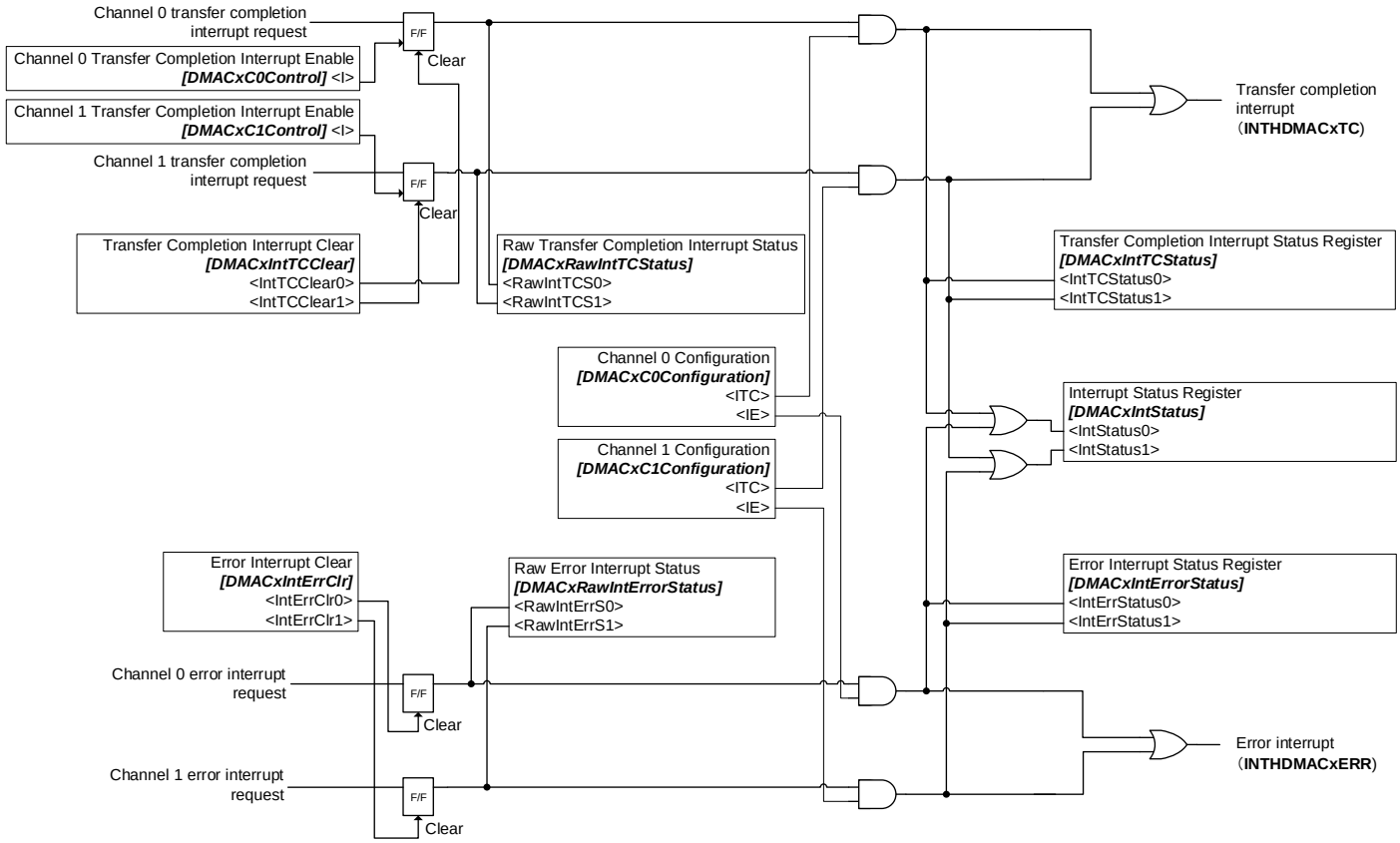

Figure 3.1 Configuration of the interrupt circuit

3.8. Chain Transfer

HDMAC has a chain transfer function to transfer data of discontinuous addresses.

The chain transfer is performed by setting transfer parameters (a transfer source address, a transfer destination address, a transfer count, a transfer bit width, and others) per block in Linked list (LLI). One LLI can control the transfer of only one block. But next LLI can be specified in the current LLI, which enables continuous transfer (data transfer of discontinuous addresses) of blocks with different transfer parameter values.

3.8.1. Linked List Item (LLI) Setting

LLI of one block consists of the following registers. The values are stored in the registers in a memory with continuous addresses.

1. ***[DMACxCnSrcAddr]*** Transfer source address
2. ***[DMACxCnDestAddr]*** Transfer destination address
3. ***[DMACxCnLLI]*** Start address of the memory where the next LLI is stored.
4. ***[DMACxCnControl]*** Bit width, Burst size, Total transfer count, and others

The chain transfer is enabled by specifying the start address of the first LLI in ***[DMACxCnLLI]***. When a transfer request is received, the first block is transferred according to the settings in DMA setting register, and the next block is transferred according to the settings in the LLI which is assigned by ***[DMACxCnLLI]***.

The blocks which have discontinuous addresses can be transferred continuously. As many LLIs as the transfer count should be prepared. Each LLI specifies the start address of the next LLI. So, when one block transfer completes, the next LLI is loaded and the transfer continues. In the LLI of the last block, the next start address in ***[DMACxCnLLI]*** should be set to "0".

3.8.2. Interrupt Operation

When transfer completion interrupt is generated only at the last block in a chain transfer, ***[DMACxCnControl]***<I>=0 and ***[DMACxCnConfiguration]***<ITC>=1 should be set to start the data transfer and ***[DMACxCnControl]***<I>=1 should be set in the LLI of the last block.

In order to clear Transfer completion interrupt, ***[DMACxIntTCClear]*** should be controlled.

3.8.3. Example of Chain Transfer

An example of the chain transfer is shown as follows.

1. The first data transfer is executed according to the settings in DMA registers.
2. The next transfer or after is executed according to the settings in the LLI which is assigned by ***[DMACxCnLLI]***.
3. In the LLI of the last transfer, the value in ***[DMACxCnLLI]*** should be "0".

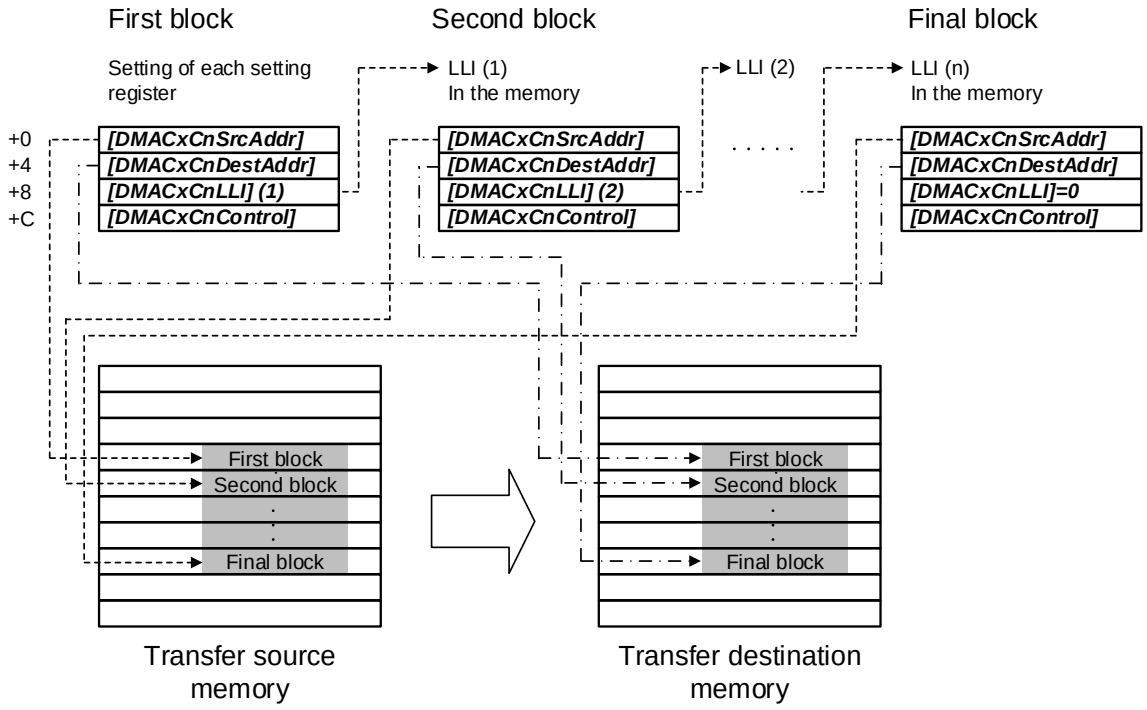

Figure 3.2 Chain transfer

4. Registers

4.1. List of Registers

The control registers of HDMAC and their addresses are shown as follows:

Peripheral function		Channel/Unit	Base address (Base)
			TYPE1
High speed DMAC	HDMAC	Unit A	0x40000000
		Unit B	0x40001000

Note: The Channel/Unit and Base address type are different by products. Please refer to "Product Information" of the reference manual for the details.

Register name		Base address (Base+)
Interrupt Status Register	***[DMACxIntStatus]***	0x0000
Transfer Completion Interrupt Status Register	***[DMACxIntTCStatus]***	0x0004
Transfer Completion Interrupt Clear Register	***[DMACxIntTCClear]***	0x0008
Error Interrupt Status Register	***[DMACxIntErrorStatus]***	0x000C
Error Interrupt Clear Register	***[DMACxIntErrClr]***	0x0010
Raw Transfer Completion Interrupt Status Register	***[DMACxRawIntTCStatus]***	0x0014
Raw Error Interrupt Status Register	***[DMACxRawIntErrorStatus]***	0x0018
Channel Enable Register	***[DMACxEnbldChns]***	0x001C
Software Burst Transfer Request Register	***[DMACxSoftBReq]***	0x0020
Software Single Transfer Request Register	***[DMACxSoftSReq]***	0x0024
Reserved.	-	0x0028
Reserved.	-	0x002C
Configuration Register	***[DMACxConfiguration]***	0x0030
Channel 0 Transfer Source Address Register	***[DMACxC0SrcAddr]***	0x0100
Channel 0 Transfer Destination Address Register	***[DMACxC0DestAddr]***	0x0104
Channel 0 Linked List Item Register	***[DMACxC0LLI]***	0x0108
Channel 0 Control Register	***[DMACxC0Control]***	0x010C
Channel 0 Configuration Register	***[DMACxC0Configuration]***	0x0110
Channel 1 Transfer Source Address Register	***[DMACxC1SrcAddr]***	0x0120
Channel 1 Transfer Destination Address Register	***[DMACxC1DestAddr]***	0x0124
Channel 1 Linked List Item Register	***[DMACxC1LLI]***	0x0128
Channel 1 Control Register	***[DMACxC1Control]***	0x012C
Channel 1 Configuration Register	***[DMACxC1Configuration]***	0x0130

Note1: All registers can be accessed only with Word unit (32 bits).
Note2: "Reserved" address should not be accessed.
Note3: When a register prepared per channel is written and a register which is not prepared per channel is read, the read should be done after one cycle wait or more, or done twice.

4.2. Detail of Register

4.2.1. *[DMACxIntStatus]* (Interrupt Status Register)

Bit	Bit Symbol	After Reset	Type	Description
31:2	-	Undefined.	R	Read as "Undefined".
1	IntStatus1	0	R	Transfer channel 1 interrupt status interrupt status 0: No interrupt is generated. 1: An interrupt is generated. The status of Transfer completion interrupt or Error interrupt in after going through mask setting is shown. (Note1)
0	IntStatus0	0	R	Transfer channel 0 interrupt status 0: No interrupt is generated. 1: An interrupt is generated. The status of Transfer completion interrupt or Error interrupt in after going through mask setting is shown. (Note1)

Note1: In order to know which interrupt is generated, ***[DMACxIntTCStatus]*** and ***[DMACxIntErrorStatus]*** should be checked.

4.2.2. *[DMACxIntTCStatus]* (Transfer Completion Interrupt Status Register)

Bit	Bit Symbol	After Reset	Type	Description
31:2	-	Undefined.	R	Read as "Undefined".
1	IntTCStatus1	0	R	Transfer channel 1 transfer completion interrupt status 0: No interrupt is generated. 1: Interrupt is generated. The status of Transfer completion interrupt in after going through mask setting is shown.
0	IntTCStatus0	0	R	Transfer channel 0 transfer completion interrupt status 0: No interrupt is generated. 1: Interrupt is generated. The status of Transfer completion interrupt in after going through mask setting is shown.

4.2.3. *[DMACxIntTCClear]* (Transfer Completion Interrupt Clear Register)

Bit	Bit Symbol	After Reset	Type	Description
31:2	-	0	W	Write as "0".
1	IntTCClear1	0	W	Transfer channel 1 transfer completion interrupt clear 0: Invalid. 1: Interrupt clear When "1" is written, the interrupt and the status are cleared.
0	IntTCClear0	0	W	Transfer channel 0 transfer completion interrupt clear 0: Invalid. 1: Interrupt clear When "1" is written, the interrupt and the status are cleared.

4.2.4. *[DMACxIntErrorStatus]* (Error Interrupt Status Register)

Bit	Bit Symbol	After Reset	Type	Description
31:2	-	Undefined.	R	Read as "Undefined".
1	IntErrStatus1	0	R	Transfer channel 1 error interrupt status 0: No interrupt is generated. 1: Interrupt is generated. The status of Error interrupt in after going through mask setting is shown.
0	IntErrStatus0	0	R	Transfer channel 0 error interrupt status 0: No interrupt is generated. 1: Interrupt is generated. The status of Error interrupt in after going through mask setting is shown.

4.2.5. *[DMACxIntErrClr]* (Error Interrupt Clear Register)

Bit	Bit Symbol	After Reset	Type	Description
31:2	-	0	W	Write as "0".
1	IntErrClr1	0	W	Transfer channel 1 error interrupt clear 0: Invalid. 1: Interrupt clear When "1" is written, the interrupt and the status are cleared.
0	IntErrClr0	0	W	Transfer channel 0 error interrupt clear 0: Invalid. 1: Interrupt clear When "1" is written, the interrupt and the status are cleared.

4.2.6. *[DMACxRawIntTCStatus]* (Raw Transfer Completion Interrupt Status Register)

Bit	Bit Symbol	After Reset	Type	Description
31:2	-	Undefined.	R	Read as "Undefined".
1	RawIntTCS1	0	R	Transfer channel 1 transfer completion interrupt status 0: No interrupt is generated. 1: Interrupt is generated. The status of Transfer completion interrupt of before going through mask setting is shown.
0	RawIntTCS0	0	R	Transfer channel 0 transfer completion interrupt status 0: No interrupt is generated. 1: Interrupt is generated. The status of Transfer completion interrupt of before going through mask setting is shown.

4.2.7. *[DMACxRawIntErrorStatus]* (Raw Error Interrupt Status Register)

Bit	Bit Symbol	After Reset	Type	Description
31:2	-	Undefined.	R	Read as "Undefined".
1	RawIntErrS1	0	R	Transfer channel 1 error interrupt status 0: No interrupt is generated. 1: Interrupt is generated. The status of Error interrupt of before going through mask setting is shown.
0	RawIntErrS0	0	R	Transfer channel 0 error interrupt status 0: No interrupt is generated. 1: Interrupt is generated. The status of Error interrupt of before going through mask setting is shown.

4.2.8. *[DMACxEnbldChns]* (Channel Enable Register)

Bit	Bit Symbol	After Reset	Type	Description
31:2	-	Undefined.	R	Read as "Undefined".
1	EnabledCH1	0	R	Transfer channel 1 transfer enable status 0: Transfer completion 1: Transfer enable When all data transfers set in ***[DMACxCnControl]*** completes, this bit becomes "0".
0	EnabledCH0	0	R	Transfer channel 0 transfer enable status 0: Transfer completion 1: Transfer enable When all data transfers set in ***[DMACxCnControl]*** completes, this bit becomes "0".

4.2.9. *[DMACxSoftBReq]* (Software Burst Transfer Request Register)

Bit	Bit Symbol	After Reset	Type	Description
31:16	-	Undefined.	W	Write as "0".
15	SoftBReq15	0	R/W	[Read] Burst transfer status per request number 0: Burst transfer completion 1: Burst transfer execution [Write] Burst transfer request by software 0: Invalid 1: Burst transfer request should be generated.
14	SoftBReq14	0		
13	SoftBReq13	0		
12	SoftBReq12	0		
11	SoftBReq11	0		
10	SoftBReq10	0		
9	SoftBReq9	0		
8	SoftBReq8	0		
7	SoftBReq7	0		
6	SoftBReq6	0		
5	SoftBReq5	0		
4	SoftBReq4	0		
3	SoftBReq3	0		
2	SoftBReq2	0		
1	SoftBReq1	0		
0	SoftBReq0	0		

Note1: The transfer requests by software and a peripheral device should not be generated simultaneously.
Note2: Refer to "Product Information" in the reference manual for the connection between transfer request and request channel.
Note3: Write "0" to the bit corresponding to the request channel without burst transfer request.

4.2.10. *[DMACxSoftSReq]* (Software Single Transfer Request Register)

Bit	Bit Symbol	After Reset	Type	Description
31:16	-	Undefined.	W	Write as "0".
15	SoftSReq15	0	R/W	[Read] Single transfer status per request number 0: Single transfer completion 1: Single transfer execution [Write] Single transfer request by software 0: Invalid. 1: Single transfer request should be generated.
14	SoftSReq14	0		
13	SoftSReq13	0		
12	SoftSReq12	0		
11	SoftSReq11	0		
10	SoftSReq10	0		
9	SoftSReq9	0		
8	SoftSReq8	0		
7	SoftSReq7	0		
6	SoftSReq6	0		
5	SoftSReq5	0		
4	SoftSReq4	0		
3	SoftSReq3	0		
2	SoftSReq2	0		
1	SoftSReq1	0		
0	SoftSReq0	0		

Note1: The transfer requests by software and a peripheral device should not be generated simultaneously.
Note2: Refer to "Product Information" in the reference manual for the connection between transfer request and request channel.
Note3: Write "0" to the bit corresponding to the request channel without single transfer request.

4.2.11. *[DMACxConfiguration]* (Configuration Register)

Bit	Bit Symbol	After Reset	Type	Description
31:2	-	Undefined.	W	Write as "0".
1	-	0	R/W	Write as "0".
0	E	0	R/W	HDMAC operation 0: Stop 1: Operating When HDMAC is used, this bit should be set to "1".

4.2.12. *[DMACxCnSrcAddr]* (Channel n Transfer Source Address Register) (n=0,1)

Bit	Bit Symbol	After Reset	Type	Description
31:0	SrcAddr[31:0]	0x00000000	R/W	Transfer source address setting Transfer source address is assigned.

Note1: When the channel n is enabled (***[DMACxCnConfiguration]***<E> = 1), <SrcAddr[31:0]> is updated. <SrcAddr[31:0]> should be set before the channel n is enabled (while the channel n is disabled: ***[DMACxCnConfiguration]***<E> = 0), and it should not be changed during data transfer.

Note2: The value in <SrcAddr[31:0]> changes one after another during transfer operation. The read value changes similarly.

4.2.13. *[DMACxCnDestAddr]* (Channel n Transfer Destination Address Register) (n=0,1)

Bit	Bit Symbol	After Reset	Type	Description
31:0	DestAddr[31:0]	0x00000000	R/W	Transfer destination address setting Transfer destination address is assigned.

Note1: <DestAddr[31:0]> should be set before the channel n is enabled (while the channel n is disabled: ***[DMACxCnConfiguration]***<E> = 0), and it should not be changed during data transfer.

4.2.14. *[DMACxCnLLI]* (Channel n Linked List Item Register) (n=0,1)

Bit	Bit Symbol	After Reset	Type	Description
31:2	LLI[31:2]	0x00000000	R/W	Start address of Linked list (LLI). <LLI>: The next LLI address is assigned. (Note1) (Note2) (Note3) This field should be set when the chain transfer function uses LLI's. <LLI> = 0x00000000 shows the block is the last one.
1:0	LLI[1:0]	Undefined.	W	"00" should be written.

Note1: The address should be assigned in Word alignment. (The values of the bit 1 and bit 0 in the address should be "00".)
Note2: <LLI[31:0]> should not be changed during the enable of the channel operation.
Note3: For the details of LLI, refer to "3.8. Chain Transfer".

4.2.15. *[DMACxCnControl]* (Channel n Control Register) (n=0,1)

Bit	Bit Symbol	After Reset	Type	Description
31	I	0	R/W	Transfer completion interrupt enable (Note1) 0: Transfer completion interrupt disable 1: Transfer completion interrupt enable Transfer completion interrupt is enabled by ***[DMACxCnConfiguration]***<ITC> = 1 and <I> = 1.
30:28	-	Undefined.	W	Write as "0".
27	DI	0	R/W	Increment of the transfer destination address 0: Fixed address 1: Increment When this bit is set to "1", the transfer destination address increments per transfer.
26	SI	0	R/W	Increment of the transfer source address 0: Fixed address 1: Increment When this bit is set to "1", the transfer source address increments per transfer.
25:24	-	Undefined.	W	Write as "0".
23:21	Dwidth[2:0]	000	R/W	Bit width of the transfer destination (Note2) 000: Byte (8 bits) 001: Halfword (16 bits) 010: Word (32 bits) 011 to 111: Reserved.
20:18	Swidth[2:0]	000	R/W	Bit width of the transfer source (Note2) 000: Byte (8 bits) 001: Halfword (16 bits) 010: Word (32 bits) 011 to 111: Reserved.
17:15	DBSize[2:0]	000	R/W	Burst size of the transfer destination (Note3) (Note4) (Note5) 000: 1 beat 100: 32 beats 001: 4 beats 101: 64 beats 010: 8 beats 110: 128 beats 011: 16 beats 111: 256 beats
14:12	SBSize[2:0]	000	R/W	Burst size of the transfer source (Note3) (Note4) (Note5) 000: 1 beat 100: 32 beats 001: 4 beats 101: 64 beats 010: 8 beats 110: 128 beats 011: 16 beats 111: 256 beats
11:0	TransferSize[11:0]	0x000	R/W	Total transfer count (Note6) (Note7) This field should be set to the total count of the data transfer with the bit width unit of the transfer source.

Note1: Transfer completion interrupt can be generated only at the last transfer completion by the setting of <I> = 1 in LLI of the final block in the chain transfer. When the interrupt is generated in the normal transfer (the chain transfer is not used), both ***[DMACxCnConfiguration]***<ITC> and <I> should be set to "1".

Note2: The settings of <Dwidth[2:0]> and <Swidth[2:0]> should be set to satisfy the following formula.

Source bit width × total transfer count = transfer destination bit width × N (N: integer)

N must be an integer.
If the source bit width is smaller than the destination bit width, care must be taken in setting the total transfer count.

Note3: The burst size is the data amount which is transferred by one burst transfer request. Unless the bit width of the transfer source or the total transfer count is changed, the total data amount of transfer is the same regardless of the burst size.

Note4: <DBSize[2:0]> and <SBSize[2:0]> are the data amounts in the burst transfer.
These values are used for the peripheral device which has storage of multiple data such as an FIFO buffer.

Note5: The burst sizes of <DBSize> and <SBSize> have nothing to do with HBURST in AHB bus.

Note6: The total transfer count has the unit of the bit width of the transfer source. (<Swidth> = 000 (8 bits – Byte unit), <Swidth> = 001 (16 bits – Halfword unit), and <Swidth> = 010 (32 bits – Word unit)

Note7: <TransferSize[11:0]> decrements to "0" during data transfer. When this field is read during data transfer, the remaining transfer count is shown. When, it is read after transfer completion, "0" returns.

4.2.16. *[DMACxCnConfiguration]* (Channel n Configuration Register) (n=0,1)

Bit	Bit Symbol	After Reset	Type	Description
31:19	-	Undefined.	W	Write as "0".
18	Halt	0	R/W	Reception control of the transfer channel n transfer request (Note1) 0: Transfer request reception enable 1: Reserved
17	Active	0	R	Data existence in FIFO of the transfer channel n 0: Data exists in FIFO. 1: No data exists in FIFO.
16	Lock	0	R/W	Lock transfer setting (undivided transfer) 0: Lock transfer disable 1: Lock transfer enable When the lock transfer is enabled, the data of the specified burst count is transferred continuously without releasing the bus.
15	ITC	0	R/W	The mask setting of Transfer completion interrupt 0: Transfer completion interrupt is masked. 1: Transfer completion interrupt is not masked. (Interrupt enable) When <ITC> = 1 and ***[DMACxCnControl]***<I> = 1 are set, Transfer completion interrupt is notified to CPU.
14	IE	0	R/W	The mask setting of Error interrupt 0: Error interrupt is masked. 1: Error interrupt is not masked. (Interrupt enable)
13:11	FlowCntrl[2:0]	000	R/W	Transfer type setting 000: Memory to Memory (Note2) 001: Memory to Peripheral device 010: Peripheral device to Memory 011: Peripheral device to Peripheral device 100 to 111: Reserved.
10	-	Undefined.	W	Write as "0".
9:6	DestPeripheral[3:0]	0000	R/W	Request channel of the transfer destination 0000 to 1111: Request channel of the peripheral device of the transfer destination (0 to 15) (Note3) When the transfer destination is a memory, this setting is ignored.
5	-	Undefined.	W	Write as "0".
4:1	SrcPeripheral[3:0]	0000	R/W	Request channel of the transfer source 0000 to 1111: Request channel of the peripheral device of the transfer source (0 to 15) (Note3) When the transfer source is a memory, this setting is ignored.
0	E	0	R/W	Transfer channel n operation (Note2) 0: transfer channel n operation disable (Note4) 1: transfer channel n operation enable Transfer channel n operation disable or enable is set. This bit becomes "0" when the data transfer of the total transfer count completes.

Note1: Both the transfer channel enable (<E> = 1) and the reception enable of the transfer request (<Halt> = 0) should be done at the same time. <Halt> = 1 should not be set.

Note2: The transfer request issued by a memory is not supported. When "Memory to Memory" is selected in <FlowCntrl[2:0]> (000), the data transfer is started by setting <E> to "1". When "Memory to Peripheral device" (001) is selected, the data transfer from the memory to FIFO is started by setting <E> to "1", and the data transfer from FIFO to the peripheral device is started by the transfer request issued by the peripheral device.

Note3: For the details of the request channel, refer to "Production Information" in Reference manual.

Note4: When <E> is set to "0" (Disabled) during data transfer (<E> = 1 and ***[DMACxEnbldChns]***<EnabledCHn> = 1), the data in the FIFO of the transfer channel disappears. If data transfer should be done again, all the settings of the transfer channel should be initialized and then the transfer should be done.

5. Revision History

Table 5.1 Revision history

Revision	Date	Description
1.0	2018-01-18	First release
2.0	2018-04-3	3.3. Transfer Type Added: Table 3.1 Note2: 3.7.4. Clear of Interrupt Corrected: Figure 3.1 "<IntTCStatus0>" -> "<RawIntTCS0>" "<IntTCStatus1>" -> "<RawIntTCS1>" "<IntErrStatus0>" -> "<RawIntErrS0>" "<IntErrStatus1>" -> "<RawIntErrS1>" 4.2.6. ***[DMACxRawIntTCStatus]*** Corrected: Bit Symbol "<IntTCStatus0>" -> "<RawIntTCS0>" "<IntTCStatus1>" -> "<RawIntTCS1>" 4.2.7. ***[DMACxRawIntErrorStatus]*** Corrected: Bit Symbol: "<IntErrStatus0>" -> "<RawIntErrS0>" "<IntErrStatus1>" -> "<RawIntErrS1>" 4.2.9. ***[DMACxSoftBReq]*** Corrected: Bit symbol by bit Deleted: Note1: 4.2.10. ***[DMACxSoftSReq]*** Corrected: Bit symbol by bit Deleted: Note1: 4.2.16. ***[DMACxCnConfiguration]*** Note 4: " ***[DMACxEnbledChns]*** " -> " ***[DMACxEnbldChns]*** "
2.1	2019-02-12	Arm trademark relationship correction. 4.2.3. ***[DMACxIntTCClear]*** Corrected: Bit0 "<IntTCSClear0>" -> " <IntTCClear0>" 4.2.11. ***[DMACxConfiguration]*** Deleted: "When this bit is "0", the other registers cannot be read nor written. " RESTRICTIONS ON PRODUCT USE Replaced

RESTRICTIONS ON PRODUCT USE

Toshiba Corporation and its subsidiaries and affiliates are collectively referred to as “TOSHIBA”.
Hardware, software and systems described in this document are collectively referred to as “Product”.

- TOSHIBA reserves the right to make changes to the information in this document and related Product without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS.**
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, lifesaving and/or life supporting medical equipment, equipment used for automobiles, trains, ships and other transportation, traffic signaling equipment, equipment used to control combustions or explosions, safety devices, elevators and escalators, and devices related to power plant. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**