Direct Memory Access (DMA)

Introduction

Direct Memory Access (DMA) is a method for transfer of data between memory and peripherals, bypassing the CPU for faster memory operations. This keeps the CPU available to perform other operations while performing the data transfer. The process is managed by the DMA Controller (DMAC).

The DMA is parameterized for upto 7 channels for managing the memory access requests from upto 16 peripherals connected to the channels. The DMA has an arbiter which manages the requests of the peripherals by selecting the channel based on the priority levels.

DMA main features

1. Upto 7 independently configurable channels.
2. Supports configurable bursts.
3. Memory to memory, memory to peripheral and peripheral to memory transfer can be done.
4. Independent source and destination transfer size. Source/destination addresses must be aligned on the data size.
5. Supports circular mode for handling circular buffer management.

DMA Transactions

When a peripheral/memory requires data to be transferred, it sends a request signal to the DMAC through the channels connected to it. The DMA checks the channel priority and responds to the request. Once the channel is selected, the DMAC sends an acknowledge signal to the source of data transfer. The DMA handles the data transfer. When the request is de-asserted by the peripheral, the DMAC releases the acknowledge signal stating the completion of data transfer between the source and the destination memory elements.

Once the DMA receives transfer request from a memory element, the CPU sets the DMA configuration registers.

- The start address in the source and the destination address registers (DMA_CPARx or DMA_CMARx, depending whether it is a peripheral or memory), are written simultaneously depending on whether it is a memory or peripheral. Here, x is the channel number. For the first transaction, the start address is the base address of the source and the destination memory elements. The start address is updated after each transaction depending upon the data size.
- The register containing the number total of transactions to be performed (DMA_CNDTRx), is configured and decremented after each transaction. Data transfer is complete when the value of DMA_CNDTRx is zero. If configured in circular mode, the register is reloaded to it’s initial value after the current transfer is finished.
- The register containing information on data transfer DMA_CCRx register is set. It is configured with information on the data transfer direction (i.e. if the source and destination are memory or peripheral), the channel priority level, the peripheral and the memory sizes, the memory and peripheral increment modes, whether the transfer is in circular mode, interrupt after half and full transfer and if the channel is Enabled.
  The CB bits of the DMA_CCRx register contains information of the configurable burst size i.e. the burst size can be independently selected.
- The registers DMA_ISR and DMA_IFCR are set after the channel is Enabled (i.e. setting the Enable bit in DMA_CCRx as 1). DMA_ISR is written once it gets TEI/HTI/TCI/GI flags. The DMA_IFCR resets clears the value of the respective error on the DMA_ISR once the flag is removed.
- DMA_CSELR is used for the mapping of DMA channels.

Programmable data size and pointer incrementation

The data transfer size is of the memory and peripheral programmed through the MSIZE and PSIZE bits respectively of the DMA_CCRx register. The start address of the DMA_CPARx and DMA_CMARx are updated according to the PINC and MINC bits of the DMA_CCRx respectively depending on the chosen data size. Start address is incremented by:

- 1, when data size is of 8 bits
- 2, when data size is of 16 bits
- 3, when data size is of 32 bits

Management of channel priority

When transfer requests from more than one channels are available, the channels are selected according to their priority levels which is managed by the arbiter.

The priority levels are managed in two stages:

- Software: The channel priority levels are set by the DMA_CCRx[13:12] and the channels are set in four priority levels:
    - 00: Low
    - 01: Medium
    - 10: High
    - 11: Very High

- If two or more channels have the same priority levels, the channel with the lowest ID number will have the highest priority.

Circular Mode

The DMA can operate in circular mode to handle circular buffers. This mode is Enabled by using the CIRC bit in the DMA_CCRx register. In this mode, the DMA_CNDTR is automatically loaded. The source and destination address registers, DMA_CPARx and DMA_CMARx are also loaded.

DMA Interrupts

There are 3 event interrupts which can be produced on a Half-transfer, Transfer complete or transfer error for a channel. Global interrupt flag is enabled when any of these flags occur simultaneously or individually.

Interrupt	Event Flag
Half-Transfer	HTIF
Transfer complete	TCIF
Transfer error	TEIF

Error Management

The DMA transfer error can occur when any reserved address is accessed for read or write operations. When a DMA transfer error occurs during a DMA read or a write access, the channel is automatically disabled through a hardware clear of its EN bit in the corresponding Channel configuration register (DMA_CCRx). The channel's transfer error interrupt flag (TEIF) in the DMA_ISR is set and an interrupt is generated if the transfer error interrupt enable bit (TEIE) in the DMA_CCRx register is set.

DMA Setup

The DMAC is connected to the Testbench and a BRAM which acts like the destination memory. The files required are:

- DMA.bsv : Contains information on the data transfer by the DMAC.
- tb_DMA.bsv : This file instantiates the DMAC and the source and the destination memory.
- AXI4_Fabric.bsv : This file contains the interface where requests and responses are handled.
- Memory_AXI4.bsv : This file contains a memory module with a slave interface, used by the testbench.

Interfaces

The DMA has a slave interface (cfg) to receive and send instructions from the CPU and a master interface (mmu) for all channels to send and receive data and response to the memory.

FIFO

- Each of the interfaces has two FIFOs, one for request and other for response.
- The FIFO destAddrFs is used to store information on the destination location. It contains the destination address, whether the destination is a memory or a peripheral and the peripheral ID. This FIFO is basically used to pass the destination address after each read operation to write side.
- The FIFO responseDataFs is used to pass the read response to the write side. It contains the data obtained after reading from the required address.

Rules

- startRead : This rule fires when there are data to transfer (i.e. the value of DMA_CNDTRx is not zero), the channel has the highest priority and the channel is not disabled. This rule enqueues the destAddrFs FIFO.
- finishRead : This rule fires when correct response data is received by the channel from the mmu. This rule enqueues the responseDataFs FIFO.
- rl_handle_circular_mode : This rule fires if the circular mode is enabled in the CIRC bit of DMA_CCRx and it loads DMA_CNDTR with it's original value.
- startWrite : This rule fires when the destination is memory. If the destination is not memory, it checks if there is interrupt from the peripheral. It fires when it receives an interrupt from the peripheral. It dequeues the FIFOs destAddrFs and responseDataFs for startRead to fire for the next transaction.
- rl_send_burst_write_data : This rule is used to perform burst write and fires when burst write is to be performed.
- finishWrite : This rule fires when there is correct write response from the mmu. This takes the response data and finishes the write.
- rl_cndtr_is_zero : It feeds a Bool value to the CReg rl_cndtr_is_zero to check if the value of DMA_CNDTR is zero.
- markTransferDone : This rule fires when the remaining data to be transferred is zero and the channel is enabled and the final write has finished. This rule indicates the end of the data transfer and restores the value of registers currentWriteRs and currentReadRs containing information on the current read and write number to zero. This rule will not fire in circular mode as DMA_CNDTR is restored with the original value once a transaction is complete.
- writeConfig : It writes the value of the configuration registers.
- readConfig : It reads the configuration registers once the registers are written by the rule writeConfig.

DMA Configuration Registers

The DMA Configuration registers are configured by the CPU once it receives an interrupt from the peripheral/memory for data transfer. Since, the DMA model is parameterized, there may be any number of channels selected between 1 and 7. So, the registers are defined for 7 channels. The the registers corresponding to the channels which are not selected are reset to zero.

1. DMA Interrupt Status Register (DMA_ISR)

Address offset: 0X00

Reset value: 0X0000 0000

This register identifies various interrupts from the channels and updates it's value according to the interrupts received.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	**Res.**	**Res.**	**Res.**	**TEIF7**	**HTIF7**	**TCIF7**	**GIF7**	**TEIF6**	**HTIF6**	**TCIF6**	**GIF6**	**TEIF5**	**HTIF5**	**TCIF5**	**GIF5**
				r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEIF4	**HTIF4**	**TCIF4**	**GIF4**	**TEIF3**	**HTIF3**	**TCIF3**	**GIF3**	**TEIF2**	**HTIF2**	**TCIF2**	**GIF2**	**TEIF1**	**HTIF1**	**TCIF1**	**GIF1**
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:28 Reserved, must be kept at reset value.

Bits 27, 23, 19, 15, 11, 7, 3 **TEIFx** : Channel x transfer error flag (x = 1..7).
This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register.
0: No transfer error (TE) on channel x
1: A transfer error (TE) occurred on channel x

Bits 26, 22, 18, 14, 10, 6, 2 **HTIFx** : Channel x half transfer flag (x = 1..7).
This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register.
0: No half transfer (HT) event on channel x
1: A half transfer (HT) event occurred on channel x

Bits 25, 21, 17, 13, 9, 5, 1 **TCIFx** : Channel x transfer complete flag (x = 1..7).
This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register.
0: No transfer complete (TC) event on channel x
1: A transfer complete (TC) event occurred on channel x

Bits 24, 20, 16, 12, 8, 4, 0 **GIFx** : Channel x half transfer flag (x = 1..7).
This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register.
0: No half transfer (HT) event on channel x
1: A half transfer (HT) event occurred on channel x

2. DMA Interrupt flag clear Register (DMA_IFCR)

Address offset: 0X04

Reset value: 0X0000 0000

This register is used to clear the values of the interrupt status register once the interrupt is absent.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15
Res.	**Res.**	**Res.**	**Res.**	**CTEIF7**	**CHTIF7**	**CTCIF7**	**CGIF7**	**CTEIF6**	**CHTIF6**	**CTCIF6**	**CGIF6**	**CTEIF5**	**CHTIF5**	**CTCIF5**	**CGIF5**	**CTEIF4**
				w	w	w	W	w	w	w	w	w	w	w	w	w

14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHTIF4	**CTCIF4**	**CGIF4**	**CTEIF3**	**CHTIF3**	**CTCIF3**	**CGIF3**	**CTEIF2**	**CHTIF2**	**CTCIF2**	**CGIF2**	**CTEIF1**	**CHTIF1**	**CTCIF1**	**CGIF1**
w	w	w	w	w	w	W	w	w	w	w	w	w	w	w

Bits 31:28 Reserved, must be kept at reset value.

Bits 27, 23, 19, 15, 11, 7, 3 **CTEIFx** : Channel x transfer error clear (x = 1..7).

This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register.

0: No transfer error (TE) on channel x

1: A transfer error (TE) occurred on channel x

Bits 26, 22, 18, 14, 10, 6, 2 **CHTIFx** : Channel x half transfer clear (x = 1..7).

This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register.

0: No half transfer (HT) event on channel x

1: A half transfer (HT) event occurred on channel x

Bits 25, 21, 17, 13, 9, 5, 1 **CTCIFx**: Channel x transfer complete clear (x = 1..7).

This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register.

0: No transfer complete (TC) event on channel x

1: A transfer complete (TC) event occurred on channel x

Bits 24, 20, 16, 12, 8, 4, 0 **CGIFx** : Channel x global interrupt clear (x = 1..7).

This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register.

0: No half transfer (HT) event on channel x

1: A half transfer (HT) event occurred on channel x

3. DMA channel x configuration register (DMA_CCRx), x is the channel number

Reset value: 0x0000 0000

This register is written once the channel is enabled. This register consists the characteristics of the data to be transferred.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15
Res.	**Res.**	**Res.**	**Res.**	**Res.**	**Res.**	**Res.**	**Res.**	**CB[23:16]**								**Res.**
								r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	

14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEM-TO-MEM	**PL[1:0]**		**MSIZE[1:0]**		**PSIZE[1:0]**		**MINC**	**PINC**	**CIRC**	**DIR**	**TEIE**	**HTIE**	**TCIE**	**EN**
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bits 31:24; 15 : Reserved, must be kept at reset value

Bits 23:16 **CB[7:0]** : Configurable burst.
These bits of the dma_ccr contains information on the configurable burst size.
These bits are set and cleared by the software.
DMA_CCRx[23:16] = x00: Configurable burst disabled

Bit 14 **MEM-TO-MEM**: Memory to memory mode
This bit is set and cleared by software.
0: Memory to memory mode disabled
1: Memory to memory mode enabled

Bits 13:12 **PL[1:0]**: Channel priority level
These bits are set and cleared by software.
00: Low
01: Medium
10: High
11: Very high

Bits 11:10 **MSIZE[1:0]**: Memory size
These bits are set and cleared by software.
00: 8-bits
01: 16-bits
10: 32-bits
11: Reserved

Bits 9:8 **PSIZE[1:0]**: Peripheral size
These bits are set and cleared by software.
00: 8-bits
01: 16-bits
10: 32-bits
11: Reserved

Bit 7 **MINC**: Memory increment mode

This bit is set and cleared by software.

0: Memory increment mode disabled

1: Memory increment mode enabled

Bit 6 **PINC**: Peripheral increment mode

This bit is set and cleared by software.

0: Peripheral increment mode disabled

1: Peripheral increment mode enabled

Bit 5 **CIRC**: Circular mode

This bit is set and cleared by software.

0: Circular mode disabled

1: Circular mode enabled

Bit 4 **DIR**: Data transfer direction

This bit is set and cleared by software.

0: Read from peripheral

1: Read from memory

Bit 3 **TEIE**: Transfer error interrupt enable

This bit is set and cleared by software.

0: TE interrupt disabled

1: TE interrupt enabled

Bit 2 **HTIE**: Half transfer interrupt enable

This bit is set and cleared by software.

0: HT interrupt disabled

1: HT interrupt enabled

Bit 1 **TCIE**: Transfer complete interrupt enable

This bit is set and cleared by software.

0: TC interrupt disabled

1: TC interrupt enabled

Bit 0 **EN**: Channel enable

This bit is set and cleared by software.

0: Channel disabled

1: Channel enabled

4. DMA channel x number of data register(DMA_CNDTRx), x is the channel number

Reset value: 0x0000 0000

The register can only be written when the channel is disabled. Read-only mode after the channel is Enabled. The register value is updated after every DMA transfer.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	**Res.**	**Res.**	**Res.**	**Res.**	**Res.**	**Res.**	**Res.**	**Res.**	**Res.**	**Res.**	**Res.**	**Res.**	**Res.**	**Res.**	**Res.**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NDT[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bit 31:16: Reserved, must be kept at reset value.

Bit 15:0 **NDT[15:0]**: Number of data left to be transferred. The register value is updated after every DMA Transfer. The transfer stops once the value is zero.
When the data transfer is complete, if configured in circular mode, the channel is loaded With the previously programmed value which otherwise is configured to zero.

5. DMA channel x memory address register(DMA_CMARx), x is the channel number

Reset value: 0x0000 0000

This register is not written when the channel is enabled.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MA[31:16]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MA[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bits 31:0 **MA [31:0]**: Memory address

Base address of the memory area from/to which the data will be read/written.
When MSIZE is 01 (16-bit), the MA[0] bit is ignored. Access is automatically aligned to a Halfword address.
When MSIZE is 10 (32-bit), MA[1:0] are ignored. Access is automatically aligned to a word address.

6. DMA channel x peripheral address register(DMA_CPARx), x is the channel number

Reset value: 0x0000 0000

This register is not written when the channel is enabled.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PA[31:16]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PA[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bits 31:0 **PA[31:0]** : Peripheral address

Base address of the peripheral data register from/to which the data will be read/written.
When PSIZE is 01 (16-bit), the PA[0] bit is ignored. Access is automatically aligned to a Halfword address.
When PSIZE is 10 (32-bit), PA[1:0] are ignored. Access is automatically aligned to a word Address.

7. DMA Channel Selection Register(DMA1_CSELR)

Address offset: 0XB0 (wrt DMA base address)

Reset value: 0x0000 0000

This register is used to manage mapping of DMA channels.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	**Res.**	**Res.**	**Res.**	**C7S [3:0]**				**C6S [3:0]**				**C5S [3:0]**			
				r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C4S [3:0]				**C3S [3:0]**				**C2S [3:0]**				**C1S [3:0]**			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Bits 31:28 Reserved, must be kept at reset value.

Bits 27:24 **C7S[3:0]**: DMA channel 7 selection.
Selects the peripherals mapped to channel 7

Bits 23:20 **C6S[3:0]**: DMA channel 6 selection.
Selects the peripherals mapped to channel 6

Bits 19:16 **C5S[3:0]**: DMA channel 5 selection.
Selects the peripherals mapped to channel 5

Bits 15:12 **C4S[3:0]**: DMA channel 4 selection.
Selects the peripherals mapped to channel 4

Bits 11:8 **C3S[3:0]**: DMA channel 3 selection.
Selects the peripherals mapped to channel 3

Bits 7:4 **C2S[3:0]**: DMA channel 2 selection.
Selects the peripherals mapped to channel 2

Bits 3:0 **C1S[3:0]**: DMA channel 1 selection.
Selects the peripherals mapped to channel 1

DMA Register Map

The table below shows the DMA register map and the register reset values.

Offset	Register	Reset value
8’h00	DMA_ISR	X0000 0000
8’h04	DMA_IFCR	
8’hB0	DMA_CSELR	
8’h08	DMA_CCR1	
8’h0C	DMA_CNDTR1	
8’h10	DMA_CPAR1	
8’h18	DMA_CMAR1	
8’h20	DMA_CCR2	
8’h24	DMA_CNDTR2	
8’h28	DMA_CPAR2	
8’h30	DMA_CMAR2	
8’h38	DMA_CCR3	
8’h3C	DMA_CNDTR3	
8’h40	DMA_CPAR3	
8’h48	DMA_CMAR3	
8’h50	DMA_CCR4	
8’h54	DMA_CNDTR4	
8’h58	DMA_CPAR4	
8’h60	DMA_CMAR4	
8’h68	DMA_CCR5	
8’h6C	DMA_CNDTR5	
8’h70	DMA_CPAR5	
8’h78	DMA_CMAR5	
8’h80	DMA_CCR6	
8’h84	DMA_CNDTR6	
8’h88	DMA_CPAR6	
8’h90	DMA_CMAR6	
8’h98	DMA_CCR7	
8’h9C	DMA_CNDTR7	
8’HA0	DMA_CPAR7	
8’HA8	DMA_CMAR7	

References

- STM Micro Reference Manual
- DMA Controller Lab by Bluespec Inc.