Chapter 12
DMA Controller Module

This chapter describes the MCF5307 DMA controller module. It provides an overview of the module and describes in detail its signals and registers. The latter sections of this chapter describe operations, features, and supported data transfer modes in detail.

12.1 Overview

The direct memory access (DMA) controller module provides an efficient way to move blocks of data with minimal processor interaction. The DMA module, shown in Figure 12-1, provides four channels that allow byte, word, or longword operand transfers. Each channel has a dedicated set of registers that define the source and destination addresses (SAR*n* and DAR*n*), byte count (BCR*n*), and control and status (DCR*n* and DSR*n*). Transfers can be dual or single address to off-chip devices or dual address to on-chip devices, such as UART, SDRAM controller, and parallel port.

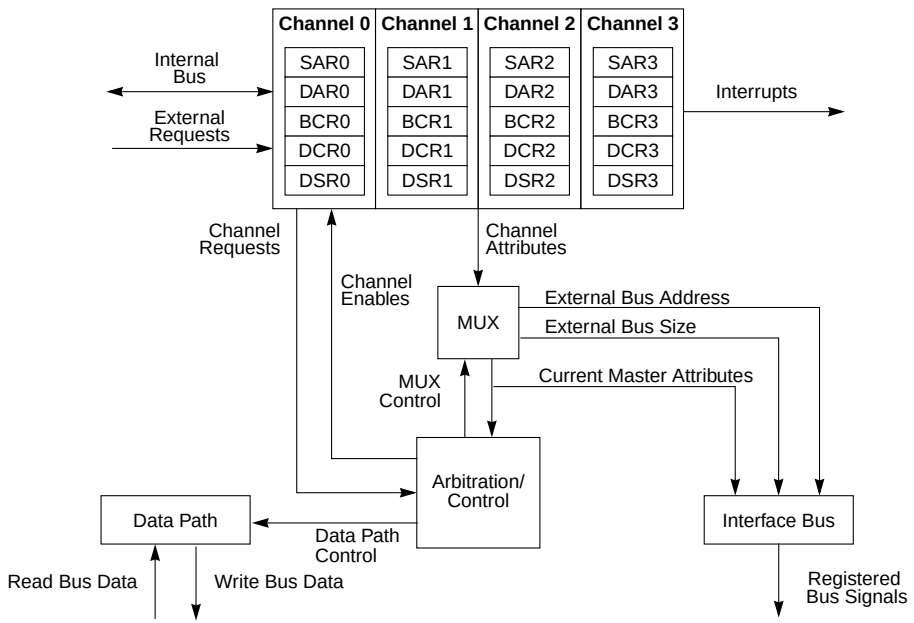

Figure 12-1. DMA Signal Diagram

12.1.1 DMA Module Features

The DMA controller module features are as follows:

- Four fully independent, programmable DMA controller channels/bus modules
- Auto-alignment feature for source or destination accesses
- Dual- and single-address transfers
- Two external request pins ($\overline{\text{DREQ}}$[1:0]) provided for channels 1 and 0
- Channel arbitration on transfer boundaries
- Data transfers in 8-, 16-, 32-, or 128-bit blocks using a 16-byte buffer
- Continuous-mode and cycle-steal transfers
- Independent transfer widths for source and destination
- Independent source and destination address registers
- Data transfer can occur in as few as two clocks

12.2 DMA Signal Description

Table 12-1 briefly describes the DMA module signals that provide handshake control for either a source or destination external device.

Table 12-1. DMA Signals

Signal	I/O	Description
$\overline{\text{DREQ}}$[1:0]/ PP[6:5]	I	External DMA request. $\overline{\text{DREQ}}$[1:0] can serve as the DMA request inputs or as two parallel port bits. They are programmable individually through the PAR. A peripheral device asserts these inputs to request an operand transfer between it and memory. $\overline{\text{DREQ}}$ signals are asserted to initiate DMA accesses in the respective channels. The system should drive unused $\overline{\text{DREQ}}$ signals to logic high. Although each channel has an individual $\overline{\text{DREQ}}$ signal, in the MCF5307 only channels 0 and 1 connect to external $\overline{\text{DREQ}}$ pins.$\overline{\text{DREQ}}$ signals for channels 2 and 3 are connected to the UART0 and UART1 bus interrupt signals.
TT[1:0]/ PP[1:0]	O	Transfer type. A DMA access is indicated by the transfer type pins, TT[1:0] = 01. The transfer modifier, TM[2:0] configurations shown below are meaningful only if TT[1:0] = 01, indicating an external master or DMA access.
TM[2:0] /PP[4:2]	O	Multiplexed transfer attribute pins. The encodings below are valid when TT[1:0] = 01 and internal DMA channels are driving the bus. DMA transfer information on TM[2:1] can be provided on every DMA transfer or only on the last transfer by programming DCR[AT]. TM[2:1] Encoding 00 DMA acknowledge information not provided 01 DMA transfer, channel 0 10 DMA transfer, channel 1 11 Reserved TM0 Encoding for DMA as master (TT = 01) 0 Single-address access negated 1 Single-address access For TT[1:0] = 01, the TM0 encoding is independent of TM[2:1]. If DCR[SAA] is set, TM0 designates a single-address DMA access.

12.3 DMA Transfer Overview

The DMA module usually transfers data faster than the ColdFire core can under software control. The term ‘direct memory access’ refers to peripheral device’s ability to access system memory directly, greatly improving overall system performance. The DMA module consists of four independent, functionally equivalent channels, so references to DMA in this chapter apply to any of the channels. It is not possible to implicitly address all four channels at once. The MCF5307 on-chip peripherals do not support single-address transfers.

The processor generates DMA requests internally by setting DCR[START]; a device can generate a DMA request externally by using $\overline{\text{DREQ}}$ pins. The processor can program bus bandwidth for each channel. The channels support cycle-steal and continuous transfer modes; see Section 12.5.1, “Transfer Requests (Cycle-Steal and Continuous Modes).”

The DMA controller supports dual- and single-address transfers as follows. In both, the DMA channel supports 32 address bits and 32 data bits.

- Dual-address transfers—A dual-address transfer consists of a read followed by a write and is initiated by an internal request using the START bit or by an external device using $\overline{\text{DREQ}}$. Two types of transfer can occur, a read from a source device or a write to a destination device; see Figure 12-2.

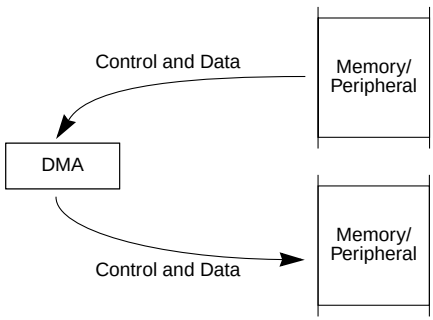

Figure 12-2. Dual-Address Transfer

- Single-address transfers—An external device can initiate a single-address transfer by asserting $\overline{\text{DREQ}}$. The MCF5307 provides address and control signals for single-address transfers. The external device reads to or writes from the specified address, as Figure 12-3 shows. External logic is required.

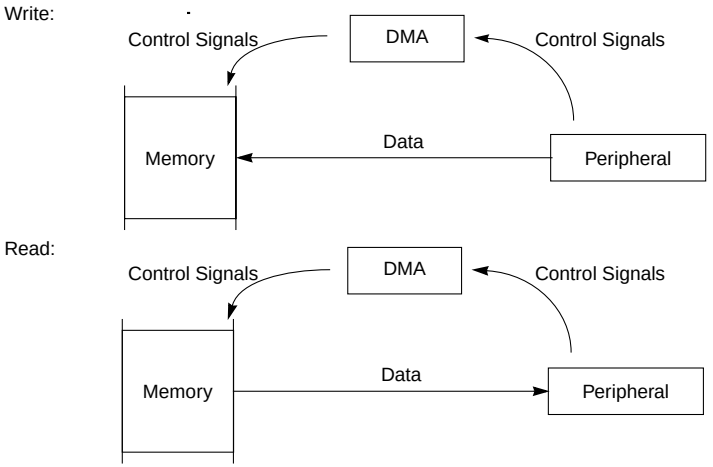

Figure 12-3. Single-Address Transfers

Any operation involving the DMA module follows the same three steps:

1. Channel initialization—Channel registers are loaded with control information, address pointers, and a byte-transfer count.
2. Data transfer—The DMA accepts requests for operand transfers and provides addressing and bus control for the transfers.
3. Channel termination—Occurs after the operation is finished, either successfully or due to an error. The channel indicates the operation status in the channel's DSR, described in Section 12.4.5, "DMA Status Registers (DSR0–DSR3)."

12.4 DMA Controller Module Programming Model

This section describes each internal register and its bit assignment. Note that there is no way to prevent a write to a control register during a DMA transfer. Table 12-2 shows the mapping of DMA controller registers. Note the differences for the byte count registers depending on the value of MPARK[BCR24BIT].

Table 12-2. Memory Map for DMA Controller Module Registers

DMA Channel	MBAR Offset	[31:24]	[23:16]	[15:8]	[7:0]
0	0x300	Source address register 0 (SAR0) [p. 12-6]			
	0x304	Destination address register 0 (DAR0) [p. 12-7]			
	0x308	DMA control register 0 (DCR0) [p. 12-8]			
	0x30C	Byte count register 0 (BCR24BIT = 0) [1]		Reserved	
	0x30C	Reserved	Byte count register 0 (BCR24BIT = 1) [1] (BCR0) [p. 12-7]		
	0x310	DMA status register 0 (DSR0) [p. 12-10]	Reserved		
	0x314	DMA interrupt vector register 0 (DIVR0) [p. 12-11]	Reserved		
1	0x340	Source address register 1 (SAR1) [p. 12-6]			
	0x344	Destination address register 1 (DAR1) [p. 12-7]			
	0x348	DMA control register 1 (DCR1) [p. 12-8]			
	0x34C	Byte count register 1 (BCR24BIT = 0) [1]		Reserved	
	0x34C	Reserved	Byte count register 1 (BCR24BIT = 1) [1] (BCR1) [p. 12-7]		
	0x350	DMA status register 1 (DSR1) [p. 12-10]	Reserved		
	0x354	DMA interrupt vector register 1 (DIVR1) [p. 12-11]	Reserved		
2	0x380	Source address register 2 (SAR2) [p. 12-6]			
	0x384	Destination address register 2 (DAR2) [p. 12-7]			
	0x388	DMA control register 2 (DCR2) [p. 12-8]			
	0x38C	Byte count register 2 (BCR24BIT = 0) [1]		Reserved	
	0x38C	Reserved	Byte count register 2 (BCR24BIT = 1) [1] (BCR2) [p. 12-7]		
	0x390	DMA status register 2 (DSR2) [p. 12-10]	Reserved		
	0x394	DMA interrupt vector register 2 (DIVR2) [p. 12-11]	Reserved		

Table 12-2. Memory Map for DMA Controller Module Registers (Continued)

DMA Channel	MBAR Offset	[31:24]	[23:16]	[15:8]	[7:0]
3	0x3C0	Source address register 3 (SAR3) [p. 12-6]			
	0x3C4	Destination address register 3 (DAR3) [p. 12-7]			
	0x3C8	DMA control register 3 (DCR3) [p. 12-8]			
	0x3CC	Byte count register 3 (BCR24BIT = 0) [1]		Reserved	
	0x3CC	Reserved	Byte count register 3 (BCR24BIT = 1) [1] (BCR3) [p. 12-7]		
	0x3D0	DMA status register 3 (DSR3) [p. 12-10]	Reserved		
	0x3D4	DMA interrupt vector register 3 (DIVR3) [p. 12-11]	Reserved		

1 On the original MCF5307 mask set (H55J), the BCR of the DMA channels can accommodate only 16 bits. However, because the revised MCF5307 supports a 24-bit byte count range, the position of the BCR in the memory map depends on whether a 16- or 24-bit byte counter is selected. The 24-bit byte count can be selected by setting BCR24BIT = 1, making DCR[AT] available. The AT bit selects whether DMA channels assert acknowledge during the entire transfer or only at the final transfer of a DMA transaction.

New applications should take advantage of the full range of the 24-bit byte counter, including the AT bit. The 16-bit byte count option (BCR24BIT = 0) retains compatibility with older MCF5307 revisions.

NOTE:

External masters cannot access MCF5307 on-chip memories or MBAR, but they can access DMA module registers.

12.4.1 Source Address Registers (SAR0–SAR3)

SAR*n*, Figure 12-4, contains the address from which the DMA controller requests data. In single-address mode, SAR*n* provides the address regardless of the direction.

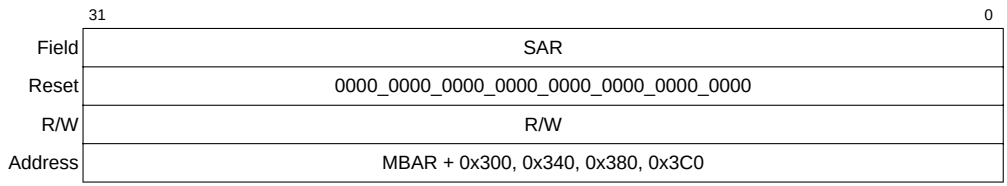

Figure 12-4. Source Address Registers (SAR*n*)

NOTE:

SAR/DAR address ranges cannot be programmed to on-chip SRAM because it cannot be accessed by on-chip DMA.

12.4.2 Destination Address Registers (DAR0–DAR3)

For dual-address transfers only, DAR*n*, Figure 12-5, holds the address to which the DMA controller sends data.

	31	0
Field	DAR	
Reset	0000_0000_0000_0000_0000_0000_0000_0000	
R/W	R/W	
Address	MBAR + 304, 0x344, 0x384, 0x3C4	

Figure 12-5. Destination Address Registers (DAR*n*)

NOTE:

On-chip DMAs do not maintain coherency with MCF5307 caches and so must not transfer data to cacheable memory.

12.4.3 Byte Count Registers (BCR0–BCR3)

BCR*n*, Figure 12-6 and Figure 12-7, holds the number of bytes yet to be transferred for a given block.The offset within the memory map is based on the value of MPARK[BCR24BIT]. BCR*n* decrements on the successful completion of the address transfer of either a write transfer in dual-address mode or any transfer in single-address mode. BCR*n* decrements by 1, 2, 4, or 16 for byte, word, longword, or line accesses, respectively.

Figure 12-6 shows BCR for BCR24BIT = 1.

	31 24	23 0
Field	—	BCR
Reset	—	0000_0000_0000_0000_0000_0000
R/W	R/W	
Address	MBAR + 0x30C, 0x34C, 0x38C, 0x3AC	

Figure 12-6. Byte Count Registers (BCR*n*)—BCR24BIT = 1

Figure 12-7 shows BCR for BCR24BIT = 0.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	BCR															
Reset	0000_0000_0000_0000															
R/W																
Addr	MBAR + 0x30C, 0x34C, 0x38C, 0x3AC															

Figure 12-7. BCR*n*—BCR24BIT = 0

DSR[DONE], shown in Figure 12-9, is set when the block transfer is complete.

When a transfer sequence is initiated and BCR*n*[BCR] is not divisible by 16, 4, or 2 when the DMA is configured for line, longword, or word transfers, respectively, DSR*n*[CE] is set and no transfer occurs. See Section 12.4.5, “DMA Status Registers (DSR0–DSR3).”

12.4.4 DMA Control Registers (DCR0–DCR3)

DCR*n*, Figure 12-8, is used for configuring the DMA controller module. Note that DCR[AT] is available only if BCR24BIT = 1.

	31	30	29	28	27	25	24	23	22	21	20	19	18	17	16
Field	INT	EEXT	CS	AA	BWC		SAA	S_RW	SINC	SSIZE		DINC	DSIZE		START
Reset	0000_0000_0000_0000														
R/W	R/W														

	15	14	0
Field	AT [1]	—	
Reset	0	N/A	
R/W	R/W		
Address	MBAR + 0x308, 0x348, 0x388, 0x3A8		

Figure 12-8. DMA Control Registers (DCR*n*)

1 Available only if BCR24BIT = 1, otherwise reserved.

Table 12-3 describes DCR fields.

Table 12-3. DCR*n* Field Descriptions

Bits	Name	Description
31	INT	Interrupt on completion of transfer. Determines whether an interrupt is generated by completing a transfer or by the occurrence of an error condition. 0 No interrupt is generated. 1 Internal interrupt signal is enabled.
30	EEXT	Enable external request. Care should be taken because a collision can occur between the START bit and $\overline{\text{DREQ}}$ when EEXT = 1. 0 External request is ignored. 1 Enables external request to initiate transfer. Internal request is always enabled. It is initiated by writing a 1 to the START bit.

Table 12-3. DCR*n* Field Descriptions (Continued)

Bits	Name	Description
29	CS	Cycle steal. 0 DMA continuously makes read/write transfers until the BCR decrements to 0. 1 Forces a single read/write transfer per request. The request may be internal by setting the START bit, or external by asserting $\overline{\text{DREQ}}$.
28	AA	Auto-align. AA and SIZE determine whether the source or destination is auto-aligned, that is, transfers are optimized based on the address and size. See Section 12.5.4.2, “Auto-Alignment.” 0 Auto-align disabled 1 If SSIZE indicates a transfer no smaller than DSIZE, source accesses are auto-aligned; otherwise, destination accesses are auto-aligned. Source alignment takes precedence over destination alignment. If auto-alignment is enabled, the appropriate address register increments, regardless of DINC or SINC.
27–25	BWC	Bandwidth control. Indicates the number of bytes in a block transfer. When the byte count reaches a multiple of the BWC value, the DMA releases the bus. For example, if BCR24BIT is 0, BWC is 001 (512 bytes or value of 0x0200), and BCR is 0x1000, the bus is relinquished after BCR values of 0x2000, 0x1E00, 0x1C00, 0x1A00, 0x1800, 0x1600, 0x1400, 0x1200, 0x1000, 0x0E00, 0x0C00, 0x0A00, 0x0800, 0x0600, 0x0400, and 0x0200. If BCR24BIT is 0, BWC is 110, and BCR is 33000, the bus is released after 232 bytes because the BCR is at 32768, a multiple of 16384. BWC BCR24BIT = 0 BCR24BIT = 1 000 DMA has priority. It does not negate its request until its transfer completes. 001 512 16384 010 1024 32768 011 2048 65536 100 4096 131072 101 8192 262144 110 16384 524288 111 32768 1048576
24	SAA	Single-address access. Determines whether the DMA channel is in dual- or single-address mode 0 Dual-address mode. 1 Single-address mode. The DMA provides an address from the SAR and directional control, bit S_RW, to allow two peripherals (one might be memory) to exchange data within a single access. Data is not stored by the DMA.
23	S_RW	Single-address access read/write value. Valid only if SAA = 1. Specifies the value of the read signal during single-address accesses. This provides directional control to the bus controller. 0 Forces the read signal to 0. 1 Forces the read signal to 1.
22	SINC	Source increment. Controls whether a source address increments after each successful transfer. 0 No change to SAR after a successful transfer. 1 The SAR increments by 1, 2, 4, or 16, as determined by the transfer size.
21–20	SSIZE	Source size. Determines the data size of the source bus cycle for the DMA control module. 00 Longword 01 Byte 10 Word 11 Line
19	DINC	Destination increment. Controls whether a destination address increments after each successful transfer. 0 No change to the DAR after a successful transfer. 1 The DAR increments by 1, 2, 4, or 16, depending upon the size of the transfer.
18–17	DSIZE	Destination size. Determines the data size of the destination bus cycle for the DMA controller. 00 Longword 01 Byte 10 Word 11 Line

Table 12-3. DCR*n* Field Descriptions (Continued)

Bits	Name	Description
16	START	Start transfer. 0 DMA inactive 1 The DMA begins the transfer in accordance to the values in the control registers. START is cleared automatically after one clock and is always read as logic 0.
15	AT	AT is available only if BCR24BIT = 1. DMA acknowledge type. Controls whether acknowledge information is provided for the entire transfer or only the final transfer. 0 Entire transfer. DMA acknowledge information is displayed anytime the channel is selected as the result of an external request. 1 Final transfer (when BCR reaches zero). For dual-address transfer, the acknowledge information is displayed for both the read and write cycles.
14–0	—	Reserved, should be cleared.

12.4.5 DMA Status Registers (DSR0–DSR3)

In response to an event, the DMA controller writes to the appropriate DSR*n* bit, Figure 12-9. Only a write to DSR*n*[DONE] results in action.

	7	6	5	4	3	2	1	0
Field	—	CE	BES	BED	—	REQ	BSY	DONE
Reset	—	0	0	0	—	0	0	0
R/W	R/W							
Address	MBAR + 0x310, 0x350, 0x390, 0x3D0							

Figure 12-9. DMA Status Registers (DSR*n*)

Table 12-4 describes DSR*n* fields.

Table 12-4. DSR*n* Field Descriptions

Bits	Name	Description
7	—	Reserved, should be cleared.
6	CE	Configuration error. Occurs when BCR, SAR, or DAR does not match the requested transfer size, or if BCR = 0 when the DMA receives a start condition. CE is cleared at hardware reset or by writing a 1 to DSR[DONE]. 0 No configuration error exists. 1 A configuration error has occurred.
5	BES	Bus error on source 0 No bus error occurred. 1 The DMA channel terminated with a bus error either during the read portion of a transfer or during an access in single-address mode (SAA = 1).
4	BED	Bus error on destination 0 No bus error occurred. 1 The DMA channel terminated with a bus error during the write portion of a transfer.
3	—	Reserved, should be cleared.

Table 12-4. DSR*n* Field Descriptions (Continued)

Bits	Name	Description
2	REQ	Request 0 No request is pending or the channel is currently active. Cleared when the channel is selected. 1 The DMA channel has a transfer remaining and the channel is not selected.
1	BSY	Busy 0 DMA channel is inactive. Cleared when the DMA has finished the last transaction. 1 BSY is set the first time the channel is enabled after a transfer is initiated.
0	DONE	Transactions done. Set when all DMA controller transactions complete normally, as determined by transfer count and error conditions. When BCR reaches zero, DONE is set when the final transfer completes successfully. DONE can also be used to abort a transfer by resetting the status bits. When a transfer completes, software must clear DONE before reprogramming the DMA. 0 Writing or reading a 0 has no effect. 1 DMA transfer completed. Writing a 1 to this bit clears all DMA status bits and can be used as an interrupt handler to clear the DMA interrupt and error bits.

12.4.6 DMA Interrupt Vector Registers (DIVR0–DIVR3)

The contents of a DMA interrupt vector register (DIVR*n*), Figure 12-10, are driven onto the internal bus in response to an interrupt acknowledge cycle.

	7	0
Field	Interrupt Vector Bits	
Reset	0000_1111	
R/W	R/W	
Address	MBAR + 0x314, 0x354, 0x394, 0x3D4	

Figure 12-10. DMA Interrupt Vector Registers (DIVR*n*)

12.5 DMA Controller Module Functional Description

In the following discussion, the term ‘DMA request’ implies that DCR[START] or DCR[EEXT] is set, followed by assertion of $\overline{\text{DREQ}}$. The START bit is cleared when the channel begins an internal access.

Before initiating a dual-address access, the DMA module verifies that DCR[SSIZE,DSIZE] are consistent with the source and destination addresses. If the source and destination are not the same size, the configuration error bit, DSR[CE], is also set. If misalignment is detected, no transfer occurs, CE is set, and, depending on the DCR configuration, an interrupt event is issued. Note that if the auto-align bit, DCR[AA], is set, error checking is performed on appropriate registers.

A read/write transfer reads bytes from the source address and writes them to the destination address. The number of bytes is the larger of the sizes specified by SSIZE and DSIZE. See Section 12.4.4, “DMA Control Registers (DCR0–DCR3).”

Source and destination address registers (SAR and DAR) can be programmed in the DCR

to increment at the completion of a successful transfer. BCR decrements when an address transfer write completes for a single-address access (DCR[SAA] = 0) or when SAA = 1.

12.5.1 Transfer Requests (Cycle-Steal and Continuous Modes)

The DMA channel supports internal and external requests. A request is issued by setting DCR[START] or by asserting $\overline{\text{DREQ}}$. Setting DCR[EEXT] enables recognition of external interrupts. Internal interrupts are always recognized. Bus usage is minimized for either internal or external requests by selecting between cycle-steal and continuous modes.

- Cycle-steal mode (DCR[CS] = 1)—Only one complete transfer from source to destination occurs for each request. If DCR[EEXT] is set, a request can be either internal or external. Internal request is selected by setting DCR[START]. An external request is initiated by asserting $\overline{\text{DREQ}}$ while EEXT is set.
- Continuous mode (DCR[CS] = 0)—After an internal or external request, the DMA continuously transfers data until BCR reaches zero or a multiple of DCR[BWC] or DSR[DONE] is set. If BCR is a multiple of BWC, the DMA request signal is negated until the bus cycle terminates to allow the internal arbiter to switch masters. DCR[BWC] = 000 specifies the maximum transfer rate; other values specify a transfer rate limit.

  The DMA performs the specified number of transfers, then relinquishes bus control. The DMA negates its internal bus request on the last transfer before the BCR reaches a multiple of the boundary specified in BWC. On completion, the DMA reasserts its bus request to regain mastership at the earliest opportunity. The minimum time that the DMA loses bus control is one bus cycle.

12.5.2 Data Transfer Modes

Each channel supports dual- and single-address transfers, described in the next sections.

12.5.2.1 Dual-Address Transfers

Dual-address transfers consist of a source operand read and a destination operand write. The DMA controller module begins a dual-address transfer sequence when DCR[SAA] is cleared during a DMA request. If no error condition exists, DSR[REQ] is set.

- Dual-address read—The DMA controller drives the SAR value onto the internal address bus. If DCR[SINC] is set, the SAR increments by the appropriate number of bytes upon a successful read cycle. When the appropriate number of read cycles complete (multiple reads if the destination size is wider than the source), the DMA initiates the write portion of the transfer.

  If a termination error occurs, DSR[BES,DONE] are set and DMA transactions stop.

- Dual-address write—The DMA controller drives the DAR value onto the address bus. If DCR[DINC] is set, DAR increments by the appropriate number of bytes at the completion of a successful write cycle. The BCR decrements by the appropriate number of bytes. DSR[DONE] is set when BCR reaches zero. If the BCR is greater than zero, another read/write transfer is initiated. If the BCR is a multiple of DCR[BWC], the DMA request signal is negated until termination of the bus cycle to allow the internal arbiter to switch masters.

  If a termination error occurs, DSR[BES,DONE] are set and DMA transactions stop.

12.5.2.2 Single-Address Transfers

Single-address transfers consist of one DMA bus cycle, allowing either a read or a write cycle to occur. The DMA controller begins a single-address transfer sequence when DCR[SAA] is set during a DMA request. If no error condition exists, DSR[REQ] is set. When the channel is enabled, DSR[BSY] is set and REQ is cleared. SAR contents are then driven onto the address bus and the value of DCR[S_RW] is driven on R/$\overline{W}$. The BCR decrements on each successful address access until it is zero, when DSR[DONE] is set.

If a termination error occurs, DSR[BES,DONE] are set and DMA transactions stop.

12.5.3 Channel Initialization and Startup

Before a block transfer starts, channel registers must be initialized with information describing configuration, request-generation method, and the data block.

12.5.3.1 Channel Prioritization

The four DMA channels are prioritized in ascending order (channel 0 having highest priority and channel 3 having the lowest) or as determined by DCR[BWC]. If BWC for a DMA channel is 000, that channel has priority only over the channel immediately preceding it. For example, if DCR3[BWC] = 000, DMA channel 3 has priority over DMA channel 2 (assuming DCR2[BWC] ≠ 000) but not over DMA channel 1.

If DCR1[BWC] = DCR2[BWC] = 000, DMA 1 has priority over DMA 0 and DMA 2. DCR2[BWC] = 000 in this case does not affect prioritization.

Prioritization of simultaneous external requests is either ascending or as determined by each channel's BWC bits as described in the previous paragraphs.

12.5.3.2 Programming the DMA Controller Module

Note the following general guidelines for programming the DMA:

- No mechanism exists to prevent writes to control registers during DMA accesses.
- If the BWC of sequential channels are equal, channel priority is in ascending order.

The SAR is loaded with the source (read) address. If the transfer is from a peripheral device to memory, the source address is the location of the peripheral data register. If the transfer

is from memory to either a peripheral device or memory, the source address is the starting address of the data block. This can be any aligned byte address. In single-address mode, this data register is used regardless of transfer direction.

The DAR should contain the destination (write) address. If the transfer is from a peripheral device to memory, or memory to memory, the DAR is loaded with the starting address of the data block to be written. If the transfer is from memory to a peripheral device, DAR is loaded with the address of the peripheral data register. This address can be any aligned byte address. DAR is not used in single-address mode.

SAR and DAR change after each cycle depending on DCR[SSIZE,DSIZE,SINC,DINC] and on the starting address. Increment values can be 1, 2, 4, or 16 for byte, word, longword, or line transfers, respectively. If the address register is programmed to remain unchanged (no count), the register is not incremented after the data transfer.

BCR*n*[BCR] must be loaded with the number of byte transfers to occur. It is decremented by 1, 2, 4, or 16 at the end of each transfer, depending on the transfer size. DSR must be cleared for channel startup.

As soon as the channel has been initialized, it is started by writing a one to DCR[START] or asserting $\overline{\text{DREQ}}$, depending on the status of DCR[EEXT]. Programming the channel for internal request causes the channel to request the bus and start transferring data immediately. If the channel is programmed for external request, $\overline{\text{DREQ}}$ must be asserted before the channel requests the bus.

Changes to DCR are effective immediately while the channel is active. To avoid problems with changing a DMA channel setup, write a one to DSR[DONE] to stop the DMA channel.

12.5.4 Data Transfer

This section includes timing diagrams that illustrate the interaction of signals in DMA data transfers. It also describes auto-alignment and bandwidth control.

12.5.4.1 External Request and Acknowledge Operation

Channels 0 and 1 initiate transfers to an external module by means of $\overline{\text{DREQ}}$[1:0]. The request for channels 2 and 3 are connected internally to the UART0 and UART1 interrupt signals, respectively. If DCR[EEXT] = 1 and the channel is idle, the DMA initiates a transfer when $\overline{\text{DREQ}}$ is asserted.

Figure 12-11 shows the minimum 4-clock cycle delay from when $\overline{\text{DREQ}}$ is sampled asserted to when a DMA bus cycle begins. This delay may be longer, depending on DMA priority, bus arbitration, DRAM refresh operations, and other factors.

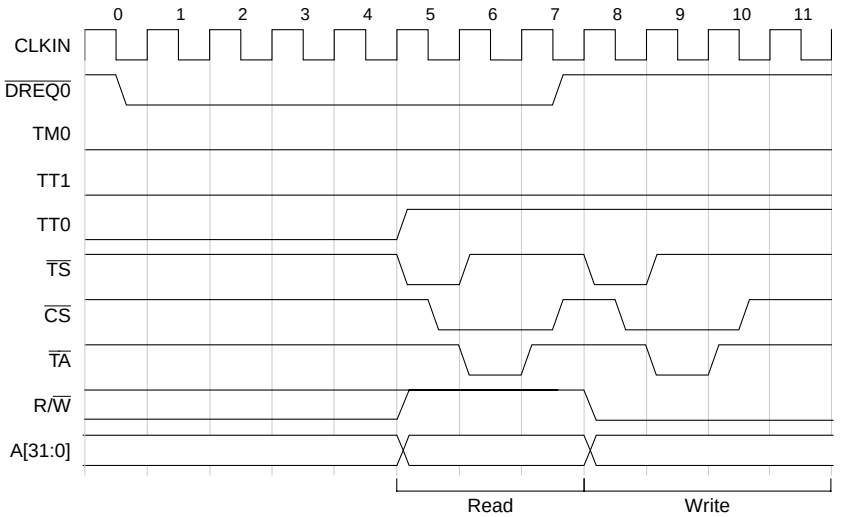

Figure 12-11. DREQ Timing Constraints, Dual-Address DMA Transfer

Although Figure 12-11 does not show TM0 signaling a DMA acknowledgement, this signal can provide an external request acknowledge response, as shown in subsequent diagrams.

To initiate a request, DREQ need only be asserted long enough to be sampled on one rising clock edge. However, note the following regarding the negation of DREQ:

- In cycle-steal mode (DCR[CS] = 1), the read/write transaction is limited to a single transfer. DREQ must be negated appropriately to avoid generating another request.
  - For dual-address transfers, DREQ must be negated before TS is asserted for the write portion, as shown in Figure 12-11, clock cycle 7.
  - For single-address transfers, DREQ must be negated before TS is asserted for the transfer, as shown in Figure 12-13, clock cycle 4.
- In burst mode, (DCR[CS] = 0), multiple read/write transfers can occur on the bus as programmed. DREQ need not be negated until DSR[DONE] is set, indicating the block transfer is complete. Another transfer cannot be initiated until the DMA registers are reprogrammed.

Figure 12-12 shows a dual-address, peripheral-to-SDRAM DMA transfer. The DMA is not parked on the bus, so the diagram shows how the CPU can generate multiple bus cycles during DMA transfers. It also shows TM0 timing. The TT signals indicate whether the CPU (0) or DMA (1) has bus mastership. TM2 indicates dual-address mode.

If DCR[AT] is 1, TM is asserted during the final transfer. If DCR[AT] is 0, TM asserts during all DMA accesses.

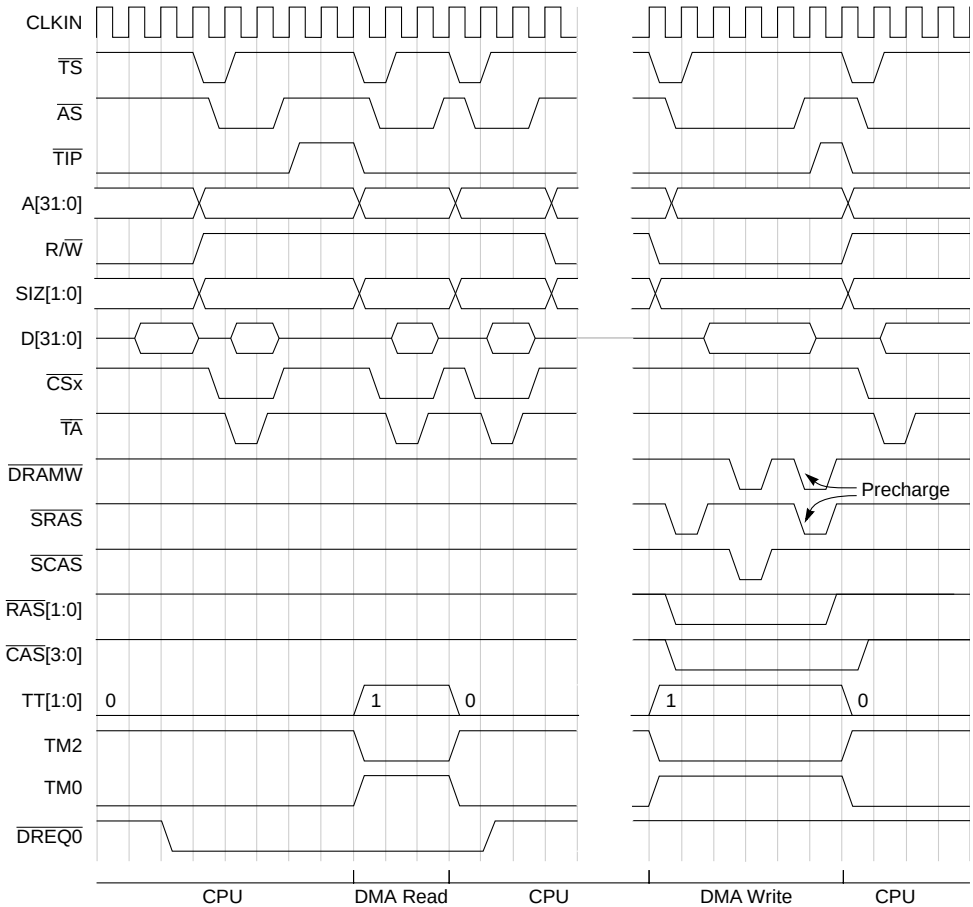

Figure 12-12. Dual-Address, Peripheral-to-SDRAM, Lower-Priority DMA Transfer

Figure 12-13 shows a single-address DMA transfer in which the peripheral is reading from memory. Note that TM2 is high, indicating a single-address transfer. Note that $\overline{\text{DREQ}}$ is negated in clock 4, before the assertion of $\overline{\text{TS}}$ in clock 6.

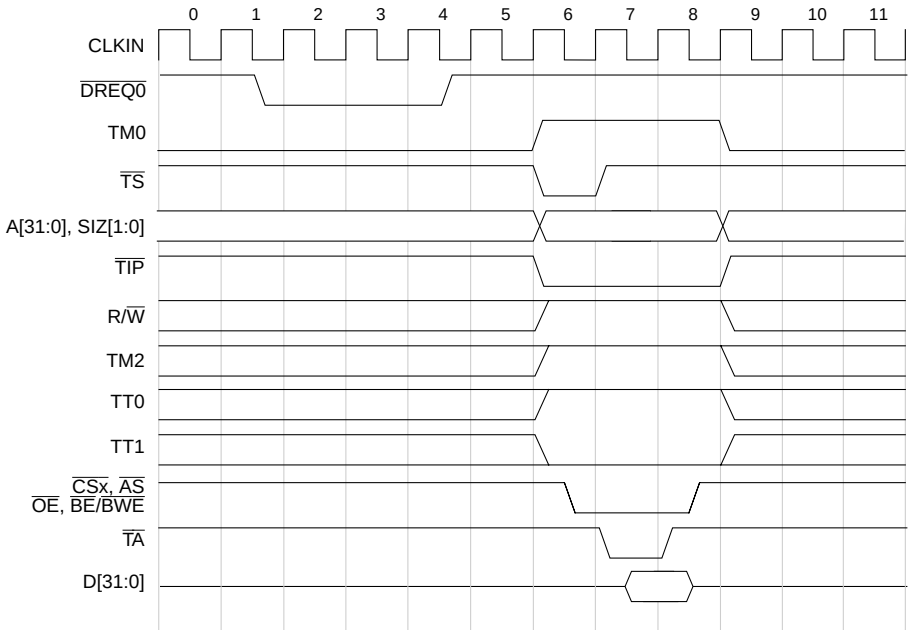

Figure 12-13. Single-Address DMA Transfer

12.5.4.2 Auto-Alignment

Auto-alignment allows block transfers to occur at the optimal size based on the address, byte count, and programmed size. To use this feature, DCR[AA] must be set. The source is auto-aligned if SSIZE indicates a transfer size larger than DSIZE. Source alignment takes precedence over the destination when the source and destination sizes are equal. Otherwise, the destination is auto-aligned. The address register chosen for alignment increments regardless of the increment value. Configuration error checking is performed on registers not chosen for alignment.

If BCR is greater than 16, the address determines transfer size. Bytes, words, or longwords are transferred until the address is aligned to the programmed size boundary, at which time accesses begin using the programmed size.

If BCR is less than 16 at the start of a transfer, the number of bytes remaining dictates transfer size. For example, AA = 1, SAR = 0x0001, BCR = 0x00F0, SSIZE = 00 (longword), and DSIZE = 01 (byte). Because SSIZE > DSIZE, the source is auto-aligned. Error checking is performed on destination registers. The access sequence is as follows:

1. Read byte from 0x0001—write 1 byte, increment SAR.
2. Read word from 0x0002—write 2 bytes, increment SAR.
3. Read longword from 0x0004—write 4 bytes, increment SAR.

4. Repeat longwords until SAR = 0x00F0.
5. Read byte from 0x00F0—write byte, increment SAR.

If DSIZE is another size, data writes are optimized to write the largest size allowed based on the address, but not exceeding the configured size.

12.5.4.3 Bandwidth Control

Bandwidth control makes it possible to force the DMA off the bus to allow access to another device. DCR[BWC] provides seven levels of block transfer sizes. If the BCR decrements to a multiple of the decode of the BWC, the DMA bus request negates until the bus cycle terminates. If a request is pending, the arbiter may then pass bus mastership to another device. If auto-alignment is enabled, DCR[AA] = 1, the BCR may skip over the programmed boundary, in which case, the DMA bus request is not negated.

If BWC = 000, the request signal remains asserted until BCR reaches zero. DMA has priority over the core. Note that in this scheme, the arbiter can always force the DMA to relinquish the bus. See Section 6.2.10.1, “Default Bus Master Park Register (MPARK).”

12.5.5 Termination

An unsuccessful transfer can terminate for one of the following reasons:

- Error conditions—When the MCF5307 encounters a read or write cycle that terminates with an error condition, DSR[BES] is set for a read and DSR[BED] is set for a write before the transfer is halted. If the error occurred in a write cycle, data in the internal holding register is lost.
- Interrupts—If DCR[INT] is set, the DMA drives the appropriate internal interrupt signal. The processor can read DSR to determine whether the transfer terminated successfully or with an error. DSR[DONE] is then written with a one to clear the interrupt and the DONE and error bits.