APR

AZ9011

Bus Master DMA Controller*

Introduction

The AZ9011 provides eight channels of Direct Memory Access (DMA). The system microprocessor programs the DMA registers via the Micro Channel® for the various modes of operation, transfer addresses and transfer counts. The DMA Controller may be programmed in one of two modes. The first programming mode emulates the operation of two 8237 DMA Controllers. The other programming mode uses extended addressing. In the DMA transfer mode, I/O devices may transfer data directly to and from memory in single transfer, burst or read verification mode. The data transfer is initiated when an I/O device requests and is granted control of the bus by the bus arbitration logic, and the DMA has been programmed to service the DMA request.

Several additional functions were included in the device to reduce the need for separate circuits in the system. These include:

- Bus Arbitration
- DRAM Refresh Logic
- Floppy Disk Interface
- Bus Master Crossover Logic

These functions will be described in separate sections within this specification.

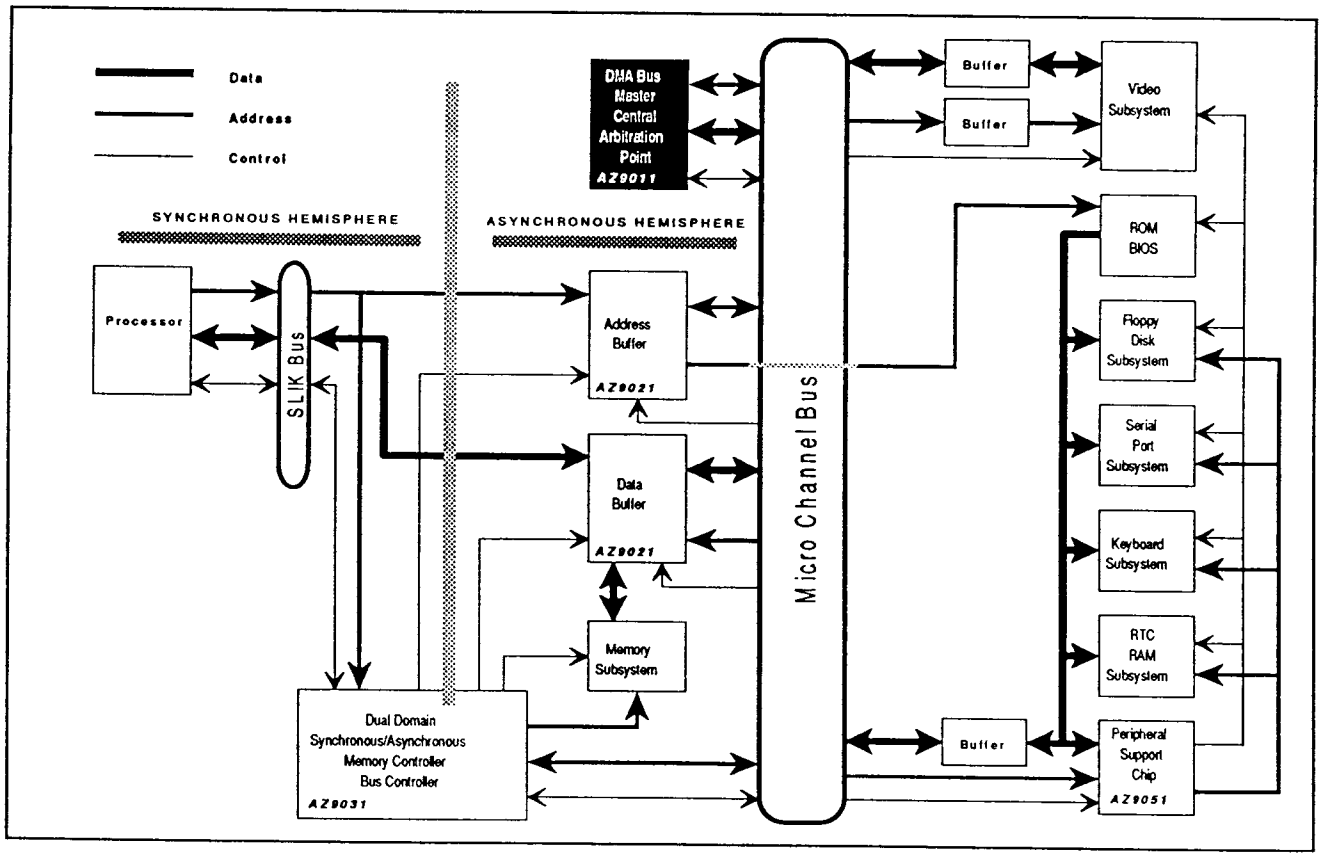


Figure 1. System Block Diagram

BMA

*Patent Pending

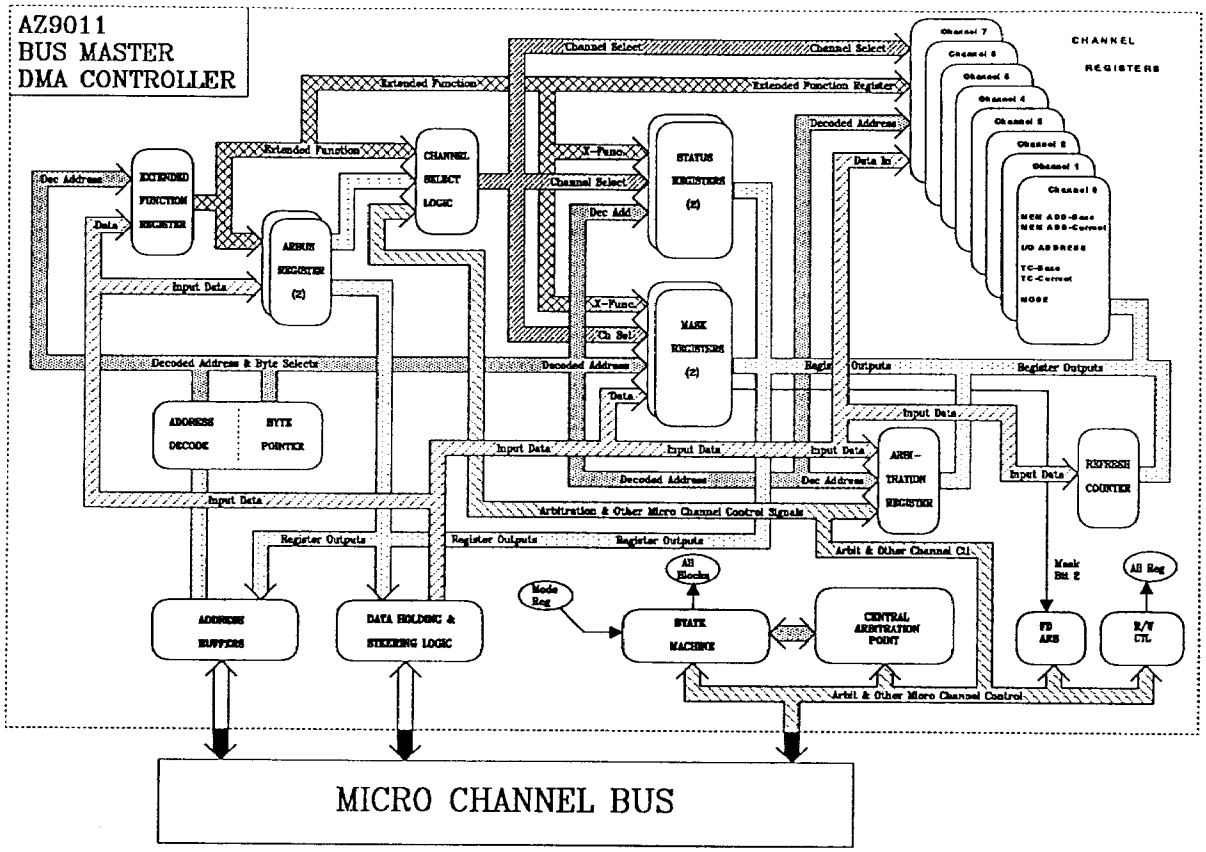


Figure 2. AZ9011 Block Diagram

SYSTEM CONFIGURATION

The AZ9011 is fully buffered and resides directly on the Micro Channel bus. During DMA transfers, it acts as a bus master and bus controller, providing all necessary control signals to facillitate the data transfer. The AZ9011 contains the data steering logic necessary to move data between memories and I/O devices of various sizes. The AZ9011 supports 8, 16 and 32 bit memories and 8 and 16 bit I/O devices.

When the CPU programs the DMA Controller, the CPU treats it like an 8 bit I/O device on the Micro Channel bus. The DMA Controller uses addresses in the range 0000H - 00DFH. In the 8237 mode, individual registers are represented by separate locations in the I/O address map. In the extended mode, one address is used to store a code which indicates the register being addressed and the function to be performed. A second address is used for the data to be read or written. This is explained in more detail in the programming discussion below.

The Bus Controller section of the AZ9011 acts as the functional control link between the DMA Controller and the Micro Channel bus. This section provides the necessary signals to control the data flow to and from the bus. It generates the control strobes necessary to facilitate communication with memories of 8, 16, or 32 bits in width. The I/O devices can have a data width of 8 or 16 bits. The devices on the channel can have data widths of 8, 16 or 32 bits. Any discrepency between the data sizes of the memory and the I/O device will be taken care of by the bus controller state machine. Data steering logic moves data to and from the appropriate byte locations.

The Bus Controller state machine in the AZ9011 first senses that an access has been requested. It then evaluates what type of access it is. The state machine evaluates the data bus width of the device which is being accessed, and determines the proper action. The software never has to be aware of data bus widths, because the bus controller automatically compensates for any differences in data widths. The AZ9011 has to broadcast to the bus, the byte request information for each individual access. If the I/O device is a different data width than the memory, the bus controller synthesizes the 4 byte enable bits as required. The bus controller calculates the number of accesses required for each transfer.

The AZ9011 drives the bus control strobes for the Micro Channel bus during the DMA transfer. A state machine also provides information internally to indicate which portion of the cycle is presently being executed. This information is necessary to the ancillary circuits which detect the data size return information, and data width information.

Operating States

The DMA controller operates in three basic states: Idle, Status and Command.

Idle: This is the state in which the microprocessor can program the DMA Controller.

Status: The DMA Controller drives the CMEM, CS0 and CS1 signals to the Micro Channel bus to indicate status.

Command: The DMA Controller controls the Micro Channel bus to produce the DMA transfer.

Operating Modes

The AZ9011 has two basic modes of operation: compatible mode and highspeed proprietary mode. The difference between the modes relates to 16 bit transfers to odd memory addresses.

In compatible mode, memory reads from an odd address are performed by doing two 8 bit memory reads followed by the 16 bit I/O write. Memory writes to an odd address are accomplished by following the 16 bit I/O read with two 8 bit memory writes. In both cases, 3 cycles are required to accomplish each transfer.

In highspeed proprietary mode, memory reads from an odd address are achieved by doing one 8 bit memory read to advance to the next even boundary followed by alternating 16 bit memory reads and 16 bit I/O writes. After the last I/O write is performed there will still be one 8 bit byte remaining which is simply discarded. Memory writes to an odd boundary are achieved by following the first 16 bit I/O read with one 8 bit memory write (to put the memory address to an even boundary). The subsequent transfers are 16 bit I/O reads followed by 16 bit memory writes to even addresses until the last memory write is performed as an 8 bit transfer to an odd location. This mode of transfer allows bursting to odd memory boundaries to take place with the number of cycles required being twice the number of transfers +one. This mode of transfer approaches being 33% more efficient than the compatible mode.

REGISTERS

The following registers are contained within the DMA:

Register	Length (bits)	Number of Registers
Memory Address (Base)	24 *	8
Memory Address (Current)	24 *	8
I/O Address	16	8
Transfer Count (Base)	16 *	8
Transfer Count (Current)	16 *	8
Mode	8	8
Status	8	2
Arbus	4	2
Mask	4	2
Temp Holding	16	1
Function	8	1
Arbitration	8	1
Refresh	12	1

*Note: These registers can be expanded to 32 bits for the Memory Address Registers and 24 bits for the Transfer Count Registers by the ADDR32 and ENLRGTCN inputs to the AZ9011 respectively. However, these expanded registers are not compatible.

Channel Registers

Each of the eight channels uses a set of four registers: Memory Address, I/O Address, Transfer Count, and Mode.

Memory Address Register: Each channel uses a set of Memory Address registers. The base register is programmed by the CPU with the starting address for the DMA transfer. The starting address is copied into the current address register at the same time. The current address register points to the memory address to be used in the next transfer and is updated (incremented or decremented) after each DMA transfer cycle. The CPU can read the status of both the base and current registers. When the autoinitialization option is selected, the base address is copied into the current address register upon completion of the transfer.

I/O Address Register: This register identifies the address for the I/O device being used for the DMA transfer. The contents of this register do not change during DMA transfers.

Transfer Count Register: The system microprocessor programs the count of transfers into the base section of this register and it is loaded into the current section. The contents of the base register do not change during DMA transfers. The current register is decremented with each transfer that is completed. When this register is decremented from 0000 to FFFF, the terminal count signal is activated. Because of this method of indicating the terminal count, this register should be loaded at setup with the value one less than the number of transfers required. When the autoinitialization option is selected, the base count is copied into the current register upon completion of the transfer.

Mode Register: The Mode register configures its associated channel for the type of operation to perform during DMA transfers. The functions programmed are the same but the methods for programming this register are different for the 8237 mode and the extended mode. This register can be read only in the extended mode.

The bit assignments for the two operating modes are as follows:

Mode Register (8237 Compatible)

Bit	Function
7,6	Reserved = 0
5	0 = Memory Address Increment 1 = Memory Address Decrement
4	0 = Autoinitialization Disabled 1 = Autoinitialization Enabled
3,2	00 = Verify Transfer 01 = Memory Write Transfer 10 = Memory Read Transfer 11 = Reserved
1,0	00 = Select Channel 0 or 4 01 = Select Channel 1 or 5 10 = Select Channel 2 or 6 11 = Select Channel 3 or 7

Mode Register (Extended Mode)

Bit	Function
7	Reserved = 0
6	0 = 8 Bit Transfer 1 = 16 Bit Transfer
5	Reserved = 0
4	0 = Memory Address Increment 1 = Memory Address Decrement
3	0 = Memory Read Transfer 1 = Memory Write Transfer
2	0 = Verify Transfer 1 = Data Transfer
1	0 = Autoinitialization Disabled 1 = Autoinitialization Enabled
0	0 = I/O Address Forced to 0000H 1 = I/O Address Taken from I/O Address Register

Common Registers

Several registers are shared among the channels or are used for other functions. These are the Status Register, Arbus Register, Mask Register, Byte Pointer, Temporary Holding Register, and Refresh Register.

Status Register: Two 8-bit Status registers are provided. One for channels 0 thru 3 and the other for channels 4 thru 7. The information in these registers tells which channels have reached a terminal count and which channels have requested the bus since the last time the status register was read (bits are cleared after each time the register is read). This register can be read using either 8237 mode or extended mode.

In the extended mode, the entire byte containing the information for the selected channel is returned. The information is retreived in conjunction with the Byte Pointer. The channel number in the Extended Function Register (address 0018H) is ignored. The value returned corresponds to the group selected by the Byte Pointer. When the Byte Pointer is cleared, it selects the status register for channels 0 - 3. Reading the status register in the extended mode toggles the Byte Pointer.

Status Register

Bit	Function
7	Channel 3 or 7 Request
6	Channel 2 or 6 Request
5	Channel 1 or 5 Request
4	Channel 0 or 4 Request
3	Terminal count for Channel 3 or 7
2	Terminal count for Channel 2 or 6
1	Terminal count for Channel 1 or 5
0	Terminal count for Channel 0 or 4

Arbus Register: The DMA Controller can work with any of the arbitration levels from 0 to 14, with eight possible channels. Channels 1 - 3 and 5 - 7 are assigned to the corresponding arbitration levels. Two 4-bit registers are provided, one register is for channel 0 and the other for channel 4. These registers provide virtual DMA operation by allowing the system microprocessor to assign the arbitration level and allow channels 0 and 4 to service devices at any arbitration level. When channel 0 or channel 4 is assigned to one of the pre-assigned levels (1-3 or 5-7), the mask bits are used to select which of the channels (or none) is to be used.

Arbus Register

Bit	Function
7-4	Reserved
3-0	Arbitration level

Mask Register: If a device requests DMA service by winning the arbitration for a channel which has the mask bit set, the DMA will not execute any DMA transfer cycles. This method may be used for a bus master to gain access to the Micro Channel bus, or to select among several DMA channels assigned to the same arbitration level. If the arbitrating device does not execute the transfer itself, a timeout will occur which will generate a non-maskable interrupt.

In the 8237 mode, two 4-bit Mask Registers are provided, one register for channels 7-4, and one for channels 3-0. In the extended mode, the individual bits are set and cleared by selecting the desired channel and using the appropriate command.

Individual mask bits may be set or cleared in either 8237 mode or extended mode. All mask bits are set by a system reset or by a DMA Controller master clear. All mask bits in the four-register group are cleared by a "Clear Mask Register" command in the 8237 mode.

The register bits are defined as:

Single Mask Register

Bit	Function
7-3	Reserved = 0
2	0 = Clear Mask Register 1 = Set Mask Register
1,0	00 = Select Channel 0 or 4 01 = Select Channel 1 or 5 10 = Select Channel 2 or 6 11 = Select Channel 3 or 7

All Mask Register

Bit	Function
7-4	Reserved = 0
3	Channel 3 or 7 0 = Clear Mask Register 1 = Set Mask Register
2	Channel 2 or 6 0 = Clear Mask Register 1 = Set Mask Register
1	Channel 1 or 5 0 = Clear Mask Register 1 = Set Mask Register
0	Channel 0 or 4 0 = Clear Mask Register 1 = Set Mask Register

Function Register: One 8-bit Function Register is provided. This receives the extended mode I/O commands and is the register which identifies the operation to be performed. The system microprocessor executes functions by first writing to the function register (address 0018) the function to be performed and the channel to use. The selected function is then executed by writing or reading port address 001A.

Address	Command
0018	Function Register
001A	Execute Function Register

The Function Register bits are assigned as follows:

Bit	Function
7-4	Command
3	Reserved = 0 (If set to 1 this function is disabled)
2-0	Channel

See Table 1 for the list of extended commands.

Command	Register	Bits	Byte Pointer
0 W/R	I/O Address Register	15-00	Yes
1 W/R	Reserved		
2 W	Base and Current Memory Address Write	23(31)-00	Yes
2 R	Base Memory Address Read	23(31)-00	Yes
3 W	Reserved		
3 W	**Test Mode 1 Base Memory Address Register Write**	23(31)-00	Yes
3 R	Current Memory Address Read	23(31)-00	Yes
4 W	Base and Current Transfer Count Write	15(23)-00	Yes
4 R	Base Transfer Count Read	15(23)-00	Yes
5 W	Reserved		
5 W	**Test Mode 1 Base Transfer Count Register Write**	15(23)-00	Yes
5 R	Current Transfer Count Read	15(23)-00	Yes
6 W	Reserved		
6 R	Status Register Read (PROGRAM COMMAND 6 DOES NOT USE CHANNEL NUMBER)	07-00	Yes
7 W/R	Extended Mode Register Write/Read		
8 W/R	Arbus Register Write/Read	04-00	
9	Mask Register Set Single Bit (Direct from Function Register)		
A	Mask Register Clear Single Bit (Direct from Function Register)		
B W	Reserved for Arbitration Test Start		
B R	Reserved		
C W	Reserved for Arbitration Test Stop		
C R	Reserved		
D	Master Clear (Direct from Function Register)		
E W/R	Reserved		
F W/R	Reserved		

Table 1. Extended Commands

Arbitration Register: This register is used to program several options for the arbitration function and to monitor the status of key parameters of the arbitration cycle. This register is located at 0090H on the I/O address map. The bit representations are different for writing and reading the register:

Arbitration Register Read Functions:

Bit	Read Function
7	1 = CPU cycles enabled during arbitration cycles 0 = CPU cycles enabled during arbitration cycles
6	1 = Arbitration masked by an NMI or Bit 6 set 0 = Normal operation
5	1 = A bus timeout has occurred 0 = Normal operation
4	1 = Interrupt request preempt enabled 0 = Normal operation
3 - 0	Value of Arbitration Bus during the previous Grant state

Arbitration Register Write Functions:

Bit	Write Function	Initial Value
7	1 = Enable CPU cycles during arbitration cycles 0 = Disable CPU cycles during arbitration cycles	0
6	1 = Force an arbitration cycle with the CPU controlling the channel until this bit is reset 0 = Continue normal arbitration operation	1
5	1 = Arbitration time set to 600 ns. minimum 0 = Arbitration time set to 300 ns. minimum	0
4	1 = Enable interrupt request preempt 0 = Normal operation	0
3 - 0	Reserved = 0	

Byte Pointer: The memory addressing range requires 24 or 32 bits of address data. The Transfer Counter uses 16 or 24 bits and I/O addresses are 16 bits. To move data on an 8-bit bus, the DMA Controller uses a Byte Pointer. Each access of a part of the system using the Byte Pointer causes it to toggle to the next state. A write to 000CH or 00D8 in 8237 mode or a master reset command will reset the Byte Pointer. In addition, any write to the extended mode Function register (0018H) will reset the Byte Pointer. When reset, the Byte Pointer will select the least significant byte of the word accessed. The next access will be to the next more significant byte. Once the most significant byte is accessed, the Byte Pointer will again toggle to the least significant byte. Note that the Byte Pointer is also used with the Status register to select between the status bits for channels 0-3 and for channels 4-7.

Temporary Holding Register: One 16-bit Temporary Holding register is provided. During a DMA transfer, data that have been read are held in this register while waiting to be written to their destination. The system microprocessor does not have access to this register.

Refresh Register: One 10-bit Refresh register is provided. This register provides memory assresses for the refresh operation. The system microprocessor does not have access to the refresh register.

Transfer Counter: The Transfer Counter is loaded from the Transfer Count register at the start of a DMA transfer. It contains a count which is one less than the number of transfers which remain. For example, if the counter contains zero, one transfer will occur. When the transfer count changes from 0000 to FFFF, the "terminal count" pulse is generated by the DMA Controller. The current value for the transfer counter is retained for each channel and may be read by the CPU.

CENTRAL ARBITRATION POINT

The AZ9011 contains all the logic necessary to implement the Central Arbitration Point (CAP) for the system. It supports up to 16 arbitrating devices and resolves bus contention between devices in an orderly fashion.

CAP Functions

The CAP is forced into the following functional states by the following qualifications:

CLEAR GRANT

1. If the bus is granted to other than the processor, and a transfer has been completed, and there has been sufficient time since the last Grant state was entered.
2. If arbitration is forced, and there is not a bus timeout, and a transfer has been completed.
3. If a bus timeout has occurred.
4. If the bus is granted to the processor, and a preempt is pending, and there has been sufficient time since the last Grant state was entered.

SET GRANT

1. If ARB is not forced, and the minimum arbitration time has passed, and LATCHED CHACK is active, and REFRESH PREEMPT is not active, and a refresh is not in progress.
2. If ARB is not forced, and the minimum arbitration time has passed, and LATCHED CHACK is not active, and REFRESH PREEMPT is not active, and a refresh is not in progress, and the ARBUS equals F.

SET REQCH

1. If ARB is not forced, and processor cycles during arbitration is not allowed, and the CAP is in arbitration, and there is a preempt pending (which is to say it was a preempt which was the cause of the arbitration), and the CAP is not at the point of awarding the bus to the processor.
2. If arbitration is occurring, a refresh is pending, and the CAP is not at the point of awarding the bus to the processor.
3. If ARB is not forced, and processor cycles during arbitration are allowed, and the minimum arbitration time has passed, and the CAP is in arbitration, and the ARBUS equals F, and there is no LATCHED CHACK.

CLEAR REQCH

1. If arbitration timeout has occurred, and the CAP is in arbitration, and the ARBUS equals F, and a refresh is not in progress.

2. If arbitration is forced, and the CAP is in arbitration, and a refresh is not in progress.
3. If processor cycles during arbitration are allowed, arbitration timeout has not occurred, and the CAP is in arbitration.

Bus Arbitration

Arbitration of the contention for access to the Micro Channel bus consists of allowing all slave devices requesting access to issue their requests in parallel and then selecting the device with the highest priority to perform its transfer. The sequence is:

1. A device requests access by pulling the PREEMPTN line low.
2. The DMA Controller begins the arbitration cycle by pulling GRANTN high.
3. The DMA Controller requests access to the Micro Channel bus from the system by pulling the REQCH line high.
4. The system completes its bus cycle, releases the bus and signals the DMA controller by issuing CHACK.
5. All devices requesting service drive the Arbitration bus with their priority code. They adjust until the code on the bus matches the code of the device with the highest priority (lowest number).
6. After allowing time for the Arbitration bus to stabilize, the DMA Controller pulls the GRANTN signal low to lock in the code of the device to be serviced. This signals the winning device that DMA transfers may begin.

ADDITIONAL FUNCTIONS

Several additional functions are provided within the AZ9011.

DRAM Refresh Logic

DRAM refresh is controlled by the DMA Controller. This is performed as if it were a request from a peripheral device for DMA service. The sequence is:

1. The AZ9011 divides the 14.3 MHz clock in order to generate an internal request for refresh every 15 microseconds.
2. Upon receiving the internal request for refresh, the AZ9011 begins the arbitration cycle by pulling GRANT high.
3. The DMA Controller requests access to the Micro Channel bus from the system by pulling the REQCH line high.
4. The system completes its bus cycle, releases the bus and signals the DMA controller by issuing CHACK.
5. The refresh request automatically wins the arbitration. The Grant mode is not entered.
6. The refresh address is transferred to the address bus and REFRESHN and Memory Read signals are issued. Wait states are introduced by external logic as needed by pulling IOCHRDY low.
7. The AZ9011 completes the refresh cycle, pulls GRANT and CHACK low, returning control of the Micro Channel bus to the system.

Floppy Disk Interface

The floppy disk subsection requests service from the DMA by issuing a Floppy Disk Request signal. The DMA contains the logic necessary to accept the request signal, arbitrate for the Micro Channel bus on behalf of the floppy disk and return a Floppy Disk Acknowledge handshake signal to indicate when the channel has been granted to the floppy disk.

Bus Master Data Crossover Logic

When a 16 bit bus master writes to a 32 bit memory, the even words are transferred directly. Bits 0 through 15 on the bus go to bits 0 through 15 in memory. When the odd words are written, it is necessary to cross the data on bits 0 through 15 on the bus over to bits 16 through 31 in memory. The AZ9011 contains the logic to sense

the condition of a 16 bit bus master writing to a 32 bit memory and intervenes to provide both the data crossover and the appropriate byte enables.

Bull Diagnostic Bus

The Bull Extended Test (BULLET) system provides the necessary viewports to efficiently verify new ASIC designs. The BULLET bus protocol has been established to allow future diagnostic growth and enhanced manufacturing test for current and future Bull ASIC chip families. See Bull Diagnostic Bus Specification.

PROGRAMMING

This section describes how to configure the DMA controller to perform transfers.

8237 Mode

The DMA Controller contains logic to model two 8237 DMA Controllers. When this mode is used for programming, the registers are accessed by selecting individual addresses and reading from or writing to them. The DMA Controller provides full 16-bit address decoding for the I/O bus.

A complete list of addresses and their functions can be found in Table 2, starting on page 11.

Extended Mode

In the extended mode, two addresses are used to access all registers and bits. The first address (0018), known as the Extended Function Register, serves as a pointer to the desired channel register or bit and identifies the function to be performed. The second address (001A) is used to pass data, if required. Refer to the Function Register description above and Table 1, found on page 6 for details on the function codes.

Autoinitialization

The DMA Controller autoinitialization feature allows the programmer to reduce the overhead associated with setup for DMA transfers by automatically initializing the current Memory Address register and the Terminal Count register with the values in their corresponding base registers after completion of a DMA transfer. To do this, the autoinitialization bit is set in the Mode register.

Arbitration Test

When ENDIAGBC is active (high), a write to the Extended Function Register (0018 Hex) with command data equal to B Hex will cause the AZ9011 to preempt for the bus and arbitrate with the value that is programmed in channel four's Arbus Register. The AZ9011 will issue -PREEMPT. When the CAP enters the arbitrate state, the AZ9011 will drive the appropriate CARB(3:0) lines and compete for the bus. When the AZ9011 gains the bus, -PREEMPT will be released. No control signals are issued when the AZ9011 is granted the bus.

When ENDIAGBC is active (high), a write to the Extended Function Register with command data equal to C Hex will stop the Arbitration Test that was started with a write to the Extended Function Register (0018 Hex) with command data equal to B Hex.

When ENDIAGBC is inactive (low), writes to the Extended Function Register with command data equal to B or C Hex will result in no action.

This Arbitration Test feature is not intended to be used by the end user of a system that employs the AZ9011. It is intended to be used by the system designer/ manufacturer to aid in system diagnostics.

Table 2. Address Decoding

Address	Function		Byte Pointer
0000	W	Base and Current Memory Address Write Channel 0	YES
0000	R	Current Memory Address Read Channel 0	YES
0001	W	Base and Current Transfer Count Write Channel 0	YES
0001	R	Current Transfer Count Read Channel 0	YES
0002	W	Base and Current Memory Address Write Channel 1	YES
0002	R	Current Memory Address Read Channel 1	YES
0003	W	Base and Current Transfer Count Write Channel 1	YES
0003	R	Current Transfer Count Read Channel 1	YES
0004	W	Base and Current Memory Address Write Channel 2	YES
0004	R	Current Memory Address Read Channel 2	YES
0005	W	Base and Current Transfer Count Write Channel 2	YES
0005	R	Current Transfer Count Read Channel 2	YES
0006	W	Base and Current Memory Address Write Channel 3	YES
0006	R	Current Memory Address Read Channel 3	YES
0007	W	Base and Current Transfer Count Write Channel 3	YES
0007	R	Current Transfer Count Read Channel 3	YES
0008	W	Not Implemented	
0008	R	Channel 0-3 Status Register Read	
0009	W	Not Implemented	
0009	R	Not Implemented	
000A	W	Channel 0-3 Single Mask Register Write	
000A	R	Not Implemented	
000B	W	Channel 0-3 Mode Register Write	
000B	R	Not Implemented	
000C	W	Clear Byte Pointer Flip/Flop	
000C	R	Not Implemented	
000D	W	Master Clear	
000D	R	Not Implemented	
000E	W	Channel 0-3 Mask Register Clear	
000E	R	Not Implemented	
000F	W	Channel 0-3 All Mask Register Bits Write	
000F	R	Not Implemented	
000F	**R**	**Test Mode 1 Read Entire Mask Register**	
0018	W	Extended Function Register	
0018	R	Not Implemented	
0018	**R**	**Test Mode 1 Read Extended Function Register**	
0019	W/R	Reserved	
001A	W/R	Extended Function Execute	**If Required**
001B	W/R	Reserved	
0080	W/R	Not Implemented	
0080	**W**	**Test Mode 1 Write Refresh Counter**	
0080	**R**	**Test Mode 1 Read Refresh Counter**	YES
0081	W/R	Channel 2, Page Table Address Register	
0082	W/R	Channel 3, Page Table Address Register	
0083	W/R	Channel 1, Page Table Address Register	
0084	W/R	Not Implemented	
0084	**W**	**Test Mode 1 Count Refresh Counter**	

0085	W/R	Not Implemented	
0086	W/R	Not Implemented	
0087	W/R	Channel 0, Page Table Address Register	
0088	W/R	Not Implemented	
0088	**W**	**Test Mode 1 Write Address Decode Select Register**	
0089	W/R	Channel 6, Page Table Address Register	
008A	W/R	Channel 7, Page Table Address Register	
008B	W/R	Channel 5, Page Table Address Register	
008C	W/R	Not Implemented	
008D	W/R	Not Implemented	
008E	W/R	Not Implemented	
008F	W/R	Channel 4, Page Table Address Register	
0090	W	Arbitration Register Write	
0090	R	Arbitration Register Read	
00C0	W	Base and Current Memory Address Write Channel 4	YES
00C0	R	Current Memory Address Read Channel 4	YES
00C1	W/R	Not Implemented	
00C2	W	Base and Current Transfer Count Write Channel 4	YES
00C2	R	Current Transfer Count Read Channel 4	YES
00C3	W/R	Not Implemented	
00C4	W	Base and Current Memory Address Write Channel 5	YES
00C4	R	Current Memory Address Read Channel 5	YES
00C5	W/R	Not Implemented	
00C6	W	Base and Current Transfer Count Write Channel 5	YES
00C6	R	Current Transfer Count Read Channel 5	YES
00C7	W/R	Not Implemented	
00C8	W	Base and Current Memory Address Write Channel 6	YES
00C8	R	Current Memory Address Read Channel 6	YES
00C9	W/R	Not Implemented	
00CA	W	Base and Current Transfer Count Write Channel 6	YES
00CA	R	Current Transfer Count Read Channel 6	YES
00CB	W/R	Not Implemented	
00CC	W	Base and Current Memory Address Write Channel 7	YES
00CC	R	Current Memory Address Read Channel 7	YES
00CD	W/R	Not Implemented	
00CE	W	Base and Current Transfer Count Write Channel 7	YES
00CE	R	Current Transfer Count Read Channel 7	YES
00CF	W/R	Not Implemented	
00D0	W	Not Implemented	
00D0	R	Channel 4-7 Status Register Read	
00D1	W/R	Not Implemented	
00D2	W	Not Implemented	
00D2	R	Not Implemented	
00D3	W/R	Not Implemented	
00D4	W	Channel 4-7 Single Mask Register Write	
00D4	R	Not Implemented	
00D5	W/R	Not Implemented	
00D6	W	Channel 4-7 Mode Register Write	
00D6	R	Not Implemented	
00D7	W/R	Not Implemented	
00D8	W	Clear Byte Pointer Flip/Flop	

00D8	R	Not Implemented
00D9	W/R	Not Implemented
00DA	W	Master Clear
00DA	R	Not Implemented
00DB	W/R	Not Implemented
00DC	W	Channel 4-7 Mask Register Clear
00DC	R	Not Implemented
00DD	W/R	Not Implemented
00DE	W	Channel 4-7 All Mask Register Bits Write
00DE	R	Not Implemented
00DE	**R**	**Test Mode 1 Read Entire Mask Register**
00DF	W/R	Not Implemented

Status Register (Bits Are Reset On Read) **(Read Only)**

Bit 0	1 Channel 0 or 4 Has Reached TC
Bit 1	1 Channel 1 or 5 Has Reached TC
Bit 2	1 Channel 2 or 6 Has Reached TC
Bit 3	1 Channel 3 or 7 Has Reached TC
Bit 4	1 Channel 0 or 4 Request
Bit 5	1 Channel 1 or 5 Request
Bit 6	1 Channel 2 or 6 Request
Bit 7	1 Channel 3 or 7 Request

Single Mask Register (Set On Master Clear) **(Write Only)**

Bits 1-0	00 Select Channel 0 or 4 Mask Bit
	01 Select Channel 1 or 5 Mask Bit
	10 Select Channel 2 or 6 Mask Bit
	11 Select Channel 3 or 7 Mask Bit
Bit 2	0 Reset Mask Bit
	1 Set Mask Bit
Bits 7-3	Not Implemented

All Mask Register Bits (Write Only)

Bit 0	0 Clear Channel 0 or 4 Mask Bit
	1 Set Channel 0 or 4 Mask Bit
Bit 1	0 Clear Channel 1 or 5 Mask Bit
	1 Set Channel 1 or 5 Mask Bit
Bit 2	0 Clear Channel 2 or 6 Mask Bit
	1 Set Channel 2 or 6 Mask Bit
Bit 3	0 Clear Channel 3 or 7 Mask Bit
	1 Set Channel 3 or 7 Mask Bit
Bits 7-4	Not Implemented

Mode Register (8237 Compatible) **(Write Only)**

Bits 1-0	00 Select Channel 0 or 4
	01 Select Channel 1 or 5
	10 Select Channel 2 or 6
	11 Select Channel 3 or 7
Bits 3-2	00 Verify Transfer
	01 Write Transfer
	10 Read Transfer
	11 Reserved
Bit 4	0 Autoinitialization Disabled
	1 Autoinitialization Enabled
Bit 5	0 Memory Address Increment Select
	1 Memory Address Decrement Select
Bits7-6	Not Implemented

Mode Register (Extended Mode)

Bit 0	0 I/O Adderss Equals 0000H
	1 Use Programmed I/O Address
Bit 1	0 Autoinitialization Disabled
	1 Autoinitialization Enabled
Bit 2	0 Verify Transfer
	1 Data Transfer
Bit 3	0 Memory Read Transfer
	1 Memory Write Transfer
Bit 4	0 Memory Address Increment Select
	1 Memory Address Decrement Select
Bit 5	Not Implemented **(Writable And Readable)**
Bit 6	0 8 Bit Transfer
	1 16 Bit Transfer
Bits7	Not Implemented **(Writable And Readable)**

Register Definitions

Arbus Register

	Bits 7 - 4	Bits 3 - 0
Ch. 0 Read	Ch. 4's Arbitration Level	Ch. 0's Arbitration Level
Ch. 0 Write	Not Writable	Ch. 0's Arbitration Level 0 - F Hex
Ch. 1 Read	Ch. 4's Arbitration Level	1 Hex
Ch. 1 Write	Not Writable	Not Writable
Ch. 2 Read	Ch. 4's Arbitration Level	2 Hex
Ch. 2 Write	Not Writable	Not Writable
Ch. 3 Read	Ch. 4's Arbitration Level	3 Hex
Ch. 3 Write	Not Writable	Not Writable
Ch. 4 Read	Ch. 4's Arbitration Level	Ch. 4's Arbitration Level
Ch. 4 Write	Not Writable	Ch. 4's Arbitration Level 0 - F Hex
Ch. 5 Read	Ch. 4's Arbitration Level	5 Hex
Ch. 5 Write	Not Writable	Not Writable
Ch. 6 Read	Ch. 4's Arbitration Level	6 Hex
Ch. 6 Write	Not Writable	Not Writable
Ch. 7 Read	Ch. 4's Arbitration Level	7 Hex
Ch. 7 Write	Not Writable	Not Writable

Arbitration Register Write

Bits 3-0	Reserved = 0 (Not Implemented)
Bit 4	Enable Interrupt Request Preempt **Default = Disabled**
Bit 5	Enable Extended Arbitration **Default = Disabled**
Bit 6	Arbitration Mask **Default = Enabled**
Bit 7	Enable System CPU Cycles **Default = Disabled**

Arbitration Register Read

Bits 3-0	Value of Arbitration Bus During Previous Grant State
Bit 4	Interrupt Request Preempt Enabled
Bit 5	Bus Timeout Occured
Bit 6	Arbitration Masked by NMI or Bit 6 Set
Bit 7	System CPU Cycles Enabled

Entire Mask Register Read

Bit 0	Ch. 0 Mask Bit
Bit 1	Ch. 1 Mask Bit
Bit 2	Ch. 2 Mask Bit
Bit 3	Ch. 3 Mask Bit
Bit 4	Ch. 4 Mask Bit
Bit 5	Ch. 5 Mask Bit
Bit 6	Ch. 6 Mask Bit
Bit 7	Ch. 7 Mask Bit

Refresh Counter

The refresh counter is a ten bit up counter. The ten bits of the counter are enabled on the address bus during refresh. The refresh counter is counted once after every refresh.

Refresh Counter Bit	Address Bit
Bit 9	Bit 11
Bit 1	Bit 10
Bit 0	Bit 9
Bit 8	Bit 8
Bit 7	Bit 7
Bit 6	Bit 6
Bit 5	Bit 5
Bit 4	Bit 4
Bit 3	Bit 3
Bit 2	Bit 2
Bit 1	Bit 1
Bit 0	Bit 0

Write Refresh Counter

Data Bus Bits 9-0 Are Written To The Refresh Counter.

Read Refresh Counter (8237 Mode Byte Pointer Toggle)

Refresh Counter Bits 7-0 Are Read When The Byte Pointer Is Byte 0.

Refresh Counter Bits 9-8 Are Read When The Byte Pointer Is Byte 1.

Count Refresh Counter

An I/O write to this address will count the refresh counter once.

Address Decode Select Register

Bits 1-0 = 0 Hex.	D15 = 000F Hex. D14 = 000E Hex. D13 = 000D Hex. D12 = 000C Hex. D11 = 000B Hex. D10 = 000A Hex. D09 = 0009 Hex. D08 = 0008 Hex. D07 = 0007 Hex. D06 = 0006 Hex. D05 = 0005 Hex. D04 = 0004 Hex. D03 = 0003 Hex. D02 = 0002 Hex. D01 = 0001 Hex. D00 = 0000 Hex.

Bits 1-0 = 1 Hex.	D15 = 008F Hex. D14 = 008E Hex. D13 = 008D Hex. D12 = 008C Hex. D11 = 008B Hex. D10 = 008A Hex. D09 = 0089 Hex. D08 = 0088 Hex. D07 = 0087 Hex. D06 = 0086 Hex. D05 = 0085 Hex. D04 = 0084 Hex. D03 = 0083 Hex. D02 = 0082 Hex. D01 = 0081 Hex. D00 = 0080 Hex.

Bits 1-0 = 2 Hex.	D15 = 00DE Hex. D14 = 00DC Hex. D13 = 00DA Hex. D12 = 00D8 Hex. D11 = 00D6 Hex. D10 = 00D4 Hex. D09 = 00D2 Hex. D08 = 00D0 Hex. D07 = 00CE Hex. D06 = 00CC Hex. D05 = 00CA Hex. D04 = 00C8 Hex. D03 = 00C6 Hex. D02 = 00C4 Hex. D01 = 00C2 Hex. D00 = 00C0 Hex.

Bits 1-0 = 3 Hex.	D15 = Not Implemented D14 = Not Implemented D13 = Not Implemented D12 = Not Implemented D11 = Not Implemented D10 = Not Implemented D09 = Not Implemented D08 = Not Implemented D07 = CHANNEL 4 ARBUS COMPARE D06 = CHANNEL 0 ARBUS COMPARE D05 = Tied High D04 = Tied High D03 = Tied High D02 = 0090 Hex. D01 = 001A Hex. D00 = 0018 Hex.

Bits 3-2	Not Used
Bits 7-4	Not Implemented

Diagnostic Multiplexer

An eight by eight diagnostic multiplexer has been implemented. DIAGIN2, DIAGIN1 and DIAGIN0 provide an octal select for internal signals to be multiplexed out of the chip on DIAG7, DIAG6, DIAG5, DIAG4, DIAG3, DIAG2, DIAG1 and DIAG0.

DIAGIN[2:0] = 0 Oct.	DIAG7 = INTINTAN DIAG6 = INTMEMWN DIAG5 = IOWRITE DIAG4 = IOREAD DIAG3 = BYT3 DIAG2 = BYT2 DIAG1 = BYT1 DIAG0 = BYT0

DIAGIN[2:0] = 1 Oct.	DIAG7 = MASKBIT7 DIAG6 = MASKBIT6 DIAG5 = MASLBIT5 DIAG4 = MASKBIT4 DIAG3 = MASKBIT3 DIAG2 = MASKBIT2 DIAG1 = MASKBIT1 DIAG0 = MASKBIT0

DIAGIN[2:0] = 2 Oct.
- DIAG7 = MADDREGWR
- DIAG6 = ADDREGDFLOPLOADN
- DIAG5 = CNTADRNDLY2N
- DIAG4 = CNTADRNDLY1N
- DIAG3 = CNTADRN
- DIAG2 = AUTOINITNDLYN
- DIAG1 = AUTOINITN
- DIAG0 = DMAMASKSET

DIAGIN[2:0] = 3 Oct.
- DIAG7 = PRSELECT
- DIAG6 = ARSELECT
- DIAG5 = WCSELECT
- DIAG4 = TCREGWR
- DIAG3 = TCRDFLOPLOADN
- DIAG2 = DECWCDLY2N
- DIAG1 = DECWCDLY1N
- DIAG0 = DECWCN

DIAGIN[2:0] = 4 Oct.
- DIAG7 = READSELECTEN
- DIAG6 = DMA
- DIAG5 = LOADN
- DIAG4 = DMALOADN
- DIAG3 = LOWTOLOW
- DIAG2 = HIGHISHIGH
- DIAG1 = LOWISLOW
- DIAG0 = DMALE2

DIAGIN[2:0] = 5 Oct.
- DIAG7 = ADDR
- DIAG6 = IOADDR
- DIAG5 = Tied Low
- DIAG4 = Tied Low
- DIAG3 = Tied Low
- DIAG2 = Tied Low
- DIAG1 = Tied Low
- DIAG0 = Tied Low

DIAGIN[2:0] = 6 Oct.
- DIAG7 = Tied Low
- DIAG6 = Tied Low
- DIAG5 = Tied Low
- DIAG4 = STMCYCLE5
- DIAG3 = STMCYCLE4
- DIAG2 = STMCYCLE3
- DIAG1 = STMCYCLE2
- DIAG0 = STMCYCLE1

DIAGIN[2:0] = 7 Oct.
- DIAG7 = RESTRTPLS
- DIAG6 = ODD
- DIAG5 = STMCP6
- DIAG4 = STMCP5
- DIAG3 = STMCP4
- DIAG2 = STMCP3
- DIAG1 = STMCP2
- DIAG0 = STMCP1

Modes of Operation

Normal Operating Mode

TESTN	= 1
T0	= 1
T1	= 1
T2	= 1
ENCAP	= 1
ENREFRSH	= 1

Test Modes

Test Mode 0 (Enable Address Decode On The Data Bus)

TESTN = 0
T0 = 0

Test Mode 1 (Read And Write Extra Registers)

TESTN = 0
T0 = 1

Parametric Test And Turn Off All Drivers Except PAROUT

T1 = 0

Other Operating Modes

Proprietary versus compatible mode

T2 = 0 DMA controller in compatible mode
= 1 DMA controller in proprietary mode

Disable Internal Central Arbitration Point (This Allows For An External Central Arbitration Point)

ENCAP = 0

Disable Internal Refresh Request (If No External Refresh Request is Provided This Disables Refresh)

ENREFRSH = 0

Micro Channel Interface

D31 - D0: These signals make up the bi-directional, active high, three state Data Bus.

A15 - A0: Address lines 15 through 0. These are bi-directional, active high, three state signals are used to address the memory and I/O devices and used as inputs to address the DMA registers.

A31 - A16: Address lines 31 through 16. These are active high, three state outputs are used to address the memory and I/O. Note that these addresses are not used as inputs to the DMA Controller.

CADLN: Address Decode Latch. This active low bi-directional signal is used by channel devices to latch valid address and status bits.

CCMDN: Channel Command. An active low bi-directional signal used to qualify data on the data bus.

CMEM: Channel Memory / -I/O. An active high bi-directional signal is used to differentiate between Memory (high) and I/O (low) operations.

CS0N: Channel Status Bit 0. An active low bi-directional signal used to differentiate between I/O and Memory writes and reads.

CS1N: Channel Status Bit 1. An active low bi-directional signal used to differentiate between I/O and Memory writes and reads.

CS0N, CS1N, and CMEM are interpreted as follows:

CMEM	CS0N	CS1N	Operation
0	0	1	I/O Write
0	1	0	I/O Read
1	0	1	Memory Write
1	1	0	Memory Read

CBE3N - CBE0N: Channel Bus Enables. These active low bi-directional signals are used during 32-bit data transfers to indicate which bytes will be valid on the bus.

CBHEN: Channel Byte High Enable. This is active low bi-directional signal is used to enable transfer of data on the high byte of the data bus.

IOCHRDY: I/O Channel Ready (Micro Channel signal - CD CHRDY). This input is used by a channel device to extend the time needed to complete an operation.

DS16N: Data Size 16 Return. This active low input allows channel bus masters to monitor the data size information. When active, it indicates that the device being accessed has a data width of 16 bits.

DS32N: Data Size 32 Return. This active low input allows channel bus masters to monitor the data size information. When active, it indicates that the device being accessed has a data width of 32 bits.

CMADE24: Channel Memory Address Enable 24. This output indicates that the most significant 8 bits of address are all equal to zero. Indicates that the address is under 16 MBytes.

ARBITRATION BUS: The following signals are used in the arbitration cycle:

CARB3 - CARB0: Arbitration Bus. These bi-directional signals are driven high with the priority level of the arbitrating bus participants.

BURSTN: Burst. This active low input indicates that the winner of the arbitration is requesting operation in the burst mode.

BUSTON: Bus Time-out. An active low output which indicates an arbitration time-out has occurred.

GRANTN: Grant. An active low bi-directional signal indicating the end of an arbitration cycle and authorizing the winning DMA device to begin its DMA cycle.

PREEMPTIN: Preempt Input. This active low input is driven by any bus participant to request use of the DMA channel.

PREEMPTN: Preempt. An active low output indicating a bus arbitration request generated within the DMA Controller. The possible sources of this signal are a refresh request or an interrupt request.

TCN: Terminal Count. An active low output indicating that the transfer count of the current DMA transfer has reached zero.

DMA Controller Signals

DMACSN: DMA Chip Select. An active low output which indicates that the DMA Controller is responding to a command directed to it. This signal is used to control bus drivers to properly direct the data bus signals.

BUSYN: DMA Busy. An active low output which indicates that a DMA or refresh transfer is in progress.

REQCH: Request Channel. This output indicates to the synchronous hemisphere of the system that the DMA Controller is ready to begin a data transfer.

CHACK: Channel Acknowledge. This input is the handshaking signal from the synchronous hemisphere indicating that the channel is free and the DMA transfer can begin.

MASK2: Mask Register bit 2. This active high output indicates that the mask bit for channel 2 is set. This signal may used by external logic to disable the arbitration for this channel when the mask bit is set.

NMIN: Non-maskable Interrupt. A bi-directional, three-state signal used in an open collector configuration. This signal is generated by the DMA Controller at the end of an arbitration time-out to indicate an error on the bus. As an input, this signal forces the central arbitration control point into the forced arbitration mode.

GCSENAN: Global Chip Select Enable. Active low input which enables the internal address decode for access to internal registers. Normally tied to the system CHACK.

ADDR32: This active high input expands the Memory Address Registers of the DMA Controller from 24 bits to 32 bits. Non-compatible.

ENLRGTCN: This active high input expands the Transfer Count Registers of the DMA Controller from 16 bits to 24 bits. Non-compatible.

TR32: A bi-directional tri-state driver that is driven low by the AZ9011 to indicate that the DMA contoller is a 32 bit bus master. This line is monitored by the bus master crossover logic to determine if a crossover needs to be supplied for 16 bit bus masters.

Memory Controller Interface

The following signals are used to execute the DRAM refresh:

REFRESHN: Refresh. An active low output indicating that a refresh operation is in progress.

FASTREFN: Fast Refresh. An active low input used to select the fast refresh rate used for testing.

Floppy Disk Interface

FDREQ: Floppy Disk Request. This input indicates a request for DMA service from the system board floppy disk controller.

FDCDACKN: Floppy Disk Acknowledge. This active low output indicates a grant of the DMA service request to the system board floppy disk controller.

FDPREMN: Floppy Disk Preempt Request. This active low output is generated when the system board floppy disk controller has a DMA request pending and the bus is in use. This pin must be tied to the Micro Channel PREEMPT line.

Miscellaneous

CLKDMA: This active high input is the 40 Mhz clock that provides timing for the DMA transfers

CLK143: This active high input is the 14.3 Mhz clock generated on the system board and used by the DMA to generate timing for the refresh circuits.

INTR: Interrupt Request. This active high input is used to indicate the presence of any interrupt. This signal is the logical OR of all interrupt requests.

ENXDLN: This active low input enables data transfer on the upper 16 bits of the data bus.

DMARSTN: System Reset. Active low.

MBCMDDLYN: An active low input necessary to provide the channel command timing required by the bus master crossover logic.

DLYBUSTON: An active low input which when driven low will prevent the Bus Time-out (BUSTON) signal from appearing until after DLYBUSTON is driven back to a one.

Diagnostic / Test

ENDIAGBC: An active high input which enables the DMA to respond to diagnostic commands for testing the bus time out and arbitration circuitry.

DOREF: Test pin that indicates that a refresh is in progress for arbitration test purposes.

REFREQN: Test point to request refresh from an external source. Must be left unconnected in normal operation.

DIS100ARB: This input disables the 100 nanosecond arbitration period option if pulled high. Tied high in normal operation.

ENREFRSH: This input enables the internal refresh arbitration circuitry. Provided as a system diagnostic feature. Tied high in normal operation.

ENCAP: Enable Central Arbitration Point. Test and diagnostic input which enables the central arbitration function in the chip. Tied high in normal operation.

EOT: This output indicates an end of transfer condition which allows arbitration to proceed.

DLYGNTN: Delay Grant. If this input is high, the grant for a bus access will be delayed during the arbitration sequence. Tied low in normal operation.

DLYARBN: Delay Arbitration. If this input is high, the Grant state will be extended and the next arbitration cycle will be delayed. Tied low in normal operation.

DIAG 7-DIAG0: Diagnostic Output Bus.

DIAGIN 2-DIAGIN0: Diagnostic Inputs.

TESTN: This active low input is used for testing only and is tied high for normal operation.

T0: This active high input is used for testing only and is tied high for normal operation.

T1: When low, this input turns off all drivers except EOT for parametric testing.

T2: This input determines if the DMA Controller is in High Speed Mode (high) or Compatible Mode (low).

Bull Diagnostic Bus

TDI: Test Data In. Used as the serial input for Bull Extended Test mode.

TDO: Test Data Out. Used as the serial output for Bull Extended Test mode.

TCLK: Test Clock. Controlled by automatic test equipment when verifying new designs or for enhanced manufacturing tests when in Bull Extended Test mode.

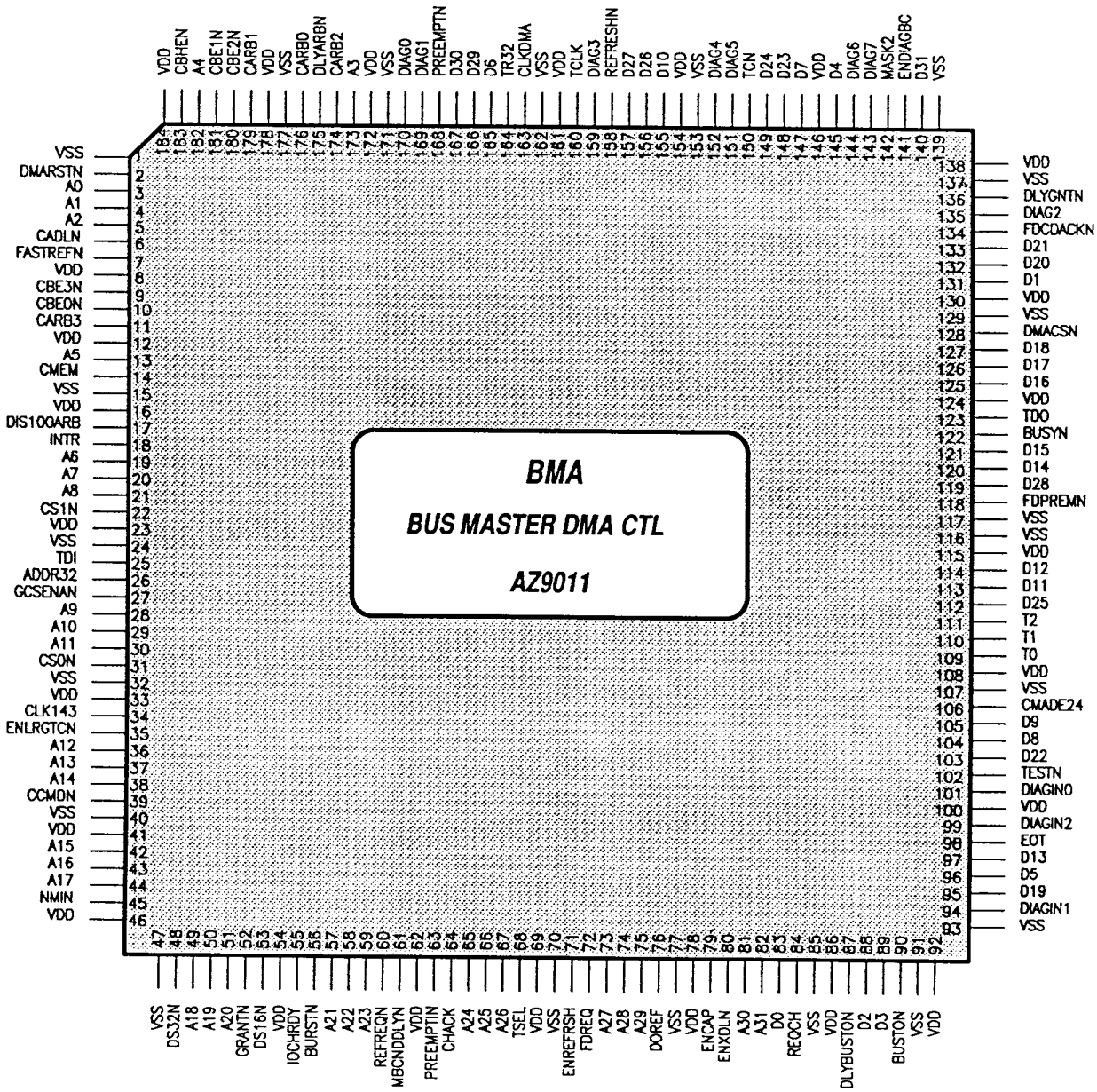
TSEL: Test Select. Selects the mode of TDI and TDO when in Bull Extended Test mode.

Other

VDD: Power.

VSS: Ground.

Figure 3. AZ9011 Package Outline



Buffer	Type	Attribute	Parameter	Condition	Pins
	I	TTL	VIL=(VSS,0.8V)	4.75V<VDD<5.25V	CLKDMA
			VIH=(2.0V,VDD)	4.75V<VDD<5.25V	
			IIL=(-10UA,N/A)	VIN=0.00V	
			IIH=(N/A,10UA)	VIN=VDD	

Buffer	Type	Attribute	Parameter	Condition	Pins
	I	CMOS,PU			T1

Buffer	Type	Attribute	Parameter	Condition	Pins
	I	TTL	VIL=(VSS,0.8V)	4.75V<VDD<5.25V	FASTREFN
			VIH=(2.0V,VDD)	4.75V<VDD<5.25V	DMARSTN
			IIL=(-10UA,N/A)	VIN=0.00V	ENLRGTCN
			IIH=(N/A,10UA)	VIN=VDD	ADDR32
					T2
					DIAGIN2
					DIAGIN1
					DLYGNTN
					ENDIAGBC
					DLYARBN
					DS16N
					DS32N
					FDREQ
					DLYBUSTON
					INTR
					DIS100ARB
					GCSENAN
					CLK143
					TESTN
					DIAGIN0
					T0
					BURSTN
					CHACK
					IOCHRDY
					PREEMPTIN
					ENCAP
					ENXDLN
					ENREFRSH

Electrical Specification (Continued)

Buffer	Type	Attribute	Parameter	Condition	Pins
	B	TTL	VIL=(VSS,0.8V)	4.75V<VDD<5.25V	CS1N
			VIH=(2.0V,VDD)	4.75V<VDD<5.25V	A5
			IIL=(-10UA,N/A)	VIN=0.00V	A6
			IIH=(N/A,10UA)	VIN=VDD	A0
	Z		VOL=(VSS,0.4V)	0.0MA<IOL<24.0MA	A2
			VOH=(2.4V,VDD)	-24MA<IOH<-0.0MA	A10
			IOZL=(-10UA,N/A)	VO=VSS	A14
			IOZH=(N/A,10UA)	VO=VDD	A15
			IOSL=(40MA,170MA)	VO=VDD	A13
			IOSH=(-100MA,-280MA)	VO=VSS	D12
					D9
					D19
					D13
					D18
					D15
					D21
					D20
					D27
					D4
					D31
					A3
					D30
					CARB1
					A4
					CARB0
					CBE1N
					CBE2N
					D3
					D2
					D0
					CADLN
					A1
					CBE0N
					CBE3N
					A7
					CMEM
					CARB3
					A8
					CCMDN
					A12
					A11
					CS0N
					A9
					D5

					D8
					D25
					D22
					D28
					D11
					D1
					D16
					D14
					D17
					D23
					D26
					D7
					D10
					D24
					D6
					CARB2
					CBHEN
					D29
					GRANTN

Buffer	Type	Attribute	Parameter	Condition	Pins
	B	TTL	VIL=(VSS,0.8V)	4.75V<VDD<5.25V	NMIN
			VIH=(2.0V,VDD)	4.75V<VDD<5.25V	
			IIL=(-260UA,-30UA)	VIN=0.00V	
			IIH=(N/A,40UA)	VIN=VDD	
		PU,Z	VOL=(VSS,0.4V)	0.0MA<IOL<24.0MA	
			VOH=(2.4V,VDD)	-24MA<IOH<-0.0MA	
			IOZL=(-10UA,N/A)	VO=VSS	
			IOZH=(N/A,10UA)	VO=VDD	
			IOSL=(40MA,170MA)	VO=VDD	
			IOSH=(-100MA,-280MA)	VO=VSS	

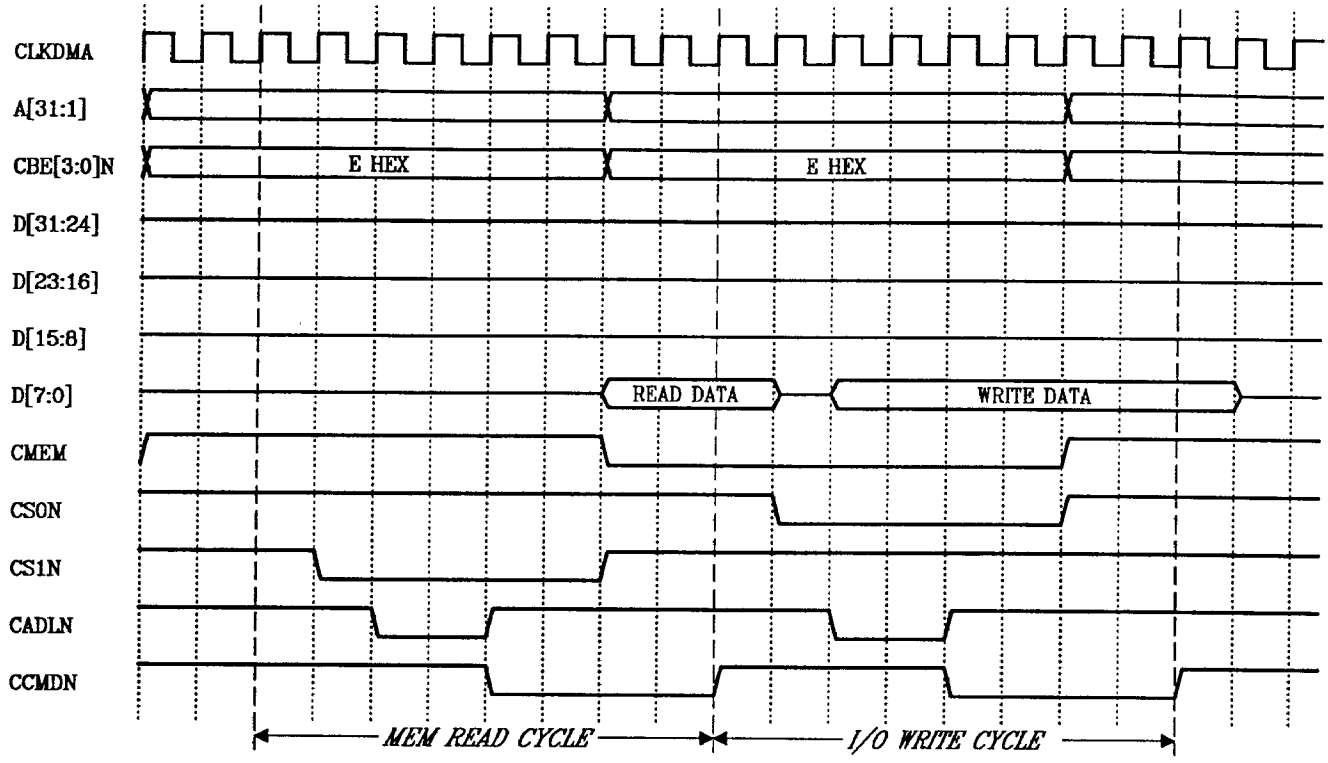
Buffer	Type	Attribute	Parameter	Condition	Pins
	B	TTL	VIL=(VSS,0.8V)	4.75V<VDD<5.25V	REFREQN
			VIH=(2.0V,VDD)	4.75V<VDD<5.25V	DOREF
			IIL=(-10UA,N/A)	VIN=0.00V	
			IIH=(N/A,10UA)	VIN=VDD	
		Z	VOL=(VSS,0.4V)	0.0MA<IOL<6.0MA	
			VOH=(2.4V,VDD)	-6.0MA<IOH<-0.0MA	
			IOZL=(-10UA,N/A)	VO=VSS	
			IOZH=(N/A,10UA)	VO=VDD	

Buffer	Type	Attribute	Parameter	Condition	Pins
	O		VOL=(VSS,0.4V)	0.0MA<IOL<6.0MA	EOT
			VOH=(2.4V,VDD)	-6.0MA<IOH<-0.0MA	

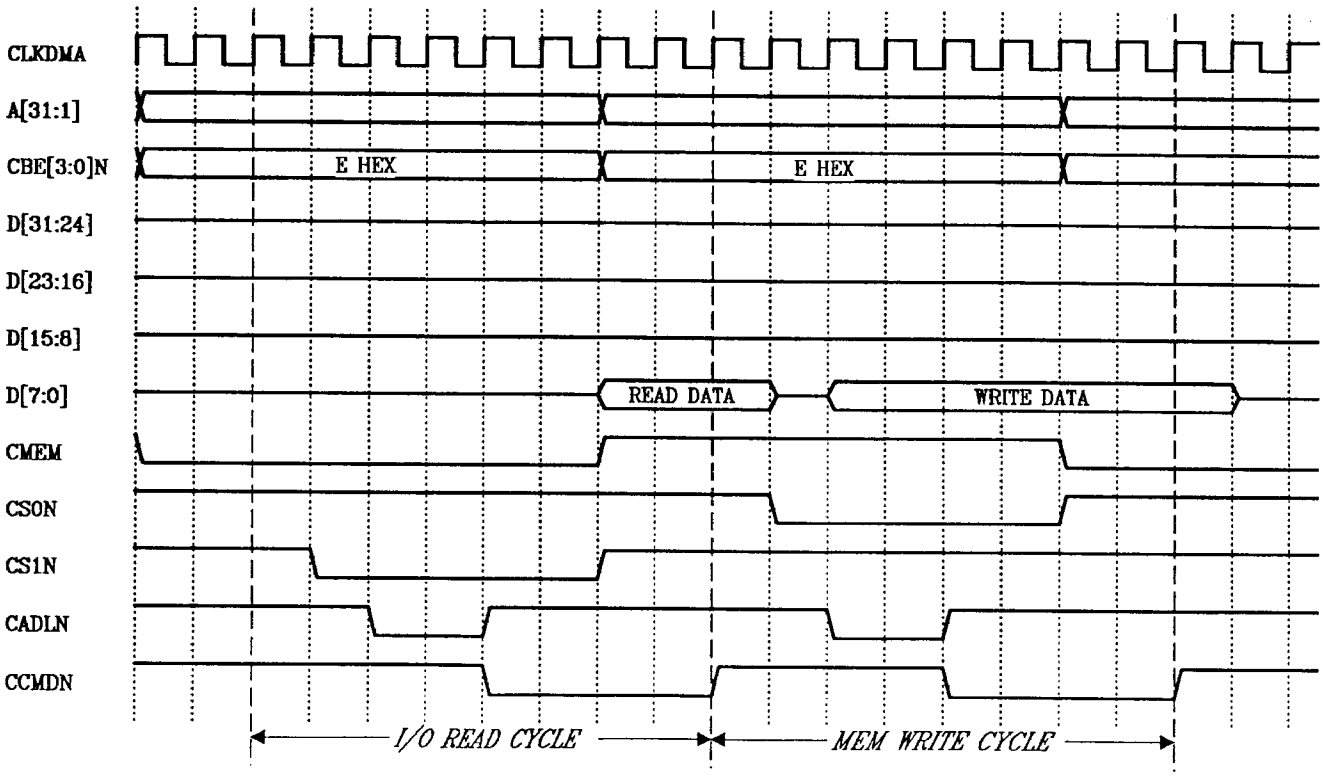
Buffer	Type	Attribute	Parameter	Condition	Pins
	O		VOL=(VSS,0.4V)	0.0MA<IOL<24.0MA	A17
			VOH=(2.4V,VDD)	-24MA<IOH<-0.0MA	A16
			IOZL=(-10UA,N/A)	VO=VSS	A23
			IOZH=(N/A,10UA)	VO=VDD	A24
			IOSL=(40MA,170MA)	VO=VDD	A18
			IOSH=(-100MA,-280MA)	VO=VSS	A20
					A30
					A29
					CMADE24
					BUSYN
					TCN
					REFRESHN
					A19
					A22
					A25
					A21
					A27
					A26
					A31
					A28

Buffer	Type	Attribute	Parameter	Condition	Pins
	O		VOL=(VSS,0.4V)	0.0MA<IOL<6.0MA	DIAG2
			VOH=(2.4V,VDD)	-6.0MA<IOH<-0.0MA	DIAG5
			IOZL=(-10UA,N/A)	VO=VSS	DIAG7
			IOZH=(N/A,10UA)	VO=VDD	DIAG0
					BUSTON
					DIAG6
					MASK2
					DIAG4
					DIAG3
					DIAG1

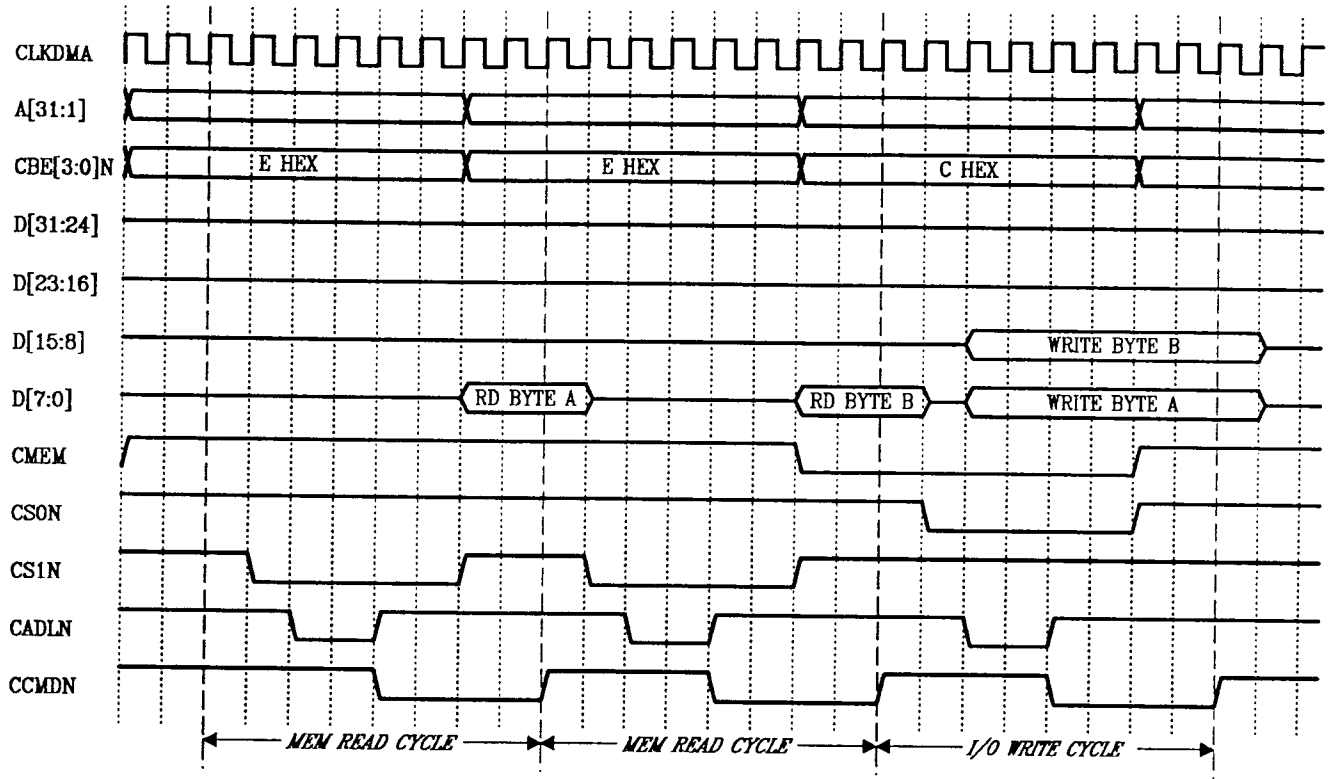
Buffer	Type	Attribute	Parameter	Condition	Pins
	O		VOL=(VSS,0.4V)	0.0MA<IOL<6.0MA	FDCDACKN
			VOH=(2.4V,VDD)	-6.0MA<IOH<-0.0MA	REQCH
			IOZL=(-10UA,N/A)	VO=VSS	FDPREMN
			IOZH=(N/A,10UA)	VO=VDD	DMACSN
					PREEMPTN



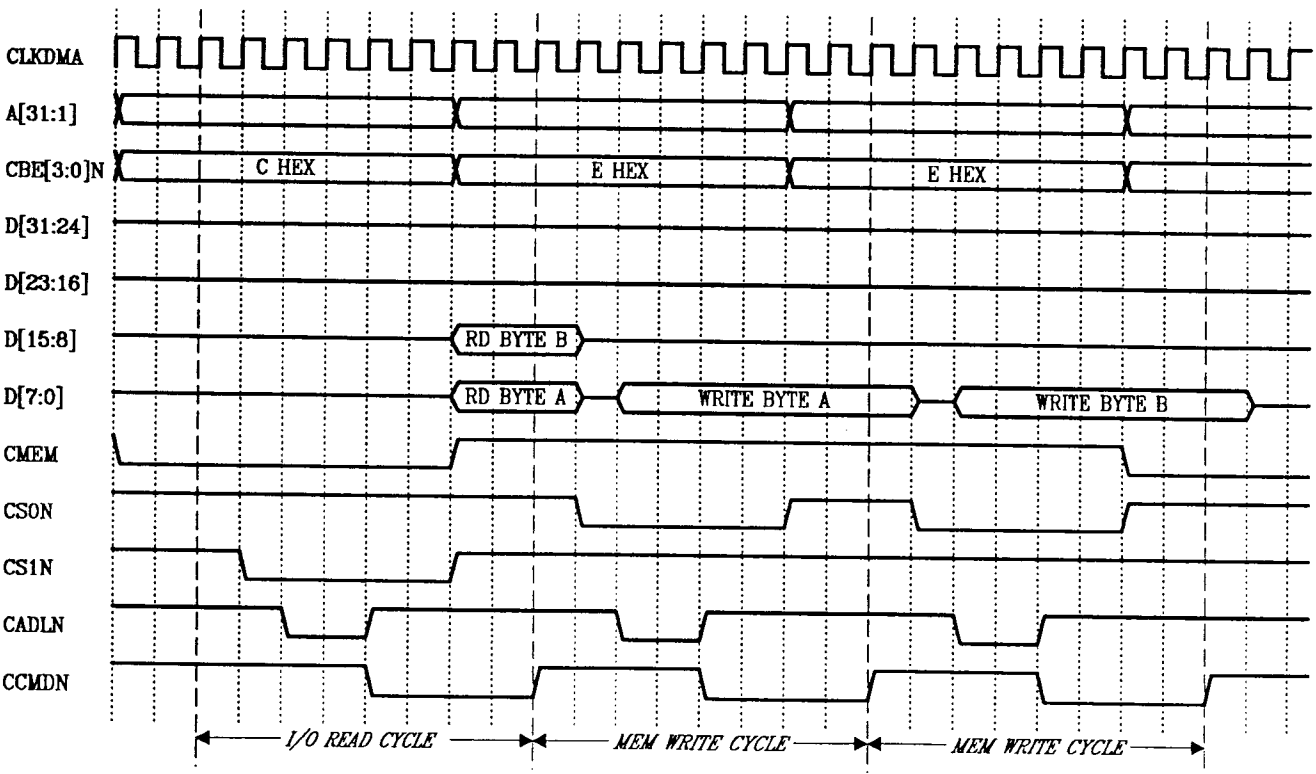
8 BIT MEMORY READ TRANSFER



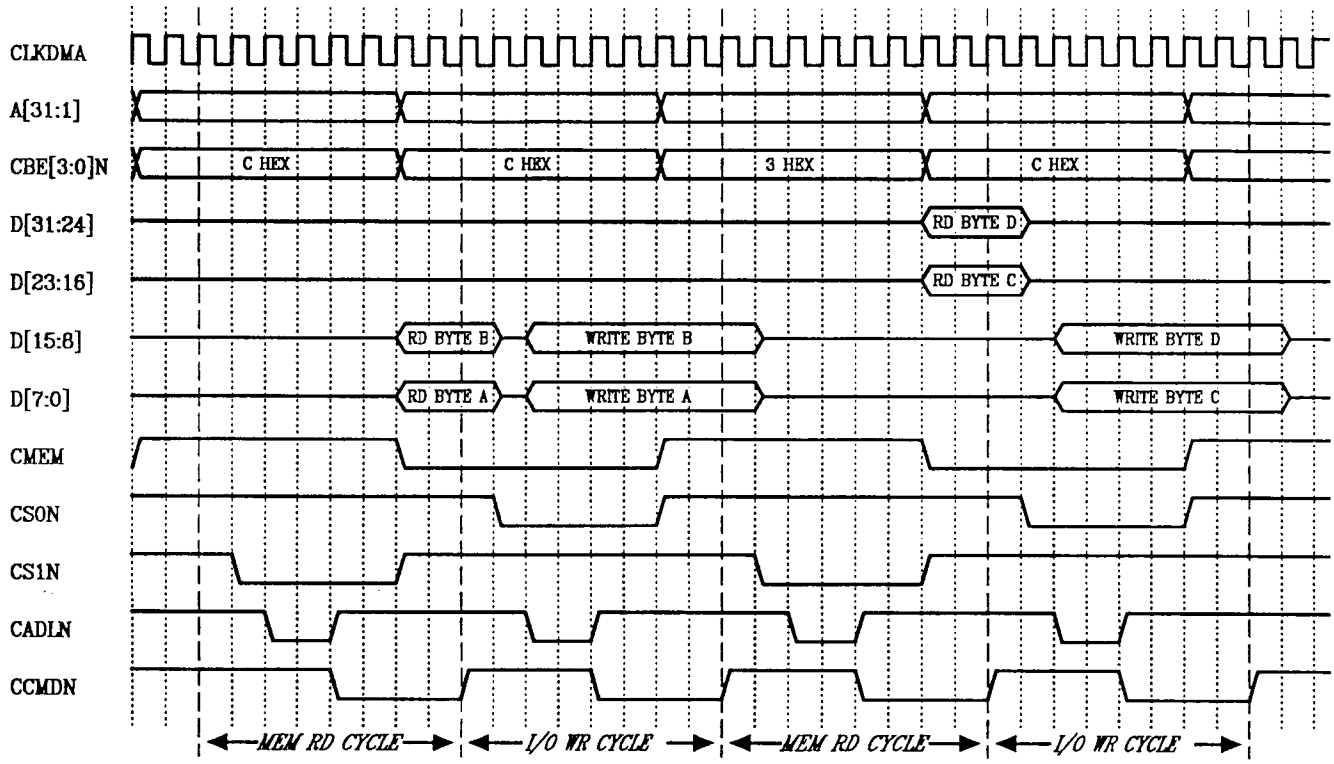
8 BIT MEMORY WRITE TRANSFER



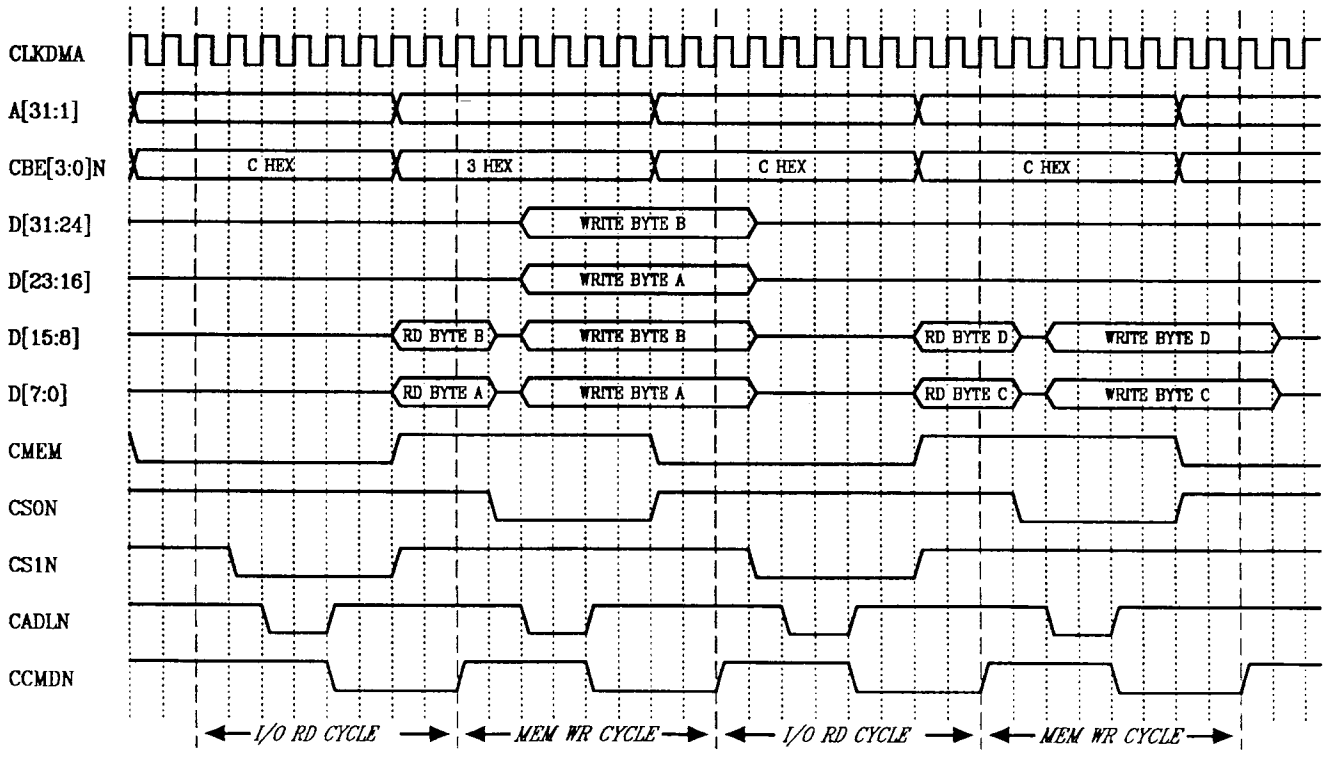
8 BIT MEM TO 16 BIT I/O TRANSFER



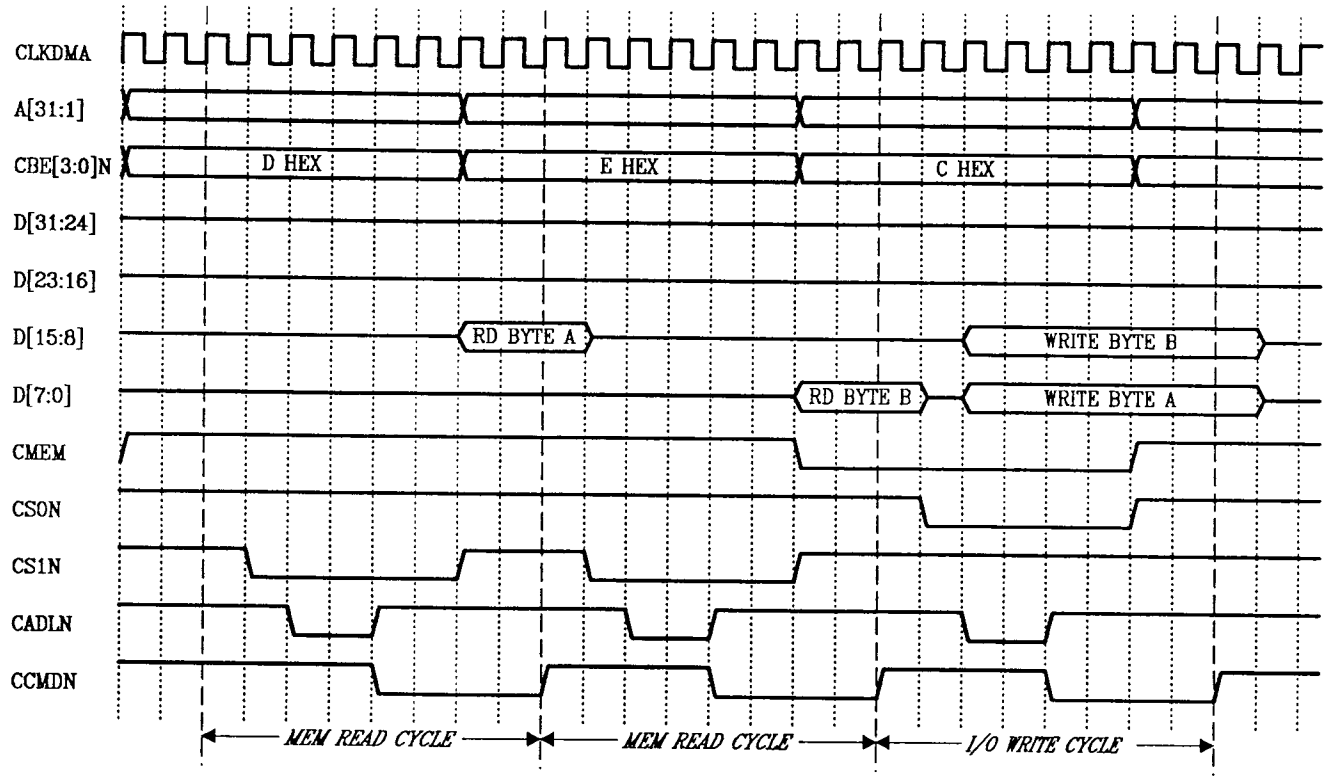
16 BIT I/O TO 8 BIT MEM TRANSFER



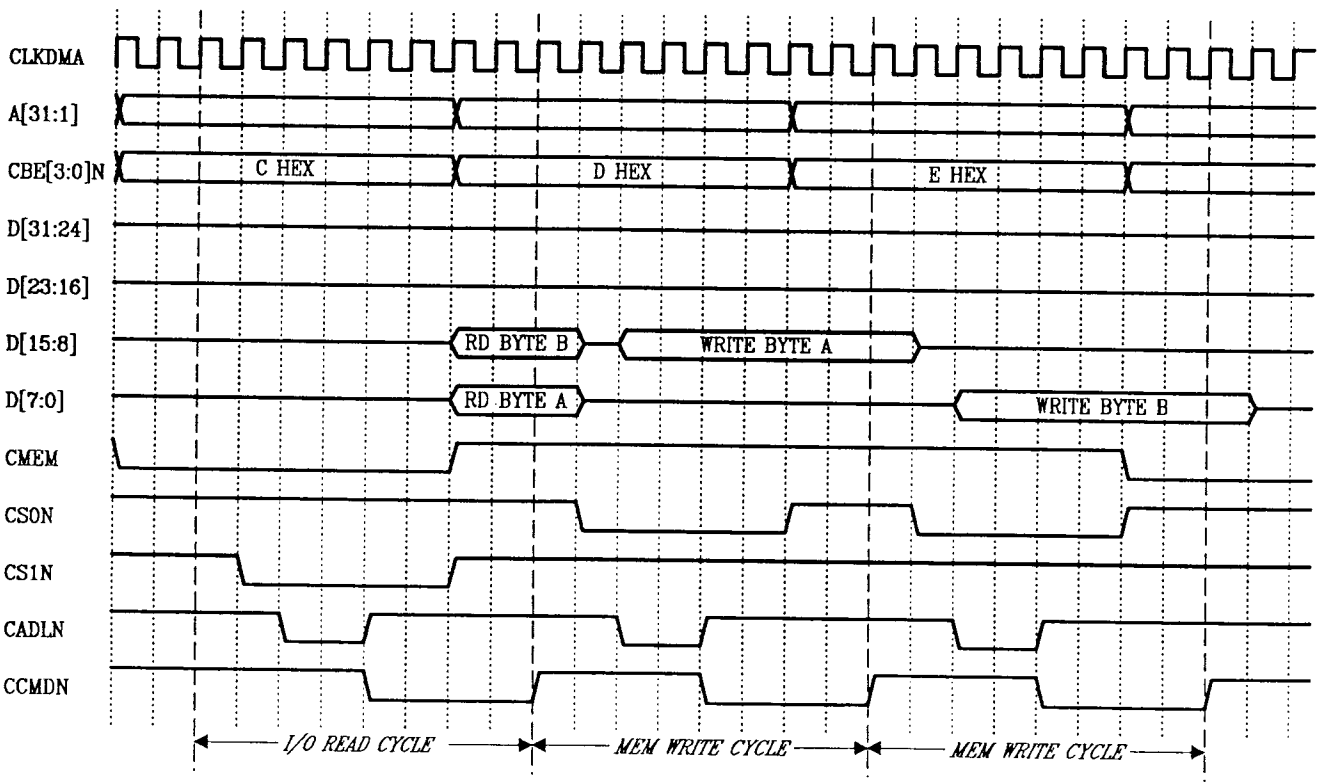
32 BIT MEM TO 16 BIT I/O TRANSFER



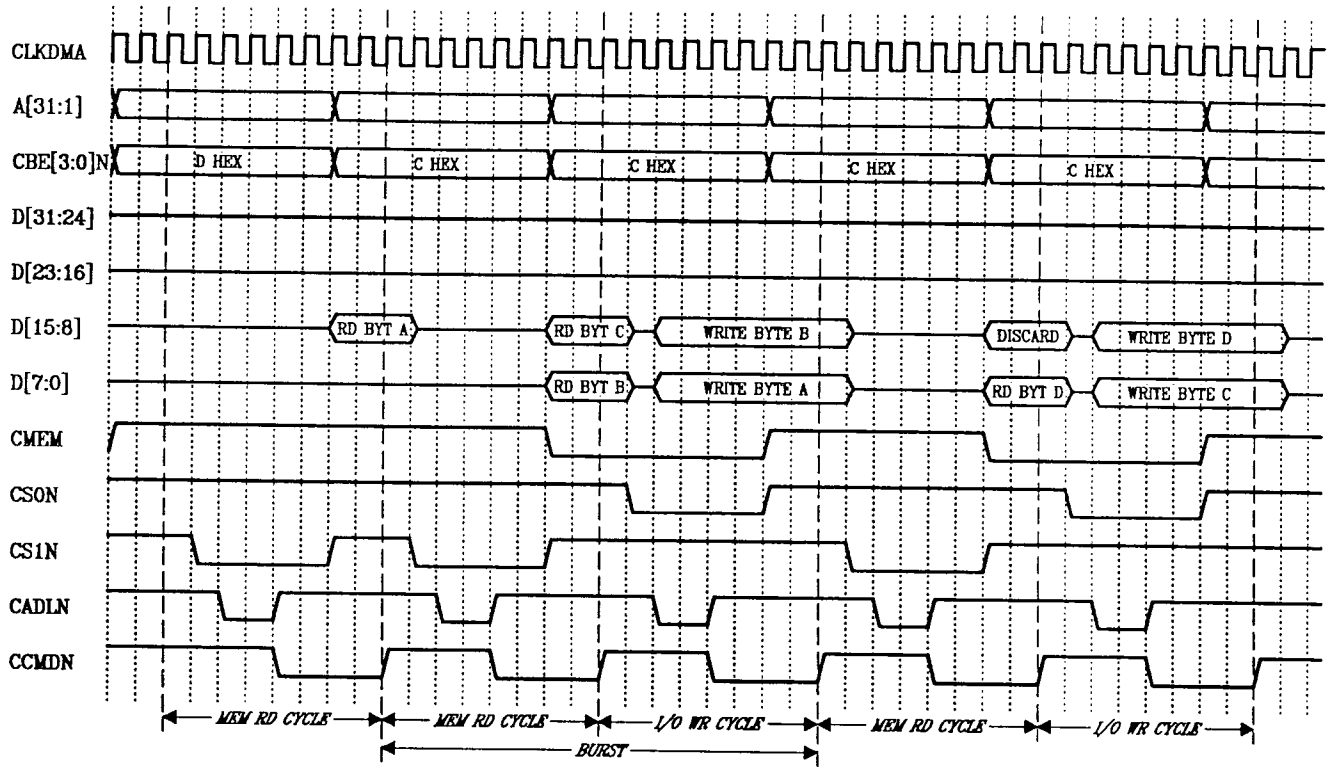
16 BIT I/O TO 32 BIT MEM TRANSFER



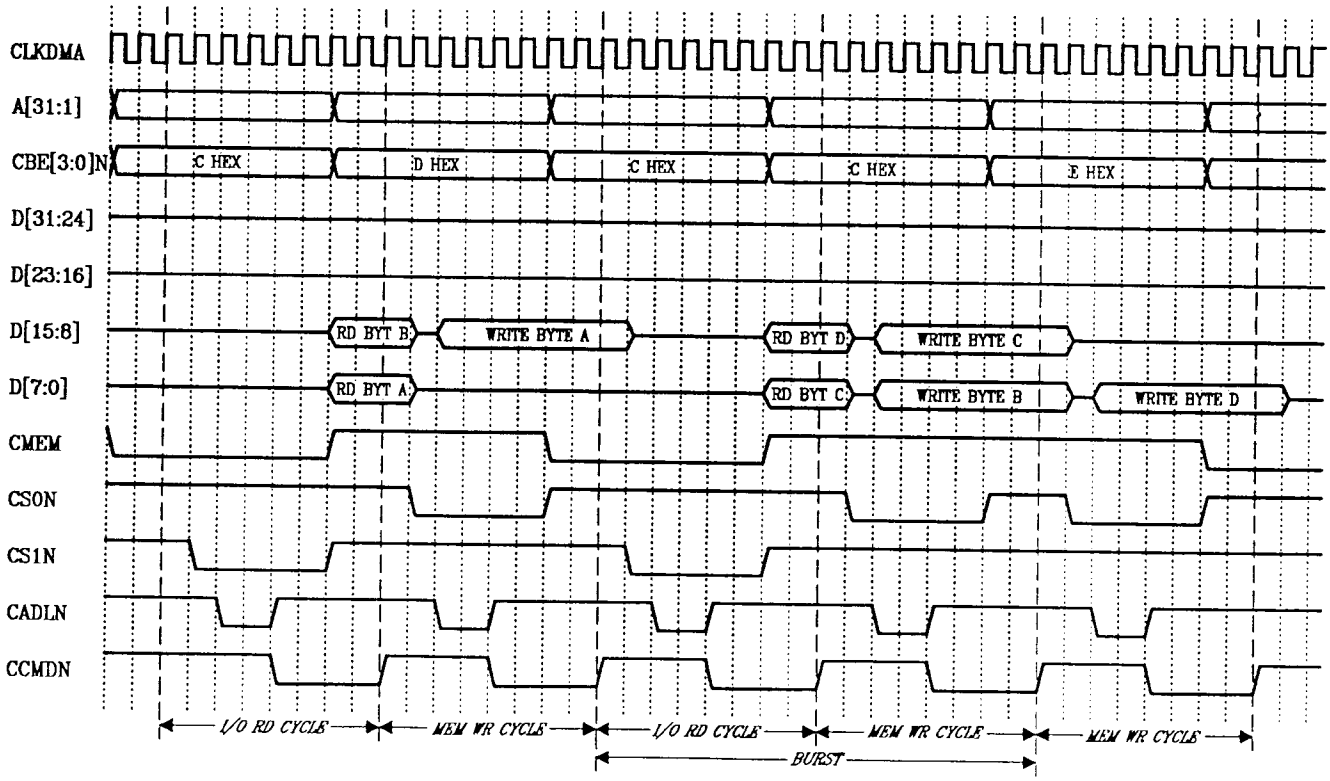
COMPATIBLE MODE 16 BIT TRANSFER FROM AN ODD MEMORY ADDRESS



COMPATIBLE MODE 16 BIT TRANSFER TO AN ODD MEMORY ADDRESS



HIGH SPEED MODE 16 BIT TRANSFER FROM AN ODD MEMORY ADDRESS



BULL MODE 16 BIT TRANSFER TO AN ODD MEMORY ADDRESS

Table 3. Timing Information

INPUT TIMINGS		BCCOM	WCCOM
BURSTN	INPUT SETUP TO CLKDMA	7.5	10.0
BURSTN	INPUT HOLD FROM CLKDMA	7.5	10.0
CADLN	INPUT HIGH FROM ADDRESS STABLE	6.8	20.8
CADLN	INPUT MINIMUM PULSE WIDTH	20.7	
CCMDN	INPUT MINIMUM PULSE WIDTH	*90	
CHACK	INPUT SETUP TO CLKDMA	1.6	0.0
CHACK	INPUT HOLD FROM CLKDMA	2.0	3.0
CMEM	INPUT SETUP TO DMAADLN HIGH	0.0	0.0
CMEM	INPUT HOLD FROM DMAADLN HIGH	2.5	4.5
CS[1:0]N	INPUT SETUP TO CCMDN HIGH	0.0	0.0
CS[1:0]N	INPUT HOLD FROM CCMDN HIGH	2.5	3.5
D[15:00] (Program)	INPUT SETUP TO CCMDN HIGH	7.5	10.0
D[15:00] (Program)	INPUT HOLD FROM CCMDN HIGH	7.5	10.0
D[31:00] (DMA)	INPUT SETUP TO CLKDMA	0.0	0.0
D[31:00] (DMA)	INPUT HOLD FROM CLKDMA	6.0	14.0
DMARSTN	INPUT MINIMUM PULSE WIDTH	2 CLKDMA	
DS16N	INPUT MINIMUM PULSE WIDTH	*90	
DS32N	INPUT MINIMUM PULSE WIDTH	*90	
FDREQ	INPUT SETUP TO GRANTN HIGH	0.0	0.0
FDREQ	INPUT HOLD FROM GRANTN HIGH	2.5	3.0
GCSENAN	INPUT SETUP TO CLKDMA	1.6	0.0
GCSENAN	INPUT HOLD FROM CLKDMA	2.0	3.0
GRANTN	INPUT SETUP TO CLKDMA	5.5	14.0
GRANTN	INPUT HOLD FROM CLKDMA	0.0	0.0
INTR	INPUT SETUP TO CLKDMA	1.0	1.0
INTR	INPUT HOLD FROM CLKDMA	1.0	1.0
IOCHRDY	INPUT MINIMUM PULSE WIDTH	*90	
NMIN	INPUT MINIMUM PULSE WIDTH	13.6	
PREEMPTIN	INPUT SETUP TO CLKDMA	4.0	6.5
PREEMPTIN	INPUT HOLD FROM CLKDMA	6.0	13.0
READYN	INPUT SETUP TO CLKDMA	2.0	2.0
READYN	INPUT HOLD FROM CLKDMA	2.0	3.5
CLK143	ASYNCHRONOUS		
CLKDMA	MASTER INPUT		
ADDR32	NOT INTENDED TO BE CHANGED WITH POWER ON		
DIAGIN[2:0]	NOT INTENDED TO BE CHANGED WITH POWER ON		
DIS100ARB	NOT INTENDED TO BE CHANGED WITH POWER ON		
ENCAP	NOT INTENDED TO BE CHANGED WITH POWER ON		
ENDIAGBC	NOT INTENDED TO BE CHANGED WITH POWER ON		
ENLRGTCN	NOT INTENDED TO BE CHANGED WITH POWER ON		
ENREFRSH	NOT INTENDED TO BE CHANGED WITH POWER ON		
ENXDLN	NOT INTENDED TO BE CHANGED WITH POWER ON		
FASTREFN	NOT INTENDED TO BE CHANGED WITH POWER ON		
T0	NOT INTENDED TO BE CHANGED WITH POWER ON		
T1	NOT INTENDED TO BE CHANGED WITH POWER ON		
T2	NOT INTENDED TO BE CHANGED WITH POWER ON		
TESTN	NOT INTENDED TO BE CHANGED WITH POWER ON		

* PER MICRO CHANNEL SPECIFICATION

Timing values are preliminary and reflect measurements taken from prototype silicon.

OUTPUT TIMINGS		BCCOM	WCCOM
DLYARBN	RESERVED FOR FUTURE IMPLEMENTATION		
DLYGNTN	RESERVED FOR FUTURE IMPLEMENTATION		
DLYBUSTON	RESERVED FOR FUTURE IMPLEMENTATION		
A[31:01]	OUTPUT DELAY DRIVEN FROM CLKDMA (DMA)	9.3	30.0
A[31:01]	OUTPUT DELAY HIGH-Z FROM CLKDMA (DMA)	13.6	42.9
A[31:01]	OUTPUT DELAY DRIVEN FROM CLKDMA(Refresh)	15.2	49.5
A[31:01]	OUTPUT DELAY HIGH-Z FROM CLKDMA(Refresh)	13.6	43.6
A[31:01]	OUTPUT DELAY INVALID FROM CLKDMA (Memory Address Count)	10.0	35.0
A[31:01]	OUTPUT DELAY VALID FROM CLKDMA (Memory Address Count)	12.0	42.0
A[31:01]	OUTPUT DELAY VALID FROM CLKDMA (Mem/IO Address Switch)	9.0	27.0
A[31:01]	OUTPUT DELAY VALID FROM CLKDMA (IO/Mem Address Switch)	8.0	26.0
A[31:01]	OUTPUT DELAY INVALID FROM CLKDMA (Refresh Address Count)	6.6	20.9
CBE[3:0]N (DMA)	OUTPUT DELAY DRIVEN FROM CLKDMA	9.3	30.0
CBE[3:0]N (DMA)	OUTPUT DELAY HIGH-Z FROM CLKDMA	13.6	42.9
CBE[3:0]N (Refresh)	OUTPUT DELAY DRIVEN FROM CLKDMA	15.2	49.5
CBE[3:0]N (Refresh)	OUTPUT DELAY HIGH-Z FROM CLKDMA	13.6	43.6
CBE[3:0]N	OUTPUT DELAY INVALID FROM CLKDMA (Memory Address Count)	13.6	44.4
CBE[3:0]N	OUTPUT DELAY VALID FROM CLKDMA (Memory Address Count)	9.5	30.5
CBE[3:0]N	OUTPUT DELAY INVALID FROM CLKDMA (Mem/IO OR IO/Mem Address Switch)	12.3	40.5
CBE[3:0]N	OUTPUT DELAY VALID FROM CLKDMA (Mem/IO OR IO/Mem Address Switch)	6.9	21.6
CBE[3:0]N	OUTPUT DELAY INVALID FROM CLKDMA (Refresh Address Count)	7.2	26.8
CBE[3:0]N	OUTPUT DELAY VALID FROM CLKDMA (Refresh)	6.9	21.6
CBHEN (DMA)	OUTPUT DELAY DRIVEN FROM CLKDMA	9.3	30.0
CBHEN (DMA)	OUTPUT DELAY HIGH-Z FROM CLKDMA	13.6	42.9
CBHEN (Refresh)	OUTPUT DELAY DRIVEN FROM CLKDMA	15.2	49.5
CBHEN (Refresh)	OUTPUT DELAY HIGH-Z FROM CLKDMA	13.6	43.6
CBHEN	OUTPUT DELAY INVALID FROM CLKDMA (Memory Address Count)	13.6	44.4
CBHEN	OUTPUT DELAY VALID FROM CLKDMA (Memory Address Count)	9.5	30.5
CBHEN	OUTPUT DELAY INVALID FROM CLKDMA (Mem/IO OR IO/Mem Address Switch)	12.3	40.5
CBHEN	OUTPUT DELAY VALID FROM CLKDMA (Mem/IO OR IO/Mem Address Switch)	6.9	21.6
CBHEN	OUTPUT DELAY INVALID FROM CLKDMA (Refresh Address Count)	7.2	26.8
CBHEN	OUTPUT DELAY VALID FROM CLKDMA (Refresh)	6.9	21.6
CMADE24 (DMA)	OUTPUT DELAY DRIVEN FROM CLKDMA	9.3	30.0
CMADE24 (DMA)	OUTPUT DELAY HIGH-Z FROM CLKDMA	13.6	42.9
CMADE24 (Refresh)	OUTPUT DELAY DRIVEN FROM CLKDMA	15.2	49.5
CMADE24 (Refresh)	OUTPUT DELAY HIGH-Z FROM CLKDMA	13.6	43.6
CMADE24	OUTPUT DELAY INVALID FROM CLKDMA	13.6	44.4

OUTPUT TIMINGS		BCCOM	WCCOM
	(Memory Address Count)		
CMADE24	OUTPUT DELAY VALID FROM CLKDMA	9.5	30.5
	(Memory Address Count)		
CMADE24	OUTPUT DELAY INVALID FROM CLKDMA	12.3	40.5
	(Mem/IO OR IO/Mem Address Switch)		
CMADE24	OUTPUT DELAY VALID FROM CLKDMA	6.9	21.6
	(Mem/IO OR IO/Mem Address Switch)		
CMADE24	OUTPUT DELAY INVALID FROM CLKDMA	7.2	26.8
	(Refresh Address Count)		
CMADE24	OUTPUT DELAY VALID FROM CLKDMA	6.9	21.6
	(Refresh)		
BUSTON	OUTPUT DELAY FROM CLKDMA (HIGH/LOW)	5.8	18.5
BUSTON	OUTPUT DELAY FROM CLKDMA (LOW/HIGH)	5.4	17.0
D[31:00] (DMA)	OUTPUT DELAY DRIVEN FROM CLKDMA	7.2	22.8
D[31:00] (DMA)	OUTPUT DELAY HIGH-Z FROM CLKDMA	6.6	24.4
D[31:00] (DMA)	OUTPUT DELAY VALID FROM CLKDMA	7.2	22.8
D[31:00] (DMA)	OUTPUT DELAY INVALID FROM CLKDMA	7.0	21.0
D[31:00] (DMA)	OUTPUT DELAY VALID FROM CLKDMA	6.6	21.4
	(MEM. WRITE TO MEM. WRITE)		
D[31:00] (DMA)	OUTPUT DELAY INVALID FROM CLKDMA	7.6	24.4
	(MEM. WRITE TO MEM. WRITE)		
D[31:00]	OUTPUT DELAY DRIVEN FROM CCMDN LOW	7.5	23.4
	(Program Read)		
D[31:00]	OUTPUT DELAY HIGH-Z FROM CCMDN HIGH	6.3	20.6
	(Program Read)		
D[31:00]	OUTPUT DELAY VALID FROM CCMDN LOW	7.5	23.4
	(Program Read)		
D[31:00]	OUTPUT DELAY INVALID FROM CCMDN HIGH	6.3	20.6
	(Program Read)		
BUSYN (DMA)	OUTPUT DELAY FROM GRANTN LOW (HIGH/LOW)	14.7	48.6
BUSYN (DMA)	OUTPUT DELAY FROM GRANTN LOW (LOW/HIGH)	14.0	45.5
BUSYN (Refresh)	OUTPUT DELAY FROM GRANTN LOW (HIGH/LOW)	8.8	29.1
BUSYN (Refresh)	OUTPUT DELAY FROM GRANTN LOW (LOW/HIGH)	15.4	50.0
DMACSN	OUTPUT DELAY FROM DMAADLN LOW (HIGH/LOW)	7.4	24.6
DMACSN	OUTPUT DELAY FROM DMAADLN LOW (LOW/HIGH)	7.2	24.4
FDCDACKN	OUTPUT DELAY FROM CLKDMA (HIGH/LOW)	14.8	49.4
FDCDACKN	OUTPUT DELAY FROM CLKDMA (LOW/HIGH)	12.6	40.8
CARB[3:0]	OUTPUT DELAY FROM CLKDMA (HIGH/LOW)	14.5	47.6
CARB[3:0]	OUTPUT DELAY FROM CLKDMA (LOW/HIGH)	12.2	39.6
FDPREMN	OUTPUT DELAY FROM FDREQ HIGH (HIGH/LOW)	6.6	22.6
FDPREMN	OUTPUT DELAY FROM FDREQ LOW (LOW/HIGH)	4.6	15.3
GRANTN	OUTPUT DELAY FROM CLKDMA (HIGH/LOW)	7.1	23.1
GRANTN	OUTPUT DELAY FROM CLKDMA (LOW/HIGH)	6.4	20.6
REQCH	OUTPUT DELAY FROM CLKDMA (HIGH/LOW)	5.7	17.0
REQCH	OUTPUT DELAY FROM CLKDMA (LOW/HIGH)	5.2	14.6
NMIN	OUTPUT DELAY FROM CLKDMA (HIGH/LOW)	10.4	33.5
NMIN	OUTPUT DELAY FROM CLKDMA (HIGH-Z)	?	?
PREEMPTN (Refresh)	OUTPUT DELAY FROM CLKDMA (HIGH/LOW)	6.4	20.6
PREEMPTN (Refresh)	OUTPUT DELAY FROM CLKDMA (LOW/HIGH)	6.0	19.5
PREEMPTN (INTR)	OUTPUT DELAY FROM CLKDMA (HIGH/LOW)	6.1	19.5
PREEMPTN (INTR)	OUTPUT DELAY FROM CLKDMA (LOW/HIGH)	5.3	17.3
FDPREMN (DIAG)	OUTPUT DELAY FROM CLKDMA (HIGH/LOW)	6.1	19.5
FDPREMN (DIAG)	OUTPUT DELAY FROM CLKDMA (LOW/HIGH)	5.3	17.3
REFRESHN	OUTPUT DELAY FROM CLKDMA (HIGH/LOW)	7.2	23.6

OUTPUT TIMINGS		BCCOM	WCCOM
REFRESHN	OUTPUT DELAY FROM CLKDMA (LOW/HIGH)	13.2	42.5
CS0N	OUTPUT DELAY FROM CLKDMA (HIGH/LOW)	5.8	18.6
CS0N	OUTPUT DELAY FROM CLKDMA (LOW/HIGH)	5.4	17.1
CS0N (Refresh)	OUTPUT DELAY FROM CLKDMA (DRIVEN)	8.0	25.9
CS0N (Refresh)	OUTPUT DELAY FROM CLKDMA (HIGH-Z)	16.2	52.8
CS0N (DMA)	OUTPUT DELAY FROM CLKDMA (DRIVEN)	13.9	45.4
CS0N (DMA)	OUTPUT DELAY FROM CLKDMA (HIGH-Z)	14.8	48.3
CS1N	OUTPUT DELAY FROM CLKDMA (HIGH/LOW)	5.8	18.5
CS1N	OUTPUT DELAY FROM CLKDMA (LOW/HIGH)	5.4	16.1
CS1N (Refresh)	OUTPUT DELAY FROM CLKDMA (DRIVEN)	8.0	25.9
CS1N (Refresh)	OUTPUT DELAY FROM CLKDMA (HIGH-Z)	16.2	52.8
CS1N (DMA)	OUTPUT DELAY FROM CLKDMA (DRIVEN)	13.9	45.4
CS1N (DMA)	OUTPUT DELAY FROM CLKDMA (HIGH-Z)	14.8	48.3
TCN	OUTPUT DELAY FROM CCMDN LOW (HIGH/LOW)	5.7	18.5
TCN	OUTPUT DELAY FROM CCMDN HIGH (LOW/HIGH)	5.0	15.9
DIAG[7:0]	OUTPUT (USED FOR TEST ONLY)	N/A	N/A
DOREF	OUTPUT DELAY FROM CLKDMA (HIGH/LOW)	5.5	17.7
DOREF	OUTPUT DELAY FROM CLKDMA (LOW/HIGH)	4.9	15.8
EOT	OUTPUT DELAY FROM CLKDMA (HIGH/LOW)	TBD	TBD
EOT	OUTPUT DELAY FROM CLKDMA (LOW/HIGH)	TBD	TBD
MASK2	OUTPUT DELAY FROM CCMDN HIGH (HIGH/LOW)	TBD	TBD
MASK2	OUTPUT DELAY FROM CCMDN HIGH (LOW/HIGH)	TBD	TBD
PAROUT	OUTPUT (USED FOR TEST ONLY)	N/A	N/A
REFREQN	OUTPUT DELAY FROM CLKDMA (HIGH/LOW)	7.9	25.2
REFREQN	OUTPUT DELAY FROM CLKDMA (LOW/HIGH)	6.7	21.1

BMA, Inc. does not assume any responsibility for use of any circuitry described, no circuit patent licences are implied, and BMA, Inc. reserves the right, at any time without notice, to change said circuitry or specifications. Trademarks are property of their respective holders. BMA, Inc. disclaims any proprietary interest in trademarks and trade names other than its own.

900 LONG LAKE ROAD • NEW BRIGHTON, MN 55112-6400
TEL (612) 633-5660 • TELEX 6971084 BMA UW • FAX (612) 633-6387

BMA

© 1990 BMA, Inc.

12/18/90