AmZ8016

DMA Transfer Controller

DISTINCTIVE CHARACTERISTICS

- Two independent multi-function channels
- Automatic loading/reloading of control parameters by each channel
- Optional automatic chaining of operations
- Channel interleave operations
- Masked data pattern matching for search operations
- Vectored interrupts on selected transfer conditions
- Base registers for repetitive operations

GENERAL DESCRIPTION

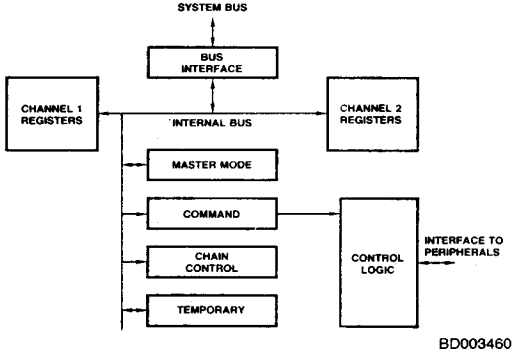
The AmZ8016* DMA Transfer Controller (DTC) is a high performance peripheral interface circuit for Z8000 processor systems. In addition to providing data block transfer capability between memory and peripherals, each of the DTC's two channels can perform peripheral-to-peripheral as well as memory-to-memory transfer. A special Search Mode of Operation compares data read from a memory or peripheral source to the content of a pattern register.

For all DMA operations (search, transfer, and transfer-and-search), the DTC can operate with either byte or word data sizes. In some system configurations, it may be necessary to transfer between word-organized memory and a byte-oriented peripheral. The DTC provides a byte packing/unpacking capability through its byte-word funneling transfer or transfer-and-search option. Some DMA applications may continuously transfer data between the same two locations. These applications may not require the flexibility inherent in reloading registers from memory tables. To service these repetitive DMA operations, base registers are provided on each channel which reinitialize the current source and destination Address and Operation Count registers. To change the data transfer direction under CPU control, provision is made for reassigning the source address as a destination and the destination as a source, eliminating the need for actual reloading of these address registers.

Frequently, DMA devices must interface to slow peripherals or slow memory. In addition to providing a hardware $\overline{WAIT}$ input, the AmZ8016 DTC allows the user to select independently, for both source and destination addresses, automatic insertion of 0, 1, 2 or 4 wait states. The user may even disable the $\overline{WAIT}$ input pin function altogether and use these software programmed wait states exclusively.

High throughput and powerful transfer options are of limited usefulness if a DMA requires frequent reloading by the host CPU. The AmZ8016 DTC minimizes CPU interactions by allowing each channel to load its control parameters from memory into the channel's control registers. The only action required of the CPU is to load the address of the control parameter table into the channel and issue an instruction to start this register loading operation.

BLOCK DIAGRAM



BD003460

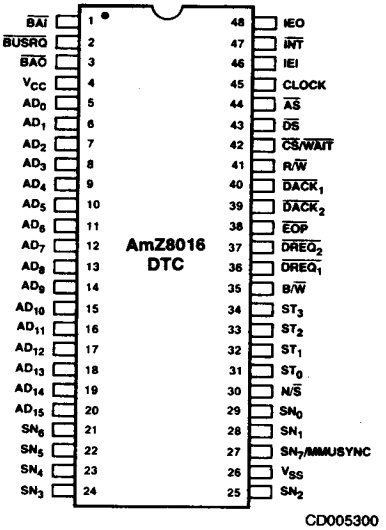
RELATED PRODUCTS

Part No.	Description
AmZ8016	Application Manual (contains detailed application configuration and software example).

*Z8000 is a trademark of Zilog, Inc.

Refer to page 7-1 for Essential Information on Military Devices

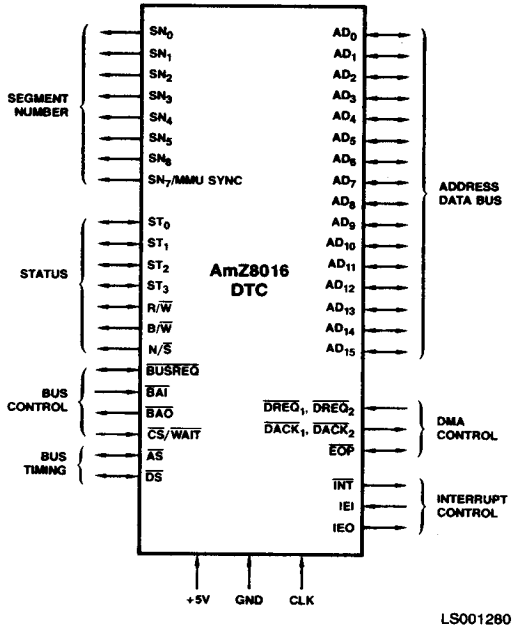
CONNECTION DIAGRAM
Top View
D-48, P-48



CD005300

Note: Pin 1 is marked for orientation

LOGIC SYMBOL

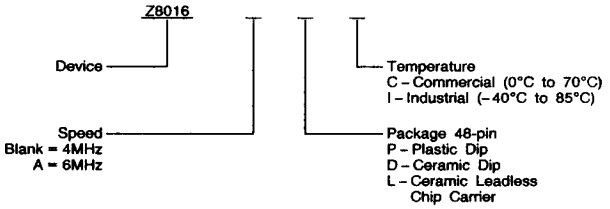


LS001280

Refer to page 7-1 for Essential Information on Military Devices

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).



Valid Combinations	
Z8016	DC, PC ADC, APC

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, check for newly released valid combinations and/or obtain additional data on AMD's standard military grade product.

2

Refer to page 7-1 for Essential Information on Military Devices

PIN DESCRIPTION

Pin No.	Name	I/O	Description
4	VCC		+5 Power Supply.
26	VSS		Ground.
45	CLOCK	I	(Clock). The Clock signal controls the internal operations and the rates of data transfers. It is usually derived from a master system clock or the associated CPU clock. The Clock input requires a high voltage input signal. When the DTC is used with an MMU, they must both be driven from the same clock signal. Many DTC input signals can make transitions independent of the DTC clock; these signals can be asychronous to the DTC clock. On other signals, such as $\overline{WAIT}$ inputs, transitions must meet set-up and hold requirements relative to the DTC clock. See the timing diagrams for details.
5-20	AD_0-AD_{15}	I/O	(Address/Data Bus). The Address/Data Bus is a time-multiplexed, bidirectional, active High, three-state bus used for all I/O and memory transactions. HIGH on the bus corresponds to 1 and LOW corresponds to O. AD_0 is the least significant bit position and AD_{15} is the most significant. The presence of addresses is defined by the timing edge of $\overline{AS}$, and the asserted or requested presence of data is defined by the $\overline{DS}$ signal. The status output lines ST_0-ST_3 indicate the type of transaction, either memory or I/O. When the DTC is in control of the system bus, it dominates the AD Bus; when the DTC is not in control of the system bus, the CPU or other external devices dominate the AD Bus. The presence of address or data on the AD_0-AD_{15} bus is defined only by $\overline{AS}$ and $\overline{DS}$. When the DTC is not in control of the bus; there is no required relation between the presence of address or data and the DTC clock. This allows the DTC to be used with a system bus which does not have a bussed clock signal.
44	$\overline{AS}$	I/O	(Address Strobe). Address Strobe is a bidirectional, active-low, three-state signal. A LOW-to-HIGH transition on this signal while $\overline{DS}$ is HIGH indicates that the AD_0-AD_{15} bus contains address information. During a DMA operation when the DTC is in control of the system, $\overline{AS}$ is an output generated by the DTC to indicate that a valid address is on AD_0-AD_{15}. The address information output by the DTC is stable prior to the LOW-to-HIGH $\overline{AS}$ transition. When the DTC is not in control of the system bus and the external system is transferring information to the DTC or from it, the DTC samples address information from the AD_0-AD_{15} bus on the LOW-to-HIGH $\overline{AS}$ transition. There are no timing requirements between $\overline{AS}$ as an input and the DTC clock; this allows use of the DTC with a system bus which does not have a bussed clock. If $\overline{AS}$ and $\overline{DS}$ are simultaneously LOW, the DTC is reset.
43	$\overline{DS}$	I/O	(Data Strobe). Data Strobe is a bidirectional, active-low, three-state signal. A LOW on this signal while $\overline{AS}$ is HIGH indicates that the AD_0-AD_{15} bus is being used for data transfer. When the DTC is not in control of the system bus and the external system is transferring information to or from the DTC, $\overline{DS}$ is a timing input used by the DTC to move data to or from the AD_0-AD_{15} bus. Data is written into the DTC by the external system on the LOW-to-HIGH $\overline{DS}$ transition. Data is read from DTC by the external system while $\overline{DS}$ is LOW. There are no timing requirements between $\overline{DS}$ as an input and the DTC clock; this allows use of the DTC with a system bus which does not have a bussed clock. During a DMA operation when the DTC is in control of the system, $\overline{DS}$ is an output generated by the DTC and used by the system to move data to or from the $AD_0 - AD_{15}$ bus. When the DTC has bus control, it writes to the external system by placing data on the $AD_0 - AD_{15}$ bus before the HIGH-to-LOW $\overline{DS}$ transition and holding the data stable until after the LOW-to-HIGH $\overline{DS}$ transition; while reading from the external system, the LOW-to-HIGH transition of $\overline{DS}$ latches data into the temporary register of the DTC (see timing diagram).
31-34	$ST_0 - ST_3$	I/O	(Status). The four Status lines are three-state, bidirectional signals containing coded information regarding the current bus transaction. When the DTC is not in control of the system bus, $ST_0 - ST_3$ are inputs and are used to detect interrupt and segement trap acknowledge cycles. There are not timing requirements between transitions on the $ST_0 - ST_3$ input and the DTC clock; input transitions on $ST_0 - ST_3$ are only defined relative to $\overline{AS}$ and $\overline{DS}$. When the DTC is in control of the system bus, the $ST_0 - ST_3$ lines are outputs which indicate the type of memory of I/O transition being performed. The status codes decoded and generated by the DTC are indicated in Figure 1 by the letters D and G, respectively.
41	R/$\overline{W}$	I/O	(Read/$\overline{Write}$).Read/Write is a bidirectional, three-state signal. Read polarity is HIGH and WRITE polarity is LOW. R/$\overline{W}$ indicates the data direction of the current bus transaction, and is stable starting when $\overline{AS}$ goes LOW until the bus transaction ends (see timing diagram). When the DTC is not in control of the system bus and the external system is transferring information to or from the DTC, R/$\overline{W}$ is a status input used by the DTC to determine if data is entering or leaving on the $SD_0 - AD_{15}$ bus during $\overline{DS}$ time. In such a case, Read (HIGH) indicates that the system is requesting data from the DTC and Write (LOW) indicates that the system is presenting data to the DTC. There are no timing requirements between R/$\overline{W}$ as an input and the DTC clock; transitions on R/$\overline{W}$ as an input are only defined relative to $\overline{AS}$ and $\overline{DS}$. When DTC is in control of the system bus, R/$\overline{W}$ is an output generated by the DTC, with Read indicating that data is being requested from the addressed location or devices and Write indicating that data is being presented to the addressed location or device. Flyby DMA operations are a special case where R/$\overline{W}$ is valid for the normally addressed memory or peripheral locations and must be interpreted in reverse by the "Flyby" peripheral that uses it.
30	N/$\overline{S}$	O	(Normal/$\overline{System}$, 3-State). Normal/System is a three-state output activated only when the DTC is in control of the system bus. This signal is used to indicate which memory space is being accessed. The N/$\overline{S}$ pin is HIGH for normal memory and LOW for system memory. System space is always indicated for I/O cycles.
35	B/$\overline{W}$	O	(Byte/$\overline{Word}$, 3-State). This output indicates the type of data transferred on the AD bus. HIGH indicates a byte (8-byte) and LOW indicates a word (16-bit) transfer. This output is activated when $\overline{AS}$ goes LOW and remains valid for the duration of the whole transaction (see timing diagram). The address generated by the DTC is always a byte address, even though the memory organized as 16-bit words. All word-sized data are word aligned and must be addressed by even addresses ($A_0 = 0$). When addressing byte transactions, the least significant address bit determines which byte is needed; an even address specifies the most significant byte ($AD_8 - AD_{15}$), and an odd address specifies the least significant byte ($AD_0 - AD_7$). (Note that the higher address specifies the less significant byte!) This addressing mechanism applies to memory accesses as well as I/O and special I/O accesses. When the DTC is a slave, it ignores the B/$\overline{W}$ signal.

Refer to page 7-1 for Essential Information on Military Devices

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
42	$\overline{CS}/\overline{WAIT}$	I	(Chip Select/Wait). When the DTC is not in control of the system bus, this pin serves as an active-low Chip Select ($\overline{CS}$) input. A CPU or other external device uses $\overline{CS}$ to activate the DTC for reading and writing of its internal registers. $\overline{CS}$ may be held LOW for multiple transfers to and/or from the DTC, provided $\overline{AS}$ and $\overline{DS}$ are toggled for each transfer. There are no timing requirements between the $\overline{CS}$ input and the DTC clock; the $\overline{CS}$ input timing requirements are only defined relative to $\overline{AS}$. When the DTC is in control of the system bus, this pin serves as an active-low $\overline{WAIT}$ input. Slow memories and peripheral devices may use $\overline{WAIT}$ to extend $\overline{DS}$ during bus transfers. Unlike the $\overline{CS}$ input, transitions on the $\overline{WAIT}$ input must meet certain timing requirements relative to the DTC clock. See the timing diagram for details. The Wait function may be disabled using a control bit in the Master Mode register; in which case, the input is treated as an active-low Chip Select only and is ignored when the DTC is in control of the system bus.
2	$\overline{BUSRQ}$	I/O	(Bus Request). Bus Request is an active-low, open-drain, bidirectional signal used by the DTC to obtain control of the bus from the CPU. Before driving $\overline{BUSRQ}$ active, the DTC samples this line to insure that another request is not already being made by another device. $\overline{BUSRQ}$ lines from multiple devices are wire-ORed together externally with a common pull-up resistor of 1.8 kΩ or more. Since the DTC internally synchronizes the sampled $\overline{BUSRQ}$ signal, transitions on $\overline{BUSRQ}$ may be asynchronous to the DTC clock.
1	$\overline{BAI}$	I	(Bus Acknowledge In). $\overline{BAI}$ is an active-low asynchronous input indicating that the CPU has relinquished the bus and that no higher priority device has assumed bus control. Since $\overline{BAI}$ is internally synchronized by the DTC before being used, transitions on $\overline{BAI}$ do not have to be synchronous with the DTC clock. The $\overline{BAI}$ input is usually connected to the $\overline{BUSAK}$ line from the CPU or to the $\overline{BAO}$ output from a higher-priority device in the Bus Request daisy chain. $\overline{AS}$ and $\overline{DS}$ must both be HIGH during the HIGH-to-LOW transition of $\overline{BAI}$.
3	$\overline{BAO}$	O	(Bus Acknowledge Out). $\overline{BAO}$ is an active-low output which indicates that $\overline{BAI}$ is active and that the DTC is not currently in control of the bus. This signal is intended for use by lower priority devices on the Bus Request daisy chain.
47	$\overline{INT}$	O	(Interrupt). Interrupt is an active-low, open-drain output used to interrupt the CPU. It may be connected to any of the CPU interrupt inputs and may be wire-ORed with other sources of interrupts. An external pull-up resistor of 1.8 kΩ or greater is required.
46	IEI	I	(Interrupt Enable In). IEI is an active-high input which allows the DTC to activate the $\overline{INT}$ output and to respond to interrupt acknowledge operations. It is used with other signals to implement the interrupt daisy chain. Transitions on IEI do not have to be synchronous with the DTC clock.
48	IEO	O	(Interrupt Enable Out). IEO is an active-high output that enables devices lower in the chain when higher priority interrupts are not pending or under service. It is used in conjunction with other signals to implement the Interrupt daisy chain. See the Interrupt section of this document for further details on $\overline{INT}$, IEI and IEO.
36, 37	$\overline{DREQ_1}$, $\overline{DREQ_2}$	I	(DMA Request). The DMA Request lines are two active-low inputs, one per channel. They may make transitions independent of the DTC clock and are used by external logic to indicate and control DMA operations performed by the DTC.
40, 39	$\overline{DACK_1}$, $\overline{DACK_2}$	O	(DMA Acknowledge). The DMA Acknowledge lines are active-low outputs, one per channel, which indicate that the channel is performing a DMA operation. $\overline{DACK}$ is pulsed, held active or held inactive during DMA transfers, as programmed in the Channel Mode register. For Flowthru operations, the peripheral is fully addressed using the conventional I/O addressing protocols and therefore may choose to ignore $\overline{DACK}$. $\overline{DACK}$ is always output as programmed in the Channel Mode register for a DMA operation, even when the operation is initiated by a CPU software request command or as a result of chaining. $\overline{DACK}$ is not output during the actual chaining operations.
38	$\overline{EOP}$	I/O	(End of Process). $\overline{EOP}$ is an active-low, open-drain, bidirectional signal. It must be pulled up with an external resistor of 1.8 kΩ or more. The DTC emits an output pulse on $\overline{EOP}$ when a TC or MC termination occurs, as defined later. An external source may terminate a DMA operation in progress by driving $\overline{EOP}$ LOW. $\overline{EOP}$ always applies to the active channel; if no channel is active, $\overline{EOP}$ is ignored. The Suppress output of the MMU may be connected to $\overline{EOP}$ to terminate DMA accesses which violate the MMU protection settings. To provide full access protection, an external $\overline{EOP}$ is accepted even during chaining.
29, 28, 25, 24-21	SN_0-SN_6	O	(Segment Number). The segment lines are three-state outputs activated only when the DTC is controlling the system bus. SN_0 is the least significant bit of the segment number and SN_6 is the most significant. The Z8001 and Z8002 CPUs access I/O by outputting a 16-bit I/O address on AD_0-AD_{15}. When the AmZ8016 DTC is operated in Logical Address space, the I/O address space is increased to 23 bits. The lower 16 bits of I/O address appear on AD_0-AD_{15}. An additional 7 bits of I/O addresses appear on SN_0-SN_6. Users of the DTC in the Logical Address space configuration may choose to disregard the SN_0-SN_6 I/O address information or may use it to increase the DTC's I/O address space beyond that of the CPU. When the AmZ8016 DTC is configured for Physical Address space, signals SN_0-SN_6 specify the 17th (SN_0) through 23rd (SN_6) bits of a 24-bit linear address. The lower 16 address bits appear on AD_0 through AD_{15} respectively; the 24th address bit is output on SN_7/MMUSync. This 24-bit linear address allows the DTC to access anywhere within 16 Megabytes of memory. Users of the DTC in the physical address space configuration may choose to disregard the extended I/O addressing capability of the DTC by disregarding SN_0-SN_6 and SN_7/MMUSync during I/O operations, or may use the extended addressing to increase the number of I/O ports accessible by the DTC beyond the number of I/O ports accessible by the CPU.

Refer to page 7-1 for Essential Information on Military Devices

PIN DESCRIPTION (Cont.)

Pin No.	Name	I/O	Description
27	SN_7/MMUSync	O	(Segment Number 7/MMUSync). When DTC is programmed in Logical Address space, this line outputs an active-HIGH MMUSYNC pulse prior to each machine cycle. The MMU uses this signal to synchronize access to its translation table and to differentiate between CPU and DTC control. The MMU ignores MMUSYNC if ST_0-ST_3 indicate I/O. This output is LOW when DTC is a bus slave and the MM_1 bit is set. In Physical address space, this line outputs SN_7 which becomes the 24th address bit in a linear address space. This bit can be used to address both memory and I/O – see the SN_0-SN_6 pin description for details. With this output SN_7, a HIGH represents 1 and a LOW represents 0. This pin floats to high impedance state when DTC is a bus slave and the MM_1 bit is cleared.

ST3	ST2	ST1	ST0	Transaction/Operation	DTC Action (Note)
L	L	L	L	Internal Operation	
L	L	L	H	Memory Refresh	
L	L	H	L	I/O Transaction	G
L	L	H	H	Special I/O Transaction	G
L	H	L	L	Segment Trap Acknowledge	D
L	H	L	H	Non-Maskable Interrupt Acknowledge	D
L	H	H	L	Non-Vectored Interrupt Acknowledge	D
L	H	H	H	Vectored Interrupt Acknowledge	D
H	L	L	L	Memory Transaction for Data/DTC Chaining	G
H	L	L	H	Memory Transaction for Stack	G
H	L	H	L	Reserved	
H	L	H	H	Reserved	
H	H	L	L	Memory Transaction for Program Fetch (Subsequent Word)	G
H	H	L	H	Memory Transaction for Program Fetch (First Word)	
H	H	H	L	Reserved	
H	H	H	H	Reserved	

Notes: D = Status code is decoded by DTC when not in control of system bus.
G = Status code is generated by DTC when in control of system bus.

Figure 1. Status Code

DETAILED DESCRIPTION

Any given DMA operation, be it a transfer, a search or a transfer-and-search operation, consists of three phases. In the first phase, the channel's registers are initialized to specify and control the desired DMA operation. In the second phase, the DMA operation itself is started and performed. The final phase involves terminating the DMA operation and performing any actions selected to occur on termination. Each of these different phases is described in detail in the following sections.

Reset

The DTC can be reset either by hardware or software. The software reset command is described in the "Commands" section. Hardware resets are applied by pulling both $\overline{AS}$ and $\overline{DS}$ LOW. Because the DTC may be in control of the bus when a reset is applied, it is important that $\overline{BAI}$ be driven HIGH when applying a reset to avoid possible bus contention between the applied LOW signals on $\overline{AS}$ and $\overline{DS}$ and the DTC's driving of these pins. As soon as $\overline{BAI}$ goes inactive, the DTC places the $AD_0 - AD_{15}$, $SN_0 - SN_6$, $ST_0 - ST_3$, $R/\overline{W}$, $N/\overline{S}$, $B/\overline{W}$, $\overline{AS}$ and $\overline{DS}$ signals in the high impedance state. If the DTC is programmed for Physical Address Space, SN_7/MMUSync will also be driven into the high impedance state when $\overline{BAI}$ goes HIGH. Figure 21 shows the suggested method of generating hardware resets for the DTC.

Both software and hardware resets clear the Master Mode register, clear CIE, IP, SIP, and WFB and set the CA and NAC in each Channel's Status register. The contents of all other DTC registers will be unchanged for a software reset. Since a hardware reset may have been applied part-way through a DMA operation being performed by a DTC channel, the channel's registers should be assumed to contain indeterminate data following a hardware reset.

Because the CA and NAC bits in the Status register are set by reset, the channel will be prevented from starting a DMA operation until its Chain Address register's Segment, Tag and offset fields are programmed and the channel is issued a "Start Chain Command".

Channel Initialization

The philosophy behind the AmZ8016 DTC design is that the DTC should be able to operate with a minimum of interaction with the host CPU. This goal is achieved by having the DTC load its own control parameters from memory into each channel. The CPU has to program only the Master Mode register and each Channel's Chain Address register. All other registers are loaded by the channels themselves from a table located in System Data memory and pointed to by the Chain Address register. This reloading operation is called chaining and the table is called the Chain Control Table.

The offset and segment fields of the Chain Address Register form a 24-bit address or a 23-bit address, which points to a location in system data memory space. Chaining is performed by repetitively reading words from memory. Note that the Chain Address register should always be loaded with an even offset; loading an odd offset will cause unpredictable results. The 2-bit Tag field facilitates interfacing to slow memory by allowing the user to select 0, 1, 2 or 4 programmable wait states. The DTC will automatically insert the programmed number of wait states in each memory access during chaining.

The Chain Address register points to the first word in the Chain Control Table. This word is called the Reload Word. See Figure 2. The purpose of the Reload Word is to specify which registers in the channel are to be reloaded. Reload Word bits 10 – 15 are undefined and may be 0 or 1. Each of bits 0 through 9 in the Reload Word correspond to either one or two registers in the channel (see Figure 3). When a Reload Word bit is '1', it means that the register or registers corresponding to that bit are to be reloaded. If a Reload Word bit is '0', the register or registers corresponding to that bit are not to be reloaded. The data to be loaded into the selected register(s) follow(s) the Reload Word in memory (i.e., the data are at successively larger memory addresses). The Chain Control Table is a variable length table. Only the data to be loaded are in the table and the data are packed together.

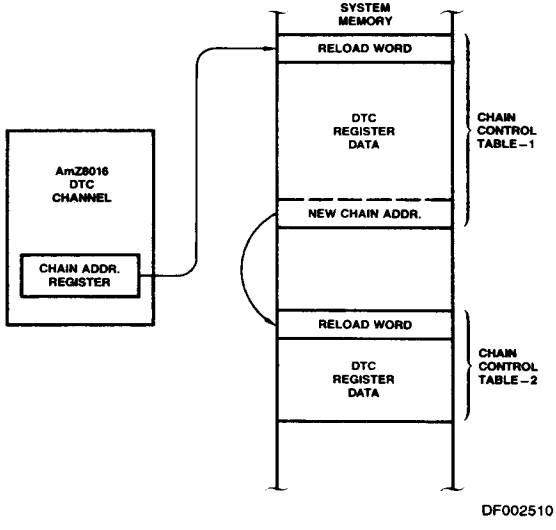


Figure 2. Chaining and Chain Control Tables

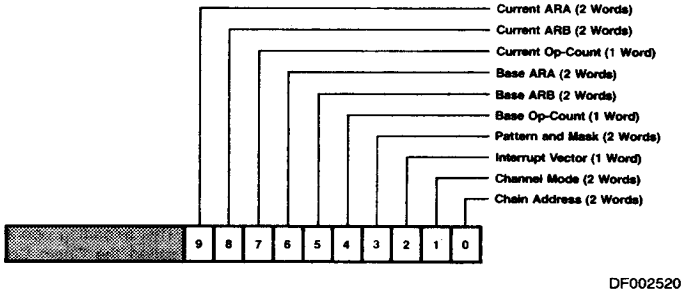


Figure 3. Reload Word

When the channel is to reload itself, it first uses the Chain Address register contents to load the Reload Word into the DTC's Chain Control Register. Next, the Chain Address register contents are incremented by two to point to the next word in memory. The channel then scans to Reload Word register from bit 9 down to bit 0 to see which registers are to be reloaded. If no registers are specified (bits 9 – 0 are all 0), no registers will be reloaded. If at least one of bits 9 – 0 are set to '1', the register(s) corresponding to the most significant set bit are reloaded, the bit is cleared and the Chain Address register is incremented by 2. The channel continues this operation of scanning the bits from the most significant to least significant bit position clearing each set bit after reloading its associated registers and incrementing the Chain Address register by 2. If all of bits 9 to 0 are set, all the registers will be reloaded in the order: Current ARA, Current ARB, Current Operation Count, . . . Channel Mode and Chain Address. Figure 4 shows two examples of Chain Control Tables. Example 1 shows the ordering of data when all register are to be reloaded. In example 2 only some registers are reloaded. Once the channel is reloaded, it is ready to perform a DMA operation. Note when loading Address Registers the Segment and Tag Word are loaded first, then the Offset Word.

Initiating DMA Operations

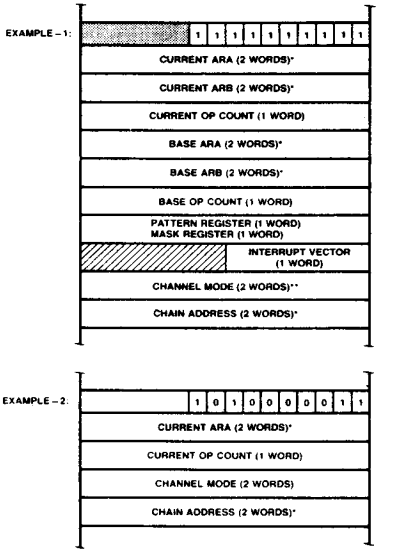
DMA Operations can be initiated in one of three ways – by software request, by hardware request and by loading a set software request bit into the Channel Mode register during Chaining.

Starting After Chaining

If the software request bit of the Channel Mode register is loaded with a '1' during chaining, the channel will perform the programmed DMA operation at the end of chaining. If the channel is programmed for Single Operation or Demand, it will perform the operation immediately. The channel will give up the bus after chaining and before the operation if the CPU Interleave bit in the Master Mode register is set. See the "Channel Response" section for details. Note that once a channel starts a chaining operation by fetching a Reload Word, it retains bus control at least until chaining of the last register's data is performed.

Software Requests

The CPU can issue Software Request commands to start DMA Operations on a channel. This will cause the channel to request the bus and perform transfers. See the description of the software request command for details.



*Note: Load the segment and tag word first, then the offset word.

**Most significant word first, then least significant.

Figure 4. Examples of Chain Control Table

Hardware Requests

DMA operations will often be started by applying a LOW on the channel's $\overline{DREQ}$ input. The "Channel Response" section describes when LOW $\overline{DREQ}$ signals are sampled and when $\overline{DREQ}$ requests can be applied to start the next DMA operation after chaining.

Bus Request/Grant

Before the DTC can perform a DMA Operation, it must gain control of the system bus. The $\overline{BUSRQ}$, $\overline{BAI}$ and $\overline{BAO}$ interface pins provide connections between the DTC and the host CPU and other DMA devices, to arbitrate which device has control of the system bus. When the DTC wants to gain bus control, it drives $\overline{BUSRQ}$ LOW.

Refer to page 7-1 for Essential Information on Military Devices

Some period of time after the DTC drives $\overline{BUSRQ}$ LOW, the CPU will relinquish bus control and drive its $\overline{BUSAK}$ signal LOW. This passes down the $\overline{BAI}$, $\overline{BAO}$ daisy chain. When the DTC's $\overline{BAI}$ input goes LOW, it may begin performing operations on the system bus. When the DTC finishes its operation, it stops driving $\overline{BUSRQ}$ LOW and allows $\overline{BAO}$ to follow $\overline{BAI}$.

Listed below are the rules followed by the DTC to request, acquire, and release the system bus. A description of the significance of the steps follows.

1. The DTC requests control of the system bus by driving $\overline{BUSRQ}$ LOW. The DTC may only drive $\overline{BUSRQ}$ LOW if $\overline{BUSRQ}$ is HIGH and $\overline{BAI}$ is HIGH.
2. After driving $\overline{BUSRQ}$ LOW, the DTC waits for its request to be acknowledged on the $\overline{BAI}$ input. When $\overline{BAI}$ goes LOW, the DTC has bus control, performs its operations and continues to drive $\overline{BUSRQ}$ until it completes.
3. When the DTC is finished with the system bus, it stops driving $\overline{BUSRQ}$ LOW and passes the LOW on $\overline{BAI}$ through to $\overline{BAO}$.
4. If the DTC is not requesting use of the bus, $\overline{BAO}$ always follows $\overline{BAI}$. If the DTC receives a LOW on $\overline{BAI}$ and $\overline{BUSRQ}$ is LOW and the DTC is not requesting use of the bus, the DTC drives $\overline{BAO}$ LOW. This situation would occur if some lower priority device was pulling $\overline{BUSRQ}$ LOW. The DTC simply passes the LOW $\overline{BAI}$ grant signal through to the lower priority device.

Note that $\overline{BAO}$ will always be LOW if $\overline{BAI}$ is LOW providing the DTC is not driving $\overline{BUSRQ}$ LOW. If the DTC is driving $\overline{BUSRQ}$ LOW, $\overline{BAO}$ will go LOW when the DTC finishes using the bus and stops applying a LOW to $\overline{BUSRQ}$. Note also that $\overline{BUSRQ}$ is a bidirectional signal. Since the DTC can only drive $\overline{BUSRQ}$ LOW if $\overline{BUSRQ}$ was previously HIGH, the DTC is able to sample $\overline{BUSRQ}$. Because the DTC may be on a different card than other DTCs and the CPU, some means must be provided to bidirectionally buffer $\overline{BUSRQ}$. Figure 21 shows a representative system with two DTC chips and a CPU. Figure 22 shows the logic used to bidirectionally buffer $\overline{BUSRQ}$. Note that the buffer and gates in the logic are both open collector (o.c.) devices.

It is necessary to ensure that all DTCs will behave identically, regardless of whether they are on the same card or different cards. Also, it is undesirable to require users to provide to Detail A logic on all DTCs, except where the logic is needed to provide buffering to drive a backplane. For this reason, each DTC incorporates identical logic to Detail A inside the chip. Thus, even if no external logic is used, the bus request-grant protocol will follow the above description. Note that when external buffering is used, the design of Detail A is such that when it is placed in series with the replicated Detail A logic inside the chip, the operation of the bus request protocol remains unchanged.

DMA Operations

There are three types of DMA operations: Transfer, Search and Transfer-and-Search. Transfers move data from a source location to a destination location. Two types of transfers are provided: Flowthru and Flyby. Searches read data from a source and compare the read data to the contents of the Pattern register. A Mask register allows the user to declare "don't care" bits.

The user can program that the search is to stop either when the read data matches the masked pattern or when the read data fails to match the masked pattern. This capability is called Stop-on-Match and Stop-on-no-Match. Transfer-and-Search combines the two functions to facilitate the transferring of variable length data blocks. Like transfer, Transfer-and-Search can be performed in either Flowthru or Flyby mode.

Transfers

Transfers use four of the Channel registers to control the transfer operation: the Current ARA and ARB register; the Current Operation Count register; and the Channel Mode register. Channel Mode register bit CM_4 is called the Flip bit and is used to select whether ARA is to point to the source and ARB is to point to the destination or vice-versa. The Current Operation Count register specifies the number of words or bytes to be transferred.

Bits $CM_3 - CM_0$ in the Channel Mode register program whether Flowthru or Flyby transfer is to be performed. Flowthru transfers are performed in either two or three steps. First, the channel outputs the address of the source and reads the source data into the DTC's Temporary register. In two-step Flowthru Transfer, the channel will then address the destination and write the Temporary register data to the destination location. The three-step Flowthru operation is described later in this section. The source and destination for Flowthru Transfers can both be memory locations or both peripheral devices or one may be a memory location and the other a peripheral device. The $\overline{DACK}$ output for the transferring channel may be programmed to be inactive throughout the transfer or active during the transfer. This is controlled by bit CM_{18} in the Channel Mode register.

Flyby transfers provide improved transfer throughput over Flowthru but are restricted to transfers between memory and peripherals or between two peripherals. Flyby operations are described in detail in the "Flyby Transactions" section.

Transfers can use both byte- and word-sized data. Flowthru byte-to-byte transfers are performed by reading a byte from the source and writing a byte to the destination. The Current Operation Count register must be loaded with the number of bytes to be transferred. Both the Current ARA and Current ARB registers, if programmed to increment/decrement, will change by ±1 if the register points to memory space and by ±2 if the register points to I/O space.

Flowthru word-to-word transfers require that the Current Operation Count specify the number of words to be transferred. Both the Current ARA and Current ARB registers, if programmed to increment/decrement, will change by ±2 regardless of whether the register points to memory or I/O space.

Byte-word funneling provides packing and unpacking of byte data to facilitate high speed transfers between byte and word peripherals and/or memory. This funneling option can only be used in Flowthru mode. Funneled Flowthru transfers are performed in three steps. For transfers from a byte source to a word destination, two consecutive byte reads are performed from the source address. The data read is assembled into the DTC's Temporary register. In the third step, the Temporary register data is written to the destination address in a word transfer. Funneled transfers from a word source to a byte destination are performed by first loading a word from the source into the DTC's Temporary register. The word is then written out to the destination in two byte writes. For funnel operations, the byte-oriented address must be in the Current ARA register and the word-oriented address must be in the Current ARB register. The Flip bit (CM_4) in the Channel Mode register is used to specify which address is the source and which is the destination. When the byte address is to be incremented or decremented, the increment/decrement operation occurs after each of the two reads or writes. The increment/decrement is by ±1.

Refer to page 7-1 for Essential Information on Military Devices

In byte-to-word funneling operations it is necessary to specify which half of the Temporary register (upper or lower byte) is loaded with the first byte of data. Similarly, for word-to-byte funneling operations it is necessary to define which half of the Temporary register is written out first. Figure 5 summarizes these characteristics for both byte-to-word and word-to-byte funneling operations. The criteria used to determine the packing/unpacking order is based on whether the Current ARB register is programmed for incrementing or decrementing of the address. Note that if the address is to remain unchanged (i.e., if bit TG_4 in the Tag Field of the Current ARB register is 1), the increment/decrement bit (bit TG_3) still specifies the packing order.

Search

Searches use five of the Channel registers to control the transfer operation: either the Current ARA or ARB; the Operation Count; the Pattern and Mask registers; and the Channel Mode register. Channel Mode register bit CM_4 is called the Flip bit and is used to select either Current ARA or ARB as the register specifying the source for the search. Only one of the Current Address registers is used for search operations since there is no destination address required. Channel mode register bit CM_2 is an enable for the output of the comparator and allows the MC (match condition) signal to be generated. The Current Operation Count register specifies the maximum number of words or bytes to be searched.

Search operations involve repetitive reads from the peripheral or memory until the specified match condition is met. The search then stops. This is called a Match Condition or MC termination. Each time a read is performed, the Source address, if so programmed, is incremented or decremented and the Operation Count is decremented by 1. If the match condition has not been met by the time the Operation Count reaches zero, the zero value will force a TC termination, ending the search. Searches can also stop due to a LOW being applied to the $\overline{EOP}$ interface pin. During a search operation, the channel's $\overline{DACK}$ output will be either inactive or active throughout the search. This is controlled by bit CM_{18} in the Channel Mode register. The reads from the peripheral or memory performed during search follow the timing sequences described in the "Flowthru Memory Transactions" and "Flowthru I/O Transactions" sections.

On each read during a Search operation, the DTC's Temporary register is loaded with data and compared to the Pattern register. The user can select that the search is to stop when the Pattern and Temporary register contents match or when they don't match. This Stop-On-Match/Stop-On-No-Match feature is programmed in bit CM_{17} of the Channel Mode register. A Mask register allows the user to exclude or mask selected Temporary register bits from the comparison by setting the corresponding Mask register bit to "1." The masked bits are defined to always match. Thus, in Stop-On-Match, successful matching of the unmasked bits, in conjunction with the always-matched masked bits, will cause the search to stop. For Stop-On-No-Match, the always-matched masked bits are by definition excluded from not matching and therefore excluded from stopping the search.

For word reads the user may select either 8-bit or 16-bit compares through Channel Mode register bit CM_{16}. In an 8-bit, Stop-On-Match, word-read operation, successful matching of either the upper or lower byte of unmasked Pattern and Temporary registers bits will stop the search. Both bytes do not have to match. In 16-bit Stop-On-Match with word reads, all unmasked Pattern and Temporary register bits must match to stop the search. In an 8-bit or 16-bit, Stop-On-No-Match, word-read Search operation, failure of any bit to match will terminate the Search operation.

In an 8-bit Stop-On-Match the byte-reads, the Search will Stop if either the upper or lower byte of unmasked Pattern and Temporary register bits match. For an 8-bit Stop-On-No-Match with byte reads, failure of matching in any unmasked Pattern and Temporary register bit will cause the search to stop.

For 8-bit searches, the upper and lower bytes of the Pattern and Mask register should usually be programmed with the same data. Failure to set the upper and lower bytes of the Pattern and Mask registers to identical values will result in different comparison criteria being used for the upper an lower bytes of the Temporary register. Users failing to program identical values for the upper and lower bytes can predict the results by recognizing that in 8-bit Stop-On-Match, the search will end if all the unmasked bits in either the upper or lower byte matches, and for 8-bit Stop-On-No-Match, the failure of any unmasked bit to match will end the search. For accurate predictions, it is also necessary to know that for word reads the Temporary register high and low bytes are loaded from $AD_{15} - AD_8$ and $AD_7 - AD_0$ respectively. In byte reads, the read byte is duplicated in both halves of the Temporary register except in funneling.

Funneling Direction	Current ARB Tag Field		Increment/Decrement and Packing/Unpacking Rules
	TG4	TG3	
Word-to-Byte (Flip-bit = 1)	0	0	Increment ARB, Write High Byte First
	0	1	Decrement ARB, Write Low Byte First
	1	0	Hold ARB, Write High Byte First
	1	1	Hold ARB, Write Low Byte First
Byte-to-Word (Flip-bit = 0)	0	0	Increment ARB, Read High Half of Word Written First
	0	1	Decrement ARB, Read Low Half of Word Written First
	1	0	Hold ARB, Read High Half of Word Written First
	1	1	Hold ARB, Read Low Half of Word Written First

Figure 5. Byte/Word Funneling

Refer to page 7-1 for Essential Information on Military Devices

Transfer-and-Search

Transfer-and-Search combines the operations of Transfer and Search functions. The registers used to control Transfer-and-Searches are the Current ARA and ARB register, the Operation Count register, the Pattern and Mask register, and the Channel Mode register.

A Transfer-and-Search operation will end when the data transferred meets the match condition specified in Channel Mode register bits $CM_{17} - CM_{16}$. The Mask and Pattern registers indicate those bits being compared with the Temporary register contents. Like Transfers and Searches, Transfers-and-Searches will also be terminated if the operation count goes to zero or if a LOW is applied to the $\overline{EOP}$ pin. Regardless of whether Transfer-and-Search stops because of a TC, MC or EOP, it will always complete the iteration by writing to the destination address before ending (writing twice for word-to-byte funneling).

In Flowthru mode, Transfer-and-Search the timing is identical to Flowthru Transfer. While the data is in the Temporary register, it is masked by the Mask register and compared to the Pattern register. For word Transfer and Transfer-and-Search, the high and low bytes of the Temporary register are always written to and read from $AD_{15} - AD_8$ and $AD_7 - AD_0$ respectively. For byte Transfer and Transfer-and-Search, the byte read is always loaded into both halves of the Temporary register and the entire register is driven directly out onto the $AD_{15} - AD_0$ bus. Transfer-and-Search can also be used with byte word funneling. In funneling, the match is an 8-bit match or 16-bit match as determined by the setting of bit CM_{16} and CM_{17}.

Flyby Transfer-and-Search can be used to increase throughput for transfer between two peripherals or between memory and a peripheral. In this operation, the operand sizes of the source and destination must be the same. A complete discussion of Flyby timing is given in the "Flyby Transactions" section. During a Flyby Transfer-and-Search, data is loaded into the Temporary register to facilitate the comparison operation and at the same time data is transferred from the source to the destination. When byte operands are used, data is loaded into both bytes of the Temporary register, from the $AD_{15} - AD_8$ bus if the Current ARA register is even and from $AD_7 - AD_0$ line if the Current ARA register is odd. This will alternate for memory bytes so the user must drive both halves of the bus to use the search. When word operands are used, data is loaded directly from $AD_{15} - AD_8$ and $AD_7 - AD_0$ into the Temporary register's high and low bytes respectively.

Channel Response

Channel Mode register bits $CM_6 - CM_5$ select the channel's response to the request to start a DMA operation. The response falls into either of two types: Single Operation or Demand. There are three subtypes for Demand operations: Demand Dedicated with Bus Hold, Demand Dedicated with Bus Release, and Demand Interleave. To make the discussions clear, it is necessary to define the term "single iteration of a DMA operation". For Search operations, one iteration consists of a single read operation and a comparison of the read data to the unmasked Pattern register bits. The Operation Count will be decremented by 1 and the Current Address register used incremented or decremented if so programmed. For Transfer and Transfer-and-Search operations, a single iteration comprises reading a datum from the source, writing it to the destination, comparing the read datum to the unmasked Pattern register bits (Transfer-and-Search only), decrementing the Operation Count by 1 and incrementing/decrementing the Current ARA and ARB registers if so programmed. In byte-word funneling, a single iteration consists of two reads followed by a write (Byte-to-Word funneling) or one read followed by two writes (Word-to-Byte funneling). In all Transfer and Transfer-and-Search cases the iteration will not stop until the data in the Temporary register is written to the destination.

Single Operation

The Single Operation response is intended for use with peripherals which transfer single bytes or words at irregular intervals. Each application of a Software request command will cause the channel to perform a single iteration of the DMA operation. Similarly, if the Software request bit is set by chaining, at the end of chaining the channel will perform a single iteration of the DMA operation. Each application of a HIGH-to-LOW transition on the $\overline{DREQ}$ input will also cause a single iteration of the DMA operation. If the Hardware mask bit is set when the transition is made, the iteration will be performed when the mask is cleared, providing the DMA operation has not terminated. See the Set/Clear Hardware mask bit command for details. Each time a Single Operation ends, the channel will give up control of the bus unless a new transition has occurred on $\overline{DREQ}$. The new transition can occur anytime after the LOW-to-HIGH $\overline{AS}$ transition on the first memory or I/O access of the DMA iteration. Timing Diagram 1 shows the times after which a new transition can be applied and recognized to avoid giving up the bus at the end of the current iteration.

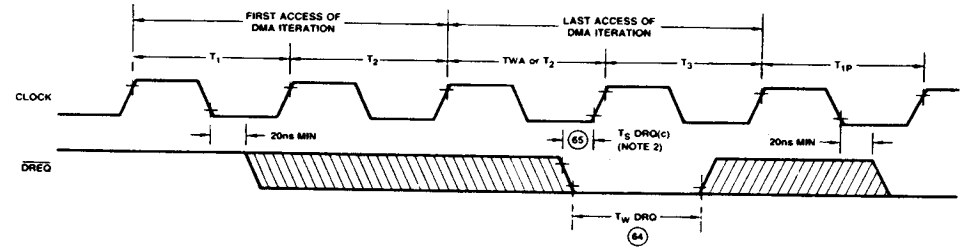
Demand Dedicated with Bus Hold

In Demand Dedicated with Bus Hold (abbreviated Bus Hold), the application of a Software request command or the setting of the software request bit during chaining or applying a LOW level on the $\overline{DREQ}$ input will cause the channel to acquire bus control.

If $\overline{DACK}$ is programmed as a level output ($CM_{18} = 0$), $\overline{DACK}$ will be active from when the channel acquires bus control to when it relinquishes control.

Once the channel gains bus control due to a LOW $\overline{DREQ}$ level, it samples $\overline{DREQ}$ as shown in Timing Diagram 2. If $\overline{DREQ}$ is LOW, an iteration of the DMA operation is performed. If $\overline{DREQ}$ is HIGH, the channel retains bus control and continues to drive all bus control signals active or inactive, but performs no DMA operation. This the user can start or stop execution of DMA operations by modulating $\overline{DREQ}$. Once TC, MC or EOP occurs, the channel will either release the bus or, if chaining or Base-to-Current reloading is to occur, perform the desired operation. After chaining or Base-to-Current reloading, if the channel is still in Bus Hold mode and does not have a set software request bit (set either by chaining or command), the channel will relinquish bus control unless a LOW $\overline{DREQ}$ level occurs within the time limits.

Refer to page 7-1 for Essential Information on Military Devices



WF005310

Notes: 1. HIGH-to-LOW $\overline{DREQ}$ transitions will only be recognized after the HIGH-to-LOW transition of the clock during T_1 of the first access of the DMA iteration.
2. A HIGH-to-LOW $\overline{DREQ}$ transition must meet the conditions in Note 1 and must occur $T_SDRQ(c)$ before state T_3 of the last access of the DMA iteration if the channel is to retain bus control and immediately start the next iteration. $\overline{DREQ}$ may go HIGH before $T_SDRQ(c)$ if it has met the T_WDRQ parameter.
3. Flyby and Search transactions have only a single access; parameter $T_SDRQ(c)$ should be referenced to the start of T_3 of the access. All other operations will always have two or three accesses per iteration.

Timing Diagram 1. Sampling $\overline{DREQ}$ During Single Transfer DMA Operations

Demand Dedicated with Bus Release

In Demand Dedicated with Bus Release (abbreviated Bus Release), the application of a Software Request command will cause the channel to request the bus and perform the programmed DMA operation until TC, MC or EOP. If the channel was programmed for Bus Release, and the software request bit was set during chaining, the channel will start the DMA operation as soon as chaining ends, without releasing the bus, and will continue performing the operation until TC, MC or EOP.

When an active $\overline{DREQ}$ is applied to a channel programmed for Bus Release, the channel will acquire the bus and perform DMA operations until (a) TC, MC or EOP or (b) until $\overline{DREQ}$ goes inactive. Timing Diagram 2 (b) shows when $\overline{DREQ}$ is sampled to determine if the channel should perform another cycle or release the bus. Note that this sampling also occurs on the last cycle of a chaining operation. If a channel has an active $\overline{DREQ}$ at the end of chaining, it will begin performing DMA operations immediately, without releasing the bus. When a TC, MC or EOP occurs, terminating a Bus Release mode operation, the channel, if enabled for chaining and/or Base-to-Current reloading, will perform chaining and/or reloading (assuming the Status register's SIP bit is clear) without releasing the bus.

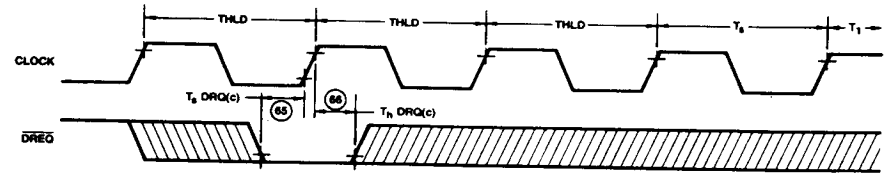
If an active request is not applied and the channel is in Demand Dedicated with Bus Hold, the channel will go into state THLD (see Timing Diagram 2 (a)). If an active request is not applied and the channel is in Demand Dedicated with Bus Release or Demand Interleave mode, it will release the bus. Note that even if an active request is applied in Demand Interleave, the channel may still release the bus. The request for Demand Interleave should continue to be applied to ensure that the channel eventually responds to the request by acquiring the bus (i.e., the request is not latched by the channel).

Demand Interleave

Demand Interleave behaves in different ways depending on the setting of Master Mode register bit MM_2. If MM_2 is set, the DTC will always relinquish bus control and then re-request it after each DMA iteration. This permits the CPU and other devices to gain bus control. For instance, if MM_2 is clear, control can pass from one DTC channel to the other without requiring the DTC to release bus control. If both channels have active requests, control will pass to the channel which did not just have control. If MM2 is clear and both channels have active requests and are in Demand Interleave mode, control will toggle between the channels after each DMA operation iteration and the DTC will retain bus control until both channels are finished with the bus. If MM2 is set and both channels have active requests and are in Demand Interleave mode, each channel will relinquish control to the CPU after each iteration resulting in the following control sequence: channel 1, CPU, channel 2, CPU, etc. Note that if there are other devices on the bus request daisy chain, they may gain control during the part of the sequence labeled CPU.

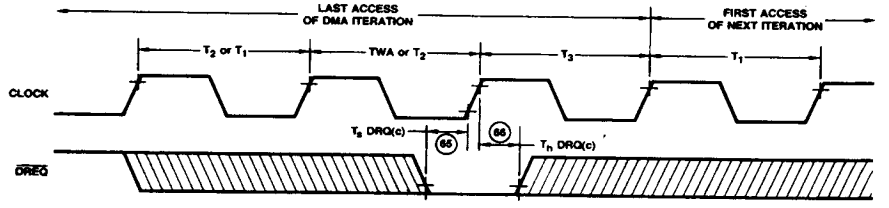
A software or hardware request will cause a channel programmed for Demand Interleave to perform interleaved DMA operations until TC, MC or EOP. If the Software request bit is set during chaining, the channel will retain the bus after chaining and will immediately start performing DMA iteration and will interleave all DMA iterations after the first. If $\overline{DREQ}$ is LOW on the last cycle during chaining, the channel will perform a single iteration immediately after chaining and interleave thereafter until (a) TC, MC or EOP or (b) $\overline{DREQ}$ goes HIGH. If (b) occurs, the channel will relinquish the bus until $\overline{DREQ}$ goes LOW again and the channel again starts performing interleaved operations. If (a) occurs, the channel will not interleave before first performing chaining and/or Base-to-Current reloading (assuming SIP is cleared).

The waveform of $\overline{DACK}$ is programmed in Channel Mode Register (CM_{18}). The Pulsed $\overline{DACK}$ is for flyby transaction only. See Timing Diagram 3. Note: This figure shows a single Search or Flyby iteration. State TWA is optionally inserted if programmed. For more than one iteration, the level $\overline{DACK}$ output would stay active during the time the channel had bus control. When CM_{18} is set, the $\overline{DACK}$ output will be inactive for all non-flyby modes.



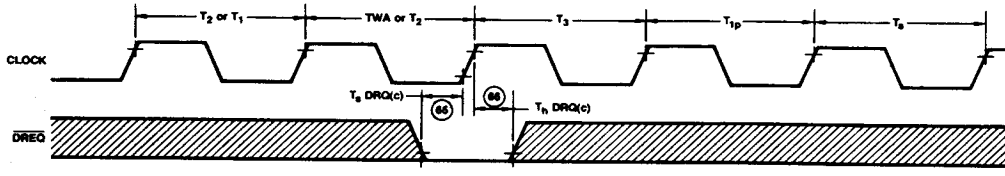
WF005320

a) Sampling of $\overline{DREQ}$ while in Bus Hold Mode



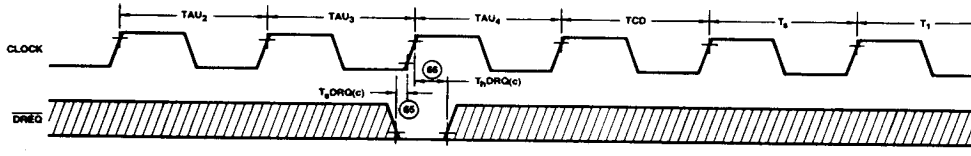
WF005330

b) $\overline{DREQ}$ Sampling in Demand Mode During DMA Operations



WF005340

c) Sampling $\overline{DREQ}$ at the End of Chaining



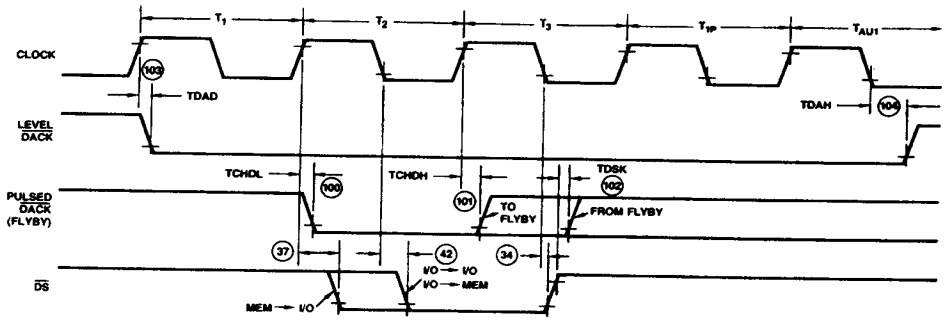
WF005350

d) Sampling $\overline{DREQ}$ at the End of Base-to-Current Reloading

Notes: 1. $\overline{DREQ}$ must be LOW from the start of T_SDRQ(c) to the end of T_hDRQ(c) to ensure that the request is recognized.
2. Failure to meet this set-up time will result in the channel releasing the bus.
3. T_S is a set-up state, generated before entering DMA operation cycle.
4. TAU_2 through TAU_4 are auto-reloading states, followed by TCD (chain decision) state.

Timing Diagram 2. $\overline{DREQ}$ Sampling in Demand Mode

Refer to page 7-1 for Essential Information on Military Devices



WF005360

Note: LEVEL $\overline{\text{DACK}}$ RE occurs as shown if auto-reloading is not programmed. LEVEL $\overline{\text{DACK}}$ stays LOW for three additional clocks for reloading.

Timing Diagram 3. $\overline{\text{DACK}}$ Timing

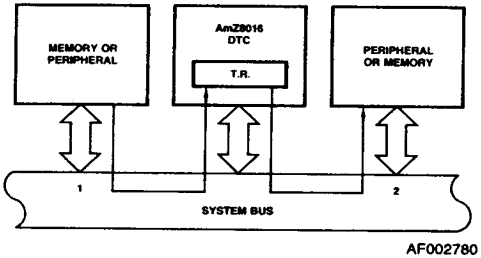
Wait States

The DTC has a $\overline{\text{WAIT}}$ input which is multiplexed with Chip Select ($\overline{\text{CS}}$) yielding a $\overline{\text{CS}}/\overline{\text{WAIT}}$ input. This pin functions as a $\overline{\text{CS}}$ for the DTC when the DTC is not in control of the bus and as a $\overline{\text{WAIT}}$ input when the DTC is bus master. Because multiplexing $\overline{\text{CS}}$ and $\overline{\text{WAIT}}$ requires external logic, (see Figure 23), the user can select that wait states are automatically inserted when the DTC, as bus master, accesses I/O or memory addresses. The number of wait states to be added to the memory or I/O transfer can be programmed by the user as 0, 1, 2 or 4 and can be separately programmed for the Current Address registers A and B and for the Chain Address register. This allows different speed memories and peripheral to be associated with each of these addresses. The Base Address registers A and B also have a Tag Field which is loaded into the Current ARA and ARB registers during Base-to-Current reloading. Because many users utilizing the software programmable wait states will not need the ability to generate hardware wait states through the $\overline{\text{CS}}/\overline{\text{WAIT}}$ pin, the wait function can be disabled, yielding a Chip Select input only, by clearing the Wait Line Enable bit (MM_3) in the Master Mode register.

During memory transactions, the $\overline{\text{WAIT}}$ input is sampled in the middle of the T_2 state. If $\overline{\text{WAIT}}$ is HIGH, and if no programmable wait states are selected, the DTC will proceed to state T_3. Otherwise, at least one wait state will be inserted. The flowthru I/O transaction should be programmed to have one wait state inserted (TWA) for Z8000 peripherals, otherwise timing is the same as memory transactions. The $\overline{\text{WAIT}}$ line is then sampled in the middle of state TWA. If $\overline{\text{WAIT}}$ is HIGH the DTC will proceed to state T_3. Otherwise additional wait states will be inserted.

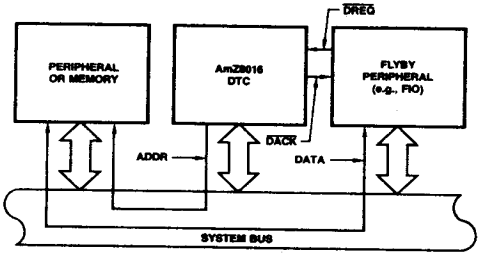
Consider what happens in a transaction when both hardware and software wait states are inserted. Each time the $\overline{\text{CS}}/\overline{\text{WAIT}}$ line is sampled, if it is LOW, a hardware wait state will be inserted in the next cycle. The software wait state insertion will be suspended until $\overline{\text{CS}}/\overline{\text{WAIT}}$ is sampled and is HIGH. The hardware wait states may be inserted anytime during the software wait state sequence. It is important to note that hardware wait states are served consecutively rather than concurrently with software wait states. For example, assume for a Flowthru I/O Transaction that a user has programmed 4 software wait states. Driving a LOW on the $\overline{\text{CS}}/\overline{\text{WAIT}}$ input during T_2 for 2 cycles would insert 2 hardware wait states. Driving $\overline{\text{CS}}/\overline{\text{WAIT}}$ HIGH for 3 cycles would allow insertion of three of the four software wait states. Driving $\overline{\text{CS}}/\overline{\text{WAIT}}$ LOW for 2 more cycles would insert 2 more hardware wait states. Finally, driving $\overline{\text{CS}}/\overline{\text{WAIT}}$ HIGH would allow the final software wait state to be inserted. During this last software wait state, the $\overline{\text{CS}}/\overline{\text{WAIT}}$ pin would be sampled for the last time. If it is HIGH, the channel will proceed to state T_3.

If the pin is LOW, the channel will insert hardware wait states until the pin goes HIGH and the channel would then enter state T_3 to complete the I/O transaction.



AF002780

Figure 6. Configuration of Flowthru Transaction



AF002790

Figure 7. Configuration of Flyby Transaction

Refer to page 7-1 for Essential Information on Military Devices

DMA Transactions

There are three types of transactions performed by the AmZ8016 DTC: Flowthru, Flyby, and Search. Figures 6 and 7 show the configurations of Flowthru and Flyby Transactions.

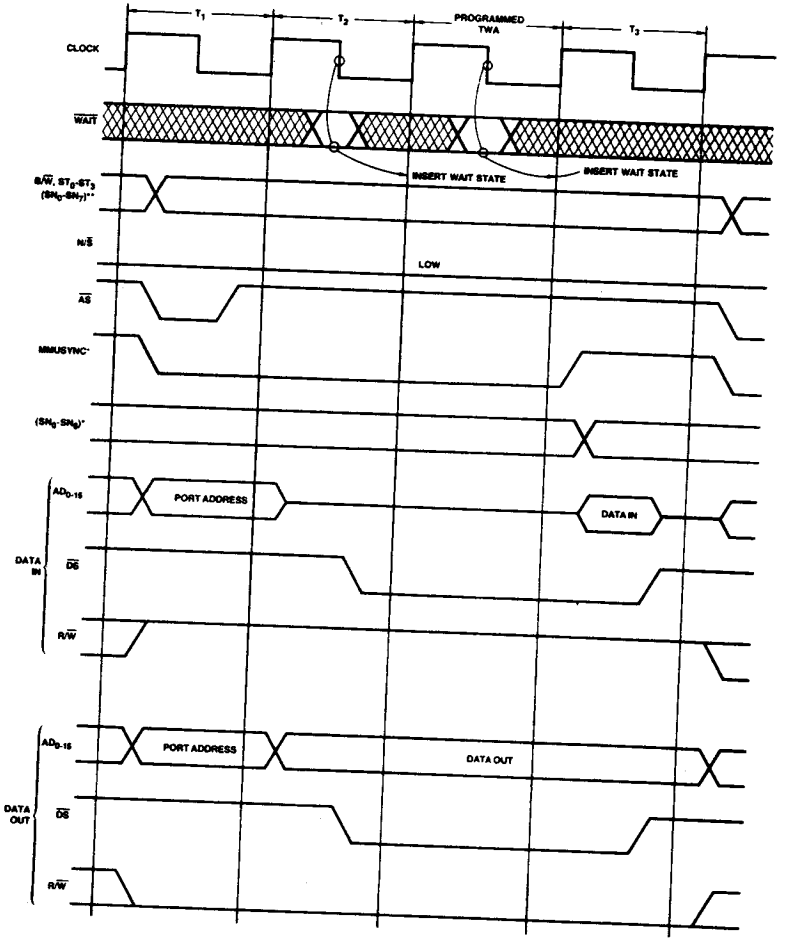
Flowthru I/O Transactions

There are Two types of I/O space on the AmZ8016: I/O and Special I/O. Status lines $ST_0 - ST_3$ specify when an I/O operation is being performed and which of the two I/O spaces are being accessed, as shown in Figure 1. During an I/O transaction, status signal N/$\overline{S}$ will be LOW to indicate a System Level operation.

Each I/O space is addressed by the host CPU by a 16-bit address. The DTC allows an extended I/O address of 24- or 23-bits to be used at the option of the user. When the Master Mode register bit MM_1 is cleared, the DTC is configured for Physical Address space. If MM_1 is set, the DTC is configured for Logical Address space.

The timing for I/O and Special I/O operations are identical. An I/O cycle consists of three states: T_1, T_2, and T_3 as shown in Timing Diagram 4. The TWA state is a wait state programmed to be inserted by the user. The user may select to insert additional software wait states through the Tag fields of the Current ARA and ARB registers. In addition, if Master Mode register bit $MM_3 = 1$, hardware wait states may be inserted by driving a LOW signal on the $\overline{CS}/\overline{WAIT}$ pin.

The $ST_0 - ST_3$ lines will reflect the appropriate code for the current cycle (I/O or Special I/O) early in T_1 and the AS output will be pulsed LOW to mark the beginning of the cycle. The



WF005650

*For logical addressing only
**For physical addressing only

Timing Diagram 4. Flowthru I/O Transactions

Refer to page 7-1 for Essential Information on Military Devices

offset portion of the address for the peripheral being accessed will appear on $AD_0 - AD_{15}$ during T_1. The N/$\overline{S}$ line will be set LOW (system) and the R/$\overline{W}$ line and B/$\overline{W}$ line will select a read or write operation for bytes or words. The N/$\overline{S}$, R/$\overline{W}$ and B/$\overline{W}$ lines will become stable during T_1 and will remain stable until after T_3. I/O address space is byte-address but both 8- and 16-bit data sizes are supported. During I/O transactions the B/$\overline{W}$ output signal will be HIGH for byte transactions and LOW for word transactions. For I/O transactions, both even and odd addresses can be output, hence the address bit output on AD_0 may be 0 or 1.

The channel can perform both I/O read and I/O write operations. During an I/O read, the R/$\overline{W}$ output will be HIGH. The $AD_0 - AD_{15}$ bus will be placed in the high impedance state by the DTC during T_2. The DTC drive the $\overline{DS}$ output LOW to signal the peripherals that data can be gated onto the bus. The DTC will strobe the data into it's Temporary register during T_3. $\overline{DS}$ will be driven HIGH to signal the end of the I/O transaction.

For byte I/O writes, the channel will drive the same data on data bus lines $AD_0 - AD_7$ and $AD_8 - AD_{15}$. During byte I/O reads when the address bit on AD0 is 0, the DTC will strobe data in from data lines $AD_8 - AD_{15}$. During byte I/O reads when the address bit on AD_0 is 1, the DTC will strobe data in from data lines $AD_0 - AD_7$. Thus, when an 8-bit peripheral is connected to the bus, it's internal registers will typically be mapped at all even or all odd addresses. To simplify access to 8-bit peripherals, byte oriented I/O address are incremented/decremented by 2.

Flowthru Memory Transactions

There are six status codes which can be generated by a DTC channel while it is accessing memory. See Figure 2. Thus, if a user segregates memory into different banks by decoding the NORMAL/$\overline{SYSTEM}$ and $ST_0 - ST_3$ lines, the DTC can be used to move data from space to space.

The timing for all Flowthru memory transactions is the same, regardless of the status code being output. During chaining operations the DTC reads words from an address in System Data memory pointed to by the active channel's Chain Address register. Those chaining operations are performed identically to the Flowthru memory read transactions, except that the data is loaded into an internal DTC channel register rather than the Temporary register. Note that chaining never causes a write or a byte read; thus, all memory writes or all byte accesses are due to DMA operations. A typical memory operation consists of three cycles: T_1, T_2 and T_3, as shown in Timing Diagram 5. The user may select to insert 1, 2, or 4 software wait states after state T_2 and before state T_3 by programming the Current Address register Tag field. If the Wait Line Enable bit in the Master Mode register is set, the user may also insert hardware wait states after state T_2 and before state T_3 by driving a LOW on the $\overline{CS}/\overline{WAIT}$ signal.

The operation of Flowthru memory transaction are performed identical to the Flowthru I/O transactions except for $\overline{DS}$ width. (See Timing Diagrams.)

Flyby Transactions

Flyby transfers and transfer-and-search operations are performed in a single step, providing a transfer rate significantly faster than that available from Flowthrus. In Flyby, operations can only be performed between memory and peripherals or between peripherals and peripherals. Memory-to-memory operations can not be performed in Flyby mode; these must be done using Flowthru.

Flyby Memory-peripheral operations can only be used with peripherals having a special Flyby signal input. This peripheral input is connected to the channel's $\overline{DACK}$ output. For memory-peripheral Flyby, the address of the source memory location, must be programmed in the Current ARA register. The Current ARB register must be programmed with the destination memory location for peripheral-memory Flyby. Flyby peripheral-to-peripheral operations can only be performed if one of the peripherals has a Flyby signal input. This peripheral input must be connected to the channel's $\overline{DACK}$ output. If both peripherals have a Flyby input, only one should be connected to $\overline{DACK}$; the other peripheral's Flyby input should be held high during the Flyby operation. The address of the peripheral not connected to the channel's $\overline{DACK}$ output should be programmed in the Current ARB register. Note that Flip bit is set (CM4 = 1) for I/O to memory write transactions and cleared (CM4 = 0) for memory read to I/O transaction.

A Flyby operation is performed using three states: T_1, T_2, and T_3. During T_1 the channel pulses AS and outputs the address information. See Timing Diagram 6. The R/$\overline{W}$ line is HIGH if the Current ARA specifies the source and Low if the Current ARB specifies the destination.

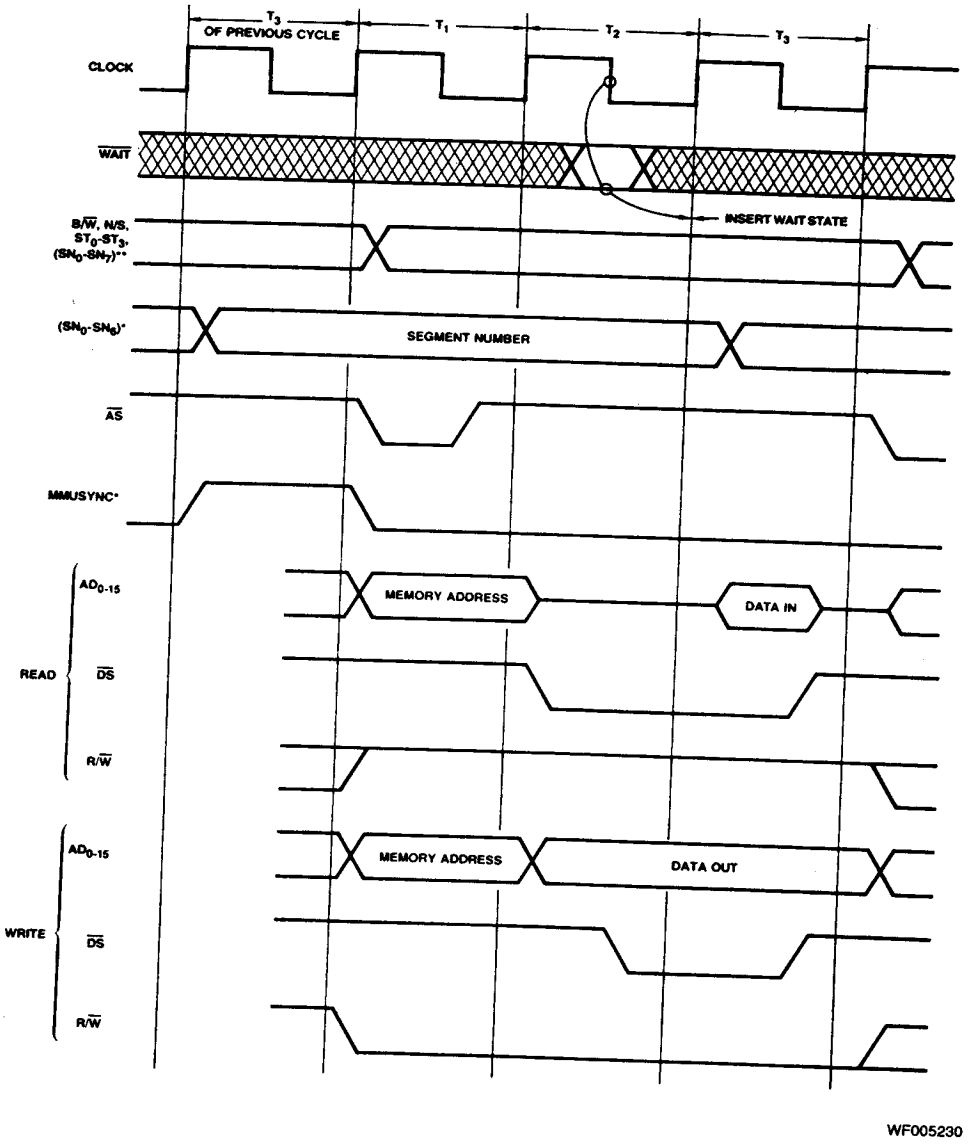
The channel's status lines ($ST_0 - ST_3$) and the N/S line are coded as specified by the Current ARA or ARB Tag field. The B/$\overline{W}$ line indicates the operand size programmed in the Channel Mode registers Operations field. During state T_2 the channel drives both $\overline{DS}$ and $\overline{DACK}$ active. The "Flyby Peripheral" connected to $\overline{DACK}$ inverts R/$\overline{W}$ to determine whether it is being read from or written to.

The $\overline{DACK}$ input serves two purposes: To select the peripheral for the Read/Write, and to provide timing information on when to drive data onto or input data from the $AD_0 - AD_{15}$ bus. Note that because the "Flyby Peripheral" never gets explicitly addressed by $AD_0 - AD_{15}$, it must know which internal register to load from or drive onto the $AD_0 - AD_{15}$ bus. On state T_3, the $\overline{DS}$ and $\overline{DACK}$ lines are driven inactive to conclude the transfer. In Transfer-and-Search mode, data is loaded into the DTC's Temporary register on the LOW-to-HIGH $\overline{DS}$ transition in order to perform the search function.

To provide adequate data setup time the rising edge of $\overline{DS}$ or $\overline{DACK}$ should be the edge used to perform the write to the transfer destination. To extend the active time of $\overline{DS}$ and $\overline{DACK}$, wait states can be inserted between T_2 and T_3. Software wait states can be inserted by programming the appropriate code in the Tag field of the Current ARA or ARB registers. Hardware wait states can be inserted by pulling $\overline{CS}/\overline{WAIT}$ LOW if the Wait Line Enable bit in the Master Mode register is set. The $\overline{CS}/\overline{WAIT}$ line is sampled in the middle of the T_2 and TWA states.

Termination

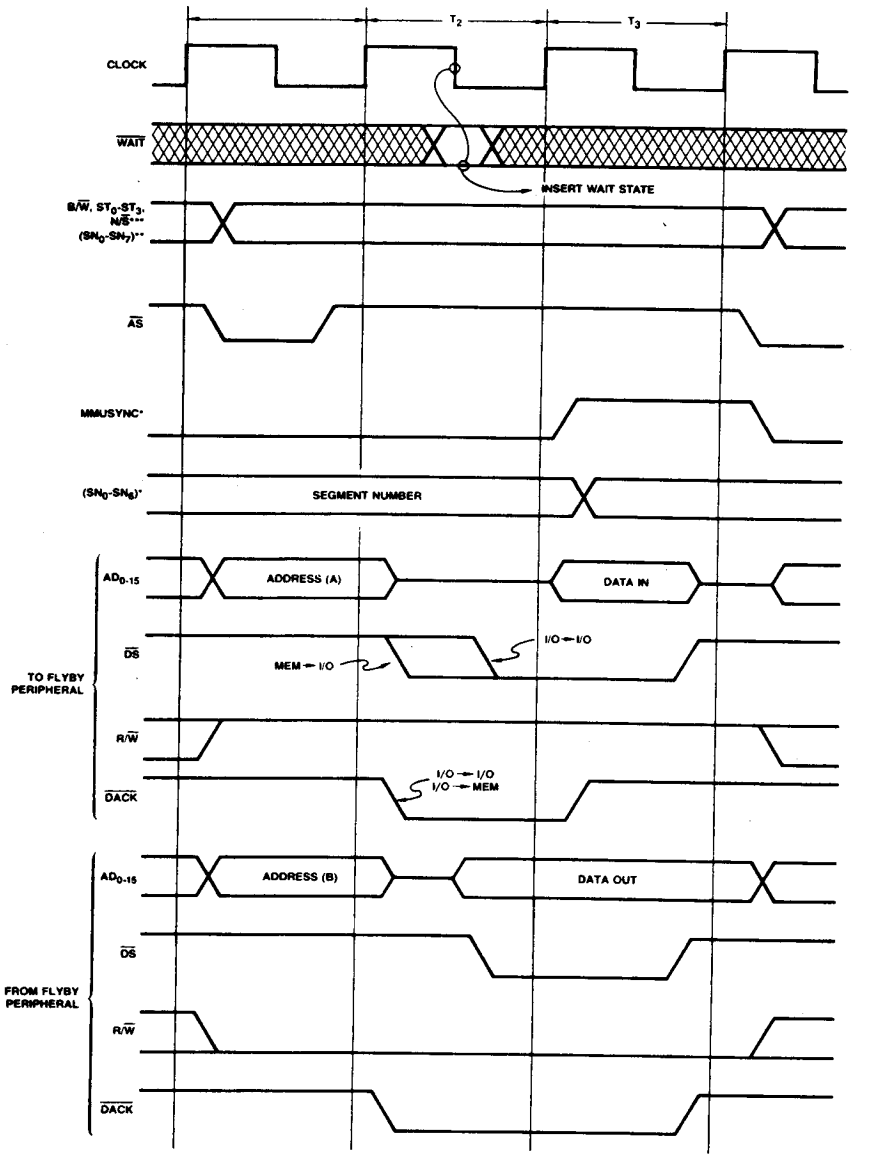
There are three ways a Transfer-and-Search or Search operation can end and two ways a Transfer operation can end. When a channel's Current Operation Count goes to 0, the DMA operation being performed will end. This is called a TC or Terminal Count termination. A DMA operation can also be stopped by driving the $\overline{EOP}$ pin LOW with external logic. This is called an EOP termination. Search and Transfer-and-Search operations have a third method of terminating called Match Condition or MC termination. An MC termination occurs when the data being Transferred-and-Searched or Searched meets the match condition programmed in Channel Mode register bits $CM_{17} - CM_{16}$. These bits allow the user to stop when a match occurs between the unmasked Pattern register bits and the data read from the source, or when a no-match occurs. Both byte and word matches are supported. MC terminations do not apply to Transfer operations since the pattern matching logic is disabled in Transfer mode.



*For logical addressing only
**For physical addressing only

Timing Diagram 5. Flowthru Memory Transactions

Refer to page 7-1 for Essential Information on Military Devices



WF005390

*For logical addressing only
**For physical addressing only
***N/S will be LOW for I/O to I/O Transactions
(A) Address is current ARA
(B) Address is current ARB

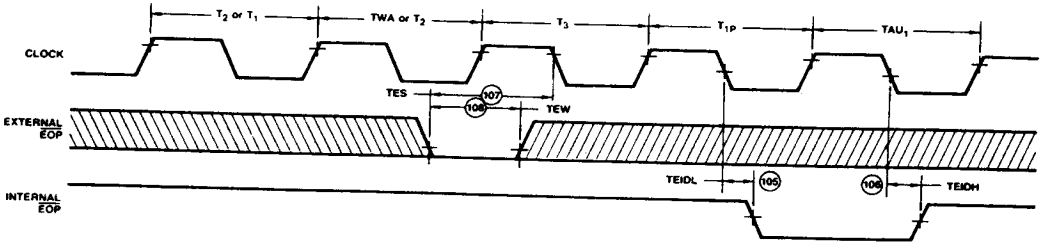
Timing Diagram 6. Flyby Transaction

Refer to page 7-1 for Essential Information on Military Devices

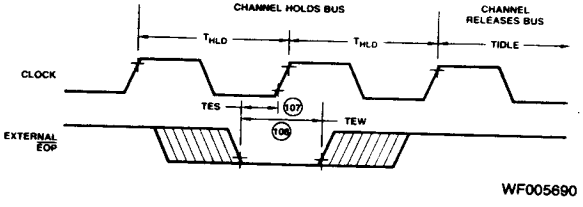
End-of-Process

The End-of-Process ($\overline{EOP}$) interface pin is a bidirectional signal. Whenever a TC, MC or $\overline{EOP}$ termination occurs, the DTC will drive the $\overline{EOP}$ pin LOW. During DMA operations, the $\overline{EOP}$ pin is sampled by the DTC to determine if $\overline{EOP}$ is being driven LOW by external logic. Timing Diagram 7 shows when internal $\overline{EOP}$s are generated marking termination of all Transfers. These figures also show the point during the DMA iteration when the $\overline{EOP}$ pin is sampled. The generation of internal $\overline{EOP}$s and sampling of external $\overline{EOP}$s for Transfer-and-Searches follows the same timing used for Transfers. Since there is a single $\overline{EOP}$ pin for both channels, $\overline{EOP}$ should only be driven LOW by a channel while that channel is being serviced. This can be accomplished by selecting a level $\overline{DACK}$ output (CMR 18 = 0) and gating each channel's $\overline{EOP}$ request with $\overline{DACK}$, as shown in Figure 8.

Some users may connect the $\overline{EOP}$ pin to the MMU's suppress ($\overline{SUP}$) output to detect the illegal memory accesses. To allow this abort feature for all memory accesses, the DTC samples $\overline{EOP}$ during chaining, as shown in Timing Diagram 7 (A). If $\overline{EOP}$ is LOW, the Chain Aborted bit in the active channel's Status register is set, the channel relinquishes bus control and the channel is inhibited from performing a new DMA operation until either a new Chain Address Segment-Tag word or offset word is loaded.



a) $\overline{EOP}$ Sampling and Generation During DMA Operations



b) Sampling of $\overline{EOP}$ During Bus Hold

Notes: 1. The diagram lists state names for both I/O and memory accesses. Sampling of $\overline{EOP}$ will occur on the falling edge of state T_3.
2. State T_{1P} is a pseudo-T_1 state, without active $\overline{AS}$, generated following termination of any DMA operation.
3. State TAU_1 is an auto-initialization state, generated following the TC, MC or EOP termination.

Timing Diagram 7. $\overline{EOP}$ Timing

Refer to page 7-1 for Essential Information on Military Devices

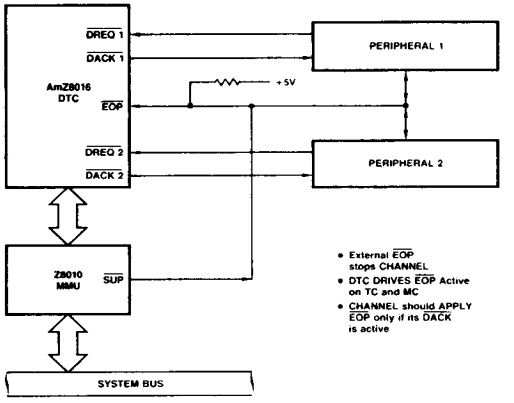


Figure 8. $\overline{EOP}$ Connection

The old Chain Address Offset and Segment may be examined to determine the error-causing address. If an $\overline{EOP}$ is detected while the channel is trying to reload the Chain Address register, the new Chain Address Offset and Segment are discarded and the old address+2 is preserved to allow inspection of the erroneous address.

Programming Completion Options

When a channel ends a DMA operation, the reason for ending is stored in the Completion Status Field of the channel's Status register. See Figure 16. This information is retained until the next DMA operation ends at which time the Status register is updated to reflect the reason(s) for the latest termination. Note that it is conceivable that more than one bit in the Completion Field could be set. As an extreme example, if a channel decremented it's Current Operation Count to zero, causing a TC termination; input data from the source generated a match causing an MC termination; and there was a LOW on $\overline{EOP}$ resulting in an EOP termination, all three of the channel's Status register completion bits would be set.

When a DMA operation ends, the channel can:

(a) Interrupt the host CPU;
(b) Perform Base-to-Current reloading;
(c) Chain reload the next DMA operation;
(d) Perform any combination of the above; or
(e) None of the above.

The user selects the action to be performed by the channel in the Completion option field of the Channel Mode register. For each type of termination (TC, MC or EOP) the user can choose which action or actions are to be taken. If no actions are selected for the type of termination that occured, the NAC bit in the Status register will be set.

More than one action can occur when a DMA operation ends. This may arise because more than one action was programmed for the applicable termination. They occur in the following order: interrupt, base-to-current reloading, and then chaining.

Interrupts

In order to allow the DTC to start executing a new DMA operation after issuing an interrupt, but before an interrupt acknowledge is received, a two-deep interrupt queue is implemented on each channel. The following discussion will describe the standard Z8000 interrupt structure and then elaborate on the additional interrupt queuing capability of the DTC.

A complete interrupt cycle consists of an interrupt request followed by an interrupt acknowledge transaction. The request, which consists of $\overline{INT}$ being pulled LOW by a peripheral, notifies the CPU that an interrupt is pending. The interrupt acknowledge transaction, which is initiated by the CPU as a result of the request, performs two functions: it selects the peripheral whose interrupt is to be acknowledged, and it obtains a vector that identifies the selected device and cause of interrupt.

A peripheral can have one or more sources of interrupt. Each interrupt source has three bits that control how it generates interrupts. These bits are a Channel Interrupt Enable bit (CIE), an Interrupt Pending bit (IP) and an Interrupt Under Service bit (IUS). On the DTC, each channel is an interrupt source. The three interrupt control bits are located in bits $CM_{15} - CM_{13}$ of each channel's Status register.

Each channel has its own vector register for identifying the source of the interrupt during an interrupt acknowledge transaction. There are two bits in the Master Mode register used for controlling interrupt behavior for the whole device. These are a Disable Lower Chain bit (DLC), and a No Vector bit (NV).

Peripherals are connected together via an interrupt daisy chain formed with their IEI and IEO pins (See Figure 9). The interrupt sources within a device are similarly connected into this chain. The overall effect is a daisy chain connecting all the interrupt sources. The daisy chain has two functions: during an interrupt-acknowledged transaction it determines which interrupt source is being acknowledged; at all other times it determines which interrupt sources can initiate an interrupt request.

Figure 10 is a state diagram for interrupt processing for an interrupt source. An interrupt source with an interrupt pending (IP = 1) makes an interrupt request (by pulling $\overline{INT}$ LOW) if, all of the following conditions are met: It is enabled (CIE = 1), is does not have an interrupt under service (IUS = 0), no higher priority interrupt is being serviced (IEI is HIGH), and no interrupt-acknowledge transaction is in progress. IEO is not pulled down by the interrupt source at this time; IEO continues to follow IEI until an interrupt-acknowledge transaction occurs.

Refer to page 7-1 for Essential Information on Military Devices

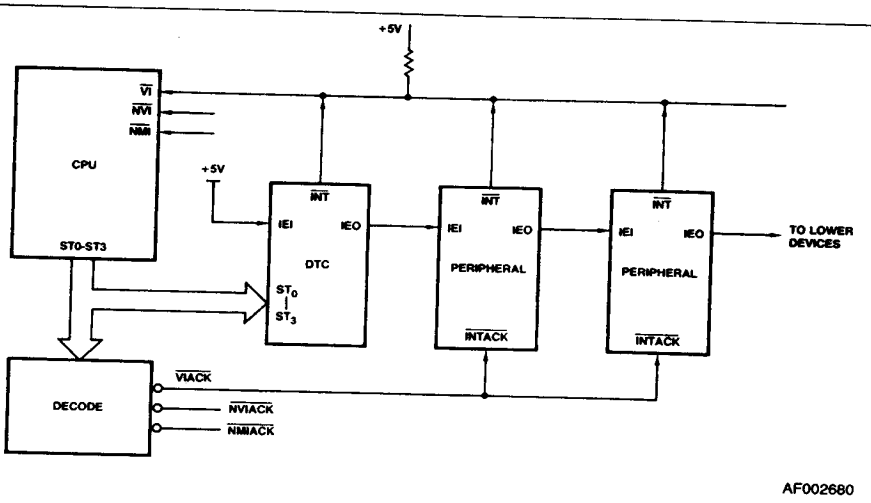


Figure 9. Interrupt Daisy Chain

Most Z8000 peripherals have an $\overline{\text{INTACK}}$ pin to signal when an interrupt acknowledge cycle is being performed. The DTC uses a different approach of monitoring status lines $ST_0 - ST_3$ to detect interrupt acknowledge cycles. It is important that the Master Mode register bits MM_6 and MM_7 be programmed to select the IEI, IEO daisy chain that the DTC is located on. Some time after $\overline{\text{INT}}$ has been pulled LOW, the CPU initiates an interrupt-acknowledge transaction. Between the rising edge of $\overline{\text{AS}}$ and the falling edge of $\overline{\text{DS}}$, the IEI/IEO daisy chain settles.

Once a channel issues an interrupt, it is desirable to allow the channel to proceed with the next DMA operation before the interrupt is acknowledged. This could lead to problems if the DTC channel attempted to chain reload the Vector register contents. In such a situation, it may not be clear whether the old or new vector would be returned during the acknowledge. This dilemma is resolved in the DTC by providing each channel with an Interrupt Save Register. When the channel sets IP as part of the procedure followed to issue an interrupt, the contents of the Vector register and some of the Status register bits are saved in an Interrupt Save register. See Figure 17. When an Interrupt Acknowledge cycle is performed, the contents of the Interrupt Save register are driven onto the bus. Although the use of an Interrupt Save register allows the channel to proceed with a new task, problems can still potentially arise if a second interrupt is to be issued by the channel before the first interrupt is acknowledged. To avoid conflicts between the first and second interrupt, each channel has a Second Interrupt Pending (SIP) bit in its Status register. When a second interrupt is to be issued before the first interrupt is acknowledged, the SIP bit is set and the channel relinquishes the bus until an acknowledge occurs. For compatibility with polled interrupt schemes, the Interrupt save register can be read by the host CPU without wait states. As an aid to debugging a system's interrupt logic, whenever IP is set, the Interrupt Save register is loaded from the Vector and Status register.

Note that the SIP bit is transferred to the IP bit when IP is cleared by the host CPU. Whenever IEI is HIGH, CIE is set and IUS is cleared, $\overline{\text{INT}}$ will go LOW as soon as IP is set. IP can be cleared as soon as the first interrupt is acknowledged. The acknowledge will, as always, automatically set IUS.

The DTC stops driving $\overline{\text{INT}}$ low as soon as IUS is set. IUS must be cleared by the CPU before $\overline{\text{INT}}$ can be driven low for the second time.

Base-to-Current Reloading

When a channel finishes a DMA operation, the user may select to perform a Base-to-Current Reload. (Base-to-Current reloading is also referred to as Auto-reloading is this document.) In this type of reload, the Current Address Registers A and B are loaded with the data in the Base Address Registers A and B respectively, and the Current Operation Count register is loaded with the data in the Base Operation Count. The Base-to-Current reload operation facilitates repetitive DMA operations without the multiple memory accesses required by chaining. Although the channel must have bus control to perform Base-to-Current reloading, the complete reloading operation occurs in four clock cycles. Note that if the channel had to relinquish the bus because two unacknowledged interrupts were queued, it will have to regain bus control to perform any Base-to-Current reloading (or chaining, for that matter). In this case it acquires the BUS once an interrupt acknowledge is received, even if it immediately afterward will relinquish the bus because no hardware or software request is present.

Chaining

If the channel is programmed to chain at the end of a DMA operation, it will use the Chain Address register to point to a Chain Control Table in memory. The first word in the table is a Reload word, specifying the register(s) to be loaded. Following the Reload word are the data values to be transferred into register(s). Chaining is described in detail in the "Channel Initialization" section.

Because chaining occurs after Base-to-Current reloading, it is possible to reset the Current Address registers A and B and the Current Operation Count register to the values used for previous DMA operations and then chain reload one or two of these registers to some special value to be used perhaps for this DMA operation only. If the Base values are not reloaded during chaining, the channel can revert back to the Base values at a later cycle.

Refer to page 7-1 for Essential Information on Military Devices

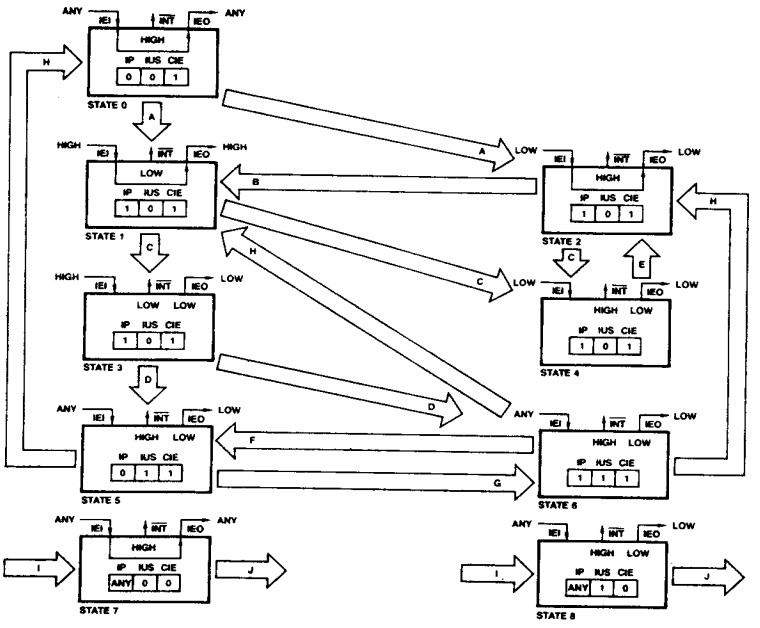


Figure 10. State Diagram for Interrupt Processing

Transition Legend

A The peripheral detects an interrupt condition and sets Interrupt Pending.

B All higher priority peripherals finish interrupt service, thus allowing IEI to go HIGH.

C An interrupt acknowledge transaction starts and the IEI/IEO daisy chain settles.

D The interrupt acknowlege transaction terminates with the peripheral selected. Interrupt Under Service (IUS) is set to 1, and Interrupt Pending (IP) may not be reset.

E The interrupt acknowlege transaction terminates with a higher priority device having been selected.

F The Interrupt Pending bit in the peripheral is reset by an I/O operation.

G A new interrupt condition is detected by the peripheral, causing IP to be set again.

H Interupt service is terminated for the peripheral by resetting IUS.

I CIE is reset to zero, causing interrupts to be disabled (Note 1).

J CIE is set to one, re-enabling interrupts (Note 2).

State Legend

0 No interrupts are pending or under service for this peripheral.

1 An interrupt is pending, and interrupt request has been made by pulling $\overline{INT}$ LOW.

2 An interrupt is pending, but no interrupt request has been made because a higher priority peripheral has an interrupt under service, and this has forced IEI LOW.

3 An interrupt acknowledge sequence is in process, and no higher priority peripheral has a pending interrupt.

4 An interrupt acknowledge sequence is in progress, but a higher priority peripheral has a pending interrupt, forcing IEI LOW.

5 The peripheral has an interrupt under service. Service may be temporarily suspended (indicated by IEI going LOW) if a higher priority device generates an interrupt.

6 This is the same as State 5 except that an interrupt is also pending in the peripheral.

7 Interrupts are disabled from this source because CIE = 0.

8 Interrupts are disabled from this source and lower priority sources because CIE = 0 and IUS = 1.

Notes: 1. Transition I to state 6 or 7 can occur from any state except 3 or 4 (which only occur during interrupt acknowledge).
2. Transition J from state 6 or 7 can be to any state except 3 or 4, depending on the value of IEI, IP, and IUS.

If an all zero Reload word is fetched during chaining, the chain operation will not reload any registers but in all other respects it will perform like any other chaining operation. Thus, the Chain Address will be incremented by 2 point to the next word in memory and at the end of the all Zero-Reload word chain operation the channel will be ready to perform a DMA operation. All zero Reload words are useful as "Stubs" to start or terminate linked lists of DMA operations traversed by chaining. On the other hand, care must be taken in their use since the channel may perform an erroneous operation if it is unintentionally started after the chaining operation.

Command Descriptions

Figure 11 shows a list of DTC commands. The commands are executed immediately after being written by the host CPU into the DTC's Command register. A description of each command follows.

Reset

This command causes the DTC to be set to the same state generated by a Hardware Reset. The Master Mode register is set to all zeros and the NAC and CA bits in each channel's Status register are set. The Chain Address should be programmed since its state may be indeterminate after a Reset. The lockout preventing channel activity is cleared by programming the Segment/Tag word or the Offset word and then issuing a "Start Chain" command.

Start Chain Channel 1/Channel 2

This command causes the selected channel to clear the No Auto-Reload or Chain (NAC) bit in the channel's Status register, and to start a chain reload operation of the channel's registers, as described in the "Channel Initialization" section. These effects will take place even if the Reload word fetched is all zeros. This command will only be honored if the Chain Abort (CA) bit and Second Interrupt Pending (SIP) bit in the Channel's Status register are clear. If either the CA or SIP bit is set, this command is disregarded.

Software Request Channel 1/Channel 2

This command sets the software request bit in the selected channel's Mode register. If the Second Interrupt Pending (SIP) bit and No Auto-Reload or Chain (NAC) bit in the channel's Status register are both clear, the channel will start executing the programmed DMA operation. If either the SIP or NAC bit is set, the channel will not start executing a DMA operation until both bits are cleared. The SIP bit will clear when the channel receives an interrupt acknowledge. One way to clear the NAC bit is to issue a Start Chain command to the channel. If the fetched Reload Word is all zeros, the channel's registers will remain unchanged and the software request bit, if set earlier by command, will cause the programmed DMA operation to start immediately. If during chaining new information is loaded into the Channel Mode register this new information will, of course, overwrite the software request bit.

Command	Opcode Bits 7654	Opcode Bits 3210	Example Code (HEX)
Reset	000X	XXXX	00
Start Chain Channel 1	101X	XXX0	A0
Start Chain Channel 2	101X	XXX1	A1
Set Software Request Channel 1	010X	XX10	42
Set Software Request Channel 2	010X	XX11	43
Clear Software Request Channel 1	010X	XX00	40
Clear Software Request Channel 2	010X	XX01	41
Set Hardware Mask Channel 1	100X	XX10	82
Set Hardware Mask Channel 2	100X	XX11	83
Clear Hardware Mask Channel 1	100X	XX00	80
Clear Hardware Mask Channel 2	100X	XX01	81
Set CIE, IUS, IP Channel 1	001E	SP10	32*
Set CIE, IUS, IP Channel 2	001E	SP11	33*
Clear CIE, IUS, IP Channel 1	001E	SP00	30*
Clear CIE, IUS, IP Channel 2	001E	SP01	31*
Set Flip Bit Channel 1	011X	XX10	62
Set Flip Bit Channel 2	011X	XX11	63
Clear Flip Bit Channel 1	011X	XX00	60
Clear Flip Bit Channel 2	011X	XX01	61

*Notes:
1. E = Set to 1 to perform set/clear on CIE. Clear to 0 for no effect on CIE.
2. S = Set to 1 to perform set/clear on IUS. Clear to 0 for no effect on IUS.
3. P = Set to 1 to perform set/clear on IP. Clear to 0 for no effect on IP.
4. X = "don't care" bit. This bit is not decoded and may be 0 or 1.

Figure 11. DTC Command Summary

Set/Clear Hardware Mask 1/Mask 2

This command sets or clears the Hardware Mask bit in the selected channel's Mode register. This command always takes effect. The Hardware Mask bit inhibits recognition of an active signal on the channel's $\overline{DREQ}$ input; this bit does not affect recognition of a software request. If the channel is in single transfer mode, it performs DMA operations upon receipt of a transition on $\overline{DREQ}$ rather than in response to a $\overline{DREQ}$ level; the transition occurring while the Hardware Mask bit is set ($CM_{19} = 1$) will be stored and serviced when the Hardware Mask is cleared (assuming the channel has not chained in the interim). The DTC will request the system bus 1-1/2 to 2 clocks after the receipt of any $\overline{DREQ}$, after which a minimum of one DMA iteration is unavoidable. $\overline{DREQ}$ transitions are only stored for the current DMA operation. If the channel performs a chain operation of single transfer mode, any $\overline{DREQ}$ transition stored for later servicing is cleared. Timing Diagrams 1 and 2 show the minimum times after each of these events a new $\overline{DREQ}$ edge can be applied if it is to be serviced by the new DMA operation. Note in the diagrams the notation of First iteration and Last iteration. This means, for example, $\overline{DREQ}$ may be asserted during the second T_1 of a Flow-through transaction, but may never be asserted during T_1 of a Flyby transaction because Flyby is done in one iteration.

Refer to page 7-1 for Essential Information on Military Devices

Set/Clear CIE, IUS AND IP Channel 1/Channel 2

This command allows the user to either set or clear any combination of the CIE, IUS and IP bits in the selected channel's Status register. These bits control the operation of the channel's interrupt structure and are described in detail in the "Interrupts" section. Setting the IP bit causes the Interrupt Save register to be loaded with current Vector and Status. The IP and IUS bits can be simultaneously cleared to facilitate an efficient conclusion to the processing of an interrupt.

Set/Clear Flip Bit in Channel 1/Channel 2

The Flip Bit in the selected channel's Mode register can be cleared and set by this command. This allows the user to reverse the source and destination and thereby reverse the data transfer direction without reprogramming the channel. This command will be most useful when repetitive DMA operations are being performed by the channel, using Base-to-Current reloading for channel reinitialization and using this command to control the direction of transfer. Chaining new information into the Channel Mode register will, of course, overwrite the Flip bit.

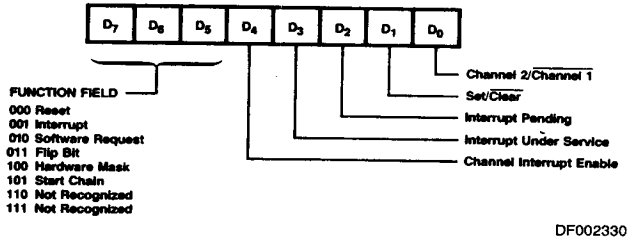


Figure 11A. Command Register

PROGRAMMING INFORMATION

Register Description

The AmZ8016 block diagram illustrates the internal registers. Figure 12 lists each register along with its size and read/write access restrictions. Registers which can be read by the CPU are either fast (F) or slow (S) readable. Fast registers can be read by an I/O operation without additional wait states. Reading slow registers requires multiple wait states (see Timing Parameters-88 for requirement). It is the responsibility of the user to supply the necessary external logic if slow readable registers are to be read. Registers can be written to by the host CPU (W) and/or can be loaded by the DMA channel itself during chaining (C). All reads or writes must be word accesses since the DTC ignores the B/$\overline{W}$ line in slave mode.

The DTC registers can be categorized into chip-level registers, which control the overall operation and configuration of the DTC, and channel-level registers which are duplicated for each channel. The four chip-level registers are the Master Mode register, the Command register, the Chain Control register and the Temporary register. The Master Mode register selects the way the DTC chip interfaces to the system. The Command register is written to by the host CPU to initiate certain operations within the DTC chip, such as resetting the unit. The Chain Control register is used by a channel while it is reloading its channel-level registers from memory. The Temporary register is used to hold data for Flowthru Transfer-and-Searches.

Master Mode Register

The 8-bit Master Mode register, shown in Figure 13, controls the chip-level interfaces. It can be read from and written to by the host CPU without wait states through pins $AD_0 - AD_7$ but it is not loadable by chaining. On a reset, the Master Mode register is cleared to all zeroes. The function of each of the Master Mode bits is described in the following paragraphs.

The Chip Enable bit CE = 1 enables the DTC to request the bus. When enabled, the DTC can perform DMA Operations and reload registers. It can always issue interrupts and respond to interrupt acknowledges. When the Chip Enable bit is cleared, the DTC is inhibited from requesting control of the system bus and, therefore, inhibited from performing chaining or DMA operations. The Chip Enable bit (MMO) should not be used as the gating item in starting or stopping the DTC. Channels should be enabled or disabled by using the Set/Clear software request or Hardware Mask Commands. The $\overline{BAO}$ signal follows $\overline{BAI}$ while the Chip Enable bit is cleared.

The Logical/$\overline{Physical}$ Address space bit selects the address space in which the DTC resides. Figure 14 shows the different configuration options. If the DTC outputs addresses which are translated by an MMU, Logical space must be selected. If the addresses output by the DTC pass directly onto the system backplane and if the host CPU is an Z8001 using an MMU, the Logical/$\overline{Physical}$ bit must be set to Physical ($MM_1 = 0$). If Z8001 addresses are not translated by an MMU, the DTC must be set to Logical ($MM_1 = 1$). In an Z8002 based system, the user may use either the Physical or Logical address space setting.

Refer to page 7-1 for Essential Information on Military Devices

Name	Size	Number	Access Type	Port Address CH-1/CH-2
Master Mode Register	8 bits	1	FW	38
Chain-Control Register	10 bits	1	C	
Temporary Register	16 bits	1	D	
Command Register	8 bits	1	W	2E/2C*
Current Address Register — A:				
Segment/Tag field	15 bits	2	CFW	1A/18
Offset field	16 bits	2	CFW	0A/08
Current Address Register — B:				
Segment/Tag field	15 bits	2	CFW	12/10
Offset field	16 bits	2	CFW	02/00
Base Address Register — A:				
Segment/Tag field	15 bits	2	CFW	1E/1C
Offset field	16 bits	2	CFW	0E/0C
Base Address Register — B:				
Segment/Tag field	15 bits	2	CFW	16/14
Offset field	16 bits	2	CFW	06/04
Current Operation Count	16 bits	2	CFW	32/30
Base Operation Count	16 bits	2	CFW	36/34
Pattern Register	16 bits	2	CSW	4A/48
Mask Register	16 bits	2	CSW	4E/4C
Status Register	16 bits	2	F	2E/2C
Interrupt Save Register	16 bits	2	F	2A/28
Interrupt Vector Register	8 bits	2	CSW	5A/58
Channel Mode Register — High	5 bits	2	CS	56/54
Channel Mode Register — Low	16 bits	2	CSW	52/50
Chain Address Register:				
Segment/Tag field	10 bits	2	CFW	26/24
Offset field	16 bits	2	CFW	22/20

Access Codes: C = Chain Loadable
D = Accessible by DTC channel
F = Fast Readable
S = Slow Readable
W = Writable

Note: Upper Register Address is determined by user's Chip Select Decode Logic. Only Lower Register Address is shown here.
*The port addresses of Command register can be used alternately for both channels except when issuing a "set/clear IP" command.

Figure 12. DTC Internal Register

When set to Logical address space, the segment and offset portions of the Current ARA and ARB registers are viewed as separate portions of the address. Incrementing and decrementing the register affects only the offset portion of the address; no carry or borrow signal is generated into the segment. Only the lower 7 bits of the segment field are used; the setting of the most significant segment bit is disregarded. The 16-bit offset portion of the address appears on pins $AD_0 - AD_{15}$ when $\overline{AS}$ is LOW and the 7-bit segment number appears on pins $SN_0 - SN_6$ for the duration of the transaction. The SN_7/MMUSync signal outputs a HIGH pulse prior to each memory transaction, and is never three-stated.

When the Logical/$\overline{\text{Physical}}$ Space bit is set to Physical, the segment and offset portions of the Current ARA and ARB registers are treated as a single linear address. All eight segment bits in the register are used. When an address is incremented or decremented, the carry/borrow signal propagates across the full 24-bit address updating both the segment and offset portions of the address. Both I/O and memory addresses in Physical space are generated by driving the offset portion of the Current Address register onto the $AD_0 - AD_{15}$ bus and driving the segment portion of the Current Address register onto the $SN_0 - SN_7$ bus. Timing Diagrams 4, 5 and 6 show how the Logical/$\overline{\text{Physical}}$ Space bit affects the DTC timing. The $AD_0 - AD_{15}$ timing is not affected. The $SN_0 - SN_7$ lines are shifted from T_3 to T_1 so that they are valid during the entire transaction. In the Logical Address mode, the MMU insures the addresses are valid during the entire transaction.

The CPU Interleave bit enables interleaving between the CPU and the DTC.

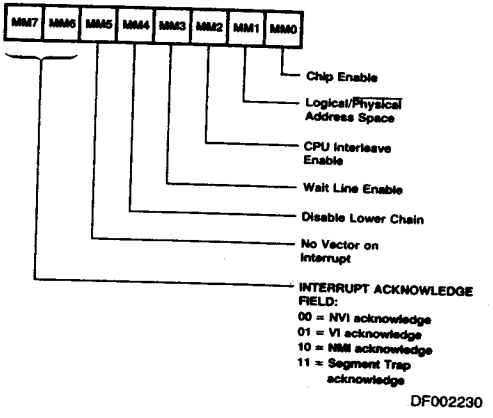
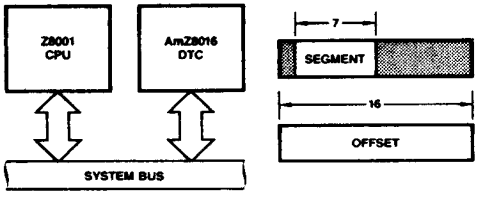
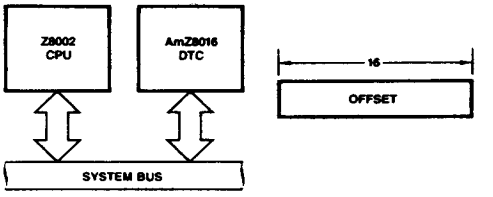


Figure 13. Master Mode Register

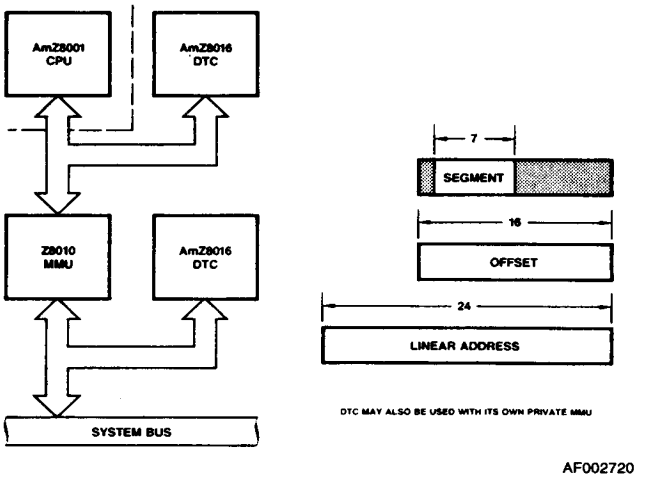
Refer to page 7-1 for Essential Information on Military Devices



a) DTC with Z8001 (Segmented) CPU



b) DTC with Z8002 (Nonsegmented) CPU



c) DTC with Z8001 (Segmented) CPU and AmZ8010 MMU

Figure 14. DTC Configuration Options

The Wait Line Enable bit is used to enable sampling of the $\overline{CS}/\overline{WAIT}$ line during Memory and I/O transactions. Because the DTC provides the ability to insert software programmable wait states, many users may disable sampling of the $\overline{CS}/\overline{WAIT}$ pin to simplify the logic driving this pin. The Wait Line Enable bit provides this flexibility. See the "Wait States" section of this document for details on wait state insertion.

The Disable Lower Chain bit is used to inhibit all lower priority devices on the interrupt daisy chain. When this bit is cleared, the DTC generates LOW and HIGH signals on the IEO output in response to IEI. When the Disable Lower Chain bit is set, IEO is forced LOW, which disables all lower priority interrupts.

The "No Vector on Interrupt" bit selects whether the DTC channel or a peripheral returns a vector during interrupt acknowledge cycles. When this bit is cleared, a channel receiving an interrupt acknowledge will drive the contents of its Interrupt Save register onto the $AD_0 - AD_{15}$ data bus while $\overline{DS}$ is LOW. If this bit is set, interrupts are serviced in an identical manner but the $AD_0 - AD_{15}$ data bus remains in a high impedance state throughout the acknowledge cycle.

The Vectored/Non-Vectored/Non-Maskable/Segment Trap Field of two bits selects which type of interrupt acknowledge cycle the DTC is to respond to. The DTC decodes from $ST_0 - ST_3$ that an interrupt acknowledge cycle is underway. The setting of this 2-bit field must correspond to the IEI/IEO daisy chain on which the DTC is located to prevent unpredictable results. For example a DTC programmed for vectored interrupts should not be placed on the non-vectored priority chain.

Chain Control Register

When a channel starts a chaining operation, it fetches a Reload word from the memory location pointed to by the Chain Address register. This word is then stored in the Chain Control register. The Chain Control register cannot be written to or read from by the CPU. Once a channel starts a chain operation, the channel will not relinquish bus control until all registers specified in the Reload word are reloaded unless an $\overline{EOP}$ signal is issued to the chip. Issuing an $\overline{EOP}$ to a channel during chaining will prevent the chain operation from resuming and the contents of the Reload Word register can be discarded.

Temporary Register

The Temporary register is used to stage data during Flowthru transfers and to hold data being compared during a Search or a Transfer-and-Search. The Temporary register cannot be written to, or read from by the CPU. In byte-word funneling, data may be loaded into or from the Temporary register on a byte-by-byte basis, with bytes sometimes moving between the low byte of the data bus and the high byte of the Temporary register or vice-versa. See the "Transfer" section for details.

Command Register

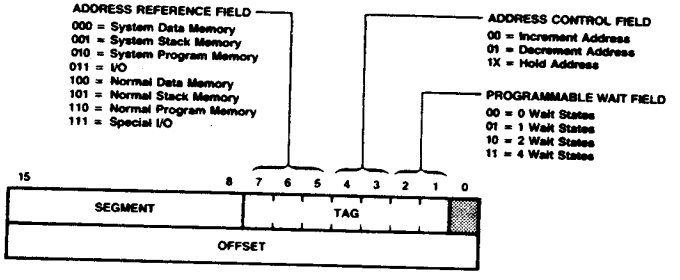
The DTC Command register is an 8-bit write-only register written to by the host CPU to execute commands. The Command register is loaded from the data on $AD_7 - AD_0$; the data on $AD_{15} - AD_8$ is disregarded. A complete discussion of the commands is given in the "Command Descriptions" section.

Current and Base Address Registers A and B

The Current Address registers A and B (Current ARA and ARB) are used to point to the source and destination addresses for DMA operations. The Base ARA and ARB register contents are transferred into the Current ARA and ARB registers at the end of a DMA operation if the user enables Base-to-Current reloading in the Completion Field of the Channel Mode register. This facilitates DMA operations without reloading of the Current registers. The ARA and ARB registers can be loaded during chaining, can be written to by the host CPU without wait states and can be read by the CPU.

Each of the Base and Current ARA and ARB registers consists of two words organized as a 7-bit Tag Field and an 8-bit segment in one word and a 16-bit offset in the other. See Figure 6. The Segment and Offset contain the actual address driven onto the bus. The Tag Field selects whether the address is to be incremented, decremented or left unchanged, and the status codes associated with the address. The Tag field also allows the user to insert 0, 1, 2 or 4 wait states into memory or I/O accesses addressed by the offset and segment fields.

The Address Reference Select Field in the Tag field selects whether the address pertains to memory space or I/O space. Note that the N/$\overline{S}$ output pin is always LOW (indicating System) for I/O space but may be either HIGH (indicating Normal) or LOW (indicating System) for memory space, as selected in the Address Space field. At the end of each iteration of a DMA Operation, the user may select to leave the address unchanged or to increment it or to decrement it. I/O addresses, if changed, are always incremented/decremented by 2. Memory addresses are changed by 1 if the address points to a byte operand (as programmed in the Channel Mode registers Operation field) and by 2 if the address points to a word operand. Note that if an I/O or memory address is used to point to a word operand, the address must be even to avoid unpredictable results. An address used to point to a byte operand may be even or odd. Since memory byte operand addresses will increment/decrement by 1, they will toggle between even and odd values. Since I/O byte operand addresses will increment/decrement by 2, once programmed to an even or odd value, they will remain even or odd, allowing consecutive I/O operations to access the same half of the data bus. High bus is for even address and low bus for odd.



DF002260

Figure 15. Address Register A and B

Current and Base Operation Count Registers

Both the Current and Base Operation Count registers may be loaded during chaining, and may be written to, and read from by the host CPU.

The 16-bit Current Operation Count register is used to specify the number of words or bytes to be transferred, searched or transferred-and-searched. For word-to-word operations and byte-word funneling, the Current Operation Count register must be programmed with the number of words to be transferred or searched.

Each time a datum is transferred or searched, the Operation Count register is decremented by 1. Once all of the data is transferred or searched, the transfer or search operation will stop, the Current Operation Count register will contain all zeros, and the TC bit in Status Register will be '1'. If the transfer or search stops before the Current Operation Count register reaches 0, the contents of the register will indicate the number of bytes or words remaining to be transferred or searched. This allows a channel which had been stopped prematurely, to be restarted where it left off without requiring reloading of the Current Operation Count register.

If the DTC is configured for Physical Address Space operation (Master Mode register bit $MM_1 = 0$), the maximum number of words that can be transferred or searched is 64K words. This is specified by setting a word count of 0000. If the DTC is configured for Logical Address Space operation (Master Mode register bit $MM_1 = 1$), the maximum number of words specified to be transferred or searched with either an incrementing or decrementing source or destination address is 32K (8000 hex). This is because in Logical Address Space, offset addresses incremented past FFFF (hex) or decremented below 0000 do not increment/decrement the segment number. Thus, after transferring or searching more than 32K words, the address wraps around within the segment over the same data previously transferred or searched.

For the byte-to-byte operations, the Current Operation Count register should specify the number of bytes to be transferred or searched. The maximum number of bytes which can be specified is 64K bytes; by setting the Current Operation Count register to 0000.

Refer to page 7-1 for Essential Information on Military Devices

Pattern and Mask Registers

The 16-bit Pattern and Mask registers are used in Search and Transfer-and-Search operations. Both the Pattern and Mask registers may be loaded by chaining, may be written to by the host CPU, and may be read from by the host CPU, provided wait states are inserted, since these registers are slow readable. The Pattern register contains the pattern that the read data is compared to. Setting a Mask register bit to '1' specifies that the bit always matches. See the "Search" and "Transfer-and-Search" sections for further details.

Status Register

The two 16-bit Status registers, depicted in Figure 16, are read-only registers which can be read by the CPU without wait states. Each of these registers reports on the status of its associated channel.

The Interrupt Status Field in the Status register contains the Channel Interrupt Enable (CIE), Interrupt Pending (IP) and Interrupt Under Service (IUS) bits. These bits are described in detail in the "Interrupt" section of this document.

The DTC status field reports on the current channel state to the CPU. The "channel initialized and waiting for request" status is not explicitly stated – it is reflected by Status register bits ST_{12} through ST_9 being all zero. The "Waiting for Bus" (WFB) status will cause bit ST_{10} to be set and indicates that the channel wants bus control to perform a DMA operation. The channel may or may not actually be asserting $\overline{BUSRQ}$ LOW, depending on the programming of the Master Mode Chip Enable bit and the state of $\overline{BUSRQ}$ and $\overline{BAI}$ when the channel decided it wanted the bus. See the "Bus Request/ Grant" section for details. If a channel completes a DMA operation and neither Base-to-Current reloading nor auto-chaining were enabled, the No Auto-Reload or Chaining (NAC) bit will be set. The NAC bit will be reset when the channel receives a "Start Chain Command." If two interrupts are queued, the Second Interrupt Pending bit (SIP) will be set and the channel will be inhibited from further activity until an interrupt acknowledge occurs. See the "Interrupt" section for details. Finally, if the channel is issued an $\overline{EOP}$ during chaining, the Chain Abort (CA) and the NAC will be set. These bits are also set when a "reset" is issued to the DTC. The CA bit holds the NAC bit in the set state. The CA is cleared when a new Chain Address Segment and Tag word or offset word is loaded into the channel.

The Hardware Interface Field provides a Hardware Request (HRQ) bit which provides a means of monitoring the channels $\overline{DREQ}$ input pin. When the $\overline{DREQ}$ pin is LOW, the HRQ bit will be '1' and vice-versa. The Hardware Mask (HM) bit, when set, prevents the DTC from responding to a LOW on $\overline{DREQ}$. Note, however, that the Hardware Request bit always reports the true (unmasked) status of $\overline{DREQ}$ regardless of the setting of the HM bit. The HM bit can be cleared by software command.

The Completion Field stores data at the end of each DMA operation. This data indicates why the DMA operation ended. When the next DMA operation ends, new data is loaded into these bits overwriting, and thereby erasing the old setting. Three bits indicate whether the DMA operation ended as a result of a TC, MC or EOP termination. The TC bit will be '1' if the Operation Count reaching zero ended the DMA operation. The MC bit will be '1' if an MC termination occurred regardless of whether Stop-on-Match or Stop-on-no-Match was selected. The EOP bit is set only when an $\overline{EOP}$ ends a DMA transfer; it is not set for EOPs issued during chaining. Note that two or even all three of MC, TC and $\overline{EOP}$ may be set if multiple reasons existed for ending the DMA operation. The MCH and MCL bits report on the match state of the upper and lower comparator bytes respectively. These bits are set when the associated comparator byte has a match and are reset otherwise, regardless of whether Stop-on-Match or Stop-on-no-Match is programmed. Regardless of the DMA operation performed, these bits will reflect the comparator status at the end of the DMA operation. These two bits are provided to help determine which byte matched or didn't match when using 8-bit matches with word searches and transfer-and-searches. The two reserved bits return zeroes during reads.

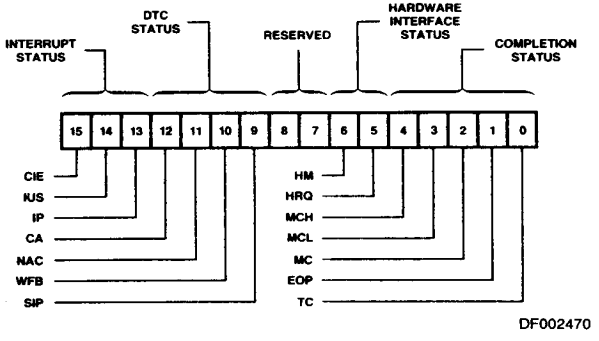


Figure 16. Status Register

Refer to page 7-1 for Essential Information on Military Devices

Interrupt Vector and Interrupt Save Registers

Each channel has an Interrupt Vector register and an Interrupt Save register. The Interrupt Vector is 8-bits wide and is written to and read from on $AD_0 - AD_7$. The Interrupt Save register may be read from by the CPU without wait states. The Interrupt Vector register contains the vector or identifier to be output during an Interrupt Acknowledge cycle. When an interrupt occurs (IP = 1), either because a DMA operation terminated or because the $\overline{EOP}$ pin was driven LOW during chaining, the contents of the Interrupt Vector register and part of the Channel Status register are stored in the 16-bit Interrupt Save register (See Figure 17). Because the vector and status are stored, a new vector can be loaded during chaining, and a new DMA operation can be performed before an interrupt acknowledge cycle occurs. If another interrupt occurs on the channel before the first is acknowledged, further channel activity is suspended.

As soon as the first clear IP command is issued, the status and vector for the second interrupt is loaded into the Interrupt Save register and channel operation resumes. The DTC can retain only two interrupts for each channel; a third operation cannot be initiated until the first interrupt has been cleared. See the "Interrupt" section for further details.

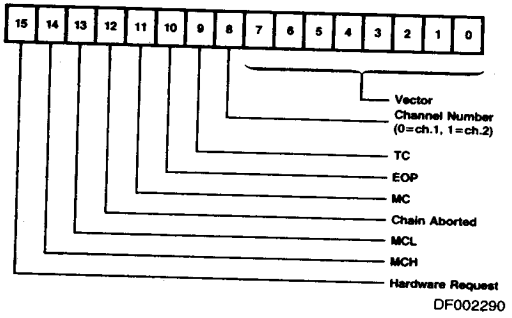


Figure 17. Interrupt Save Register

Channel Mode Register

Associated with each channel is a Channel Mode register. There are 21 bits defined in each Channel Mode register; the other 11 bits are unused. See Figure 18. The Channel Mode registers may be loaded during chaining and may be read from and written to by the host CPU. CPU reads from the Channel Mode register are slow reads and require insertion of multiple wait states. The Channel Mode register selects what type of DMA operation the channel is to perform, how the operation is to be executed, and what action, if any, is to be taken when the channel finishes.

The Data Operation Field and the Transfer Type field select the type of operation the channel is to perform. They also select the operand size of bytes or words see Figure 19 for code-definition. The different types of operations are described in detail in the "DMA Operations" section. The Flip bit is used to select whether the Current ARA register points to the source and the Current ARB register points to the destination or vice-versa.

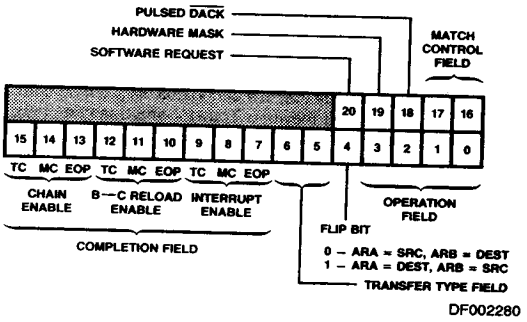


Figure 18. Channel Mode Register

DATA OPERATION FIELD			
	Operand Size		
Code/Operation	ARA	ARB	Transaction Type
Transfer			
0001	Byte	Byte	Flowthru
100X	Byte	Word	Flowthru
0000	Word	Word	Flowthru
0011	Byte	Byte	Flyby
0010	Word	Word	Flyby
Transfer-and-Search			
0101	Byte	Byte	Flowthru
110X	Byte	Word	Flowthru
0100	Word	Word	Flowthru
0111	Byte	Byte	Flyby
0110	Word	Word	Flyby
Search			
1111	Byte	Byte	N/A
1110	Word	Word	N/A
101X	Illegal		

TRANSFER FIELD AND MATCH CONTROL FIELD		
Code	Transfer Type	Match Control
00	Single Transfer	Stop on No Match
01	Demand (Bus Hold)	Stop on No Match
10	Demand (Bus Release)	Stop on Word Match
11	Demand Interleave	Stop on Byte Match

Figure 19. Channel Mode Coding

The Completion Field is used to program the action taken by the channel at the end of a DMA operation. This field is discussed in the "Completion Options" section. The 2-bit Match Control field selects whether matches use an 8-bit or 16-bit pattern and whether the channel is to stop-on-match or stop-on-no-match. See Figure 19 and the "Search" section for details. The Software Request bit and Hardware Mask bit can be set and cleared by software command. Only the lower 16 bits can be loaded in parallel with a CPU instruction. These bits are described in detail in the "Initiating DMA Operations" section.

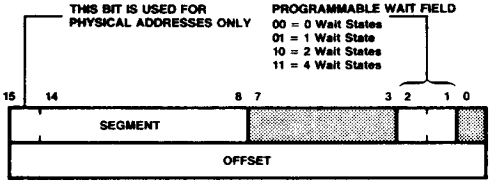
Refer to page 7-1 for Essential Information on Military Devices

The $\overline{DACK}$ Control bit is used to specify when the $\overline{DACK}$ pin is driven active. When this bit is cleared, the channel's $\overline{DACK}$ pin will be active whenever the channel is performing a DMA Operation, regardless of the type of transaction. Note that the pin will not be active while the channel is chaining. If this bit is set, the $\overline{DACK}$ pin will be inactive during chaining, during both Flowthru Transfers and Flowthru Transfer-and-Searches and during Searches, but $\overline{DACK}$ will be pulsed active during Flyby Transfers and Flyby Transfers-and-Searches at the time necessary to strobe data into or out of the Flyby peripheral. Flyby operations are discussed in detail in the "Flyby Transactions" section.

Chain Address Register

Each channel has a Chain Address register which points to the chain control table in memory containing data to be loaded into the channel's registers. The Chain Address register, as shown in Figure 20 is two words long. The first word consists of the Segment and Tag fields. The second word contains the 16-bit offset portion of the memory address. The highest bit in the segment field is not used when the DTC is configured for Logical Address space (MM1 = 1). The Tag field contains 2 bits used to designate the number of wait states to be inserted during accesses to the Chain Control Table.

The Chain Address register may be loaded during chaining and may be read from and written to by the host CPU without wait states. If an $\overline{EOP}$ is issued to the DTC during chaining, the Chain Address register holds the old address. This is true even if the access failure occurred while new Chain Address data was being loaded, since the old data is restored unless both words of the new data are successfully read. Note, however, that $\overline{EOP}$s that occur when chaining and while loading a new Chain Address cause the new data to be lost.



DF002301

Figure 20. Chain Address Register

APPLICATIONS INFORMATION

Figure 21 shows the configuration of the AmZ8016 DMA Transfer Controller (DTC) and a microprocessor system. The DTC issues a $\overline{BUSRQ}$ active low signal to the CPU to request bus control. When the CPU replies with a Bus Acknowledge $\overline{BUSAK}$ signal through the $\overline{BAI}$ $\overline{BAO}$ daisy chain to the DTC which issued the $\overline{BUSRQ}$, the DTC takes control of the Address-Data bus and the Control bus. In addition to hardware reset the 8016 can be given a software 'reset' command (i.e., loading all zero to Command Register). Two DMA channels are provided per each DTC device. The logic blocks A and B are shown in Figure 22 and 23.

Figure 22 shows the bus request logic used for bidirectionally buffering $\overline{BUSRQ}$. See 'Bus Request/Grant' section for detail. Figure 23 shows the $\overline{CS}/\overline{WAIT}$ logic for the multiplexed $\overline{CS}/\overline{WAIT}$ pin of DTC. See 'Wait States' for detailed description.

AmZ8016 DTC to AmZ8010 MMU Interface

The AmZ8010 Memory Management Unit (MMU) is a high performance LSI product that adds sophisticated address translation and memory protection capabilities to AmZ8001 CPU systems. The MMU contains table of access attributes that are individually programmable for each segment. Attributes provided are read-only, system-mode-only, DMA only, execute only, and CPU only. If the MMU detects a memory access that violates one of the attributes of a segment, the MMU interrupts the CPU or DMA to inhibit an erroneous memory write.

Figure 24 shows the AmZ8016 DTC to AmZ8010 MMU Interface Configuration. The MMUSYNC output of DTC ORed with the $\overline{BUSAK}$ signal of CPU is connected to the DMASYNC input of MMU. The MMUSYNC pin of DTC is multiplexed with SN7. If the Master Mode register bit MM1 is set (i.e., in Logical Addressing mode), this pin outputs MMUSYNC active high pulse prior to each DMA cycle when the DTC is in control of the system bus or outputs low level when the DTC is not in control of the system bus. If the MM1 is clear (i.e., in Physical Addressing mode), this pin outputs the SN7 signal when DTC is a bus master or is driven high impedance off when DTC is not on control of the system bus.

The $\overline{SUP}$ output of MMU is connected to the $\overline{EOP}$ pin of the DTC for terminating the DMA operation whenever a violation has been detected.

Refer to page 7-1 for Essential Information on Military Devices

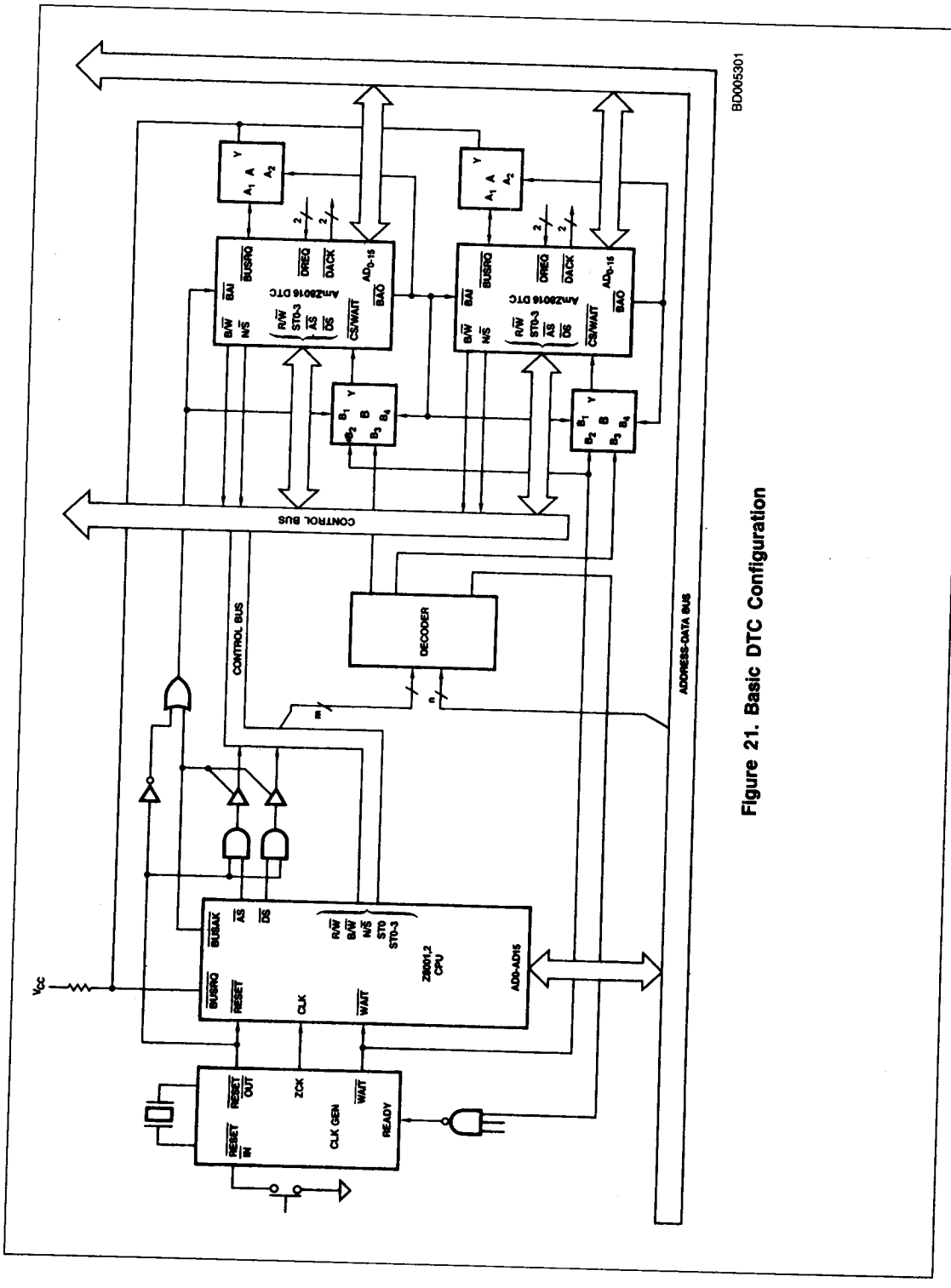
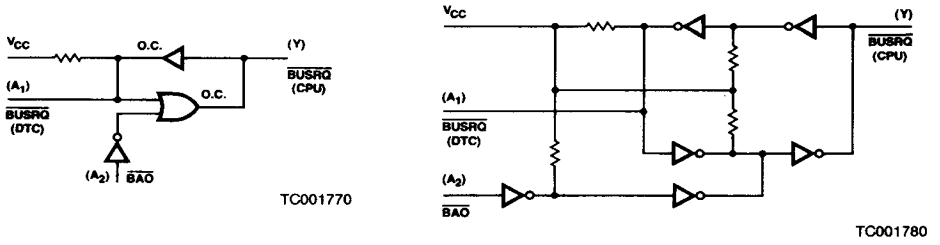


Figure 21. Basic DTC Configuration



a) Using Three Different Gates

b) Using One TTL Package (7405)

Figure 22. Logic Used to Bidirectionally Buffer $\overline{BUSRQ}$

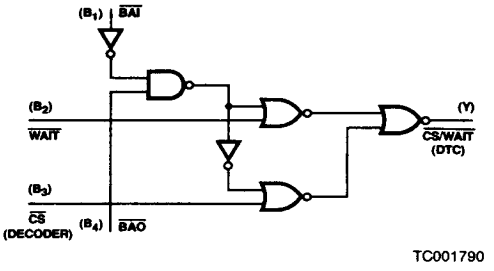
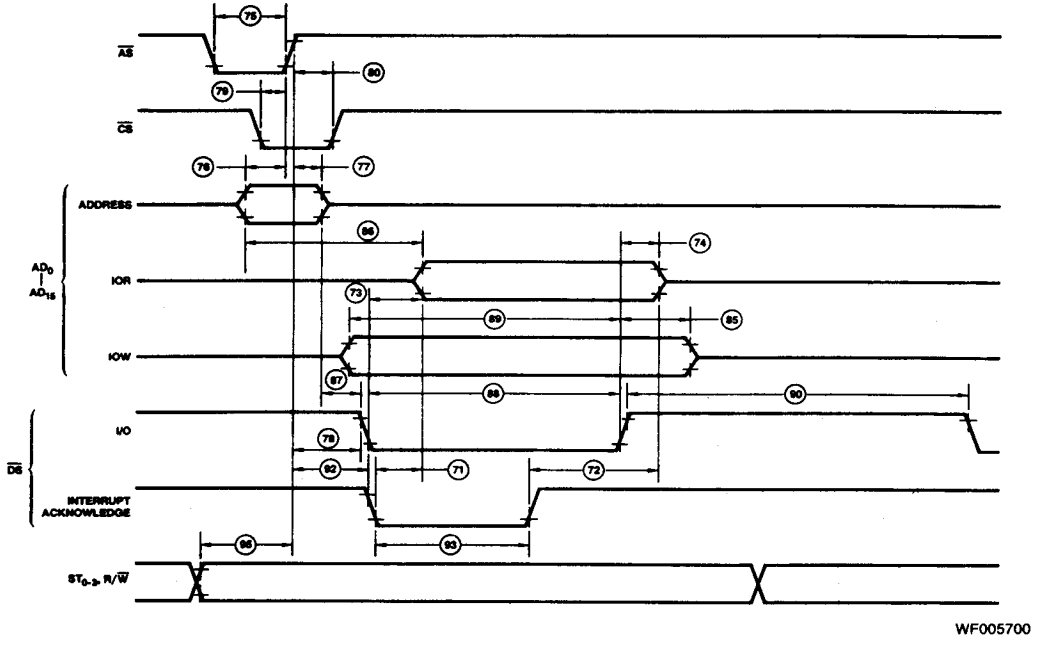


Figure 23. Logic for Multiplexed $\overline{CS}/\overline{WAIT}$ pin



Timing Diagram 8. AC Timing when DTC is Bus Slave

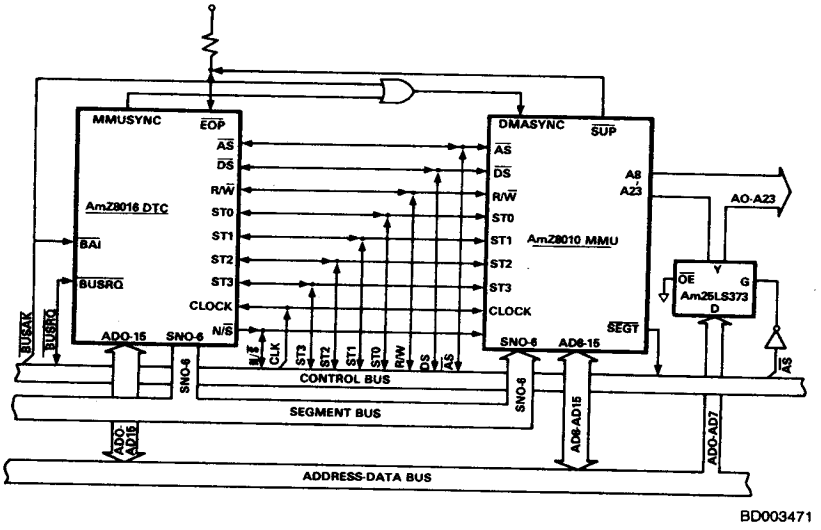
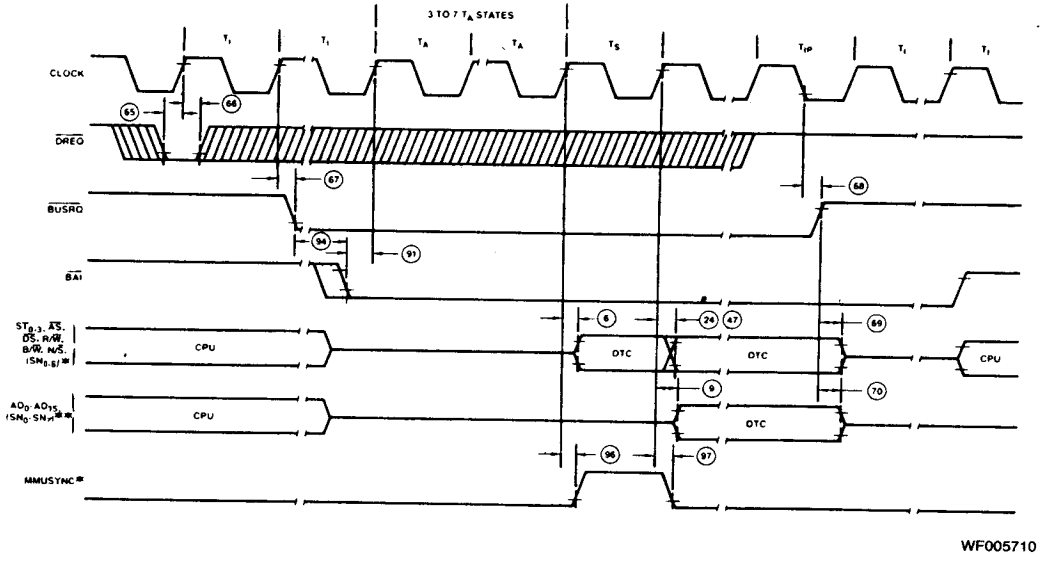


Figure 24. AmZ8016 DTC to AmZ8010 MMU Interface Configuration

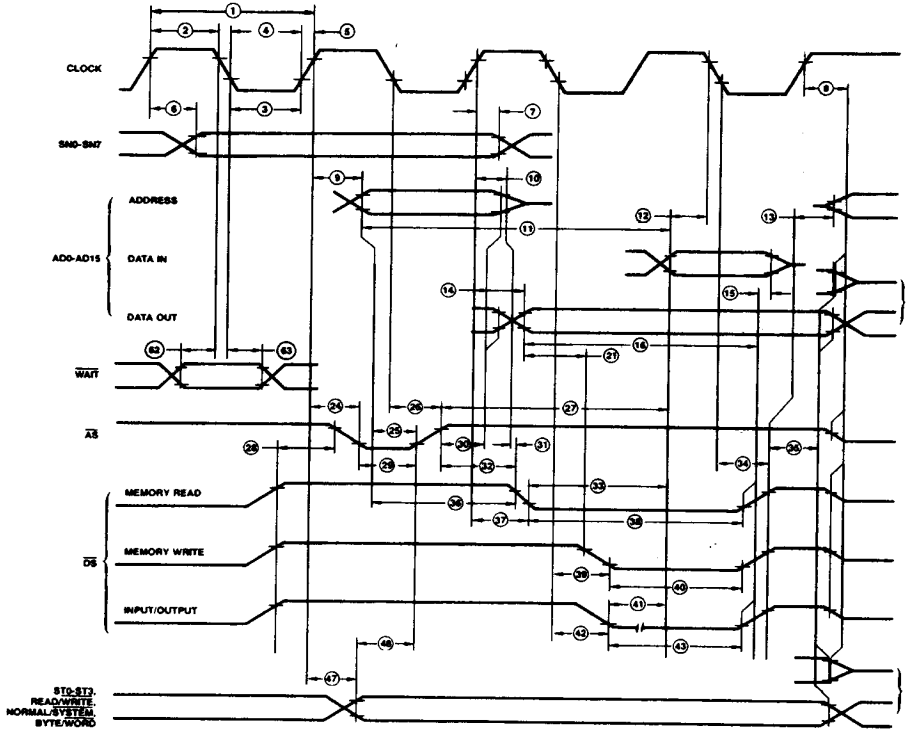


*For logical addressing only

**For physical addressing only

Timing Diagram 9. Bus Exchange Timing

Refer to page 7-1 for Essential Information on Military Devices



WF005720

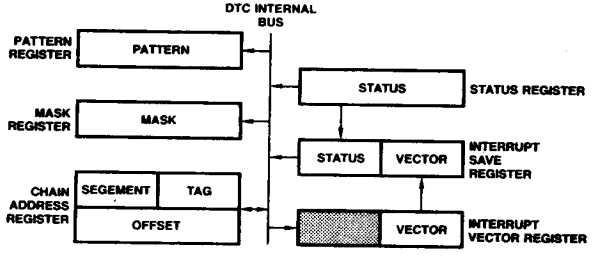
This composite timing diagram does not show actual timing sequences. Refer to this diagram only for the detailed timing relationships of individual edges. Use the preceding illustrations as an explanation of the various timing sequences.

Timing Diagram 10. AC Timing when DTC is Bus Master

APPENDIX A

AmZ8016 REGISTER SUMMARY

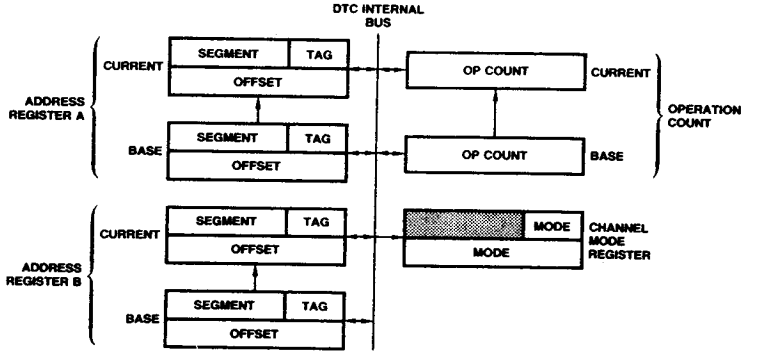
Special-Purpose Channel Registers



DF002340

Refer to page 7-1 for Essential Information on Military Devices

General-Purpose Channel Registers

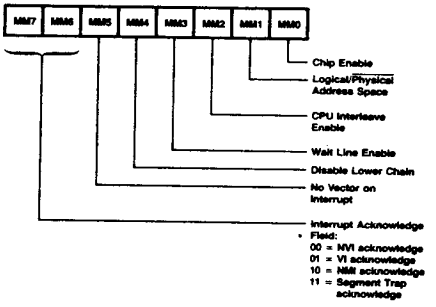


DF002350

Master Mode Register

	A7							A0
Address	X	0	1	1	1	0	0	X

Fast Readable
Writable

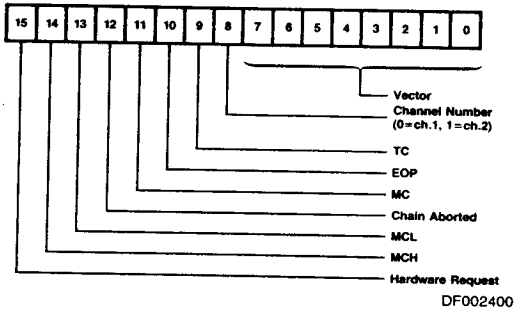


DF002390

Interrupt Save Register

	A7							A0	
Address	X	0	1	0	1	0	1	X	CH1
	X	0	1	0	1	0	0	X	CH2

Fast Readable



DF002400

Refer to page 7-1 for Essential Information on Military Devices

Miscellaneous Registers

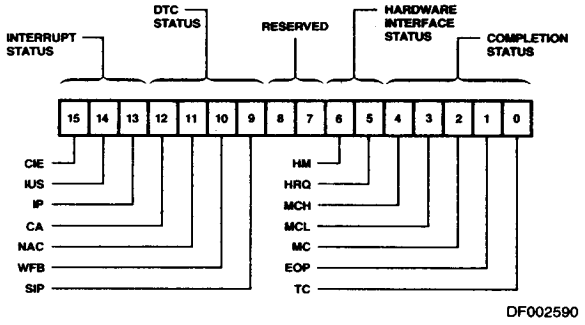
	A7						A0	
Address	X	0	1	1	0	0	1	X	Current Operation Count CH1
	X	0	1	1	0	0	0	X	Current Operation Count CH2
	X	0	1	1	0	1	1	X	Base Operation Count CH1
	X	0	1	1	0	1	0	X	Base Operation Count CH2
	X	1	0	0	1	0	1	X	Pattern CH1
	X	1	0	0	1	0	0	X	Pattern CH2
	X	1	0	0	1	1	1	X	Mask CH1
	X	1	0	0	1	1	0	X	Mask CH2

Chain Loadable
Writable
Pattern and Mask — Slow Readable
Operation Count — Fast Readable

Status Register

	A7							A0	
Address	X	0	1	0	1	1	1	X	CH1
	X	0	1	0	1	1	0	X	CH2

Fast Readable

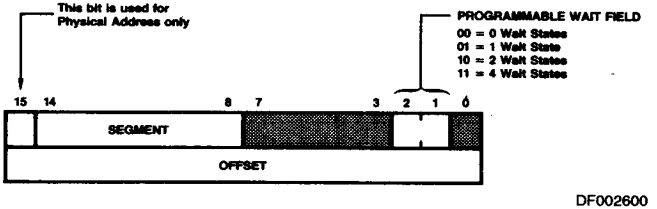


DF002590

Chain Address Register

	A7							A0	
Address	X	0	1	0	0	1	1	X	Segment/Tag CH1
	X	0	1	0	0	1	0	X	Segment/Tag CH2
	X	0	1	0	0	0	1	X	Offset CH1
	X	0	1	0	0	0	0	X	Offset CH2

Fast Readable/Writable
Chain Loadable



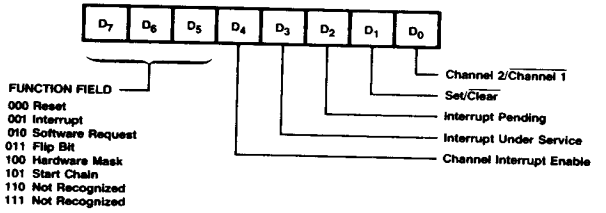
DF002600

Refer to page 7-1 for Essential Information on Military Devices

Command Register

	A7						A0
Address	X	0	1	0	1	1	X	X

Writable Only

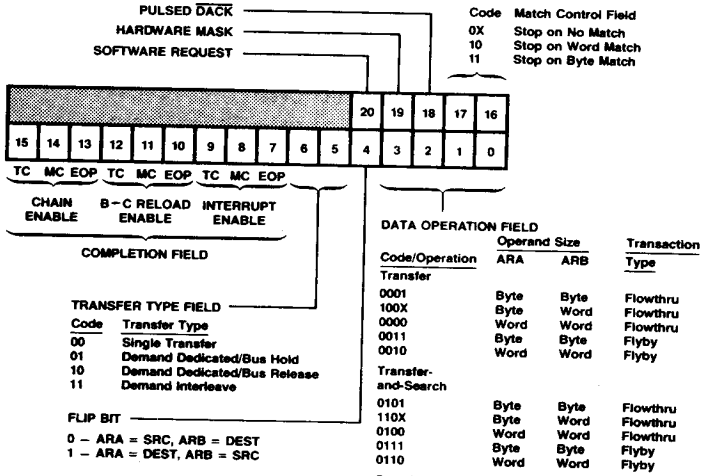


DF002610

Channel Mode Register

A7							A0	
X	1	0	1	0	1	1	X	High CH1
X	1	0	1	0	1	0	X	High CH2
X	1	0	1	0	0	1	X	Low CH1
X	1	0	1	0	0	0	X	Low CH2

Chain Loadable
Writable (Lower 16 bits)
Slow Readable



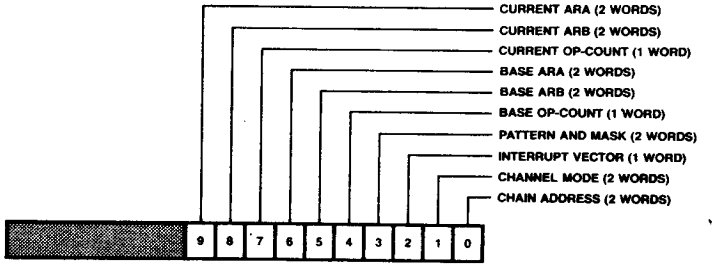
Code/Operation	Operand Size ARA	Operand Size ARB	Transaction Type
Transfer			
0001	Byte	Byte	Flowthru
100X	Byte	Word	Flowthru
0000	Word	Word	Flowthru
0011	Byte	Byte	Flyby
0010	Word	Word	Flyby
Transfer-and-Search			
0101	Byte	Byte	Flowthru
110X	Byte	Word	Flowthru
0100	Word	Word	Flowthru
0111	Byte	Byte	Flyby
0110	Word	Word	Flyby
Search			
1111	Byte	Byte	N/A
1110	Word	Word	N/A
101X	Illegal		

DF002620

Refer to page 7-1 for Essential Information on Military Devices

Chain Control Register

Chain Loadable Only

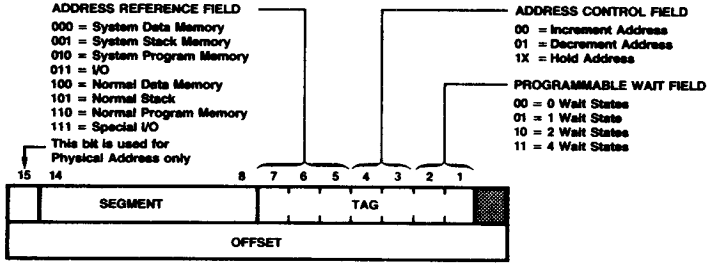


DF002630

Address Register

	A7							A0		
Address	X	0	0	1	1	0	1	X	Current ARA Segment/Tag	CH1
	X	0	0	1	1	0	0	X	Current ARA Segment/Tag	CH2
	X	0	0	0	1	0	1	X	Current ARA Offset	CH1
	X	0	0	0	1	0	0	X	Current ARA Offset	CH2
	X	0	0	1	0	0	1	X	Current ARB Segment/Tag	CH1
	X	0	0	1	0	0	0	X	Current ARB Segment/Tag	CH2
	X	0	0	0	0	0	1	X	Current ARB Offset	CH1
	X	0	0	0	0	0	0	X	Current ARB Offset	CH2
	X	0	0	1	1	1	1	X	Base ARA Segment/Tag	CH1
	X	0	0	1	1	1	0	X	Base ARA Segment/Tag	CH2
	X	0	0	0	1	1	1	X	Base ARA Offset	CH1
	X	0	0	0	1	1	0	X	Base ARA Offset	CH2
	X	0	0	1	0	1	1	X	Base ARB Segment/Tag	CH1
	X	0	0	1	0	1	0	X	Base ARB Segment/Tag	CH2
	X	0	0	0	0	1	1	X	Base ARB Offset	CH1
	X	0	0	0	0	1	0	X	Base ARB Offset	CH2

Chain Loadable
Fast Readable and Writable



DF002640

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
V_{CC} with Respect to V_{SS} ..... −0.5 to +7.0V
All Signal Voltages with
Respect to V_{SS} ..... −0.5 to +7.0V
Power Dissipation (Package Limitation) 2W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

	T_A	V_{CC}
Commercial	0°C to 70°C	5V ±5%

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

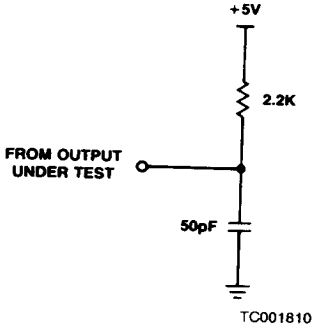
Parameters	Description	Test Conditions	Min	Max	Units
V_{CH}	Clock Input High Voltage	Driven by External Clock Generator	$V_{CC}-0.4$	$V_{CC}+0.3$	Volts
V_{CL}	Clock Input Low Voltage	Driven by External Clock Generator	−0.3	0.45	Volts
V_{IH}	Input High Voltage		2.4	$V_{CC}+0.3$	Volts
V_{IL}	Input Low Voltage		−0.3	0.8	Volts
V_{OH}	Output High Voltage	$I_{OH} = -250\mu A$	2.4		Volts
V_{OL}	Output Low Voltage	$I_{OL} = +2.0mA$		0.4	Volts
I_{IL}	Input Leakage	$V_{SS} \leq V_{IN} \leq V_{CC}$		±10	μA
I_{OL}	Output Leakage	$V_{SS} \leq V_{OUT} \leq V_{CC}$		±10	μA
I_{CC}	V_{CC} Supply Current	$T_A = 0°C$		350	mA
		$T_A = 75°C$		200	mA
C_{IN}	Input Capacitance	Unmeasured pins returned to ground. f = 1MHz over specified temperature range.		10	pF
C_{OUT}	Output Capacitance			15	pF
$C_{I/O}$	Bidirectional Capacitance			20	pF

Standard Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

$+4.75V \leq V_{CC} \leq +5.25V$
GND = 0V
$0°C \leq T_A \leq +70°C$

Open-Drain Test Load



Refer to page 7-1 for Essential Information on Military Devices

TIMING REFERENCES FOR AC TESTS

Input Waveform

WF005730

Output Waveform

WF005740

All AC parameters assume a load capacitance of 100pF max, except for parameter 6 TdC(SNv)(50pF max).

SWITCHING CHARACTERISTICS over operating range unless otherwise specified

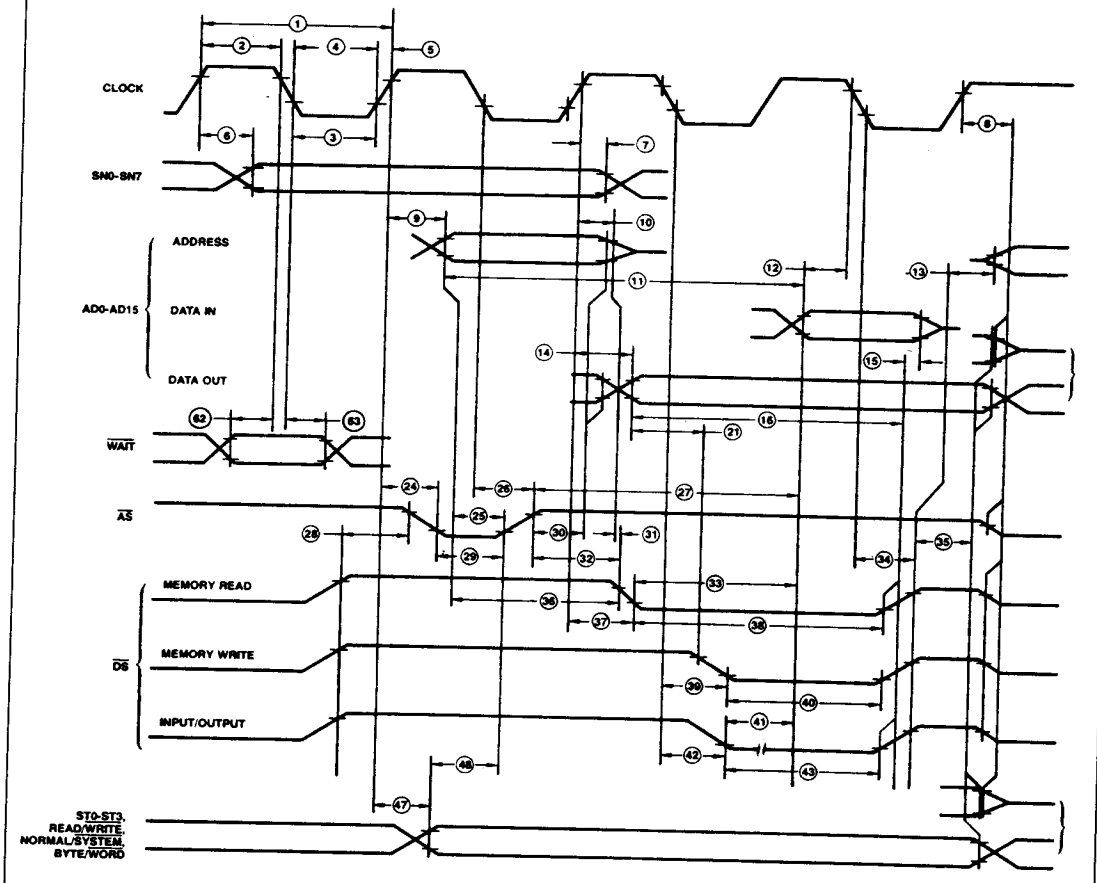
TIMING FOR DTC AS BUS MASTER

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
1	TcC	Clock Cycle Time	250	2000	165		ns
2	TwCh	Clock Width (High)	105		70		ns
3	TwCl	Clock Width (LOW)	105		70		ns
4	TfC	Clock Fall Time		20		10	ns
5	TrC	Clock Rise Time		20		15	ns
6	TdC(SNv)	Clock RE to Segment Number Valid (50pF Load) Delay***		110		90	ns
7	TdC(SNn)	Clock RE to Segment Number Valid Delay		90		60	ns
8	TdC(Bz)	Clock RE to Bus Float Delay		65		50	ns
9	TdC(A)	Clock RE to Address Valid Delay		100		90	ns
10	TdC(Az)	Clock RE to Address Float Delay		65		60	ns
11	TdA(Dl)	Address Valid to Data In Required Valid Delay	400		305		ns
12	TsDl(C)	Data In to Clock FE Set-up Time	20		15		ns
13	TdDS(A)	$\overline{DS}$ RE to Address Active Delay	80		45		ns
14	TdC(DO)	Clock RE to Data Out Valid Delay		100		90	ns
15	ThDl(DS)	$\overline{DS}$ RE to Data In Hold Time	0		0		ns
16	TdDO(DS)	Data Out Valid to $\overline{DS}$ RE Delay	230		200		ns
21	TdDO(SW)	Data Out Valid to $\overline{DS}$ FE (Write) Delay	55		35		ns
24	TdC(ASf)	Clock RE to $\overline{AS}$ FE Delay		70		60	ns
25	TdA(AS)	Address Valid to $\overline{AS}$ RE Delay	50		35		ns
26	TdC(ASr)	Clock FE to $\overline{AS}$ RE Delay		80		60	ns
27	TdAS(Dl)	$\overline{AS}$ RE to Data In Required Valid Delay	300			220	ns
28	TdDS(AS)	$\overline{DS}$ RE to $\overline{AS}$ FE Delay	75		35		ns
29	TwAS	$\overline{AS}$ Width (LOW)	80		60		ns
30	TdAS(A)	$\overline{AS}$ RE to Address Valid Delay	60		45		ns
31	TdAz(DSR)	Address Float $\overline{DS}$ (Read) FE Delay	0		0		ns
32	TdAS(DSR)	$\overline{AS}$ RE to $\overline{DS}$ FE (Read) Delay	75		40		ns
33	TdDSR(Dl)	$\overline{DS}$ (Read) FE to Data In Required Valid Delay	165		155		ns
34	TdC(DSr)	Clock FE to $\overline{DS}$ RE Delay		70		65	ns
35	TdDS(DO)	$\overline{DS}$ RE to Data Out (Write Only) and Status Valid (Read and Write) Delay	85		45		ns
36	TdA(DSR)	Address Valid to $\overline{DS}$ (Read) FE Delay	120		110		ns
37	TdC(DSR)	Clock RE to $\overline{DS}$ (Read) FE Delay		60		60	ns
38	TwDSR	$\overline{DS}$ (Read) Width (LOW)	275		185		ns
39	TdC(DSW)	Clock FE to $\overline{DS}$ (Write) FE Delay		60		60	ns
40	TwDSW	$\overline{DS}$ (Write) Width (LOW)	160		150		ns
41	TdDSl(Dl)	$\overline{DS}$ (Input) FE to Data In Required Valid Delay	325			210	ns
42	TdC(DSf)	Clock FE to $\overline{DS}$ (I/O) FE Delay		60		60	ns
43	TwDS	$\overline{DS}$ (I/O) Width (LOW)	150*		150		ns
47	TdC(S)	Clock RE to Status Valid Delay		110		80	ns
48	TdS(AS)	Status Valid to $\overline{AS}$ RE Delay	60		35		ns
62	TsWT(C)	$\overline{WAIT}$ to Clock FE Set-up Time	20		20		ns
63	ThWT(C)	$\overline{WAIT}$ to Clock FE Hold Time	30		30		ns
96	TdC(SNr)	Clock RE to SN7/MMUSync RE Delay**		110		110	ns
97	TdC(SNf)	Clock RE to SN7/MMUSync FE Delay**	20	110		110	ns

*Wait states should be inserted by programming a hardware when accessing slow peripherals.
**Logical Addressing only.
***130ns max with Logical Addressing.
Note: RE = rising edge FE = falling edge

Refer to page 7-1 for Essential Information on Military Devices

Timing Diagram 11. AC Timing when DTC is Bus Master



WF005780

This composite timing diagram does not show actual timing sequences. Refer to this diagram only for the detailed timing relationships of individual edges. Use the preceding illustrations as an explanation of the various timing sequences.

Refer to page 7-1 for Essential Information on Military Devices

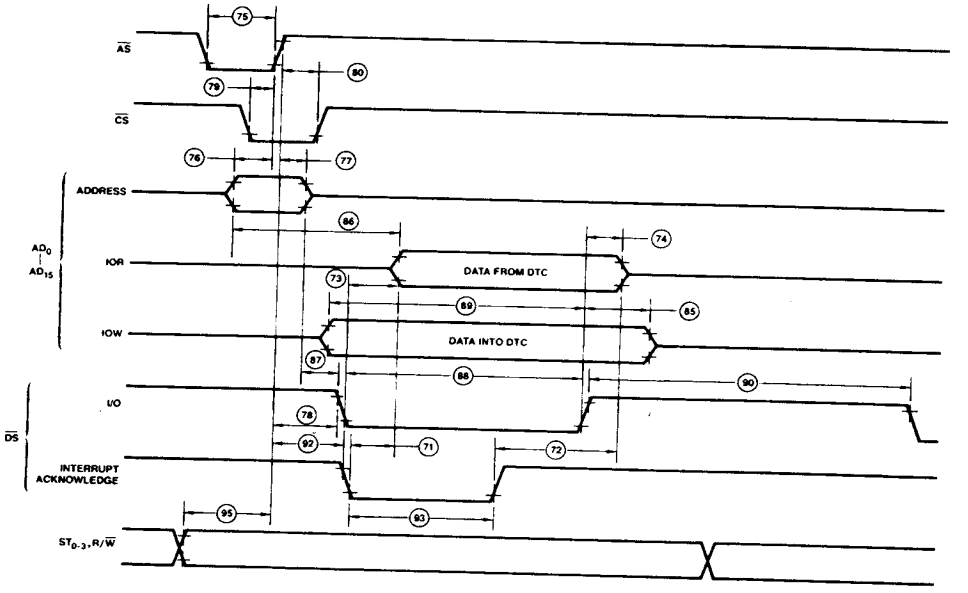
SWITCHING CHARACTERISTICS over operating range unless otherwise specified
TIMING FOR DTC AS BUS SLAVE AND CPU-DTC BUS EXCHANGE

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
64	TwDRQ	$\overline{DREQ}$ Pulse Width (Single Transfer Mode)	20		20		ns
65	TsDRQ(c)	$\overline{DREQ}$ Valid to Clock RE Set-up Time	60		50		ns
66	ThDRQ(C)	Clock RE to $\overline{DREQ}$ Valid Hold Time	20		20		ns
67	TdC(BRQf)	Clock RE to $\overline{BUSRQ}$ FE Delay		150		120	ns
68	TdC(BRQr)	Clock FE to $\overline{BUSRQ}$ RE Delay		165		150	ns
69	TdBRQ(BUSc)	$\overline{BUSRQ}$ RE to Control Bus Float Delay		140		110	ns
70	TdBRQ(BUSd)	$\overline{BUSRQ}$ RE to AD Bus Float Delay		140		110	ns
71	TdDSA(RDV)	$\overline{DS}$ FE (Acknowledge) to Data Output Valid Delay		135		120	ns
72	TdDSA(RDZ)	$\overline{DS}$ RE (Acknowledge) to Data Output Float Delay		80		75	ns
73	TdDSR(DOD)	$\overline{DS}$ FE (IOR) to Data Output Driven Delay		135		120	ns
74	TdDSR(RDZ)	$\overline{DS}$ RE (IOR) to Data Output Float Delay		80		75	ns
75	TwAS	$\overline{AS}$ Low Width	70		50		ns
76	TsA(AS)	Address Valid to $\overline{AS}$ RE Set-up Time	30		10		ns
77	ThAS(Av)	$\overline{AS}$ RE to Address Valid Hold Time	50		40		ns
78	TdAS(DS)	$\overline{AS}$ RE to $\overline{DS}$ FE Delay (I/O)	50		40		ns
79	TsCS(AS)	$\overline{CS}$ Valid to $\overline{AS}$ RE Set-up Time	0		0		ns
80	ThCS(AS)	$\overline{AS}$ RE to $\overline{CS}$ Valid Hold Time	40		30		ns
81	TwAS(DS)	$\overline{AS}$ and $\overline{DS}$ Simultaneously LOW Time (Reset)	3TcC		3TcC		ns
82	TdBAI(Az)	$\overline{BAI}$ RE to SN0 – SN7, AD0 – AD15 Float Delay (Reset)		135		120	ns
83	TdBAI(ST)	$\overline{BAI}$ RE to ST0 – ST3, R/$\overline{W}$, B/$\overline{W}$, N/$\overline{S}$ Float Delay (Reset)		100		80	ns
84	TdBAI(DS)	$\overline{BAI}$ RE to $\overline{DS}$, $\overline{AS}$ Float Delay (Reset)		100		85	ns
85	TdDS(Dn)	$\overline{DS}$ RE (IOW) to Data Valid Hold Time	40		40		ns
86	TdAC(DRV)	Address Valid to Data (IOR) Required Valid Delay		540		345	ns
87	TdAZ(DS)	Address Float to $\overline{DS}$ FE (IOR) Delay	0		0		ns
88	TwDS(IO)	$\overline{DS}$ (IO) LOW Width	150*		150		ns
89	TsD(DS)	Data (IOW) Valid to $\overline{DS}$ RE Set-up Time	40		40		ns
90	TrDS(W)	$\overline{DS}$ RE (IOW) to $\overline{DS}$ FE (IOW) (Write Recovery Time applies only for issuing Command)	4TcC		4TcC		ns
91	TsBAK(C)	$\overline{BAI}$ Valid to Clock RE Set-up Time	60		50		ns
92	TdAS(DS)	$\overline{AS}$ RE to $\overline{DS}$ FE (ACK) Delay	100		100		ns
93	TwDS(AK)	$\overline{DS}$ (ACK) LOW Width	150		150		ns
94	TdBRQ(BAI)	$\overline{BUSRQ}$ FE to $\overline{BAI}$ FE Required Delay	0		0		ns
95	TsS(AS)	Status Valid to $\overline{AS}$ RE Set-up Time	40		0		ns
98	TdBAI(BAO)	$\overline{BAI}$ RE, FE to $\overline{BAO}$ RE, FE Delay		80		70	ns
99	TdIEI(IEO)	IEI RE, FE to IEO RE, FE Delay		80		60	ns

*2000ns for reading slow-readable registers (worst case).
Note: RE = rising edge
FE = falling edge

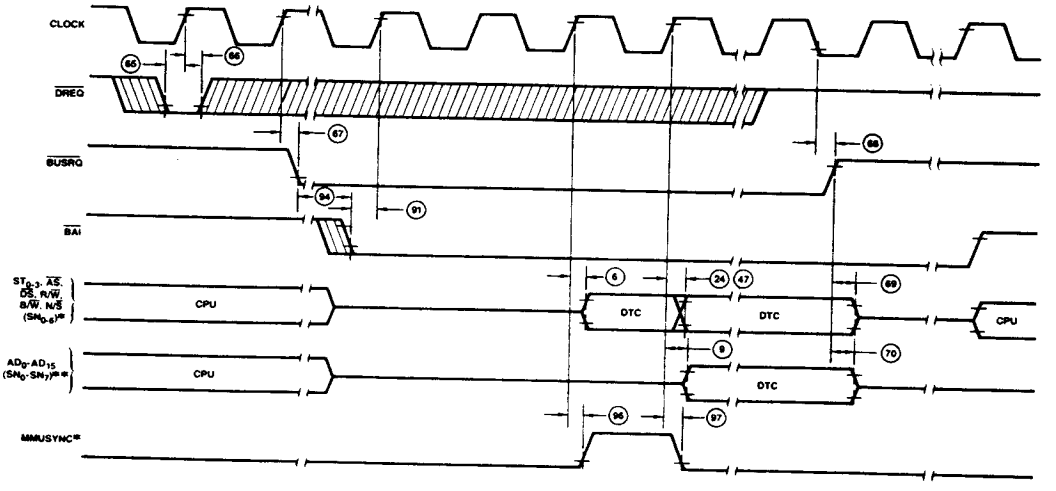
Refer to page 7-1 for Essential Information on Military Devices

Timing Diagram 12. AC Timing when DTC is Bus Slave



WF005790

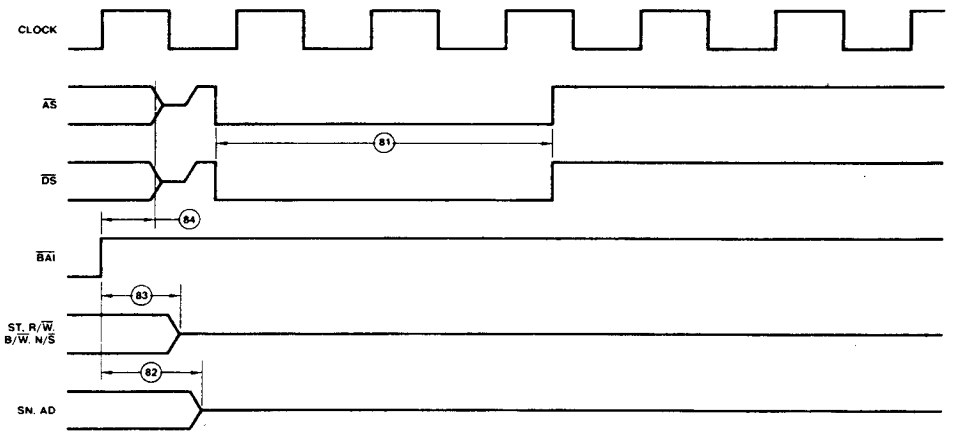
Timing Diagram 13. Bus Exchange Timing



WF005800

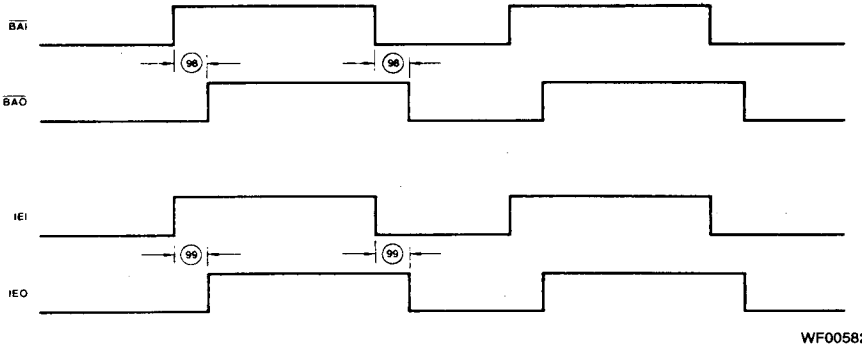
*For logical addressing only
**For physical addressing only

Timing Diagram 14. Reset Timing



WF005810

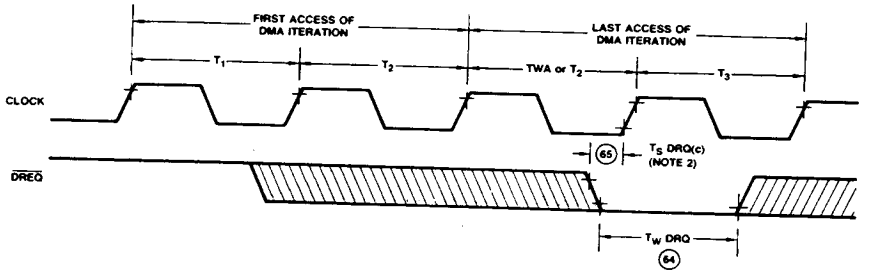
Timing Diagram 15. Delay Timings



WF005820

Refer to page 7-1 for Essential Information on Military Devices

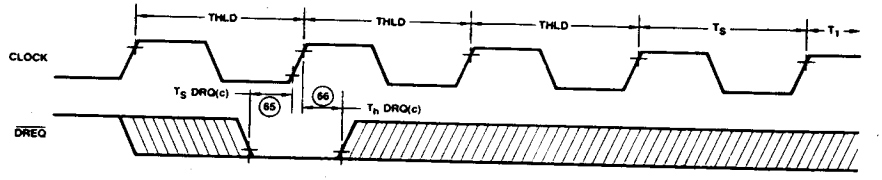
Timing Diagram 16. Sampling $\overline{DREQ}$ during Single Transfer DMA Operations



WF005830

Notes: 1. HIGH-to-LOW $\overline{DREQ}$ transitions will only be recognized after the HIGH-to-LOW transition of the clock during T_1 of the first access of the DMA iteration.
2. A HIGH-to-LOW $\overline{DREQ}$ transition must meet the conditions in Note 1 and must occur T_SDRQ(c) before state T_3 of the last access of the DMA iteration if the channel is to retain bus control and immediately start the next iteration. $\overline{DREQ}$ may go HIGH before T_SDRQ(c) if it has met the T_WDRQ parameter.
3. Flyby and Search transactions have only a single access; parameter T_SDRQ(c) should be referenced to the start of T_3 of the access. All other operations will always have two or three accesses per iteration.

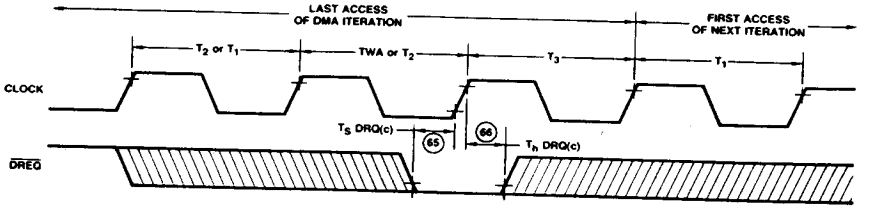
Timing Diagram 17a). Sampling $\overline{DREQ}$ at the End of Chaining



WF005840

Notes: 1. $\overline{DREQ}$ must be LOW from the start of T_SDRQ (c) to the end of T_hDRQ (c) to ensure that the request is recognized.
2. Failure to meet this setup time will result in the channel releasing the bus.

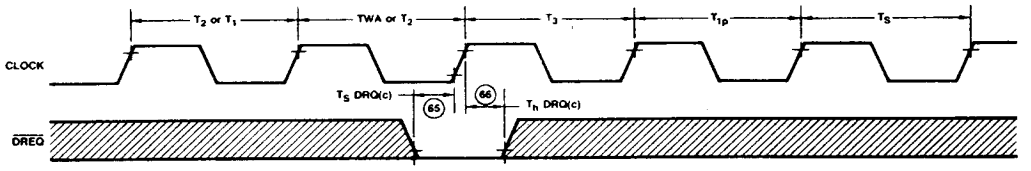
Timing Diagram 17b). Sampling of $\overline{DREQ}$ While in Bus Hold Mode



WF005850

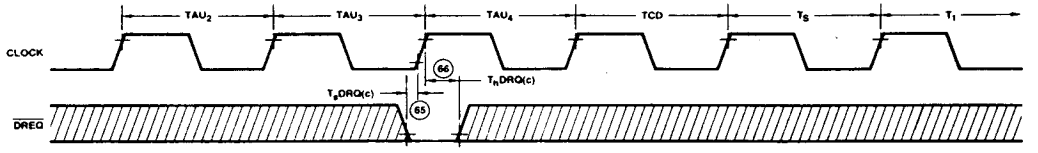
Refer to page 7-1 for Essential Information on Military Devices

Timing Diagram 17c). $\overline{DREQ}$ Sampling in Demand Mode during DMA Operations



WF005860

Timing Diagram 17d). Sampling $\overline{DREQ}$ at the End of Base-to-Current Reloading



WF005870

Notes: 1. T_S is a set-up state, generated before entering DMA operation cycle.
2. TAU_2 through TAU_4 are auto-reloading states, followed by TCD (chain decision) state.

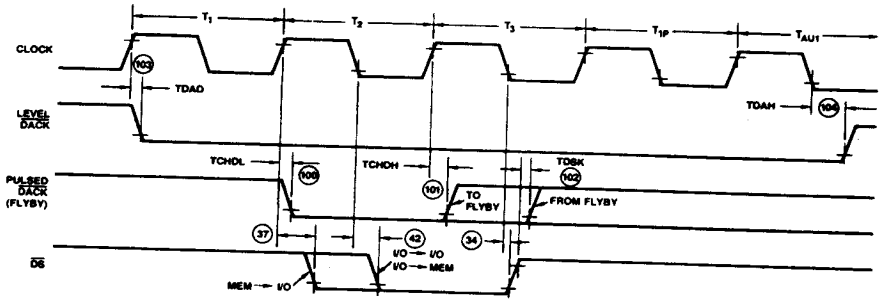
SWITCHING CHARACTERISTICS over operating range unless otherwise specified
TIMING FOR DTC-PERIPHERAL INTERFACE

Number	Parameters	Description	4MHz		6MHz		Units
			Min	Max	Min	Max	
100	TCHDL	Clock RE to Pulsed $\overline{DACK}$ FE Delay (Flyby Transactions Only)		100		100	ns
101	TCHDH	Clock RE to Pulsed $\overline{DACK}$ RE Delay (Transactions TO Flyby Peripheral Only)		100		100	ns
102	TDSK	$\overline{DS}$ RE to Pulsed $\overline{DACK}$ RE Delay (Transactions FROM Flyby Peripheral Only)	20		20		ns
103	TDAD	Clock RE to Level $\overline{DACK}$ Valid Delay		100		100	ns
104	TDAH	Clock FE to Level $\overline{DACK}$ Valid Delay		110		110	ns
105	TEIDL	Clock FE to Internal $\overline{EOP}$ FE Delay		110		100	ns
106	TEIDH	Clock FE to Internal $\overline{EOP}$ RE Delay		110		100	ns
107	TES	External $\overline{EOP}$ Valid to Clock FE Set-up Time During Operation	10		10		ns
108	TEW	External $\overline{EOP}$ Pulse Width Required During Operation	20		20		ns
109	TES(BH)	External $\overline{EOP}$ Valid to Clock RE Set-up Time During Bus Hold	10		10		ns
110	TEW(BH)	External $\overline{EOP}$ Pulse Width Required During Bus Hold	20		20		ns

Note: RE = rising edge FE = falling edge

Refer to page 7-1 for Essential Information on Military Devices

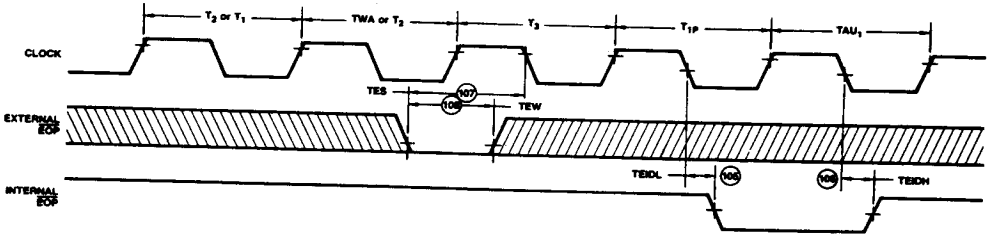
Timing Diagram 18. $\overline{DACK}$ Timing



WF005750

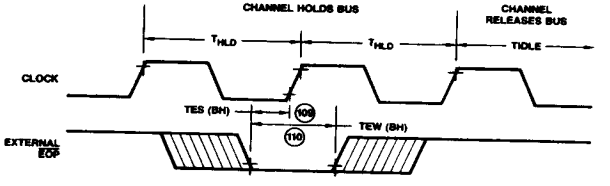
*LEVEL $\overline{DACK}$ RE occurs as shown if auto-reloading is not programmed. LEVEL $\overline{DACK}$ stays LOW for three additional clocks for reloading.

Timing Diagram 19a). $\overline{EOP}$ Sampling and Generation during DMA Operations



WF005760

Timing Diagram 19b). Sampling of $\overline{EOP}$ during Bus Hold



WF005770

Notes: 1. The diagram lists state names of both I/O and memory accesses. Sampling of $\overline{EOP}$ will occur on the falling edge of state T_3.
2. State T_{1P} is a pseudo-T_1 state, without an active $\overline{AS}$ generated, following termination of any DMA operation.
3. State TAU_1 is an auto-initialization state, generated following the TC, MC or EOP termination.

AmZ8016 DMA Derived Timings

Parameters	Description	Derivation
11	Adr. Valid to Data In	2.5 Φ − #9 − #12
13	$\overline{DS}$ ↑ to Adr. Active	.5 Φ − #34 + #9
16	Data Out to $\overline{DS}$ ↑	1.5 Φ − #14 + #34
21	Data Out to $\overline{DS}$ ↓	.5 Φ − #14 + #39
25	Adr. Valid to $\overline{AS}$ ↑	.5 Φ − #9 + (#26 − tr)
27	$\overline{AS}$ ↑ to Data In	2 Φ − #26 − #12
28	$\overline{DS}$ ↑ to $\overline{AS}$↓	.5 Φ − #34 + #24
29	$\overline{AS}$ Width	.5 Φ − #24 + (#26 − tr)
30	$\overline{AS}$ ↑ to Adr. Hold	.5 Φ − #26 + #10
31	Adr. float to $\overline{DS}$ ↓	#10 − #37
32	$\overline{AS}$↑ to $\overline{DS}$ ↓	.5 Φ − #26 + #37
33	$\overline{DS}$ ↓ to Data In (Read)	1.5 Φ − #37 − #12
35	$\overline{DS}$ ↑ to Data Out Hold	.5 Φ − #34 + #8
36	Adr. Valid to $\overline{DS}$ ↓	1 Φ − #9 + #37
38	$\overline{DS}$ Width (Read)	1.5 Φ − #37 + #34
40	$\overline{DS}$ Width (Write)	1 Φ − #39 + #34
41	$\overline{DS}$ ↓ to Data In (Input)	(1 + Wait) Φ − #42 − #12
43	$\overline{DS}$ Width (I/O)	1 Φ − #42 + #34
48	Status Valid to $\overline{AS}$ ↑	.5 Φ − #47 + (#26 − tr)

tr (nominal) = 10μS

Refer to page 7-1 for Essential Information on Military Devices