Am9516A

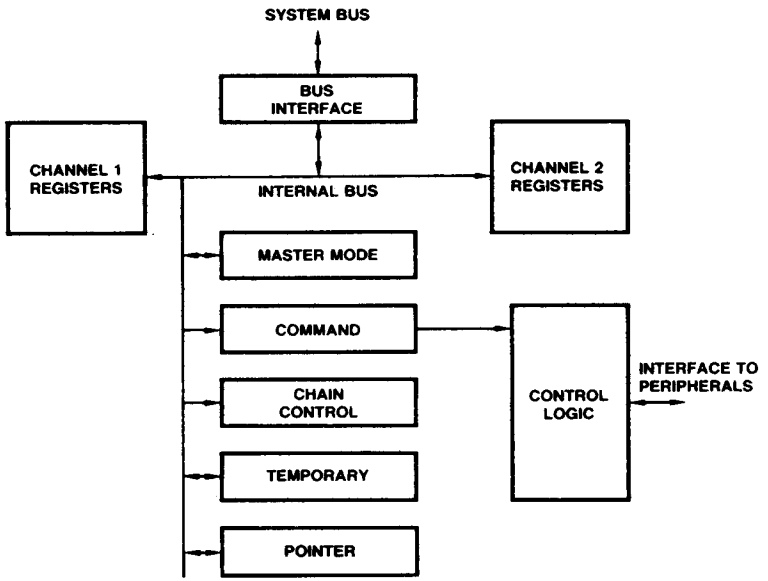
Universal DMA Controller (UDC)

FINAL

DISTINCTIVE CHARACTERISTICS

- Transfer Modes: Single, demand dedicated with bus hold, demand dedicated with bus release, demand interleave
- 16 MB physical addressing range
- Automatic loading/reloading of control parameters by each channel
- Optional automatic chaining of operations
- Channel interleave operations
- Interleave operations with system bus
- Masked data pattern matching for search operations
- Vectored interrupts on selected transfer conditions
- Software DMA request
- Software or hardware wait state insertion
- Transfer up to 6.66 MB/second at 10 MHz clock

BLOCK DIAGRAM



Publication # 03242 Rev. E Amendment /0
Issue Date: August 1989

GENERAL DESCRIPTION

The Am9516A Universal DMA Controller (UDC) is a high performance peripheral interface circuit for 8086 and 68000 CPUs. In addition to providing data block transfer capability between memory and peripherals, each of the UDC'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 UDC 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 UDC 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 memory areas. 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 re-initialize 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.

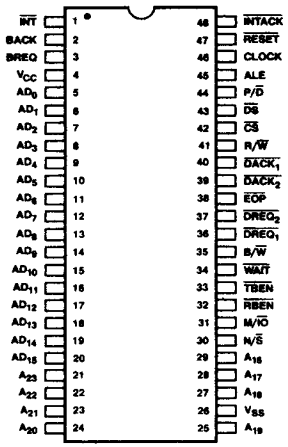
Frequently DMA devices must interface to slow peripherals or slow memory. In addition to providing a hardware $\overline{\text{WAIT}}$ input, the Am9516A UDC allows the user to select independently for both source and destination addresses and automatic insertion of 0, 1, 2 or 4 wait states. The user may even disable the $\overline{\text{WAIT}}$ 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 Am9516A UDC 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's Chain Address register and then issue a "Start Chain" Command to start the register loading operation.

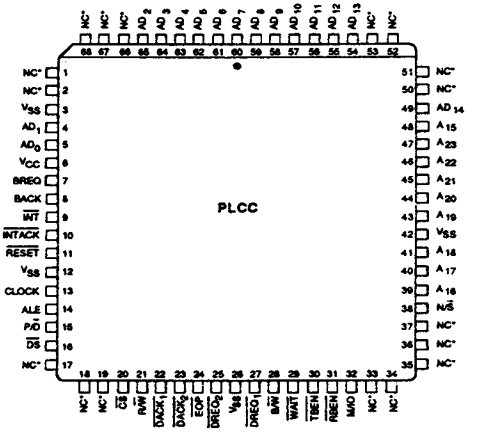
The Am9516A UDC is packaged in a 48-pin DIP and uses a single +5 V Power Supply.

CONNECTION DIAGRAM
Top View

DIPs



CD005592



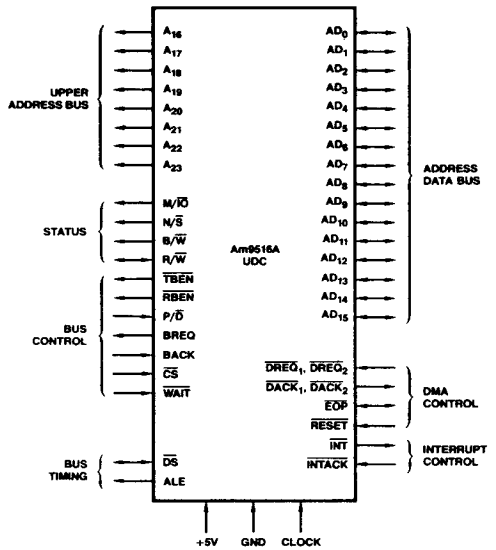
CD011970

*NC = No Connection

Note: Pin 1 is marked for orientation.

2

LOGIC SYMBOL

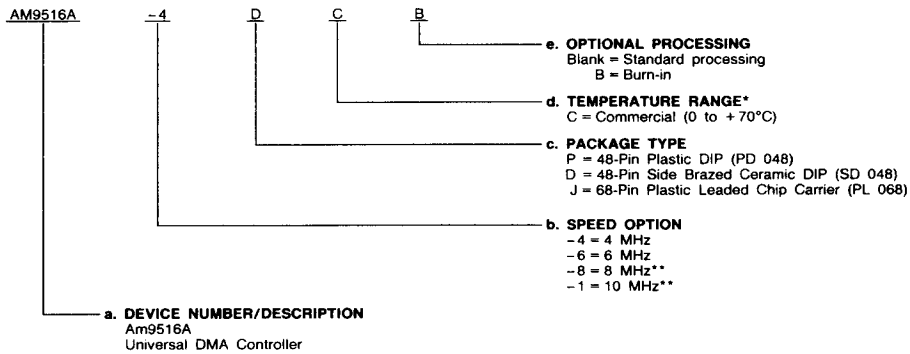


LS001331

ORDERING INFORMATION

Standard Products

AMD standard products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM9516A-4	DC, DCB, PC, JC
AM9516A-6	
AM9516A-8	
AM9516A-1	DC

*This device is also available in Military temperature range.

Valid Combinations

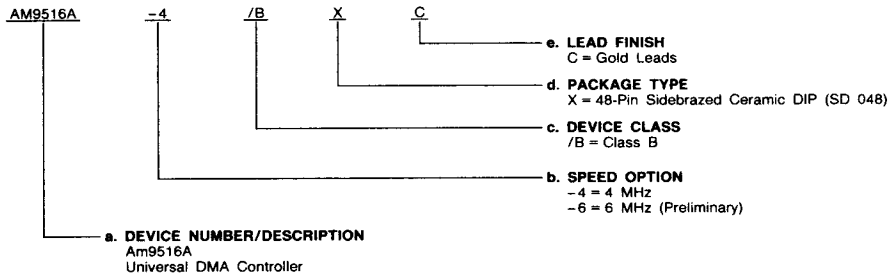
Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

**Preliminary; to be announced.

MILITARY ORDERING INFORMATION

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) for APL products is formed by a combination of: **a. Device Number**
b. Speed Option (if applicable)
c. Device Class
d. Package Type
e. Lead Finish



Valid Combinations	
AM9516A-4	/BXC

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

Group A Tests

Group A tests consist of Subgroups
1, 2, 3, 7, 8, 9, 10, 11.

PIN DESCRIPTION

Pin No.	Name	I/O	Description
4	V_{CC}		+5 V Power Supply.
26	V_{SS}		Ground.
46	CLOCK	I	Clock.
46	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. Many UDC input signals can make transitions independent of the UDC clock; these signals can be asynchronous to the UDC clock. On other signals, such as $\overline{WAIT}$ inputs, transitions must meet setup and hold requirements relative to the UDC clock. See the Timing diagrams for details.
5-20	$AD_0 - AD_{15}$	I/O	(Address-Data Bus, Three-State). The Address Data Bus is a time-mulitiplexed, 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 0. 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 ALE, and the asserted or requested presence of data is defined by the $\overline{DS}$ signal. The status output lines M/$\overline{IO}$ and N/$\overline{S}$ indicate the type of transaction, either memory or I/O. The R/$\overline{W}$ line indicates the direction of the transaction. When the UDC is in control of the system bus, it dominates the AD Bus; when the UDC is not in control of the system bus, the CPU or other external devices dominate the AD Bus. The presence of address of data on the $AD_0 - AD_{15}$ bus is defined only by ALE and $\overline{DS}$. When the UDC is not in control of the bus, there is no required relation between the presence of address or data and the UDC clock. This allows the UDC to be used with a system bus which does not have a bussed clock signal.
43	$\overline{DS}$	I/O	(Data Strobe, Three-State). Data Strobe is a bidirectional, active-low, three-state signal. A LOW on this signal indicates that the $AD_0 - AD_{15}$ bus is being used for data transfer. When the UDC is not in control of the system bus and the external system is transferring information to or from the UDC, $\overline{DS}$ is a timing input used by the UDC to move data to or from the $AD_0 - AD_{15}$ bus. Data is written into the UDC by the external ststem on the LOW-to-HIGH $\overline{DS}$ transition. Data is read from the UDC by the external system while $\overline{DS}$ is LOW. There are no timing requirements between $\overline{DS}$ as an input and the UDC clock; this allows use of the UDC with a system bus which does not have a bussed clock. During a DMA operation when the UDC is in control of the ststem, $\overline{DS}$ is an output generated by the UDC and used by the system to move data to or from the $AD_0 - AD_{15}$ bus. When the UDC 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 $\overline{DS}$ inputs data from the $AD_0 - AD_{15}$ bus into the UDC (see Timing diagram).
41	R/$\overline{W}$	I/O	(Read/Write, Three-State). 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 ALE is HIGH until the bus transaction ends (see Timing diagram). When the UDC is not in control of the system bus and the external system is transferring information to or from the UDC, R/$\overline{W}$ is a status input used by the UDC to determine if data is entering or leaving on the $AD_0 - AD_{15}$ bus during $\overline{DS}$ time. In such a case, Read (HIGH) indicates that the system is requesting data from the UDC, and Write (LOW) indicates that the system is presenting data to the UDC. There are no timing requirements between R/$\overline{W}$ as an input and the UDC clock; transitions on R/$\overline{W}$ as an input are only defined relative to $\overline{DS}$. When the UDC is in control of the system bus, R/$\overline{W}$ is an output generated by the UDC, with Read indicating that data is being requested from the addressed location or device. the addressed location or device 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.
33	$\overline{TBEN}$	O	(Transmit Buffer Enable, Open Drain). Transmit Buffer Enable is an active-low, open drain output. When UDC is a bus master, a LOW on this output indicates that the data is being transferred from the UDC to the data bus lines through the buffer. The purpose of this signal is to eliminate bus contention. When UDC is not in control of the system bus, these pins float to three-state OFF.
32	$\overline{RBEN}$	O	(Receive Buffer Enable, Open Drain). Receive Buffer Enable is an active-low, open drain output. When UDC is in control of system bus, a LOW on this output indicates that the data is being transferred from the data bus lines to the UDC through the buffer. The purpose of this signal is to eliminate bus contention. This pin floats to three-state OFF when the UDC is not in control of the system bus.
45	ALE	O	(Address Latch Enable). This active HIGH signal is provided by the UDC to latch the address signals $AD_0 - AD_{15}$ into the address latch. This pin is never floated.
44	P/$\overline{D}$	I	(Pointer/Data). Pointer/Data is an input signal to indicate the information is on the $AD_0 - AD_{15}$ bus only when the UDC is the bus slave. A HIGH on this signal indicates the information is on the AD bus is an address of the internal register to be accessed. The data on the AD bus is loaded into the Pointer register of UDC. A LOW on this signal indicates that a data transfer is taking place between the bus and the internal register designated by the Pointer register. Note that if a transaction is carried out with R/$\overline{W}$ HIGH and P/$\overline{D}$ HIGH, the contents of the Pointer register will be read.
31	M/$\overline{IO}$	O	(Memory/Input-Output, Three-State). This signal specifies the type of transaction. A HIGH on this pin indicates a memory transaction. A LOW on this pin indicates an I/O transaction. It floats to three-state OFF when UDC is not in control of the ststem bus.
30	N/$\overline{S}$	O	(Normal/System, Three-State). This output is a three-state signal activated only when the UDC is the bus master. Normal is indicated when N/$\overline{S}$ is HIGH, and System is indicated when N/$\overline{S}$ is LOW. This signal supplements the M/$\overline{IO}$ line and is used to indicate which memory or I/O space is being accessed.
35	B/$\overline{W}$	O	(Byte/Word, Three-State). This output indicates the size of data transferred on the $AD_0 - AD_{15}$ bus. HIGH indicates a byte (8-bit) and LOW indicates a word (16-bit) transfer. This output is activated when ALE is HIGH and remains valid for the duration of the whole transaction (see Timing diagram). All word-sized data are word-aligned and must be addressed by even addresses ($A_0 = 0$). When addressing byte read 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 least significant byte!) This addressing mechanism applies to memory accesses as well as I/O accesses. When the UDC is a slave, it ignores the B/$\overline{W}$ signal and this pin floats to three-state OFF.

PIN DESCRIPTION (continued)

Pin No.	Name	I/O	Description
42	$\overline{CS}$	I	(Chip Select). This pin is an active-low input. A CPU or other external device uses $\overline{CS}$ to activate the UDC for reading and writing of its internal registers. There are no timing requirements between the $\overline{CS}$ input and the UDC clock; the $\overline{CS}$ input timing requirements are only defined relative to $\overline{DS}$. This pin is ignored when UDC is in control of system bus.
34	$\overline{WAIT}$	I	(WAIT). This pin is an active-low input. Slow memories and peripheral devices may use $\overline{WAIT}$ to extend $\overline{DS}$ and $\overline{RBEN}$ or $\overline{TBEN}$ during operation. Unlike the $\overline{CS}$ input, transitions on the $\overline{WAIT}$ input must meet certain timing requirements relative to the UDC clock. See Timing Diagram 4 for details. The Wait function may be disabled using a control bit in the Master Mode register (MM2).
3	BREQ	O	(Bus Request). Bus Request is an active-HIGH signalsignal used by the UDC to obtain control of the bus from the CPU. BREQ lines from multiple devices are connected to a priority encoder.
2	BACK	I	(Bus Acknowledge). BACK is an active-HIGH, asynchronchronous input, indicating that the CPU has relinquished the bus and that no higher priority device has assumed bus control. Since BACK is internally synchronized by the UDC before being used, transitions on BACK do not have to be synchronous with the UDC clock. The BACK input is usually connected to the HLDA line from the CPU or to the output of a priority decoder.
1	$\overline{INT}$	O	(Interrupt Request, Open Drain). Interrupt Request is an active-low output used to interrupt the CPU. It is driven LOW whenever the IP and CIE bits of the Status Register are set. It is cleared by UDC after receiving a clear IP command.
48	$\overline{INTACK}$	I	(Interrupt Acknowledge). Interrupt Acknowledge is an active-low input indicating that the request for interrupt has been granted. The UDC will place a vector onto the AD bus if the No Vector or Interrupt bit (MM3) is reset.
47	$\overline{RESET}$	I	(Reset). Reset is an active-low input to disable the UDC and clear its Master Mode register.
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 UDC clock and are used by external logic to initiate and control DMA operations performed by the UDC.
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 operations 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 intiated by a CPU software request command or as a result of chaining. $\overline{DACK}$ is not output during the 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 kohm or more. The UDC 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.
29 – 27 25 – 21	$A_{16} - A_{23}$	O	(Upper Address Bus, Three-state). The $A_{16} - A_{23}$ address lines are three-state outputs activated only when the UDC is controlling the system bus. Combined with the lower 16 address bits appearing on AD_0 through AD_{15} respectively, this 24-bit linear address allows the UDC to access anywhere within 16 megabytes of memory.

Note: All inputs to the UDC, except the clock, are directly TTL compatible.

2

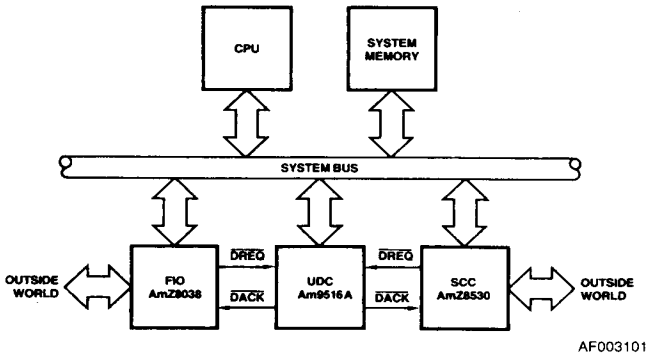


Figure 1. UDC Configurations

PRODUCT OVERVIEW

Register Description

The Am9516A UDC Block Diagram illustrates the internal registers. Figure 2 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 a normal CPU I/O operation without additional wait states. Reading slow registers requires multiple wait states. 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 UDC ignores the B/$\overline{W}$ line in slave mode. It is the responsibility of the user to supply the necessary external logic if slow readable registers are to be read.

The UDC registers can be categorized into chip-level registers, which control the overall operation and configuration of the UDC, and channel-level registers which are duplicated for each channel. The five chip-level registers are the Master Mode register the Command register, the Chain Control register, the Pointer register and the Temporary register. The Master Mode register selects the way the UDC chip interfaces to the system. The Command register is written to by the host CPU to initiate certain operations within the UDC 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 Pointer register is written to by the host CPU when the P/$\overline{D}$ input is HIGH. The data in Pointer register is the address of the internal register to be accessed. The Temporary register is used to hold data for Flowthru Transfer/Transfer-and-Searches.

The channel-level registers can be divided into two subcategories: general purpose registers, which would be found on most DMA chips, and special purpose registers, which provide additional features and functionality. The general purpose registers are the Base and Current Operation Count registers, the Base and Current Address registers A and B and the Channel Mode register. The special purpose registers are the Pattern and Mask registers, the Status register, the Interrupt Vector register, the Interrupt Save register and the Chain Address register.

The internal registers are read or written in two steps. First, the address of the register to be accessed is written to the Pointer register, when the P/$\overline{D}$ input is HIGH. Then, the data is read from or written into the desired register, which is indicated by the Pointer register, when P/$\overline{D}$ input is LOW. Note that a read with P/$\overline{D}$ HIGH causes the contents of the Pointer register to be read on AD_1 through AD_6.

Master Mode Register

The 4-bit Master Mode register, shown in Figure 3, controls the chip-level interfaces. It can be read from and written to by the host CPU without wait states through pins $AD_0 - AD_3$, 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 UDC to request the bus. When enabled, the UDC 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 UDC is inhibited from requesting control of the system bus and, therefore, inhibited from performing chaining or DMA operations.

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

The Wait Line Enable bit is used to enable sampling of the $\overline{WAIT}$ line during Memory and I/O transactions. Because the UDC provides the ability to insert software programmable wait states, many users may disable sampling of the $\overline{WAIT}$ pin to eliminate 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 "No Vector on Interrupt" bit selects whether the UDC 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 $\overline{INTACK}$ 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.

Pointer Register

The Pointer register contains the address of the internal register to be accessed. It can be read from or written to by the CPU when the P/$\overline{D}$ line is HIGH.

Name	Size	Number	Access Type	Port Address CH-1/CH-2
Master Mode Register	4 bits	1	FW	38
Pointer Register	6 bits	1	FW	
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:				
Up-Addr/Tag field	14 bits	2	CFW	1A/18
Lower Address field	16 bits	2	CFW	0A/08
Current Address Register – B:				
Up-Addr/Tag field	14 bits	2	CFW	12/10
Lower Address field	16 bits	2	CFW	02/00
Base Address Register – A:				
Up-Addr/Tag field	14 bits	2	CFW	1E/1C
Lower Address field	16 bits	2	CFW	0E/0C
Base Address Register – B:				
Up-Addr/Tag field	14 bits	2	CFW	16/14
Lower Address 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:				
Up-Addr/Tag field	10 bits	2	CFW	26/24
Lower Address field	16 bits	2	CFW	22/20

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

Note: The address of the register to be accessed is stored in the Pointer register.

*Port addresses of the Command register can be used alternately for both channels except when issuing a "set or clear IP" command.

Figure 2. UDC Internal Register

2

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 (Figure 10). 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.

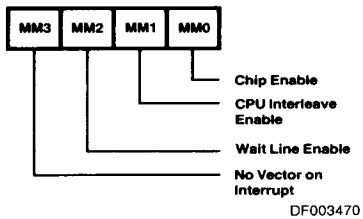


Figure 3. Master Mode Register

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 UDC Command register (Figure 19) is an 8-bit write-only register written to by the host CPU. 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 contents of the Base ARA and ARB registers are loaded 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 6-bit Tag Field and an 8-bit Upper Address in one word and a 16-bit Lower Address in the other. See Figure 5. 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 may be either HIGH (indicating Normal) or LOW (indicating System) for space. 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 register's 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.

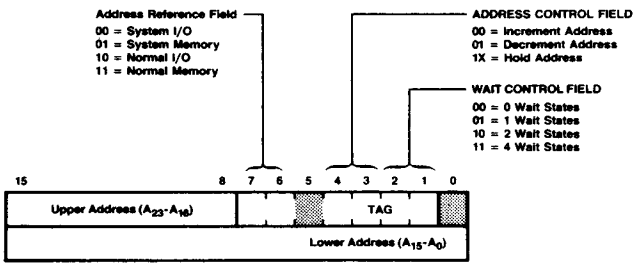
Current and Base Operation Count Registers

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 zeroes, 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.

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.



DF003390

Figure 4. Address Registers A and B

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 5, 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) and Interrupt Pending (IP) bits. These bits are described in detail in the "Interrupt" section of this document.

The UDC status field contains the current channel status. 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 BREQ HIGH, depending on the programming of the Master Mode Chip Enable bit (MMO) 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 Chaining Abort (CA) and the NAC will be set. These bits are also set when a "reset" is issued to the UDC. The CA bit holds the NAC bit in the set state. The CA bit is cleared when a new Chain Upper Address and Tag word or Lower Address 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 UDC 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 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, thereby erasing the old setting. Three bits indicate whether the DMA operation ended as a result of a TC, MC or $\overline{EOP}$ 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 external $\overline{EOP}$ ends a DMA transfer; it is not set for $\overline{EOP}$ issued during chaining. Note that two or even all three of MC, TC and EOP may be set if multiple reasons exist 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 did not match when using 8-bit matches with word searches and transfer-and-searches. The three reserved bits return zeroes during reads.

Interrupt Vector and Interrupt Save Registers

Each channel has an Interrupt Vector register and an Interrupt Save register. The Interrupt Vector is 8-bit wide and is written to and read from on $AD_0 - AD_7$. The Interrupt Save register may be read 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 $\overline{EOP}$ 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 6).

Because the vector and status are stored, a new vector can be loaded into the Interrupt Vector register 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 are loaded into the Interrupt Save register and Channel Operation resumes. The UDC 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.

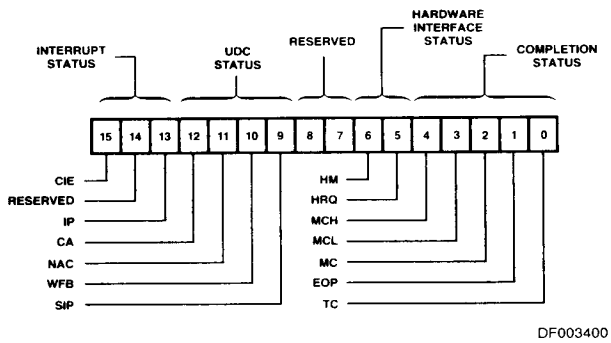


Figure 5. Status Register

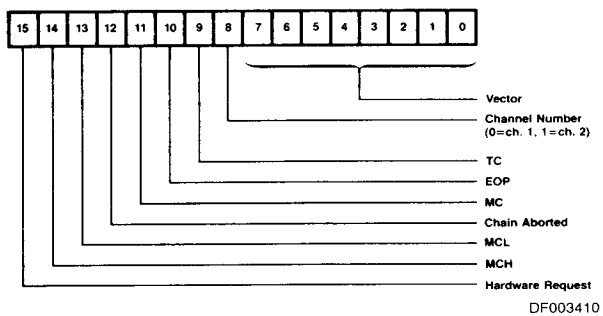


Figure 6. Interrupt Save Register

Channel Mode Register

The Channel Mode registers are two words wide. There are 21 bits defined in each Channel Mode register; the other 11 bits are unused. See Figure 7. The Channel Mode registers may be loaded during chaining and may be read by the host CPU. CPU reads of the Channel Mode register are slow reads and require insertion of multiple wait states. The Channel Mode Low word (bits 0–15) may be written to directly by the host CPU. 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. It also selects the operand size of bytes or words (see Figure 8 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.

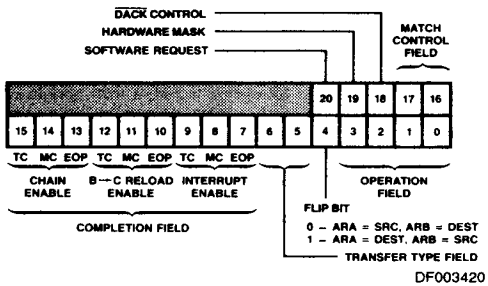


Figure 7. Channel Mode Register

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 8 and the "Search" section for details. The Software Request bit and Hardware Mask bit can be set and cleared by software command in addition to being loaded in parallel with other Channel Mode bits. These bits are described in detail in the "Initiating DMA Operations" section.

The $\overline{\text{DACK}}$ Control bit is used to specify when the $\overline{\text{DACK}}$ pin is driven active. When this bit is cleared, the channel's $\overline{\text{DACK}}$ 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 $\overline{\text{DACK}}$ pin will be inactive during chaining, during both Flowthru Transfers and Flowthru Transfer-and-Searches, and during Searches, but $\overline{\text{DACK}}$ 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.

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

Figure 8. Channel Mode Coding

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 9, is two words long. The first word consists of an Upper Address and Tag field. The second word contains the 16-bit Lower Address portion of the memory address. 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 EOP is issued to the UDC 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 EOPs that occur when chaining and while loading a new Chain Address cause the new data to be lost.

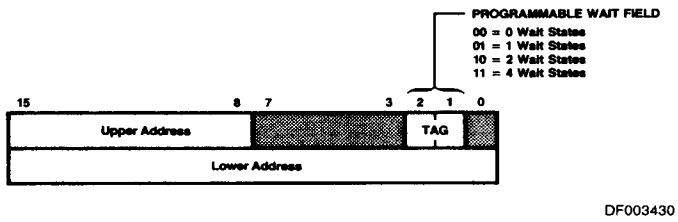


Figure 9. Chain Address Register

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 UDC can be reset either by hardware or software. The software reset command is described in the "Commands" section. Hardware resets are applied by pulling $\overline{\text{RESET}}$ LOW. The UDC may be in control of the bus when a reset is applied. BACK is removed internally causing the outputs to go tri-state. If BACK remains HIGH after reset, the UDC will not drive the bus unless BREQ is active. As soon as BACK goes inactive, the UDC places the $AD_0 - AD_{15}$, $A_{16} - AD_{23}$, R/$\overline{\text{W}}$, $\overline{\text{DS}}$, N/$\overline{\text{S}}$, M/$\overline{\text{IO}}$ B/$\overline{\text{W}}$, $\overline{\text{TBEN}}$ and $\overline{\text{RBEN}}$ signals in the high-impedance state.

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

The Master Mode register contains all zeroes after a reset. The UDC is disabled, and the CPU interleave and hardware wait are inhibited.

Because the CA and NAC bits in the Status register are set by a 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 Am9516A UDC design is that the UDC should be able to operate with a minimum of interaction with the host CPU. This goal is achieved by having the UDC 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 the System memory space and pointed to by the Chain Address register. This reloading operation is called chaining, and the table is called the Chain Control Table.

The Upper and Lower Address fields of the Chain Address register form a 24-bit address which points to a location in system memory space. Chaining is performed by repetitively reading words from memory. Note that the Chain Address register should always be loaded with an even Address; loading an odd Address 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 UDC 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 10. 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 11). 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 the Reload Word in memory (i.e., the data are stored 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.

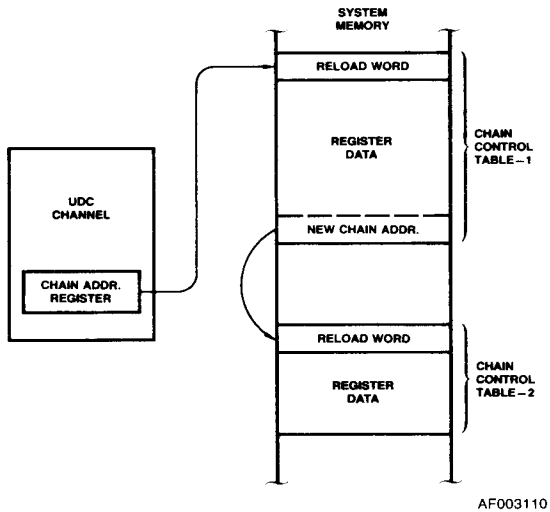


Figure 10. Chaining and Chain Control Tables

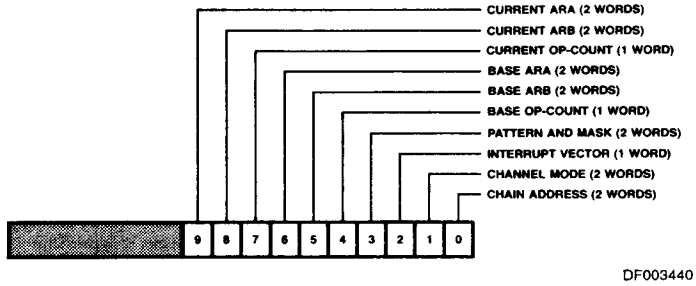


Figure 11. Reload Word/Chain Control Register

When the channel is to reload itself, it first uses the Chain Address register contents to load the Reload Word into the UDC's Chain Control register. Next, the Chain Address register contents are incremented by two to point to the next word in memory. The channel than scans the 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 is set to "1," the register(s) corresponding to the 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 12 shows two examples of Chain Control Tables. Example 1 shows the ordering of data when all registers 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 Upper Address and Tag word are loaded first, then the Lower Address word. Also, the Pattern register is loaded before the Mask register.

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.

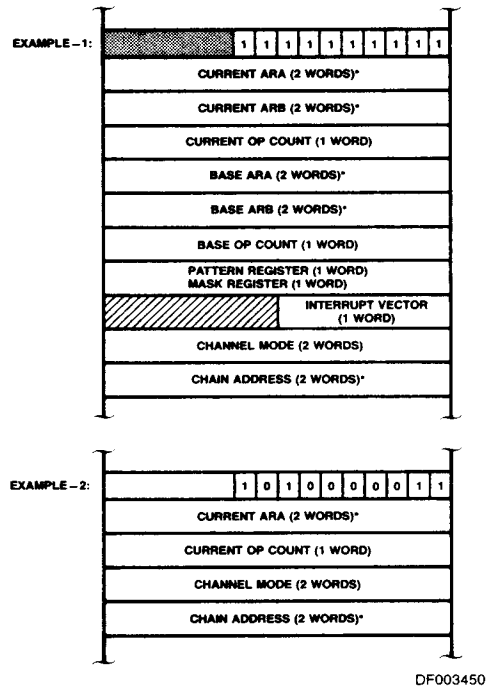


Figure 12. Examples of Chain Control Table

*Load the Upper Address and Tag Word first, then the Lower Address Word.

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 the LOW $\overline{DREQ}$ signals are sampled and when the $\overline{DREQ}$ requests can be applied to start the next DMA operation after chaining (see Timing Diagrams 1 and 2).

Bus Request/Grant

Before the UDC can perform a DMA Operation, it must gain control of the system bus. The BREQ and BACK interface pins provide connections between the UDC and the host CPU and other devices, if present, to arbitrate which device has control of the system bus. When the UDC wants to gain bus control, it drives BREQ HIGH.

Some period of time after the UDC drives BREQ HIGH, the CPU will relinquish bus control and drive its HLDA signal HIGH. When the UDC's BACK input goes HIGH, it may begin performing operations on the system bus. When the UDC finishes its operation, it stops driving BREQ HIGH.

When more than one device is used, a priority encoder and a priority decoder are used to decide the bus grant priority.

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 registers; 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 $CM_3 - M_0$ in the Channel Mode register program whether a 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 UDC'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 (i.e., the byte-word funneling) 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 a memory space ($TG_6 = 2$) and by ±2 if the register points to an I/O space ($TG_6 = 0$).

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 UDC'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 UDC'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 Current Operation Count Register must be loaded with the number of words to be transferred.

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 13 summarizes these characteristics for both byte-to-word and word-to-byte funneling operations. The criterion 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 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. 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 Transactions" sections.

On each read during a Search operation, the UDC'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. CM_2 is an enable for the output of the comparator and allows the MC signal to be generated. 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 the 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 with 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 and 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 bytes match, 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
	TG_4	TG_3	
Word-to-Byte (CM_4 = 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 (CM_4 = 0)	0	0	Increment ARB, Read High Half of Word First
	0	1	Decrement ARB, Read Low Half of Word First
	1	0	Hold ARB, Read High Half of Word First
	1	1	Hold ARB, Read Low Half of Word First

Figure 13. Byte/Word Funneling

Transfer-and-Search

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

Flyby Transfer-and-Search can be used to increase throughput for transfer between two peripherals or between memory and a peripheral. Memory-to-Memory Flyby is not supported. Also, in Flyby, the operand sizes of the source and destination must be the same, funneling is not supported. A complete discussion of Flyby timing is given 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

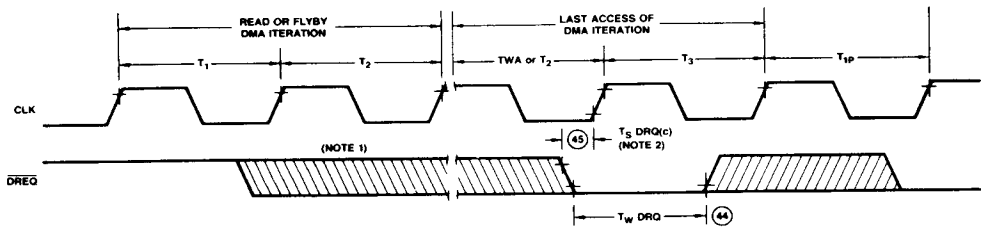
Channel Mode register bits $CM_6 - CM_5$ select the channel's response to the request to start a DMA operation. The response falls into either of two types: Single Operation or Demand. There are three subtypes for Demand operations: Demand Dedicated with Bus Hold, Demand Dedicated with Bus Release, and Demand Interleave. To make 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. See Appendix B for flowchart.

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 HIGH-to-LOW ALE transition of a read or Flyby 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 us at the end of the current iteration.

*TIMING DIAGRAM 1. Sampling $\overline{DREQ}$ During Single Transfer DMA Operations



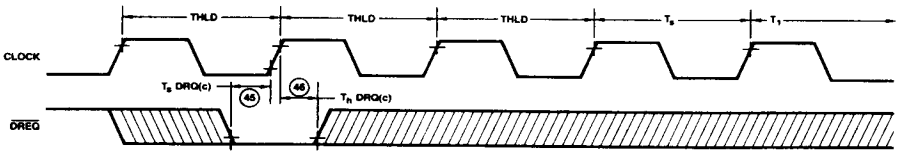
WF007460

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 a read or flyby DMA iteration.
2. A HIGH-to-LOW $\overline{DREQ}$ transition must meet the conditions in Note 1 and must occur TsDRQ(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 TsDRQ(c) if it has met the TwDRQ parameter.
3. Flyby and Search transactions have only a single access; parameter TsDRQ(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.

*See Appendix D for timing parameters.

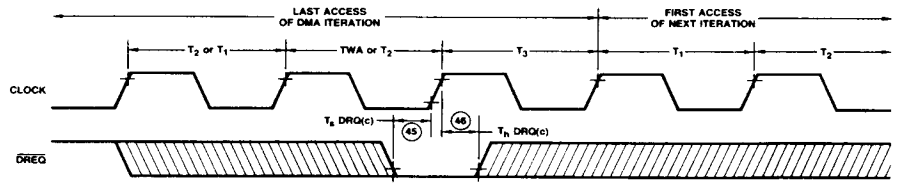
TIMING DIAGRAM 2. $\overline{DREQ}$ Sampling in Demand Mode

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



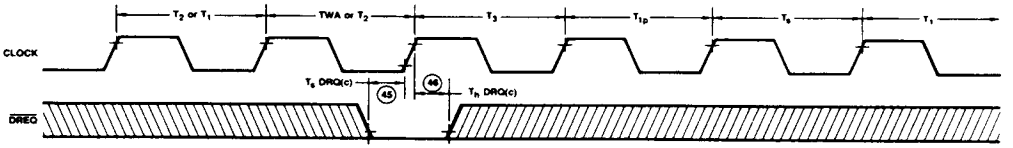
WF007480

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



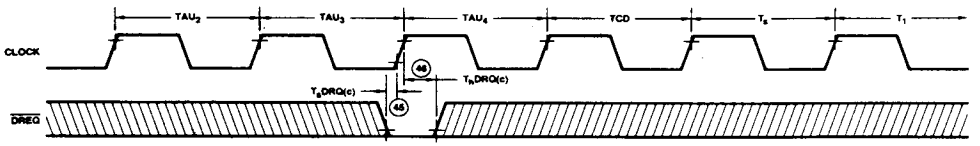
WF007490

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



WF007500

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



WF007510

Notes: 1. $\overline{DREQ}$ must be LOW from the start of TsDRQ(c) to the end of ThDREQ(c) to ensure that the request is recognized.
2. Failure to meet this setup time will result in the channel releasing the bus.
3. Ts is a setup state, generated before entering DMA operation cycle.
4. TAU_2, TAU_3 and TAU_4 are auto-reload states, followed by TCD (chain decision) state.

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. A Software Request will cause the channel to request the bus and perform the DMA operations until TC, MC or EOP.

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. Thus 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.

Demand Dedicated With Bus Release

In Demand Dedicated with Bus Release (abbreviated Bus Release), the application of a Software Request 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 LOW $\overline{DREQ}$ is applied to a channel programmed for Bus Release, the channel will acquire the bus and perform DMA operations: (a) until TC, MC or EOP or (b) until $\overline{DREQ}$ goes inactive. Timing Diagram 2 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 reloading and/or chaining (assuming the Status register's SIP bit is clear) without releasing the bus.

If the SIP bit in the Channel Status register is set when a DMA termination occurs, the channel will relinquish the bus control until an Interrupt Acknowledge has been received and the SIP bit is cleared. After an interrupt has been serviced, the channel will perform the Base-to-Current reloading and/or chaining if enabled for the termination.

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. If MM is set, the UDC 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. If MM is clear, control can pass from one UDC channel to the other without requiring the UDC to release bus control. If both channels have active requests, control will pass to the channel which did not just have control. For instance if MM 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 UDC will retain bus control until both channels are finished with the bus. If MM 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, they may gain control during the part of the sequence labelled CPU. See Appendix B for flowchart.

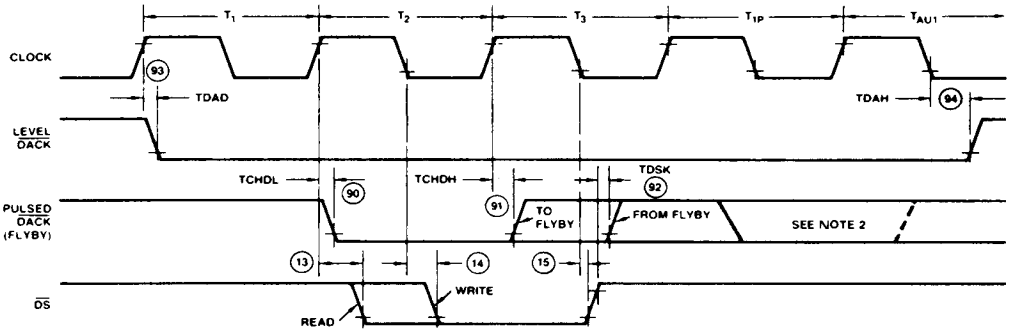
A software or hardware request will cause a channel programmed for Demand Interleave to perform interleaved DMA operations until TC, MC or EOP. If the Software request bit is set during chaining, the channel will retain the bus after chaining and will immediately start performing a DMA iteration and 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 nonflyby modes.

Wait States

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 peripherals 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 $\overline{WAIT}$ pin, the wait function can be disabled by clearing the Wait Line Enable bit (MM_2) in the Master Mode register.

TIMING DIAGRAM 3. $\overline{DACK}$ Timing



WF007521

Notes: 1. Level $\overline{DACK}$ RE occurs as shown if auto-reloading is not programmed; otherwise, it stays LOW for three additional clocks.

2. This extra $\overline{DACK}$ pulse occurs only at EOP. It should be used to distinguish which channel got the EOP.

During DMA transactions, the $\overline{WAIT}$ input is sampled in the middle of the T_2 state. If $\overline{WAIT}$ is HIGH, and if no programmable wait states are selected, the UDC will proceed to state T_3. Otherwise, at least one wait state will be inserted. The $\overline{WAIT}$ line is then sampled in the middle of state TWA. If $\overline{WAIT}$ is HIGH, the UDC will proceed to state T_3. Otherwise additional wait states will be inserted. (See Timing Diagram 4.)

Consider what happens in a transaction when both hardware and software wait states are inserted. Each time the $\overline{WAIT}$ 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 $\overline{WAIT}$ is sampled and is HIGH. The hardware wait states may be inserted anytime during the software wait state sequence. It is important to know 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 $\overline{WAIT}$ input during T_2 for 2 cycles would insert 2 hardware wait states. Driving $\overline{WAIT}$ HIGH for 3 cycles would allow insertion of three of the four software wait states. Driving $\overline{WAIT}$ LOW for 2 more cycles would insert 2 more hardware wait states. Finally, driving $\overline{WAIT}$ HIGH would allow the final software wait state to be inserted. During this last software wait state, the $\overline{WAIT}$ 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.

DMA Transactions

There are three types of transactions performed by the Am9516A UDC: Flowthru, Flyby and Search. Figures 14 and 15 show the configurations of Flowthru and Flyby Transactions.

Flowthru Transactions

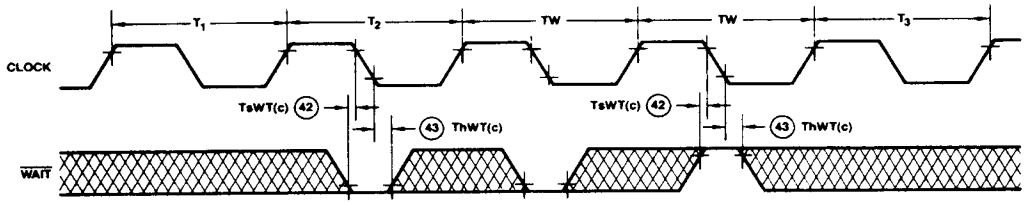
A Flowthru Transaction consists of Read and Write cycles. Each cycle consists of three states: T_1, T_2, and T_3 as shown in Timing Diagram 5. The user may select to insert software wait states through the Tag fields of the Current ARA and ARB registers. In addition, if Master Mode register bit $MM_2 = 1$, hardware wait states may be inserted by driving a LOW signal on the $\overline{WAIT}$ pin.

The M/$\overline{IO}$ and N/$\overline{S}$ lines will reflect the appropriate level for the current cycle early in T_1. The TG_6 and TG_7 bits of the current ARA and ARB registers should be programmed properly. The ALE output will be pulsed HIGH to mark the beginning of the cycle. The offset portion of the address for the peripheral being accessed will appear on $AD_0 - AD_{15}$ during T_1. The R/$\overline{W}$ and B/$\overline{W}$ lines will select a read or write operation for bytes or words. The R/$\overline{W}$, N/$\overline{S}$, M/$\overline{IO}$ and B/$\overline{W}$ lines will become stable during T_1 and will remain stable until after T_3.

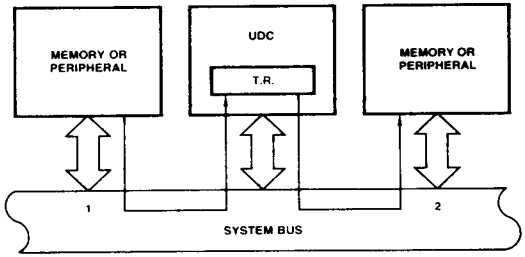
I/O address space is byte-addressed, but both 8- and 16-bit data sizes are supported. During I/O transactions the B/$\overline{W}$ output signal will be HIGH for byte transactions and LOW for word transactions. For I/O transactions, both even and odd addresses can be output; hence, the address bit output on AD_0 may be 0 or 1.

The channel can perform both I/O read and I/O write operations; the M/$\overline{IO}$ line will be LOW. During an I/O read, the $AD_0 - AD_{15}$ bus will be placed in the high-impedance state by the UDC during T_2. The UDC will drive the $\overline{DS}$ output LOW to signal the peripheral that data can be gated onto the bus. The UDC will strobe the data into its Temporary register during T_3. $\overline{DS}$ will be driven HIGH to signal the end of the I/O transaction. During I/O write, the UDC will drive the contents of the Temporary register onto the $AD_0 - AD_{15}$ bus and shortly after will drive the $\overline{DS}$ output LOW until T_3. Peripherals may strobe the data on AD bus into their internal registers on either the falling or rising edge. If the peripheral is to be accessed in a Flyby transaction also, data should be written on the rising edge of $\overline{DS}$ only.

TIMING DIAGRAM 4. $\overline{WAIT}$ Timing

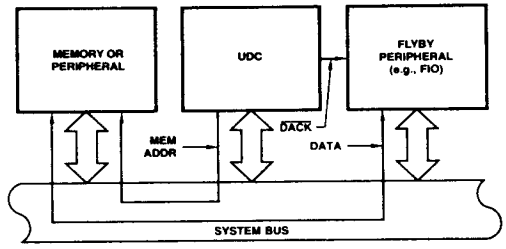


WF007680



AF003120

Figure 14. Configuration of Flowthru Transaction



AF003130

Figure 15. Configuration of Flyby 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 AD_0 is 0, the UDC 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 UDC will strobe data in from data lines $AD_0 - AD_7$. Thus, when an 8-bit peripheral is connected to the bus, its internal registers will typically be mapped at all even or all odd addresses. To simplify accesses to 8-bit peripherals, byte oriented I/O addresses are incremented/decremented by 2.

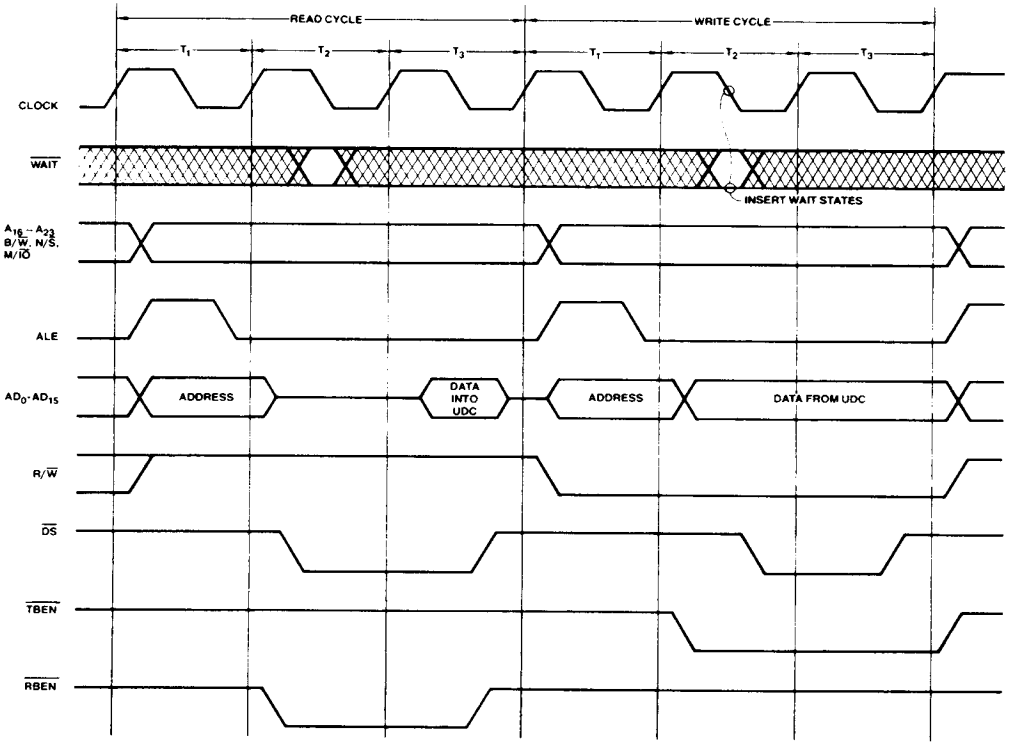
The channel can perform the I/O read and memory write operation, the memory read and I/O write operation, and the memory read and memory write operation, also. The timing for all Flowthru transactions is the same.

During chaining operations the UDC reads words from an address in System 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 UDC 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 states: 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 Tag field of the Current Address register or the Chain Address register. 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 $\overline{WAIT}$ line. The operation of Flowthru memory transactions is performed identically to the Flowthru I/O transactions. (See Timing Diagram 5.)

Flyby Transactions

Flyby Transfer and Flyby Transfer-and-Search operations are performed in a single cycle, providing a transfer rate significantly faster than that available from Flowthrus. In Flyby, operations can only be performed between memory and peripheral or between peripheral and peripheral. Memory-to-Memory operations cannot be performed in Flyby mode; these must be done using Flowthru.

TIMING DIAGRAM 5. Flowthru Transactions



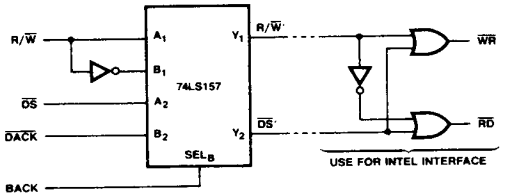
WF007540

The Flyby Transaction can only be used with peripherals having a special Flyby signal input or with external logic. This Flyby 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. For Flyby peripheral-to-peripheral transaction, if both peripherals have a Flyby input, only one (called "flyby peripheral") 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 (called "non-flyby peripheral") not connected to the channel's $\overline{DACK}$ output should be programmed in the Current ARB register when it is a destination. When the non-flyby peripheral is a source, its address should be programmed in the current ARA register. Note that a set Flip bit ($CM_4 = 1$) is for Flyby peripheral to Non-Flyby peripheral or Memory Write transaction (defined as "From Flyby Transaction"), and a clear Flip bit ($CM_4 = 0$) is for the memory or non-flyby peripheral read to Flyby peripheral transaction (defined as "To Flyby Transaction").

Transaction	CM_4	R/$\overline{W}$	Address of Memory or Non-Flyby Peripheral
To Flyby	0	HIGH	ARA
From Flyby	1	LOW	ARB

A Flyby operation is performed using three states: T_1, T_2, and T_3. During T_1 the channel pulses ALE and outputs the address information. See Timing Diagram 6. The R/$\overline{W}$ line is HIGH for "To Flyby" Transaction, and the R/$\overline{W}$ line is LOW for "From Flyby" Transaction.

The channel's M/$\overline{IO}$ and N/$\overline{S}$ lines 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 register Operation field. During state T_1 the channel drives R/$\overline{W}$ line to indicate the transaction direction. During state T_2 the channel drives both $\overline{DS}$ and $\overline{DACK}$ active. The Flyby Peripheral connected to $\overline{DACK}$ inverts the R/$\overline{W}$ signal to determine whether it is being read from or written to (see Figure 17).



$R/\overline{W}' = \overline{BACK \cdot R/\overline{W} + \overline{BACK} \cdot \overline{R/W}}$

$\overline{DS}' = \overline{BACK \cdot DACK + \overline{BACK} \cdot DS}$

$\overline{RD} = \overline{DACK \cdot R/\overline{W} \cdot BACK + DS \cdot R/\overline{W} \cdot \overline{BACK}}$

$\overline{WR} = \overline{DACK \cdot R/W \cdot BACK + \overline{DS} \cdot R/W \cdot \overline{BACK}}$

AF003140

Figure 16. Flyby Peripheral Interface

The pulsed $\overline{DACK}$ 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 is to be loaded from or driven 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 UDC's Temporary register on the LOW-to-HIGH $\overline{DS}$ transition 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 $\overline{WAIT}$ LOW if the Wait Line Enable bit in the Master Mode register is set. The $\overline{WAIT}$ line is sampled in the middle of the T_2 or TWA state.

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.

End-of-Process

The End-of-Process (EOP) interface pin is a bi-directional signal. Whenever a TC, MC or EOP termination occurs, the UDC will drive the $\overline{EOP}$ pin LOW. During DMA operations, the $\overline{EOP}$ pin is sampled by the UDC to determine if the $\overline{EOP}$ is being driven LOW by external logic. Timing Diagram 7 shows when internal EOPs are generated marking termination of all Transfers. These figures also show the point during the DMA iteration when the EOP pin is sampled. The generation of internal EOPs and sampling of external EOPs for Transfer-and-Searches follow 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 17.

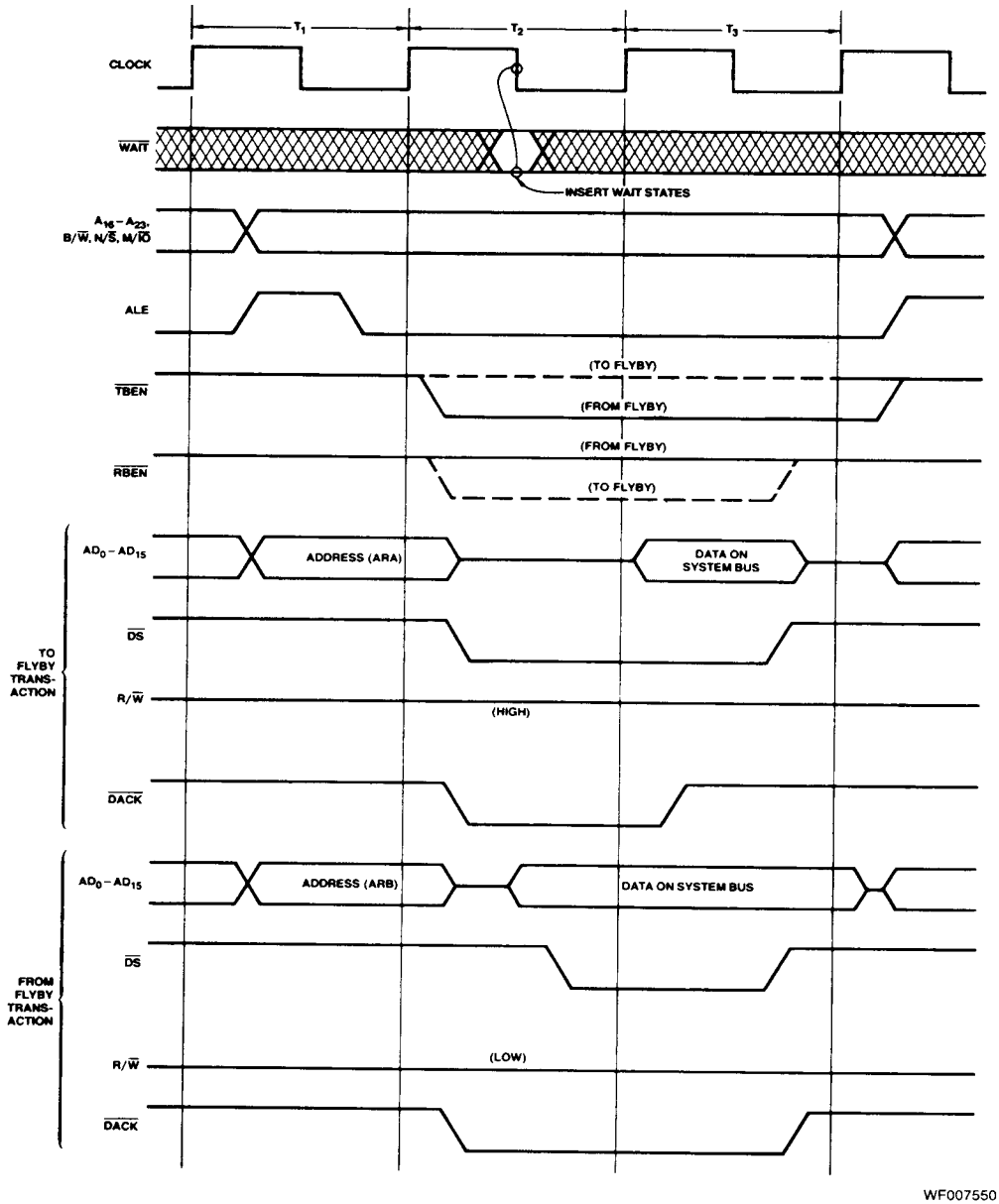
If an EOP is detected while the channel is trying to reload the Chain Address register, the new Chain Address Offset and Segment are discarded and the old address + 2 is preserved to allow inspection of the erroneous address.

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 5. 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. An as extreme example, if a channel decremented its 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 the $\overline{EOP}$ pin resulting in an EOP termination, all three of the channel's Status register completion bits would be set.

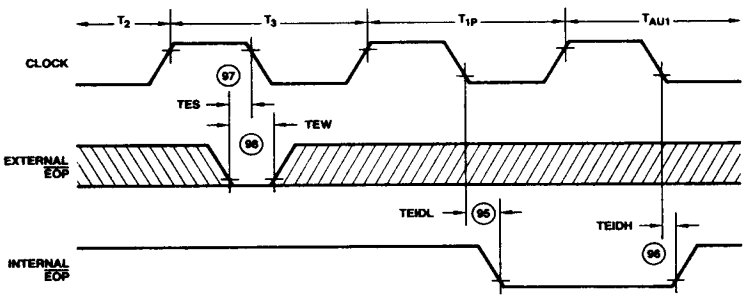
2

TIMING DIAGRAM 6. Flyby Transactions



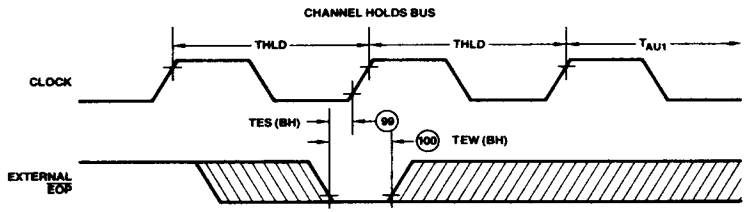
TIMING DIAGRAM 7. $\overline{EOP}$ Timing

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



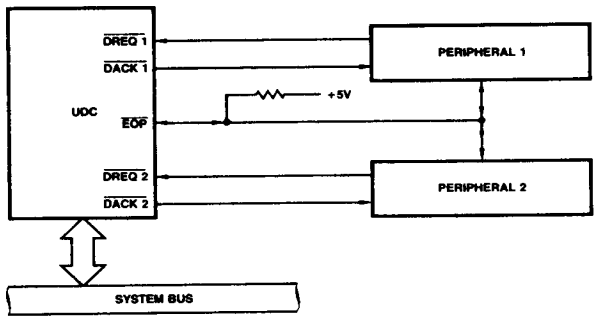
WF007560

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



WF007570

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, generated following termination of any DMA operation.
3. T_{AU1} is an auto-intialization state, generated following the TC, MC or $\overline{EOP}$ termination.



AF003150

Figure 17. $\overline{EOP}$ Connection

When a DMA operation ends, the channel can:

(a) Issue an Interrupt request (i.e., setting the IP or SIP bit of the channel's Status register);

(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 $\overline{EOP}$), the user can choose which action or actions are to be taken. If no reloading is selected for the type of termination that occurred, 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. The priorities of those actions are Interrupt request first, Base-to-Current reloading second, and then chaining. The Interrupt cannot be serviced unless the UDC has relinquished the bus.

Interrupts

To allow the UDC 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 Interrupt structure and then elaborate on the additional Interrupt queuing capability of the UDC.

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, 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 operation — the cause of the Interrupt.

A peripheral can have one or more sources of Interrupt. Each Interrupt source has two bits that control how it generates Interrupts. These bits are a Channel Interrupt Enable bit (CIE) and an Interrupt Pending bit (IP). On the UDC, each channel is an Interrupt source. The two Interrupt control bits are located in bits CM_{15} and 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 is one bit (MM_3) in the Master Mode register used for controlling Interrupt behavior for the whole device.

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 UDC 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 UDC 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 7. 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 registers.

Note that the SIP bit is transferred to the IP bit when IP is cleared by the host CPU. Whenever CIE is set, $\overline{INT}$ will go LOW as soon as IP is set.

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 in 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 (i.e., TAU_1 through TAU_4). 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 system bus once an Interrupt acknowledge is received, even if it immediately afterward will relinquish the bus because no hardware/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 the 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, 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.

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 to 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 18 shows a list of UDC commands. The commands are executed immediately after being written by the host CPU into the UDC's Command register (Figure 19). A description of each command follows.

Reset (00)

This command causes the UDC to be set to the same state generated by a Hardware Reset. The Master Mode register is set to all zeros; the CIE, IP and SIP bits are cleared; the NAC and CA bits in each channel's Status register are set; and the channel activity is forbidden. The Chain Address must be programmed since its state may be indeterminate after a Reset. The lockout preventing channel activity is cleared by issuing a Start Chain command.

Start Chain Channel 1/Channel 2 (A_0/A_1)

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 fetched Reload word is all zeros. This command will only be honored if the Chain Abort (CA) bit and the 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.

When the Waiting For Bus (WFB) bit of Status register is set, if the "Start Chain" command is issued, the channel will honor the command after one DMA iteration. It is nearly impossible for the CPU to issue a command when WFB = 1 and the UDC is enabled.

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, or, IP Channel 1	001E	XP10	32*
Set CIE, or, IP Channel 2	001E	XP11	33*
Clear CIE, or, IP Channel 1	001E	XP00	30*
Clear CIE, or, IP Channel 2	001E	XP01	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. P = Set to 1 to perform set/clear on IP; Clear to 0 for no effect on IP.
3. X = "don't care" bit. This bit is not decoded and may be 0 or 1.

Figure 18. UDC Command Summary

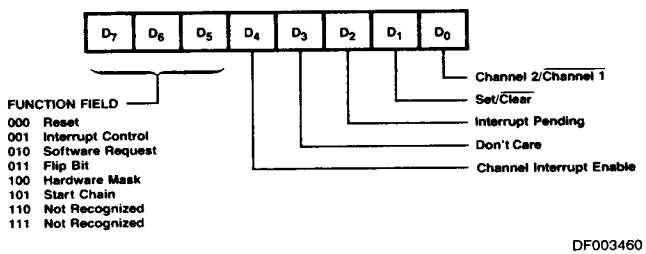


Figure 19. Command Register

Software Request Channel 1/Channel 2 (Set: 42/43, Clear: 40/41)

This command sets or clears 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 cleared, 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.

Set/Clear Hardware Mask 1/Mask 2 (Set: 82/83; Clear: 80/81)

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. Transitions occurring while the Hardware Mask bit is set will be stored and serviced when the Hardware Mask is cleared, assuming the Channel has not chained. The UDC 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 service is cleared.

Timing Diagrams 1 and 2 show the minimum times when a new $\overline{DREQ}$ can be applied if it is to be serviced by the new DMA operation. Note in Diagram 1 the notation of First iteration and Last iteration. This means, for example, $\overline{DREQ}$ may be asserted during the write cycle T_1 of a Flowthru transaction, but may never be asserted during T_1 of a Flyby transaction because Flyby is done in one iteration.

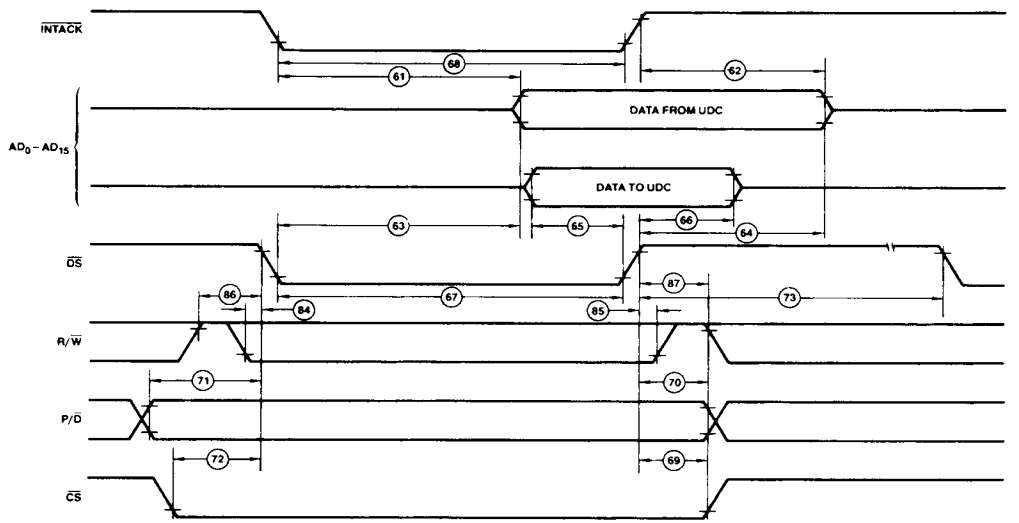
Set/Clear CIE, and IP Channel 1/Channel 2 (see Figure 18)

This command allows the user to either set or clear any combination of the CIE 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 the current Vector and Status. The IP bit is cleared to facilitate an efficient conclusion to the processing of an interrupt.

Set/Clear Flip Bit Channel 1/Channel 2 (Set: 62/63; Clear: 60/61)

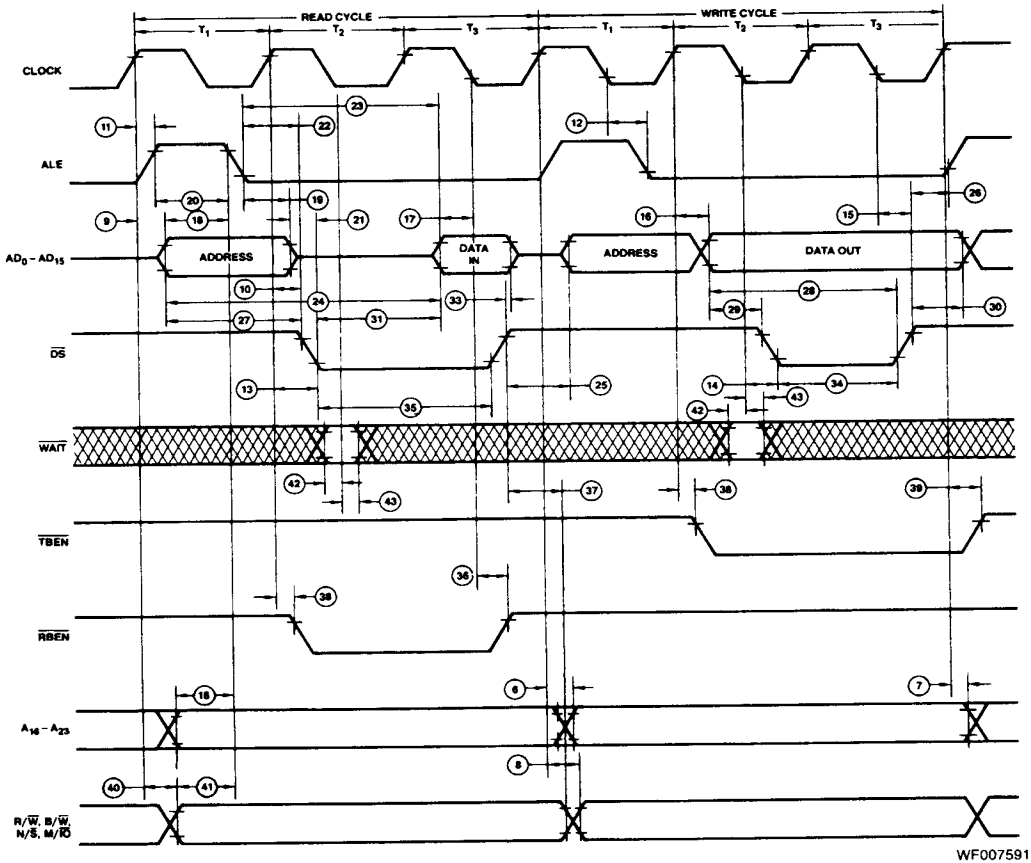
The Flip Bit in the selected channel's Mode register can be cleared and set by this command. This allows the user to reverse the source and destination and thereby reverse the data transfer direction without reprogramming the channel. This command will be most useful when repetitive DMA operations are 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.

TIMING DIAGRAM 8. AC Timing when UDC is a Bus Slave



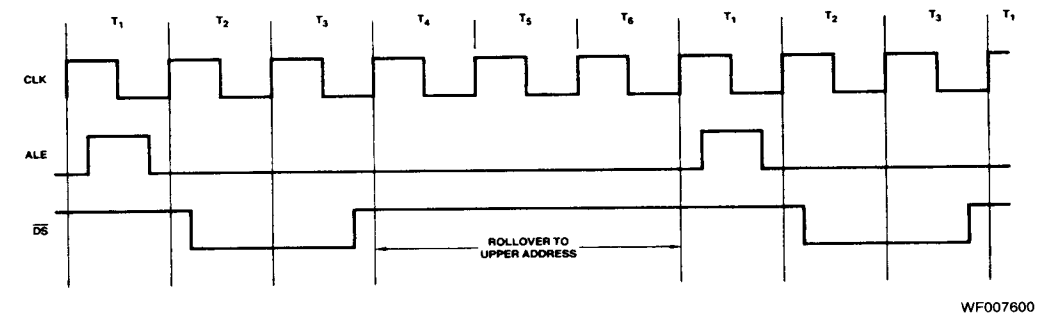
WF007580

TIMING DIAGRAM 9. AC Timing when UDC is a Bus Master



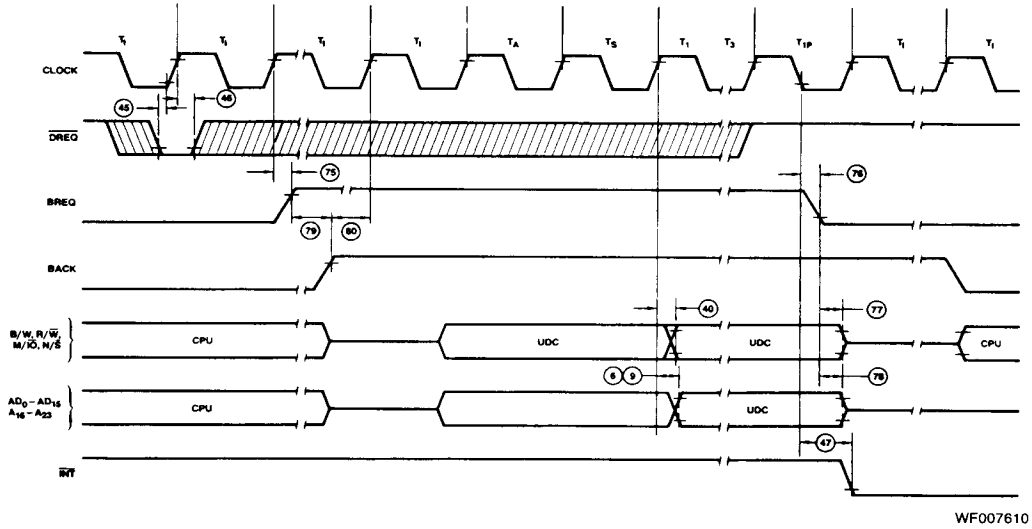
WF007591

TIMING DIAGRAM 10. Upper Address Rollover Timing



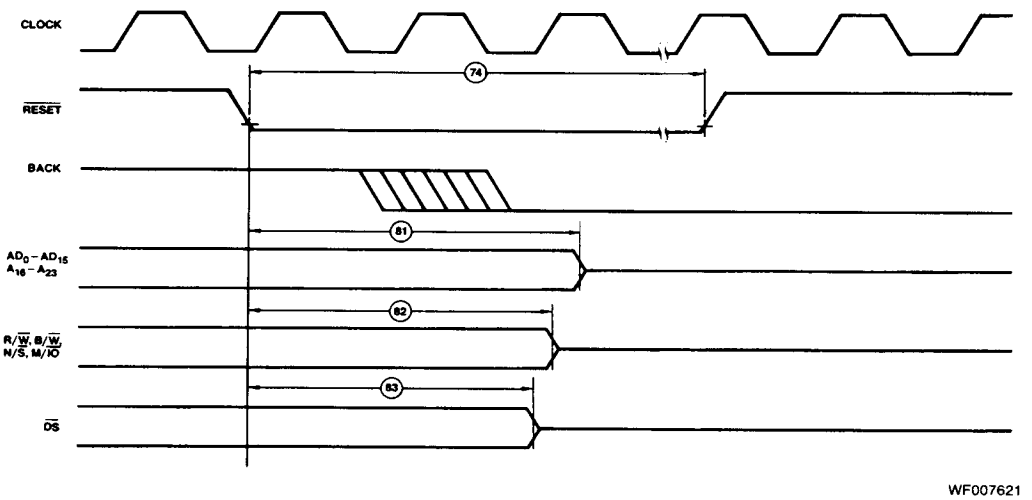
WF007600

TIMING DIAGRAM 11. Bus Exchange Timing

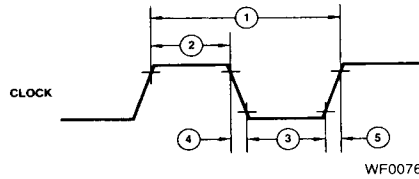


Notes: 1. Under no circumstance can BACK be removed prior to BREQ.
2. One extra ALE occurs each time the 9516 releases the bus. No $\overline{DS}$ accompanies it, so this should not present a problem.

TIMING DIAGRAM 12. Reset Timing

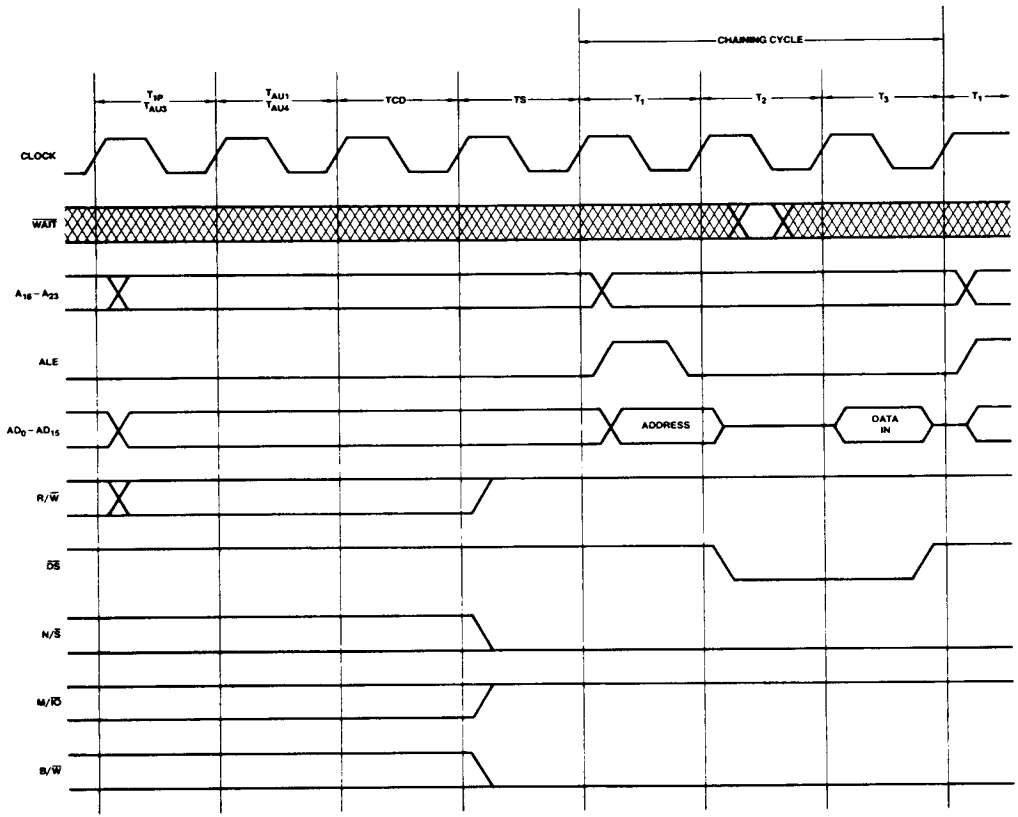


TIMING DIAGRAM 13. Clock Waveform



WF007630

TIMING DIAGRAM 14. Timing During Chaining



WF007640

2

APPLICATIONS INFORMATION

Figures 20a and 20b show the configuration of an Am9516A UDC and an Am8086 microprocessor on the same board. Figure 21 shows a configuration for them when the Am9516A UDC is on a different board. The configuration of an Am9516A UDC to 68000 CPU interface is shown in Figure 22. An example of an Am8086 initialization program is shown in Figure 23. Figure 24 shows the reload table for chaining. The details of the Programmable Array Logic (PAL*) for those interfaces are described in Appendix B.

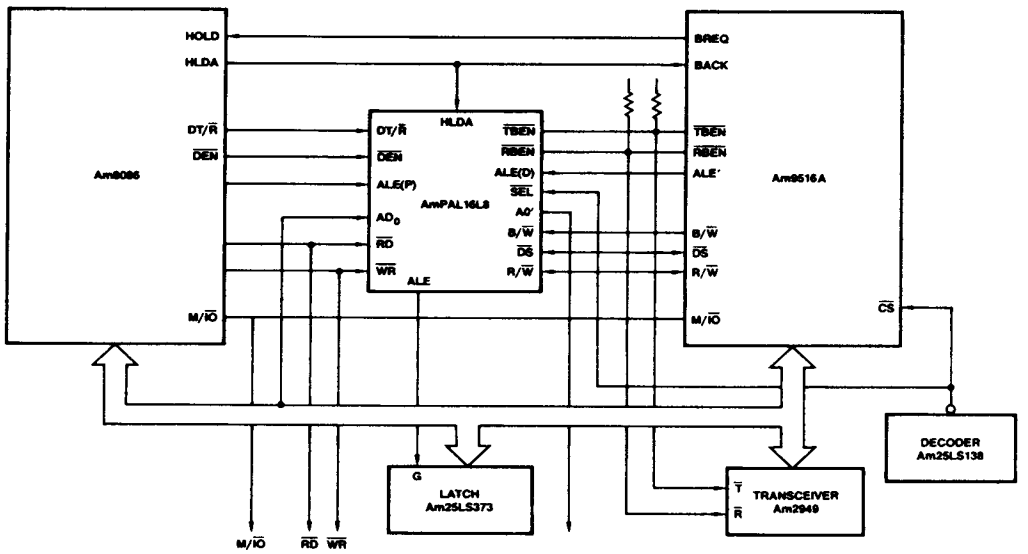


Figure 20a. Am9516A UDC to Am8086 CPU Interface (Minimum Mode)

AmPAL16L8 PALASM FILE

PAL16L8

Pat 001

Am9516A to Am8086 min mode interface chip

Advanced Micro Devices

NC ALED ALEP HLDA BW AD_0 DT/DEN/SEL GND

NC/RBEN/RD ALE A_0/RW/DS/WR/TBEN V_{CC}

If (/HLDA) DS = RD + WR

If (/HLDA) RW = DT

If (/HLDA) TBEN = /DT*/SEL*DEN

If (/HLDA) RBEN = DT*/SEL*DEN

If (HLDA) RD = /RW * DS

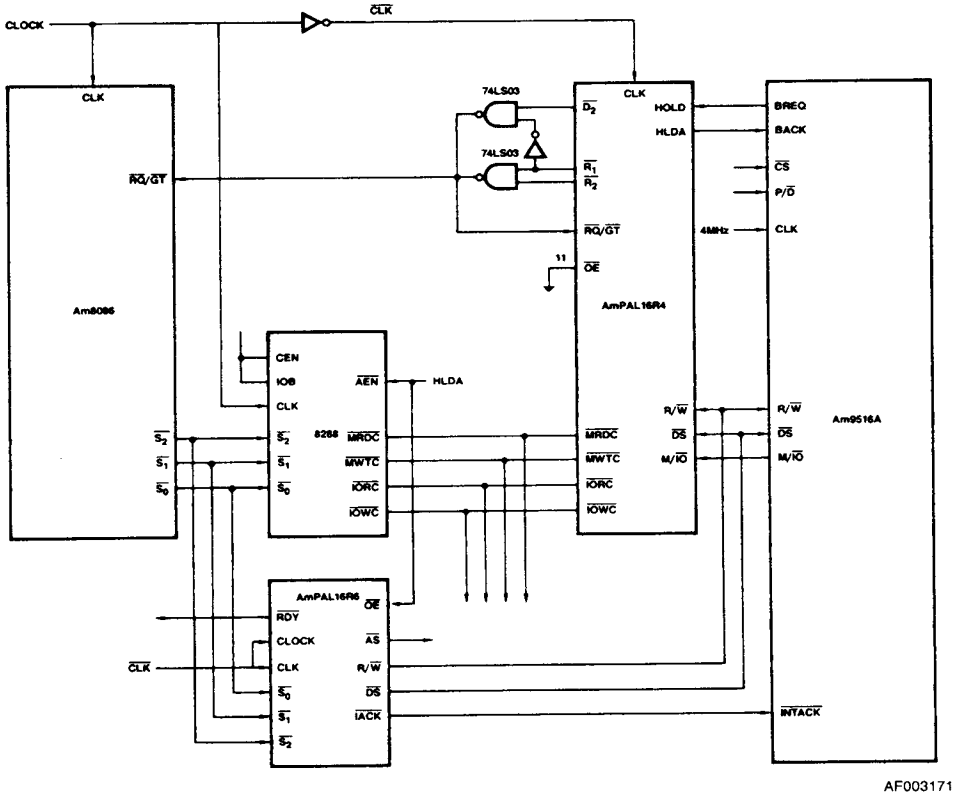
If (HLDA) WR = RW * DS

ALE = /ALEP * /ALED

A_0 = /AD_0*/BW*HLDA*ALED +
/AD_0*BW*HLDA*ALED +
/AD_0*/HLAD*ALEP + A_0*/ALEP + A_0*/ALED

DESCRIPTION

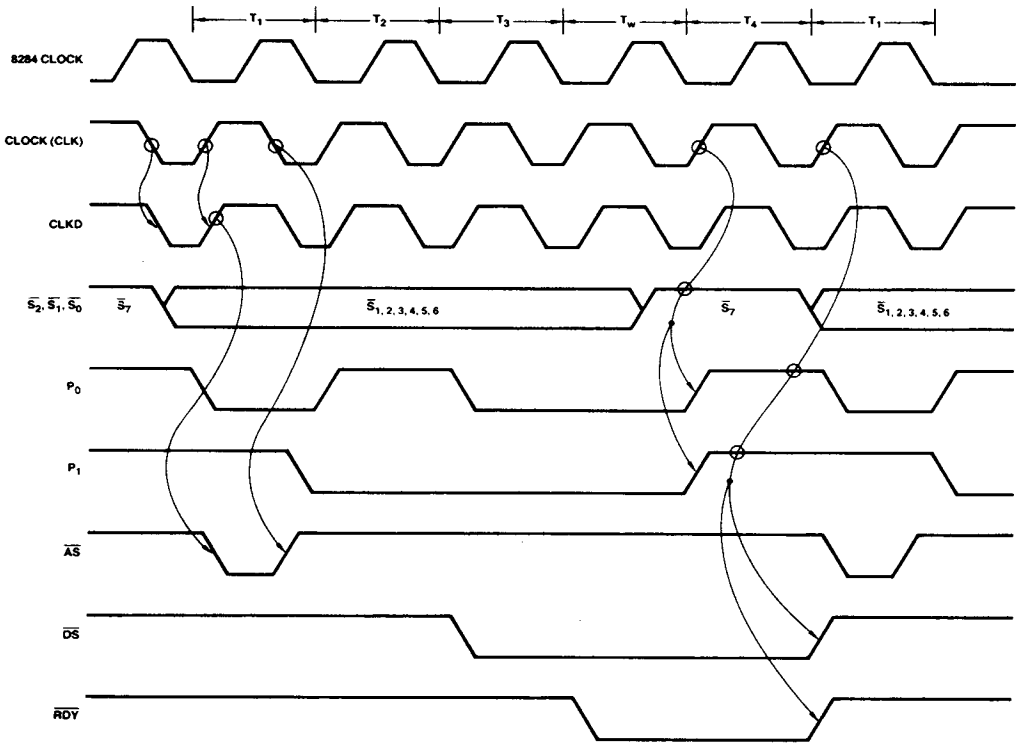
This PAL converts the control signals to interface the Am8086 in min mode to the Am9516A DMA controller. Another example shows how this is done in max mode.



AF003171

Figure 20b. Am9516A UDC to 8086 CPU Interface (Maximum Mode)

Timing Diagram of AmPAL16R6



WF007650

AmPAL16R6 PALASM FILE

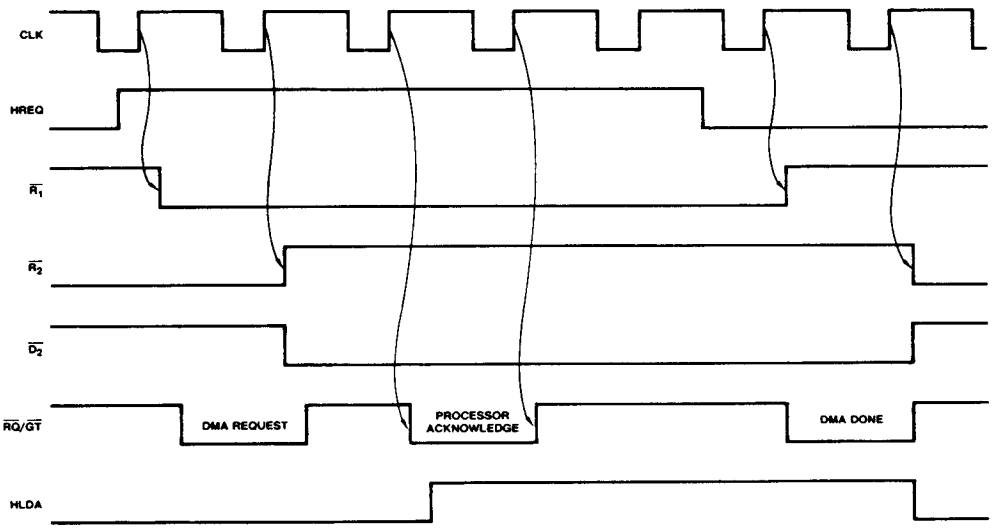
```
AmPAL16R6
PAT003
Am8086 to AmZ85XX Peripheral Interface
Advanced Micro Devices

CLOCK RESET CLK/S0/S1/S2 NC NC NC GND
/OE/AS/P1/RW/DS/PO/LACK/RDY CLKD VCC

P0: = /RESET*S0*/P0*/P1  +
      /RESET*S1*/P0*/P1  +
      /RESET*S2*/P0*/P1  +
      /RESET*S0*P1       +
      /RESET*S1*P1       +
      /RESET*S2*P1

P1: = /RESET*P0*/P1      +
      /RESET*P1*S0       +
      /RESET*P1*S1       +
      /RESET*P1*S2

DS: = /IACK*/P0*P1*S0*/S1*S2 +
      /IACK*/P0*P1*/S0*S1*S2 +
      IACK*S0*S1*S2      +
      DS*P0*P1

RW: = S0*/S1

IACK: = /RESET*S0*S1*S2 + IACK*P0*P1*/DS +
        IACK*/P0*/P1

RDY: = /RESET*S0*/S1*S2*P0*P1  +
       /RESET*/S0*S1*S2*P0*P1  +
       /RESET*DS*RDY*P0*P1

/CLKD = CLK

AS = /CLKD*P0*/P1*/IACK*CLK
```

DESCRIPTION

This PAL translates Am8086 bus signals into compatible signals for the Am9516A. It is also applicable to AmZ85XX peripherals by altering/RW and /DS to/RD and /WR. One flip-flop is available to give the necessary delay to the falling edge of/WR.

Note: The CLK signal must be externally inverted for this design.
A >

AmPAL16R4 Timing



WF007660

AmPAL16R4 PALASM FILE

B > Type Am9516A PAL
PAL16R4
Am8086 to Am9516A interface
Advanced Micro Devices
CLK/RQGT HOLD NC NC NC/RW/DS MIO GND
/OE/MWTC/MRDC HLDA/D_2/R_2/R_1/IOWC/IORC V_{CC}

If (HLDA) IORC = /MIO*DS*/RW
If (HLDA) IOWC = /MIO*DS*RW
If (HLDA) MRDC = MIO*DS*/RW
If (HLDA) MWTC = MIO*DS*RW

R_1: = HOLD
R_2: = /R_1
D_2: = R_1
/HLDA: = /R_1 + /D_2*/HLDA + /RQGT*/HLDA

DESCRIPTION

This device converts the min mode signals HOLD and HLDA to the max mode /RQGT protocol. Additionally, it generates the 8288 equivalent control outputs /MRDC, /MWTC, /IORC, and /IOWC. This PAL was used to connect the Am9516A to the Am8086 in max mode.

B >

Note: If HOLD is taken away prior to grant pulse, design will not work correctly because the release pulse will overlap the grant pulse.

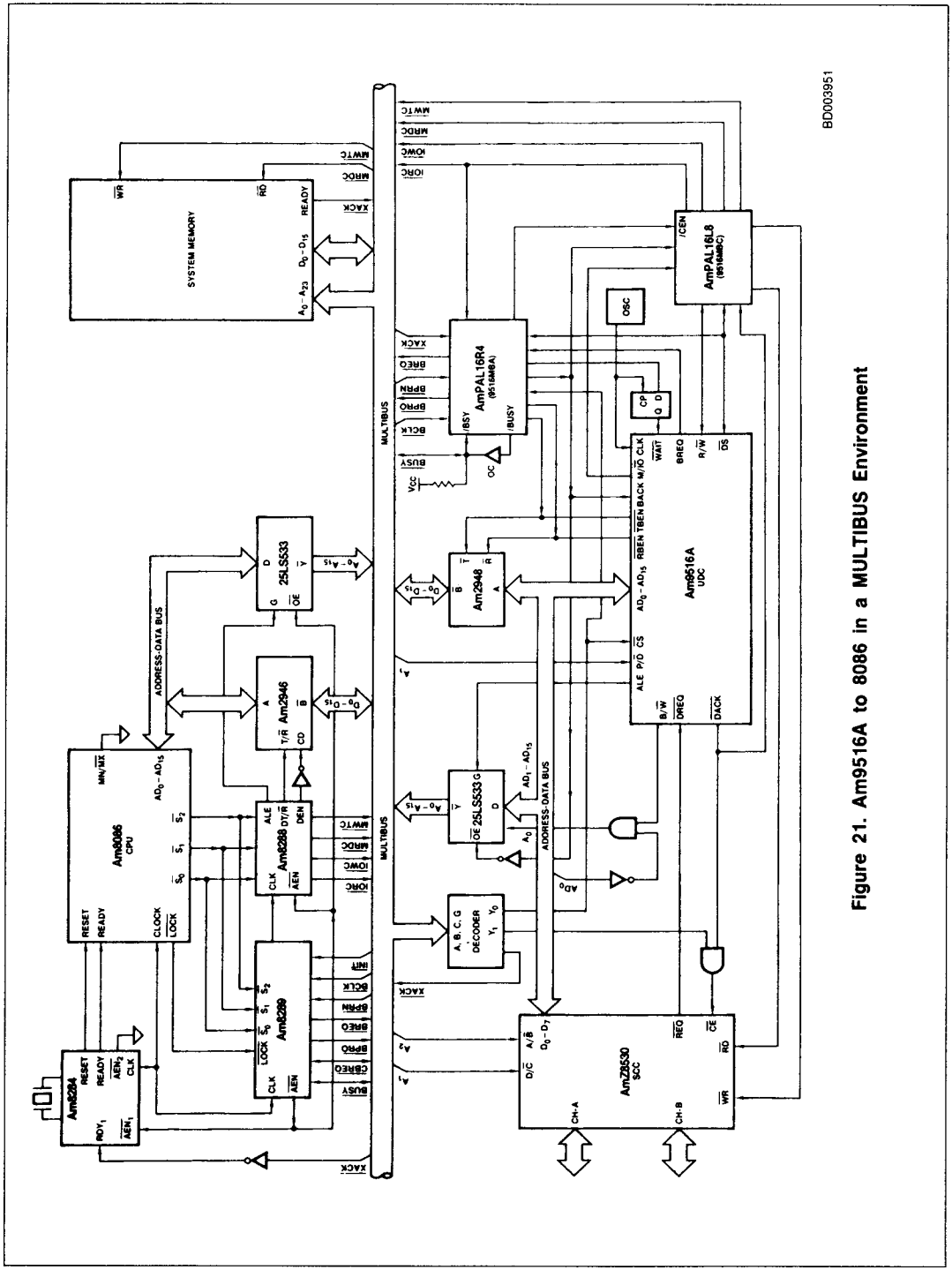
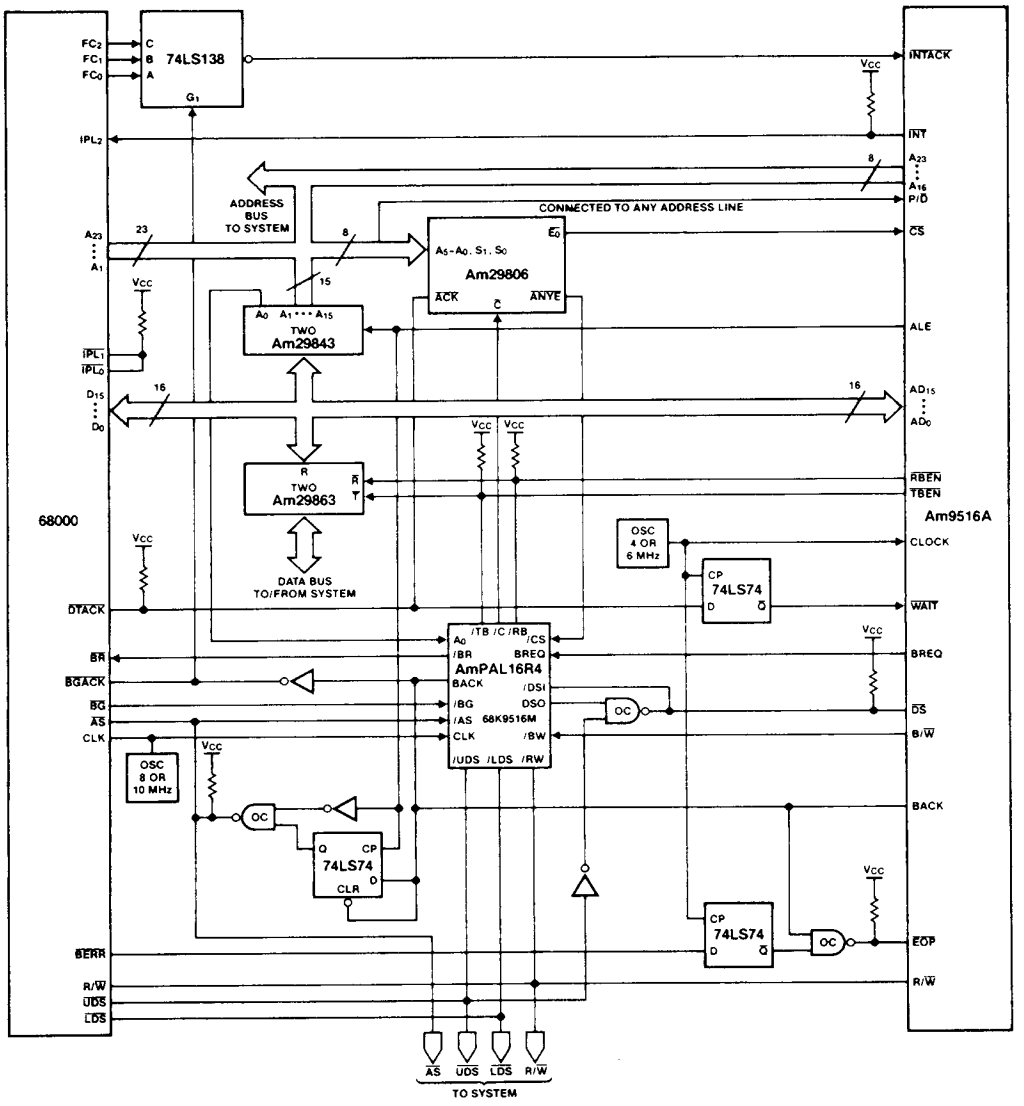


Figure 21. Am9516A to 8086 in a MULTIBUS Environment



BD003941

Figure 22. The Am9516A UDC to 68000 CPU Interface

```
PAL16R4                          PAL DESIGN SPECIFICATION
PAT006                              JOE BRCICH 9/01/83
68000 TO Am9516A INTERFACE WITH DATA HOLD CORRECTION
ADVANCED MICRO DEVICES

CLK   RW    AO    BREQ   /BG    /DSI    /AS    /BW   /CS
/OE   /LDS  /UDS  DSO    /C     BACK    /BR    /TB   /RB

IF (/BACK) RB  = /CS * RW * UDS      +
                 /CS * RW * LDS
IF (/BACK) TB  = /CS * /RW
IF (BACK) UDS  = DSI * /A0 * /BW     +
                 BW * DSI
IF (BACK) LDS  = DSI * A0 * /BW      +
                 BW * DSI
BR := BREQ * BG * BR * AS            +
      BREQ * /BG * /BACK
/BACK := /BREQ                          +
         /BREQ * /BG                    +
         /BREQ * AS                     +
         /BREQ * /BACK                  +
         /BG * /BACK                    +
         AS * /BACK
C := UDS * /BACK                     +
     LDS * /BACK
/DSO := BACK                         +
        /BACK * /RW * C
```

DESCRIPTION

IF BREQ*BACK IS TRUE THE Am9516A HAS THE BUS, OTHERWISE THE 68000 HAS THE BUS. THIS PAL CONNECTS THE Am9516A TO THE 68000 WITH ONE WAIT STATE DURING WRITES WHILE SHORTENING /DS TO ACHIEVE PROPER DATA HOLD TIME. IT ALSO CONVERTS THE BUS EXCHANGE PROTOCOL INTO 68000 FORMAT. THIS DESIGN ASSUMES NO OTHER BUS MASTERS IN THE SYSTEM. /RB AND /TB CONTROL THE TRANCEIVERS WHEN CPU IS BUS MASTER. /CS MUST BE FUNCTION OF ALL DEVICES CONNECTED TO THE CPU BUS NOT JUST THE Am9516A /CS AS SHOWN HERE.

The /CS to /DS set-up time of 30ns is met in the following ways:

(1) During a read cycle the only effect from not meeting this set-up time is that the data valid access time from the Am9516A will be delayed by a proportional amount. Since the minimum /DS Low width from the 10-MHz 68000 (during a read) is 193ns and the minimum /DS Low width to the Am9516A is 150ns, we have 43ns margin not counting gate delays which will further increase this margin.

(2) During a write cycle this is not an issue since the /DS comes later and is stretched longer due to the Wait state.

AmPAL16R4 68K9516M PALASM File

```
PAL16L8                    PAL DESIGN SPECIFICATION 9516MBC
PAT 003                          JOE BRCICH 26 JULY 84
MULTIBUS CONTROL FOR Am9516A
ADVANCED MICRO DEVICES

BACK   MIO    NC     NC     /DACK  NC     NC     NC     /CEN   GND
NC     /RD    /IORC  /DS    /MWTC  /MRDC  /IOWC  /RW    /WR    VCC

IF (BACK) IORC  = /MIO * DS * /RW * CEN

IF (BACK) IOWC  = /MIO * DS * RW * CEN

IF (BACK) MRDC  = MIO * DS * /RW * CEN

IF (BACK) MWTC  = MIO * DS * RW * CEN

RD  = DACK * RW * BACK +
      IORC * /BACK

WR  = DACK * /RW * BACK +
      IOWC * /BACK

IF (/BACK) DS  = IORC + IOWC

IF (/BACK) RW  = IOWC
```

DESCRIPTION

THIS PAL CONVERTS MULTIBUS SIGNALS INTO Am9516A COMPATIBLE SIGNALS AND VICE VERSA. IT ALSO SUPPORTS THE 8530 IN FLYBY MODE.

MULTIBUS Control for Am9516A (AmPAL16L8)

2

```
PAL16R4                    PAL DESIGN SPECIFICATION 9516MBA
PAT 004                          JOE BRCICH 30 July 84
MULTIBUS ARBITER FOR Am9516A
ADVANCED MICRO DEVICES

/BCLK  /XACK  BRQ    /BSY   /BPRN  /DS    NC     /IORC  /CS    GND
/OE    /RBEN  /TBEN  BACK   /CEN   /BREQ  /BUSY  /BPRO  /WAIT  VCC

IF (/BACK) TBEN  = IORC * CS

IF (/BACK) RBEN  = /IORC * CS

WAIT  = /XACK * BACK

BREQ  := BRQ

BPRO  = /BRQ * BPRN

/BACK  := /BUSY

BUSY  := BREQ * BPRN * /BSY * /BUSY  +
         BREQ * BUSY * BPRN          +
         BREQ * BUSY

CEN  := BACK
```

DESCRIPTION

/CEN DELAYS THE COMMANDS TO MEET THE MULTIBUS REQUIREMENT THAT ADDRESS AND DATA BE VALID AT LEAST 50NS PRIOR TO CONTROL ACTIVE. /IOWC WAS NOT USED SINCE USING /IORC IMPROVES HOLD TIME. THIS DESIGN DOES NOT SUPPORT THE /CBRQ FUNCTION.

MULTIBUS Arbiter for Am9516A (AmPAL16R4)

```
•                 •
•                 •
•                 •
B0 38             MOV     AL,38H      ;LOADING POINTER OF MASTER
E6 12             OUT     12H         ;MODE REGISTER
B8 07 00          MOV     AX,007H     ;LOADING MMR CODE
E7 10             OUTW    10H         ;
B0 26             MOV     AL,26H      ;LOADING POINTER OF CHAIN
E6 12             OUT     12H         ;ADDRESS REGISTER'S SEGMENT
B8 00 00          MOV     AX,0000H    ;LOADING SEGMENT OF CAR-1
E7 10             OUTW    10H         ;
B0 22             MOV     AL,22H      ;LOADING POINTER OF CHAIN
E6 12             OUT     12H         ;ADDRESS REGISTER'S OFFSET
B8 20 10          MOV     AX,1020H    ;LOADING OFFSET OF CAR-1
E7 10             OUTW    10H         ;
B0 2C             MOV     AL,2CH      ;LOADING POINTER OF COMMAND
E6 12             OUT     12H         ;REGISTER
B0 A0             MOV     AL,A0H      ;LOADING "START CHAIN" COMMAND
E6 10             OUT     10H         ;ISSUING "START CHAIN" COMMAND
•                 •
•                 •
•                 •
```

Figure 23. Initialization Program for 8086 CPU

Notes: The P/$\overline{D}$ input is connected to A1 line; $\overline{CS}$ is decoded from A7 through A4 (all O).

ADDRESS	0	2	4	6	8	A	C	E
1000	0000	1020	0000	1020	0007	0005	0006	0005
1010	0002	AAAA	0009	00A0	0004	0042	0042	0001
1020	03FF	0000	1F00	0000	1060	0010	0000	1F00
1030	0000	1080	0012	0000	FFFF	0001	0000	8020
1040	0000	1020	1111	1111	0000	FFFF	2004	0000
1050	0010	0000	0000	1020	0018	1020	2222	1007
1060	CACA	CACA	CACA	CACA	CACA	CACA	CACA	CACA
1070	CACA	CACA	CACA	CACA	CACA	CACA	CACA	CACA

Reload Word

TB000084

Figure 24. Reload Table for Chaining

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
V_{CC} with Respect to V_{SS} ..... −0.5 V to +7.0 V
All Signal Voltages with Respect to V_{SS} −0.5 V to +7.0 V

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

Commercial (C) Devices
Temperature (T_A) ..... 0 to +70°C
Supply Voltage (V_{CC}) ..... 5 V ± 5%

Military (M) Devices
Temperature (T_C) ..... −55 to 125°C
Supply Voltage (V_{CC}) ..... 5 V ± 10%

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

DC CHARACTERISTICS over operating range (for APL Products, Group A, Subgroups 1, 2, 3 are tested unless otherwise specified)

Parameters	Description	Test Conditions		Min	Max	Units
V_{CH}	Clock Input High Voltage	Driven by External Clock Generator		3.8	V_{CC} +0.3	Volts
V_{CL}	Clock Input Low Voltage	Driven by External Clock Generator		−0.5	0.45	Volts
V_{IH1}	Input High Voltage	All Pins Except 2, 36, 37, 38, 47, 48		2.0	V_{CC} +0.3	Volts
V_{IH2}	Input High Voltage	Pins 2, 36, 37, 38, 47, 48		2.2	V_{CC} + 0.3	Volts
V_{IL}	Input Low Voltage			−0.5	0.8	Volts
V_{OH1}	Output High Voltage	I_{OH} = −250μA (Except Pins 1, 32, 33, 38)		2.4		Volts
V_{OH2}	Output High Voltage	I_{OH} = −200 μA, Pins 1, 32, 33, 38		2.0		Volts
V_{OL}	Output Low Voltage	I_{OL} = 3.2 mA			0.45	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	MIL: T_C = −55°C		350	mA
		T_A = 70°C	MIL: T_C = +125°C		200	mA
CCLK	Input Capacitance (Clock)	Unmeasured pins returned to ground. f = 1 MHz over specified temperature range.			25*	pF
C_{IN}	Input Capacitance (Except Pin 46)	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

*Guaranteed by design; not tested.

2

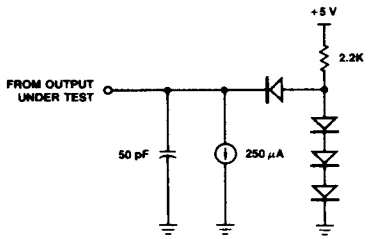
Standard Test Conditions

Commercial

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:

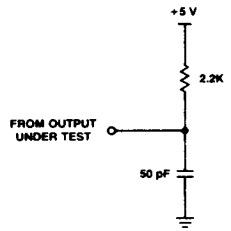
$+4.75 \leqslant V_{CC} \leqslant +5.25$ V
GND = 0 V
$0°C \leqslant T_A \leqslant +70°C$

Standard Test Load



TC002011

Open-Drain Test Load



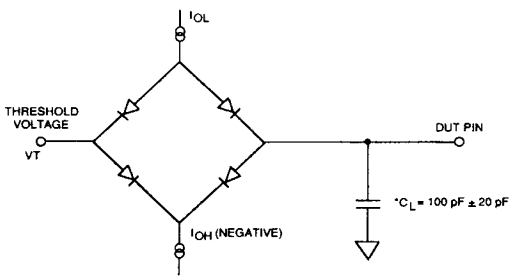
TC002021

Standard Test Conditions

Military

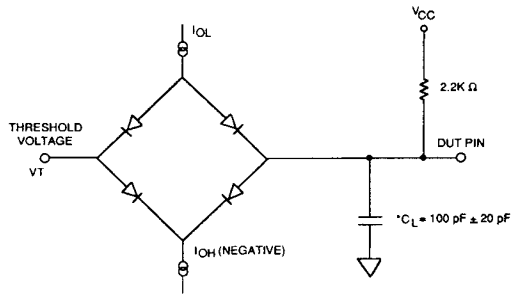
The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the referenced pin. Standard conditions are as follows:

$+4.75 \leqslant V_{CC} \leqslant +5.25$ V
GND = 0 V
$-55 \leqslant T_C \leqslant +125°C$



TC004391

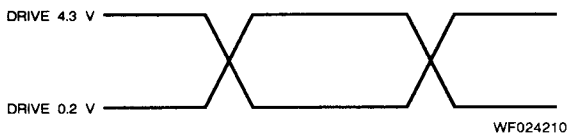
A. Standard Dynamic Load



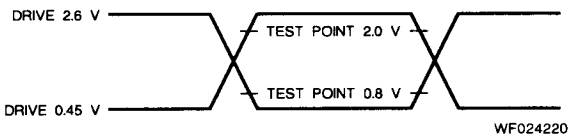
TC004381

B. Open-Drain Dynamic Load

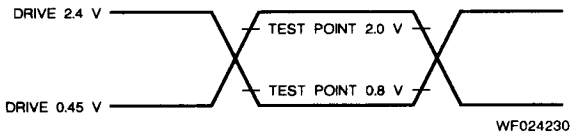
SWITCHING TEST WAVEFORMS



A. External CLOCK Generator



B. BACK, DREQ1, DREQ2, RESET, INTACK, and EOP only



C. ALL pins except BACK, DREQ1, DREQ2, RESET, INTACK, and EOP.

SWITCHING CHARACTERISTICS over **COMMERCIAL** operating range unless otherwise specified
TIMING FOR UDC AS BUS MASTER

			Preliminary								
			4 MHz		6 MHz		8 MHz		10 MHz		
Number	Parameters	Description	Min	Max	Min	Max	Min	Max	Min	Max	Units
1	TcC	Clock Cycle Time	250	2000	165	2000	125	2000	100	2000	ns
2	TwCh	Clock Width (HIGH)	105	1000	70	1000	55		45		ns
3	TwCl	Clock Width (LOW)	105		70		55		45		ns
4	TfC	Clock Fall Time		20		10		5		5	ns
5	TrC	Clock Rise Time		20		15		10		5	ns
6	TdC(AUv)	Clock RE to Upper Address (A_{16}-A_{23}) Valid Delay		90		80		60		50	ns
7	ThC(AUv)	Clock RE to Upper Address Valid Hold Time	5		5		5		5		ns
8	TdC(ST)	Clock RE to R/$\overline{W}$ and B/$\overline{W}$ Valid Delay		110		90		60		50	ns
9	TdC(A)	Clock RE to Lower Address (A_0-A_{15}) Valid Delay		90		90		60		50	ns
10	TdC(Az)	Clock RE to Lower Address (A_0-A_{15}) Float Delay		60		60		50		40	ns
11	TdC(ALr)	Clock RE to ALE RE Delay		70		60		50		30	ns
12	TdC(AL)	Clock FE to ALE FE Delay		70		60		50		40	ns
13	TdC(DS)	Clock RE to $\overline{DS}$ (Read) FE Delay		60		60		50		40	ns
14	TdC(DSf)	Clock FE to $\overline{DS}$ (Write) FE Delay		60		60		50		45	ns
15	TdC(DSr)	Clock FE to $\overline{DS}$ RE Delay		60		60		50		40	ns
16	TdC(DO)	Clock RE to Data Out Valid Delay		90		90		65		60	ns
17	TsDI(C)	Data in to Clock FE Set-up Time	20		15		10		10		ns
18	TdA(AL)	Address Valid to ALE FE Delay	50		35		20		20		ns
19	ThAL(A)	ALE FE to Lower Address Valid Hold Time	60		40		30		30		ns
20	TwAL	ALE Width (HIGH)	80		60		45		40		ns
21	TdAz(DS)	Lower Address Float to $\overline{DS}$ LOW Delay	0		0		0		0		ns
22	TdAL(DS)	ALE FE to $\overline{DS}$ (Read) FE Delay	75		35		35		35		ns
23	TdAL(DI)	ALE FE to Data in Required Valid Delay		300		215		190		150	ns
24	TdA(DI)	Address Valid to Data in Required Valid Delay		410		305		240		190	ns
25	TdDS(A)	$\overline{DS}$ RE to Address Active Delay	80		45		30		20		ns
26	TdDS(Al)	$\overline{DS}$ RE 10 ALE RL Delay	75		40		40		35		ns
27	TdA(DS)	Address Valid to $\overline{DS}$ (Read) FE Delay	160		110		90		70		ns
28	TdDO(DSr)	Data Out Valid to $\overline{DS}$ RE Delay	230		150		125		80		ns
29	TdDO(DSf)	Data Out Valid to $\overline{DS}$ FE Delay	55		35		20		15		ns
30	ThDS(DO)	$\overline{DS}$ RE to Data Out Valid Hold Time	85		45		40		25		ns
31	TdDS(DI)	$\overline{DS}$ (Read) FE to Data in Required Valid Delay		205		155		125		100	ns
33	ThDI(DS)	$\overline{DS}$ RE to Data in Hold Time	0		0		0		0		ns
34	TwDSmw	$\overline{DS}$ (Write) Width (LOW)	185		110		105		80		ns
35	TwDSmr	$\overline{DS}$ (Read) Width (LOW)	275		220		160		130		ns
36	TdC(RBr)	Clock FE to $\overline{RBEN}$ RE Delay*		70		65		50		30	ns
37	ThDS(ST)	$\overline{DS}$ RE to B/$\overline{W}$, N/$\overline{S}$, R/$\overline{W}$ and M/$\overline{IO}$ Valid Hold Time	70		45		40		25		ns
38	TdC(TRf)	Clock RE to $\overline{TBEN}$ or $\overline{RBEN}$ FE Delay		60		60		50		35	ns
39	TdC(TRr)	Clock RE to $\overline{TBEN}$ RE Delay		60		60		45		45	ns
40	TdC(ST)	Clock RE to M/$\overline{IO}$ and N/$\overline{S}$ Valid Delay		90		75		65		50	ns
41	TdS(AL)	R/$\overline{W}$, M/$\overline{IO}$, B/$\overline{W}$ and N/$\overline{S}$ Valid to ALE FE Delay	60		35		20		20		ns
42	TsWT(C)	$\overline{WAIT}$ to Clock FF Set-up Time	20		20		10		10		ns
43	ThWT(C)	$\overline{WAIT}$ to Clock FE Hold Time	20		20		35		35		ns
44	TwDRQ	$\overline{DREQ}$ Pulse Width (Single Transfer Mode)	20		20		20		20		ns
45	TsDRQ(C)	$\overline{DREQ}$ Valid to Clock RE Set-up Time	60		50		30		20		ns
46	ThDRQ(C)	Clock RE to $\overline{DREQ}$ Valid Hold Time	20		20		20		20		ns
47	TdC(INTf)	Clock FE to $\overline{INT}$ FE Delay		150		150		105		105	ns

*These must not occur simultaneously.

Note: RE = rising edge
FE = falling edge

SWITCHING CHARACTERISTICS over **MILITARY** operating range (for APL Products, Subgroups 9, 10, 11 are tested unless otherwise specified)

TIMING FOR UDC AS BUS MASTER

					Preliminary		
			4 MHz		6 MHz		
Number	Parameters	Description	Min.	Max.	Min.	Max.	Unit
1	TcC	Clock Cycle Time	250	2000	165	2000	ns
2	TwCh	Clock Width (HIGH)	105	1000	70	1000	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(AUv)	Clock RE to Upper Address (A_{16}-A_{23}) Valid Delay		90		80	ns
7	ThC(AUv)	Clock RE to Upper Address Valid Hold Time	5		5		ns
8	TdC(ST)	Clock RE to R/$\overline{W}$ and B/$\overline{W}$ Valid Delay		110		90	ns
9	TdC(A)	Clock RE to Lower Address (A_0-A_{15}) Valid Delay		90		90	ns
10	TdC(Az)	Clock RE to Lower Address (A_0-A_{15}) Float Delay		60		60	ns
11	TdC(ALr)	Clock RE to ALE RE Delay		70		60	ns
12	TdC(AL)	Clock FE to ALE FE Delay		70		60	ns
13	TdC(DS)	Clock RE to $\overline{DS}$ (Read) FE Delay		60		60	ns
14	TdC(DSf)	Clock FE to $\overline{DS}$ (Write) FE Delay		60		60	ns
15	TdC(DSr)	Clock FE to $\overline{DS}$ RE Delay		60		60	ns
16	TdC(DO)	Clock RE to Data Out Valid Delay		90		90	ns
17	TsDI(C)	Data in to Clock FE Setup Time	20		15		ns
18	TdA(AL)	Address Valid to ALE FE Delay	50		35		ns
19	ThAL(A)	ALE FE to Lower Address Valid Hold Time	60		40		ns
20	TwAL	ALE Width (HIGH)	80		60		ns
21	TdAz(DS)	Lower Address Float to $\overline{DS}$ LOW Delay	0		0		ns
22	TdAL(DS)	ALE FE to $\overline{DS}$ (Read) FE Delay	75		35		ns
23	TdAL(DI)	ALE FE to Data in Required Valid Delay		300		215	ns
24	TdA(DI)	Address Valid to Data in Required Valid Delay		410		305	ns
25	TdDS(A)	$\overline{DS}$ RE to Address Active Delay	80		45		ns
26	TdDS(Al)	$\overline{DS}$ RE to ALE RE Delay	75		40		ns
27	TdA(DS)	Address Valid to $\overline{DS}$ (Read) FE Delay	160		110		ns
28	TdDO(DSr)	Data Out Valid to $\overline{DS}$ RE Delay	230		150		ns
29	TdDO(DSf)	Data Out Valid to $\overline{DS}$ FE Delay	55		35		ns
30	ThDS(DO)	$\overline{DS}$ RE to Data Out Valid Hold Time	85		45		ns
31	TdDS(DI)	$\overline{DS}$ (Read) FE to Data in Required Valid Delay		205		155	ns
33	ThDI(DS)	$\overline{DS}$ RE to Data in Hold Time	0		0		ns
34	TwDSmw	$\overline{DS}$ (Write) Width (LOW)	185		110		ns
35	TwDSmr	$\overline{DS}$ (Read) Width (LOW)	275		220		ns
36	TdC(RBr)	Clock FE to $\overline{RBEN}$ RE Delay*		70		65	ns
37	ThDS(ST)	$\overline{DS}$ RE to B/$\overline{W}$, N/$\overline{S}$, R/$\overline{W}$ and M/$\overline{IO}$ Valid Hold Time	70		45		ns
38	TdC(TRf)	Clock RE to $\overline{TBEN}$ or $\overline{RBEN}$ FE Delay		60		60	ns
39	TdC(TRr)	Clock RE to $\overline{TBEN}$ RE Delay		60		60	ns
40	TdC(ST)	Clock RE to M/$\overline{IO}$ and N/$\overline{S}$ Valid Delay		90		75	ns
41	TdS(AL)	R/$\overline{W}$, M/$\overline{IO}$, B/$\overline{W}$ and N/$\overline{S}$ Valid to ALE FE Delay	60		35		ns
42	TsWT(C)	$\overline{WAIT}$ to Clock FF Setup Time	20		20		ns
43	ThWT(C)	$\overline{WAIT}$ to Clock FE Hold Time	20		20		ns
44	TwDRQ	$\overline{DREQ}$ Pulse Width (Single Transfer Mode)	20		20		ns
45	TsDRQ(C)	$\overline{DREQ}$ Valid to Clock RE Setup Time	60		50		ns
46	ThDRQ(C)	Clock RE to $\overline{DREQ}$ Valid Hold Time	20		20		ns
47	TdC(INTf)	Clock FE to $\overline{INT}$ FE Delay		150		150	ns

*These must not occur simultaneously.
Note: RE = rising edge
FE = falling edge

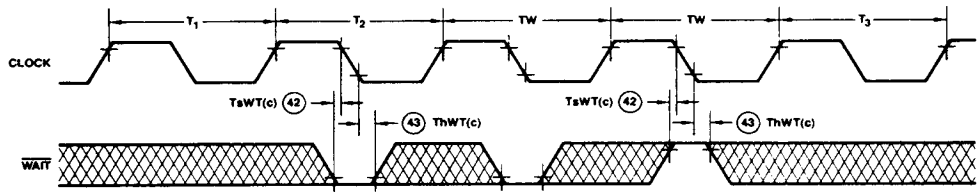
Am9516A CLOCK-CYCLE-TIME-DEPENDENT CHARACTERISTICS

The parameters listed below are also shown in the Switching specification. However, they are dependent on the actual values of the clock periods. The equations below define that dependence so that the exact limit for these parameters may be determined for any given system in relation to its specific clock characteristics.

Number	Parameters	Derivation
18	TdA (AL)	0.5TcC – #9 + (#12 – tr)
19	ThAL (A)	0.5TcC – #12 (ALE FE @ 0.8 V) + #10
21	TdAz (DS)	#13 – #10
22	TdAL (DS)	0.5TcC – #12 + #13
23	TdAL (DI)	2TcC – #12 – #17
24	TdA (DI)	2.5TcC – #9 – #17
25	TdDS (A)	0.5TcC – #15 + #9
26	TsDS (AL)	0.5TcC – #15 + #11 (ALE RE)
27	TdA (DS)	TcC – #9 + #13
28	TdDO (DSr)	1.5TcC – #16 + #15
29	TdDO (DSf)	0.5TcC – #16 + #14
30	ThDS (DO)	0.5TcC – #15 + #32
31	TdDS (DI)	1.5TcC – #13 – #17
34	TwDSmw	TcC – #14 + #15
35	TwDSmr	1.5TcC – #13 + #15
37	ThDS (ST)	0.5TcC – #15 + (#40 – tr)
41	TdS (AL)	0.5TcC – #40 + (#12 – tr)

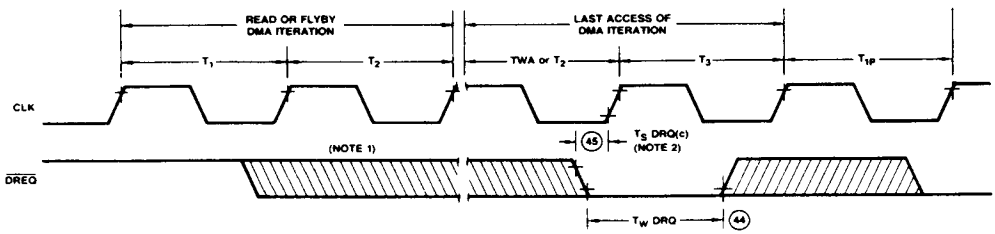
NOTE: tr (nominal) = 10ns
#32 CLCK RE to Data Out Not Valid Delay = 20ns (4 and 6 MHz)

$\overline{\text{WAIT}}$ Timing



WF007680

Sampling $\overline{\text{DREQ}}$ During Single Transfer DMA Operations



WF007670

Notes: 1. HIGH-to-LOW $\overline{\text{DREQ}}$ transitions will only be recognized after the HIGH-to-LOW transition of the clock during T_1 of a read or flyby DMA iteration.

2. A HIGH-to-LOW $\overline{\text{DREQ}}$ transition must meet the conditions in Note 1 and must occur TsDRQ(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. $\overline{\text{DREQ}}$ may go HIGH before TsDRQ(c) if it has met the TwDRQ parameter.

3. Flyby and Search transactions have only a single access; parameter TsDRQ(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.

SWITCHING CHARACTERISTICS over **COMMERCIAL** operating range unless otherwise specified
UDC AS BUS SLAVE BUS EXCHANGE

			Preliminary								
			4 MHz		6 MHz		8 MHz		10 MHz		
Number	Parameters	Description	Min	Max	Min	Max	Min	Max	Min	Max	Units
61	TdIN(DO)	$\overline{INTACK}$ FE to Data Output Valid Delay		135		135		120		110	ns
62	TdIN(DOz)	$\overline{INTACK}$ RE to Data Output Float Delay		80		80		45		35	ns
63	TdDS(DO)	$\overline{DS}$ FE (IOR) to Data Output Driven Delay		135*		135		120		110	ns
64	TdDS(DOz)	$\overline{DS}$ RE (IOR) to Data Output Float Delay		80		80		45		40	ns
65	TsDI(DS)	Data Valid to $\overline{DS}$ RE (IOW) Set-up Time	40		40		40		35		ns
66	ThDS(DI)	$\overline{DS}$ RE (IOW) to Data Valid Hold Time	40		30		0		0		ns
67	TwDS	$\overline{DS}$ Low Width	150*		150*		125		100		ns
68	TwIN	$\overline{INTACK}$ Low Width	150		150		125		100		ns
69	ThDS(CS)	$\overline{DS}$ RE to $\overline{CS}$ Valid Hold Time	20		20		15		10		ns
70	ThDS(PD)	$\overline{DS}$ RE to P/$\overline{D}$ Valid Hold Time	20		20		15		10		ns
71	TsPD(DS)	P/$\overline{D}$ Valid to $\overline{DS}$ FE Set-up Time (IOR)	10		10		10		10		ns
		P/$\overline{D}$ Valid to $\overline{DS}$ FE Set-up Time (IOW)	50		50		40		30		
72	TsCS(DS)	$\overline{CS}$ Valid to $\overline{DS}$ FE Set-up Time	30		30		20		10		ns
73	TrDS	$\overline{DS}$ RE to $\overline{DS}$ FE Recovery Time (for Commands Only)	4TcC		4TcC		4Tcc		4Tcc		ns
74	TwRST	$\overline{RESET}$ Low Width	3TcC		3TcC		3Tcc		3Tcc		ns
75	TdC(BRQf)	Clock RE to BREQ RE Delay		165		150		125		100	ns
76	TdC(BRQr)	Clock FE to BREQ FE Delay		150		150		125		100	ns
77	TdBRQ(CTRz)	BREQ FE to Control Bus Float Delay		140		140		100		60	ns
78	TdBRQ(ADz)	BREQ FE to AD Bus Float Delay		140		140		100		60	ns
79	TdBRQ(BAK)	BREQ RE to BACK RE Required Delay	0		0		0		0		ns
80	TsBAK(C)	BACK Valid to Clock RE Set-up Time	50		45		30		20		ns
81	TdRES(ADz)	(Reset) FE to A and AD Buses Float Delay		135		135		125		100	ns
82	TdRES(CTRz)	(Reset) FE to Control Bus Float Delay		100		100		100		75	ns
83	TdRES(DSz)	(Reset) FE to $\overline{DS}$ Float Delay		90		90		80		60	ns
84	TsRW(DS)	R/$\overline{W}$ Valid to $\overline{DS}$ FE Set-up Time (IOW)	2		2		2		2		ns
85	ThDS(RW)	$\overline{DS}$ RE to R/$\overline{W}$ Valid Hold Time (IOW)	-10		-10		-10		-10		ns
86	TsRW(DS)	R/$\overline{W}$ Valid to $\overline{DS}$ FE Set-up Time (IOR)	20		20		15		15		ns
87	ThDS(RW)	$\overline{DS}$ RE to R/$\overline{W}$ Valid Hold Time (IOR)	20		20		15		15		ns

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

SWITCHING CHARACTERISTICS (continued)
UDC-PERIPHERAL INTERFACE

			Preliminary								
			4 MHz		6 MHz		8 MHz		10 MHz		
Number	Parameters	Description	Min	Max	Min	Max	Min	Max	Min	Max	Units
90	TCHDL	Clock RE to Pulsed $\overline{DACK}$ FE Delay (Flyby Transactions Only)		100		85		50		40	ns
91	TCHDH	Clock RE to Pulsed $\overline{DACK}$ RE Delay (To Flyby Transactions Only)		100		85		50		40	ns
92	TDSK	$\overline{DS}$ RE to Pulsed $\overline{DACK}$ RE Delay (FROM Flyby Transactions Only)	10		10		10		10		ns
93	TDAD	Clock RE to Level $\overline{DACK}$ Valid Delay		100		85		60		50	ns
94	TDAH	Clock FE to Level $\overline{DACK}$ Valid Hold Time		100		85		60		50	ns
95	TEIDL	Clock FE to Internal $\overline{EOP}$ LOW Delay		110		90		80		70	ns
96	TEIDH	Clock FE to Internal $\overline{EOP}$ RE Delay		110		90		80		70	ns
97	TES	External $\overline{EOP}$ Valid to Clock RE Set-up Time During Operation	10		10		10		10		ns
98	TEW	External $\overline{EOP}$ Pulse Width Required During Operation	20		20		20		20		ns
99	TES(BH)	External $\overline{EOP}$ Valid to Clock FE Set-up Time During Bus Hold	10		10		10		10		ns
100	TEW(BH)	External $\overline{EOP}$ Pulse Width Required During Bus Hold	20		20		20		20		ns

Note: RE = rising edge
FE = falling edge

SWITCHING CHARACTERISTICS over **MILITARY** operating range (for APL Products, Subgroups 9, 10, 11 are tested unless otherwise specified)

UDC AS BUS SLAVE BUS EXCHANGE

					Preliminary		
			4 MHz		6 MHz		
No.	Parameter Symbol	Parameter Description	Min.	Max.	Min.	Max.	Unit
61	TdIN(DO)	$\overline{INTACK}$ FE to Data Output Valid Delay		135		135	ns
62	TdIN(DOz)	$\overline{INTACK}$ RE to Data Output Float Delay		80		80	ns
63	TdDS(DO)	$\overline{DS}$ FE (IOR) to Data Output Driven Delay		135*		135	ns
64	TdDS(DOz)	$\overline{DS}$ RE (IOR) to Data Output Float Delay		80		80	ns
65	TsDI(DS)	Data Valid to $\overline{DS}$ RE (IOW) Setup Time	40		40		ns
66	ThDS(DI)	$\overline{DS}$ RE (IOW) to Data Valid Hold Time	40		30		ns
67	TwDS	$\overline{DS}$ LOW Width	150*		150*		ns
68	TwIN	$\overline{INTACK}$ LOW Width	150		150		ns
69	ThDS(CS)	$\overline{DS}$ RE to $\overline{CS}$ Valid Hold Time	20		20		ns
70	ThDS(PD)	$\overline{DS}$ RE to P/$\overline{D}$ Valid Hold Time	20		20		ns
71	TsPD(DS)	P/$\overline{D}$ Valid to $\overline{DS}$ FE Setup Time (IOR)	10		10		ns
		P/$\overline{D}$ Valid to $\overline{DS}$ FE Setup Time (IOW)	50		50		
72	TsCS(DS)	$\overline{CS}$ Valid to $\overline{DS}$ FE Setup Time	30		30		ns
73	TrDS	$\overline{DS}$ RE to $\overline{DS}$ FE Recovery Time (for Commands Only)	4TcC		4TcC		ns
74	TwRST	$\overline{RESET}$ LOW Width	3TcC		3TcC		ns
75	TdC(BRQf)	Clock RE to BREQ RE Delay		165		150	ns
76	TdC(BRQr)	Clock FE to BREQ FE Delay		150		150	ns
77	TdBRQ(CTRz)	BREQ FE to Control Bus Float Delay		140		140	ns
78	TdBRQ(ADz)	BREQ FE to AD Bus Float Delay		140		140	ns
79	TdBRQ(BAK)	BREQ RE to BACK RE Required Delay	0		0		ns
80	TsBAK(C)	BACK Valid to Clock RE Setup Time	50		45		ns
81	TdRES(ADz)	RESET FE to A and AD Buses Float Delay		135**		135	ns
82	TdRES(CTRz)	RESET FE to Control Bus Float Delay		100**		100	ns
83	TdRES(DSz)	RESET FE to $\overline{DS}$ Float Delay		90**		90	ns
84	TsRW(DS)	R/$\overline{W}$ Valid to $\overline{DS}$ FE Setup Time (IOW)	2		2		ns
85	ThDS(RW)	$\overline{DS}$ RE to R/$\overline{W}$ Valid Hold Time (IOW)	−10		−10		ns
86	TsRW(DS)	R/$\overline{W}$ Valid to $\overline{DS}$ FE Setup Time (IOR)	20		20		ns
87	ThDS(RW)	$\overline{DS}$ RE to R/$\overline{W}$ Valid Hold Time (IOR)	20		20		ns

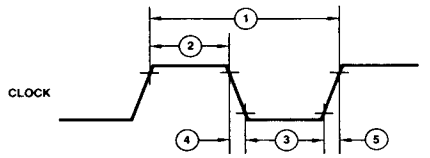
*2000 ns for slow readable registers (worst case)
**Guaranted but not tested.
Note: RE = rising edge
FE = falling edge

UDC-PERIPHERAL INTERFACE

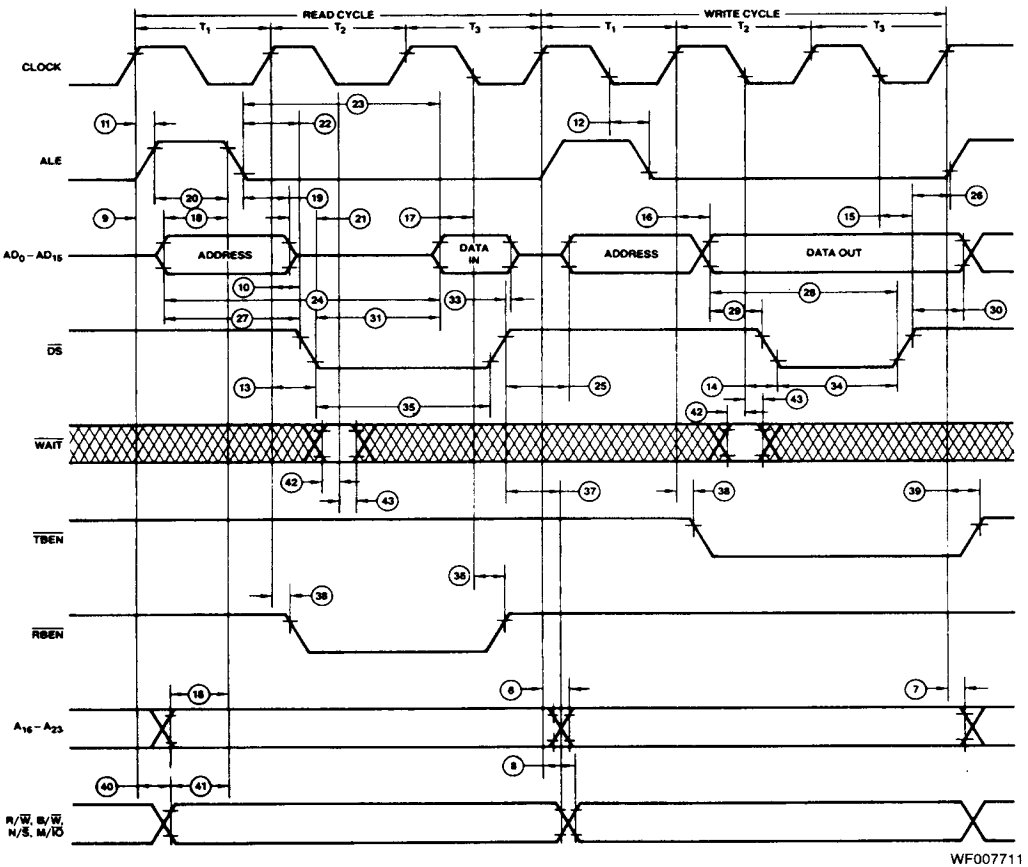
					Preliminary		
			4 MHz		6 MHz		
No.	Parameter Symbol	Parameter Description	Min.	Max.	Min.	Max.	Unit
90	TCHDL	Clock RE to Pulsed $\overline{DACK}$ FE Delay (Flyby Transactions Only)		100		85	ns
91	TCHDH	Clock RE to Pulsed $\overline{DACK}$ RE Delay (To Flyby Transactions Only)		100		85	ns
92	TDSK	$\overline{DS}$ RE to Pulsed $\overline{DACK}$ RE Delay (FROM Flyby Transactions Only)	10		10		ns
93	TDAD	Clock RE to Level $\overline{DACK}$ Valid Delay		100		85	ns
94	TDAH	Clock FE to Level $\overline{DACK}$ Valid Hold Time		100		85	ns
95	TEIDL	Clock FE to Internal $\overline{EOP}$ LOW Delay		110		90	ns
96	TEIDH	Clock FE to Internal $\overline{EOP}$ RE Delay		110		90	ns
97	TES	External $\overline{EOP}$ Valid to Clock RE Setup Time During Operation	10		10		ns
98	TEW	External $\overline{EOP}$ Pulse Width Required During Operation	20		20		ns
99	TES(BH)	External $\overline{EOP}$ Valid to Clock RE Setup Time During Bus Hold	10		10		ns
100	TEW(BH)	External $\overline{EOP}$ Pulse Width Required During Bus Hold	20		20		ns

Note: RE = rising edge
FE = falling edge

AC Timing when UDC is a Bus Master

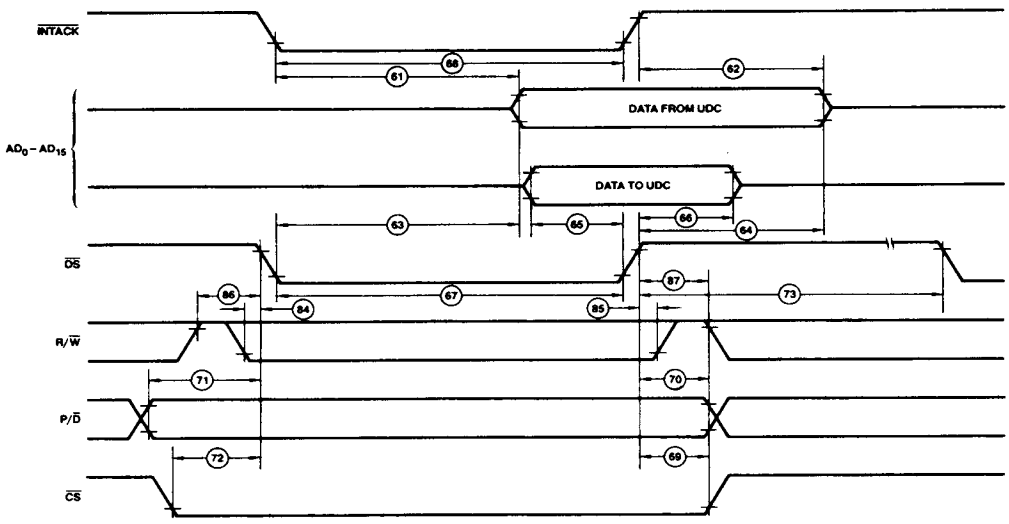


WF007630



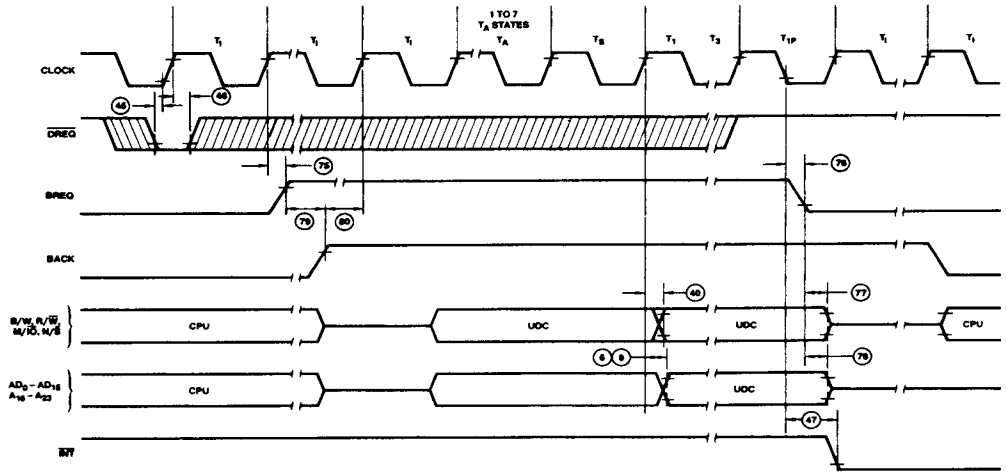
WF007711

AC Timing when UDC is a Bus Slave



WF007720

Bus Exchange Timing



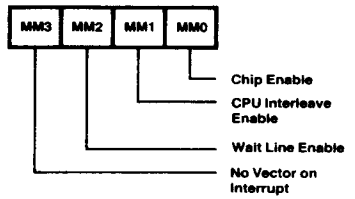
WF007730

APPENDIX A
UDC REGISTER SUMMARY

Master Mode Register

	A7							A0
Address	X	0	1	1	1	0	0	X

Fast Readable
Writable



DF003470

Miscellaneous Registers

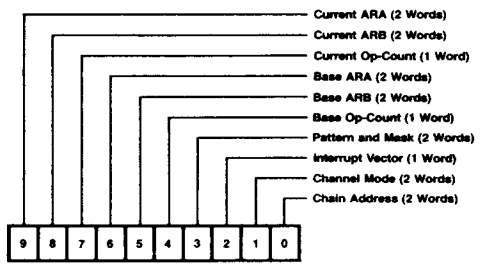
	A7							A0		
Address	X	0	1	1	0	0	1	X	Current Operation Count	CH1
	X	0	1	1	0	0	0	X	Current Operation Count	CH2
	X	0	1	1	0	1	1	X	Base Operation Count	CH1
	X	0	1	1	0	1	0	X	Base Operation Count	CH2
	X	1	0	0	1	0	1	X	Pattern	CH1
	X	1	0	0	1	0	0	X	Pattern	CH2
	X	1	0	0	1	1	1	X	Mask	CH1
	X	1	0	0	1	1	0	X	Mask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

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

D_{15}	D_{14}	D_{13}	D_{12}	D_{11}	D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

Chain Control Register

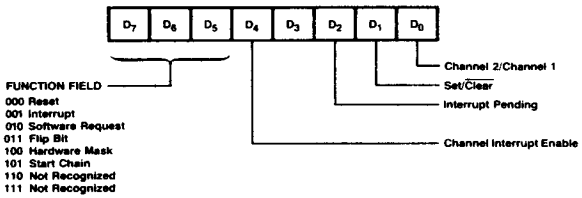
Chain Loadable Only



DF003480

Command Register

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

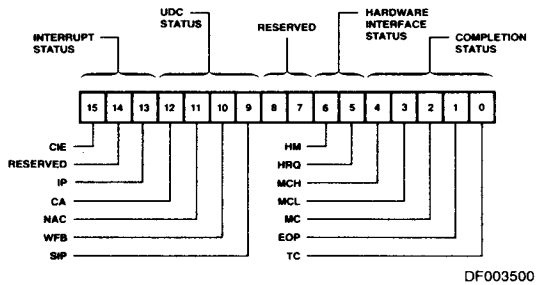


DF003490

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



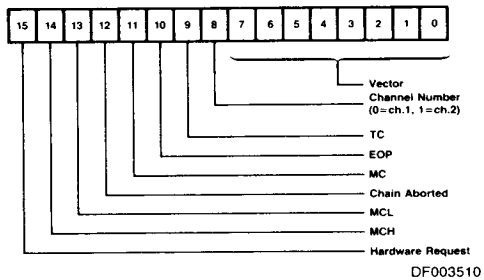
DF003500

2

Interrupt Save Register

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

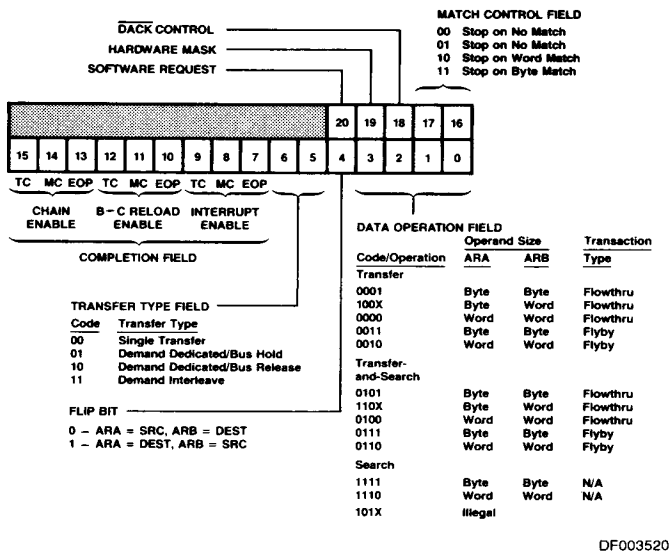
Fast Readable



Channel Mode Register

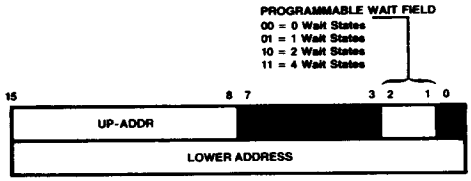
A7							A0	
X	1	0	1	0	1	1	X	High CH1
X	1	0	1	0	1	0	X	High CH2
X	1	0	1	0	0	1	X	Low CH1
X	1	0	1	0	0	0	X	Low CH2

Chain Loadable
Writable (Lower 16 bits)
Slow Readable



Chain Address Register

	A7						A0	
Address	X	0	1	0	0	1	1	X	Up-Addr CH1
	X	0	1	0	0	1	0	X	Up-Addr CH2
	X	0	1	0	0	0	1	X	Low-Addr CH1
	X	0	1	0	0	0	0	X	Low-Addr CH2

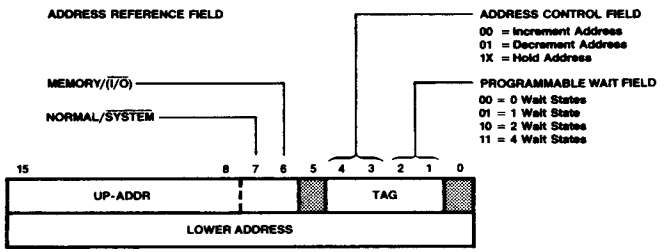


DF003530

Address Registers

	A7						A0		
Address	X	0	0	1	1	0	1	X	Current ARA Up-Addr/Tag CH1
	X	0	0	1	1	0	0	X	Current ARA Up-Addr/Tag CH2
	X	0	0	0	1	0	1	X	Current ARA Low-Addr CH1
	X	0	0	0	1	0	0	X	Current ARA Low-Addr CH2
	X	0	0	1	0	0	1	X	Current ARB Up-Addr/Tag CH1
	X	0	0	1	0	0	0	X	Current ARB Up-Addr/Tag CH2
	X	0	0	0	0	0	1	X	Current ARB Low-Addr CH1
	X	0	0	0	0	0	0	X	Current ARB Low-Addr CH2
	X	0	0	1	1	1	1	X	Base ARA Up-Addr/Tag CH1
	X	0	0	1	1	1	0	X	Base ARA Up-Addr/Tag CH2
	X	0	0	0	1	1	1	X	Base ARA Low-Addr CH1
	X	0	0	0	1	1	0	X	Base ARA Low-Addr CH2
	X	0	0	1	0	1	1	X	Base ARB Up-Addr/Tag CH1
	X	0	0	1	0	1	0	X	Base ARB Up-Addr/Tag CH2
	X	0	0	0	0	1	1	X	Base ARB Low-Addr CH1
	X	0	0	0	0	1	0	X	Base ARB Low-Addr CH2

Chain Loadable
Fast Readable and Writable



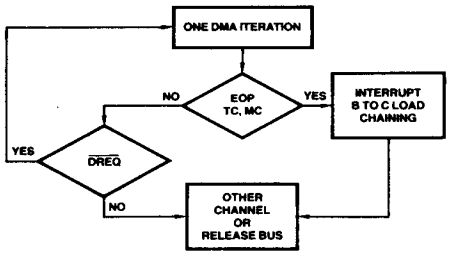
DF003540

APPENDIX B

Flow Charts of DMA Operations:

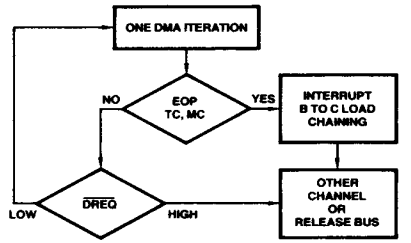
Figure B1 shows the basic DMA operations with software or hardware request. The Demand Interleave operations are shown in Figure B2.

(a) Single Operation



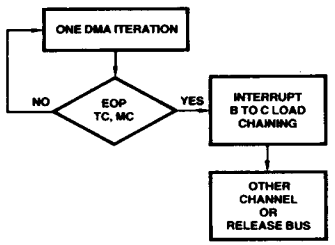
PF001250

(c) Demand Dedicated with Bus Release (Hardware Request)



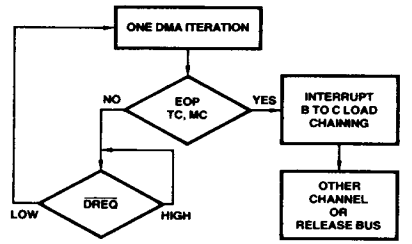
PF001260

(b) Demand Operation when Software Requesting



PF001270

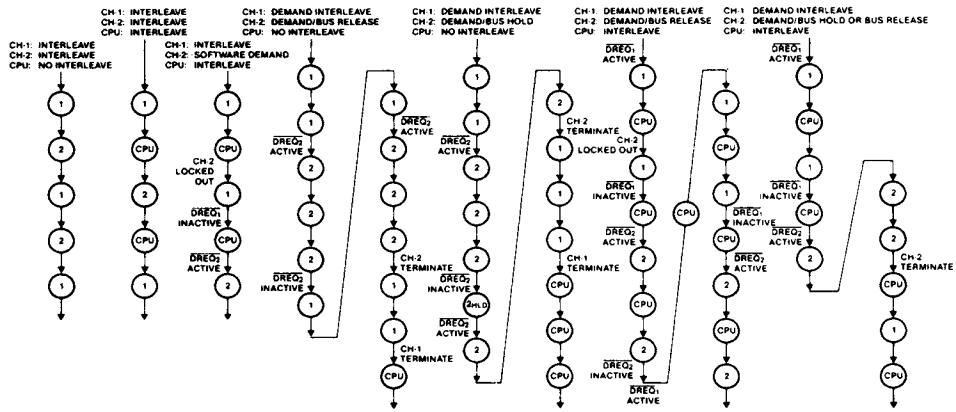
(d) Demand Dedicated with Bus Hold (Hardware Request)



PF001280

Figure B1. Basic DMA Operations of Am9516A UDC

NT**Multibus is a trademark of Intel Corp.

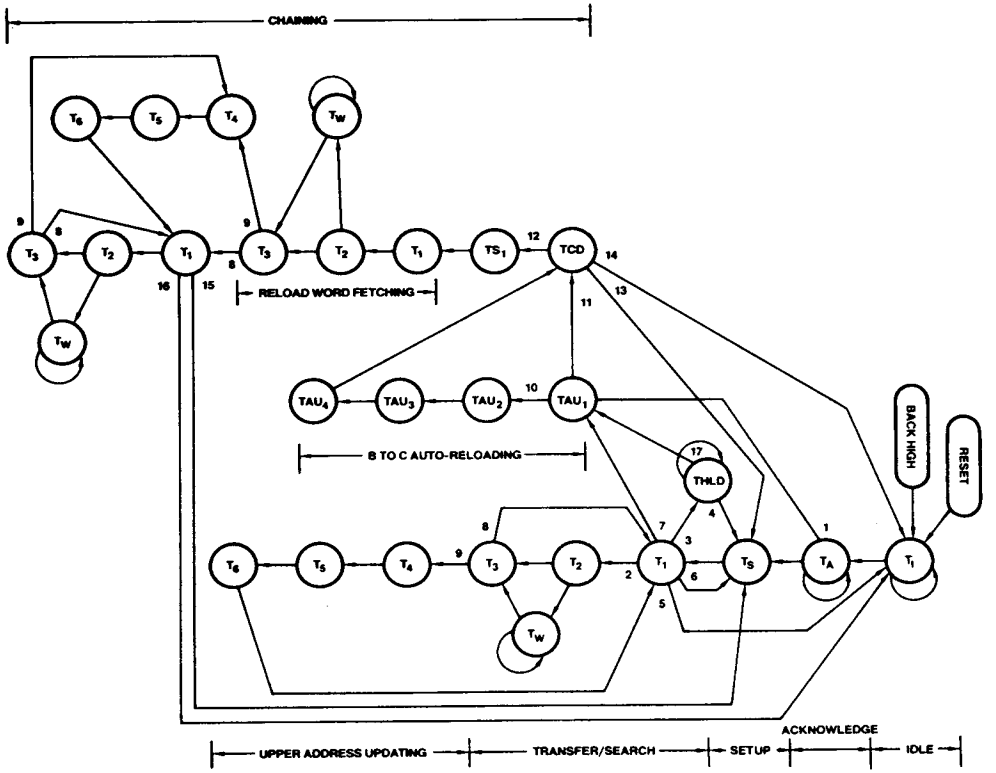


PF001300

Figure B2. Demand Interleave Operations of Am9516A UDC

APPENDIX C

Am9516A STATE DIAGRAM



AF003180

Am9516A INTERNAL OPERATION ROUTINES

1. "Start Chain" command issued or start updating routine* after an interrupt has been served.**
2. Normal DMA operation.
3. Demand with Bus hold while DREQ is inactive.
4. DREQ is active while bus held.
5. Single transfer, CPU interleave enabled, or demand with bus release while current DREQ is inactive and no DMA request is pending.
6. Single Transfer or Demand/Bus release while current DREQ is inactive, but the other DMA request is pending.
7. TC, MC or EOP termination occurs.
8. One DMA or chain transaction is done and the upper address is not changed.
9. One DMA or chain transaction is done and the upper address is changed.
10. Base-to-current auto-reloading is enabled.
11. Base-to-current auto-reloading is disabled.
12. Chaining is enabled.
13. Chaining is disabled and another DMA request is pending.
14. Chaining is disabled and no DMA request is pending.
15. Chaining ends and another DMA request is pending.
16. Chaining ends and no DMA request is pending.
17. EOP termination of Bus Hold.

*Updating routine includes base-to-current auto-reloading and chaining.

**When a second interrupt is to be issued before the first interrupt is acknowledged, the SIP bit of a Status register is set and the channel relinquishes the bus until the first interrupt has been served. If the channel was to perform the updating routine, once the SIP bit is cleared, DTC will reacquire the bus and perform the appropriate operation (i.e., 1).