MC68450

Technical Summary

Direct Memory Access Controller

M68000 microprocessors utilize state-of-the-art MOS technology to maximize performance and throughput. The MC68450 direct memory access controller (DMAC) is designed to complement the performance and architectural capabilities of M68000 Family microprocessors by moving blocks of data in a quick, efficient manner with minimum intervention from a processor. The DMAC performs memory-to-memory, memory-to-device, and device-to-memory data transfers by utilizing the following features:

- Four Independent DMA Channels with Programmable Priority
- Asynchronous M68000 Bus Structure with a 24-Bit Address and a 16-Bit Data Bus
- Fast Transfer Rates: Up to 4 Mbytes/Sec at 10 MHz, No Wait States
- Fully Supports All M68000 Bus Options such as Halt, Bus Error, and Retry
- FIFO Locked Step Support with Device Transfer Complete Signal
- Flexible Request Generation:
  Internal, Maximum Rate
  Internal, Limited Rate
  External, Cycle Steal (With or Without Hold)
  External, Burst
  Mixed Internal and External
- Programmable 8-Bit or 16-Bit Peripheral Device Types:
  Explicitly Addressed:
    M68000 Type
    M6800 Type
  Implicitly Addressed:
    Device with Request and Acknowledge
    Device with Request, Acknowledge, and Ready
- Pin and Register Compatible Functional Superset of the MC68440 DDMA

This document contains information on a new product. Specifications and information herein are subject to change without notice.

INTRODUCTION

The main purpose of a DMAC in any system is to transfer data at very high rates, usually much faster than a microprocessor under software control can handle. The term DMA is used to refer to the ability of a peripheral device to access memory in a system in the same manner as a microprocessor does. DMA operation can occur concurrently with other microprocessor operations, thus greatly boosting overall system performance.

Figure 1 illustrates a typical system configuration using a DMA interface to a high-speed disk storage device. In a system such as this, the DMAC moves blocks of data between the peripheral controllers and memory at rates approaching the limits of the memory bus since data movement is implemented in high-speed MOS hardware. A block of data consists of a sequence of byte, word, or long-word operands starting at a specific address in memory with the length of the block determined by a transfer count. A single channel operation may involve the transfer of several blocks of data between the memory and a device.

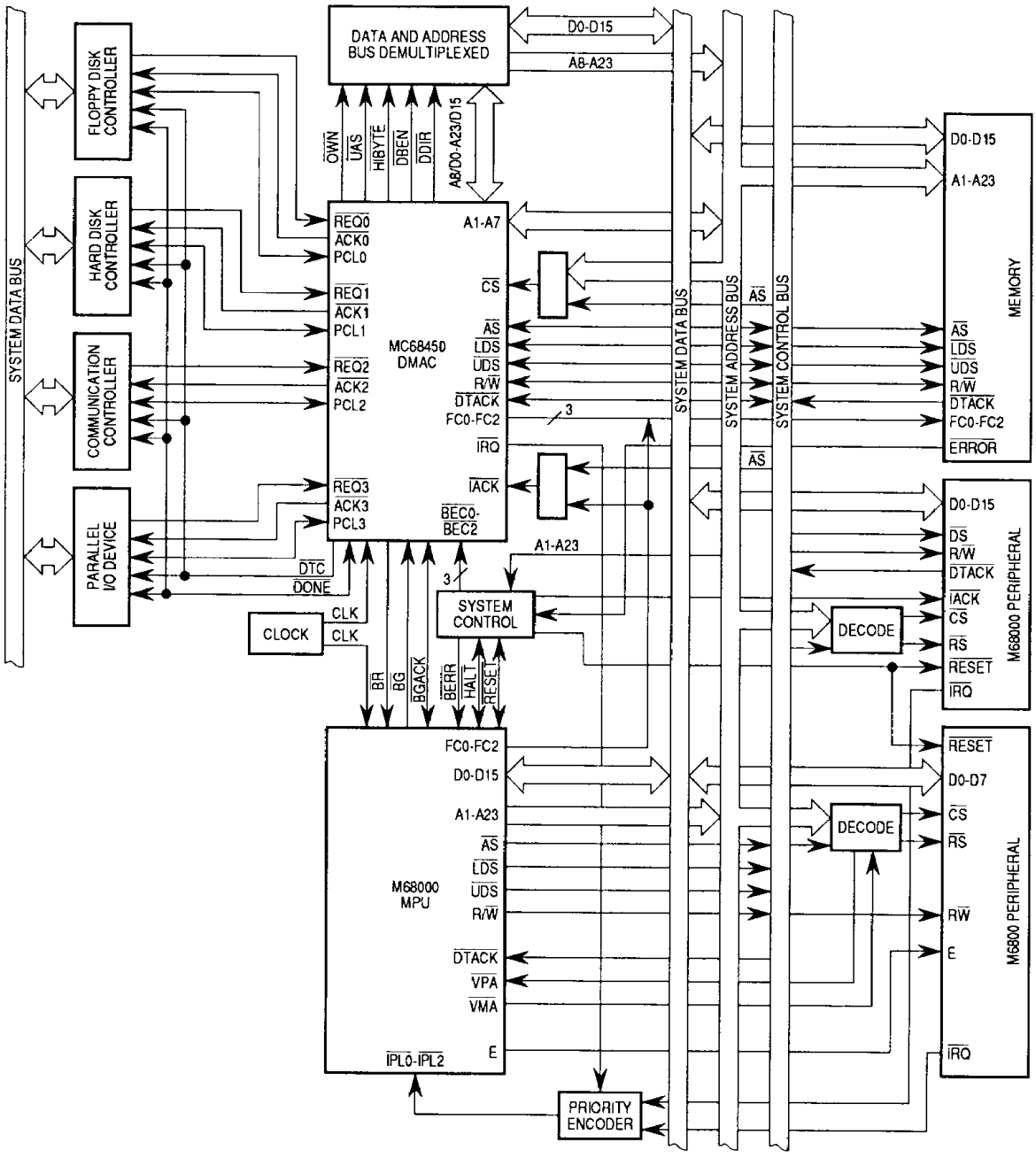


Figure 1. Typical M68000-Based System Configuration

Any operation involving the DMAC follows the same basic steps: channel initialization by the main processor, data transfer, and block termination. In the initialization phase, the host processor loads the registers of the DMAC with control information, address pointers, and transfer counts and then starts the channel. During the transfer phase, the DMAC accepts requests for operand transfers and provides addressing and bus control for the transfers. The termination phase occurs after the operation is complete when the DMAC indicates the status of the operation in the status register. During all phases of a data transfer operation, the DMAC will be in one of three operating modes:

IDLE The DMAC enters this state when it is reset by an external device and waiting for initialization by the main processor or an operand transfer request from a peripheral.

MPU The DMAC enters this state when selected by another bus master in the system (usually the main processor). In this mode, the DMAC internal registers are written or read to control channel operation or to check the status of a block transfer.

DMA The DMAC enters this state when acting as a bus master to perform an operand transfer.

In addition to fully supporting the M68000 Family bus, the DMAC also supports transfers to or from M6800 Family peripheral devices. Figure 2 illustrates a typical system configuration utilizing an M6800-type peripheral device.

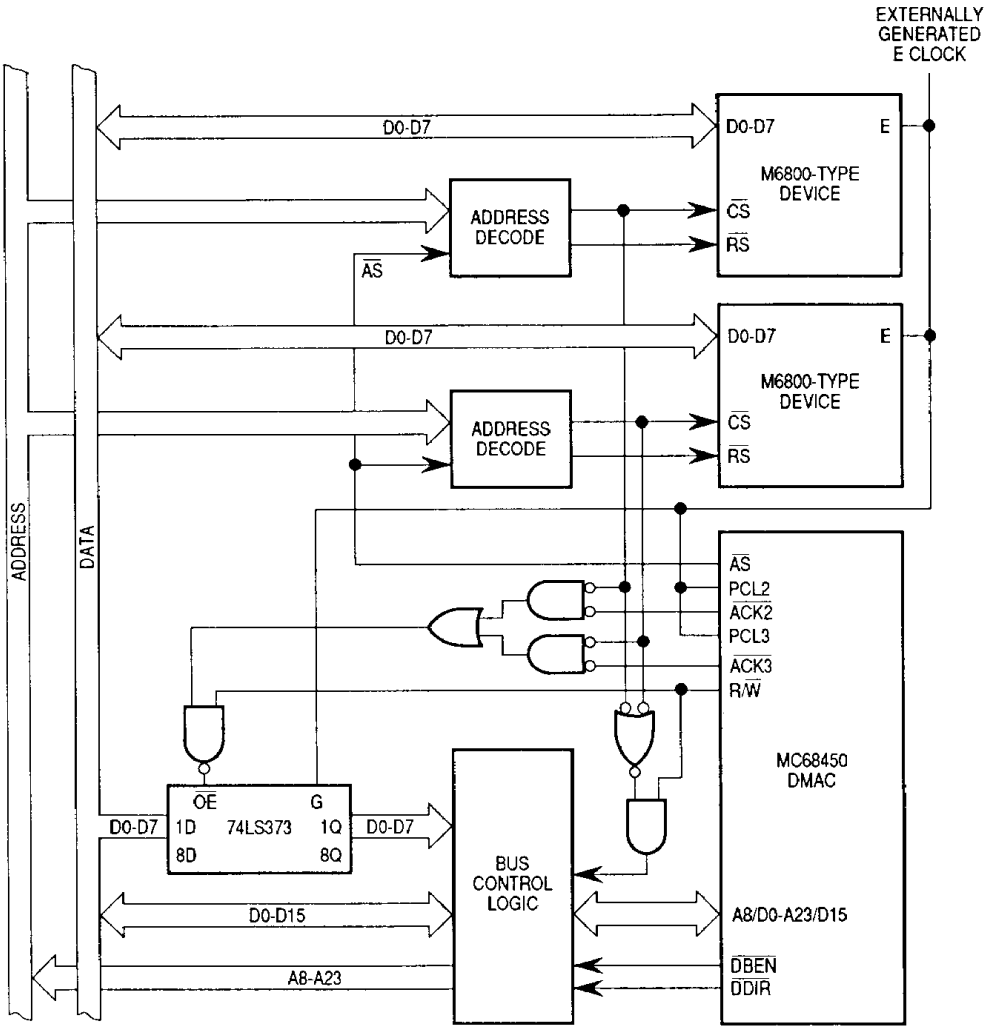


Figure 2. Typical System Configuration with M6800-Type Peripheral Devices

OPERAND TRANSFER MODES

The DMAC can perform implicitly addressed or explicitly addressed data transfers using any of the following protocols:

1. Explicitly addressed, MC68000-compatible device
2. Explicitly addressed, MC6800-compatible devices
3. Implicitly addressed, device with request and acknowledge
4. Implicitly addressed, device with request, acknowledge, and ready

During protocols 1 and 2, data is transferred from the source to an internal DMAC holding register, and then moved from the holding register to the des-

tination on the next bus cycle. Protocols 3 and 4 require only one bus cycle for data transfer since only one device needs to be addressed. With these protocols, communication is performed using a two-signal and three-signal handshake, respectively.

Implicitly addressed devices do not require the generation of a device data register address for a data transfer. Such a device is controlled by a five-signal device control interface on the DMAC during implicit address transfers (see Figure 3). Since only memory is addressed during such a data transfer, this method is called the single-address method.

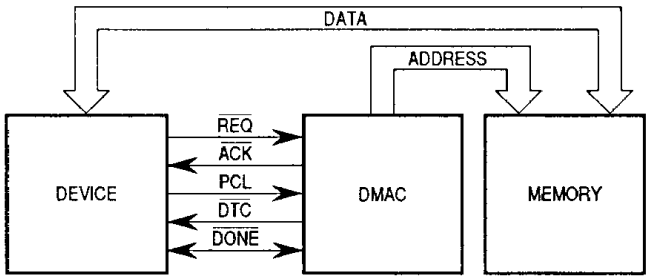


Figure 3. Implicitly Addressed Device Interface

Explicitly addressed devices require that a data register within the peripheral device be addressed. No signals other than the M68000 asynchronous bus control signals are needed to interface with such a device, although any of the five-device control signals may be used. Because the address bus is used to access the peripheral, the data cannot be directly transferred to/from memory since memory also requires addressing. Therefore, data is transferred from the source to an internal holding register in the DMAC and then transferred to the destination during a second bus transfer (see Figure 4). Since both memory and the device are addressed during such a data transfer, this method is called the dual-address method.

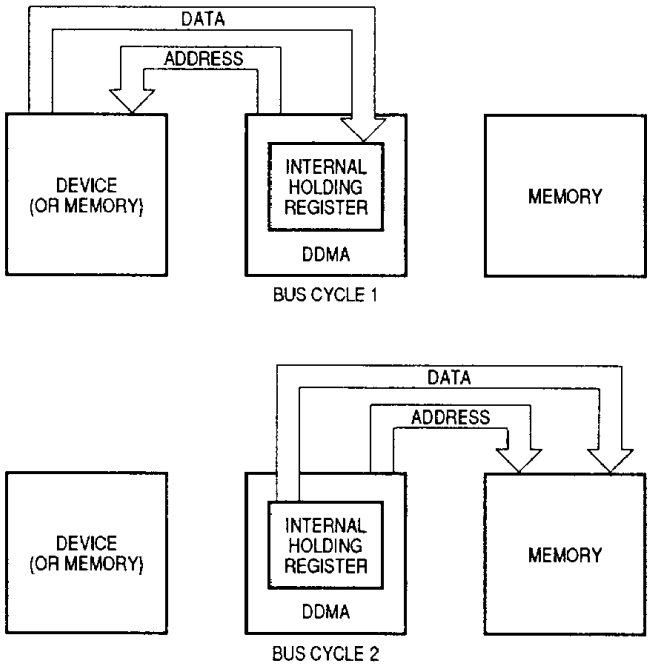


Figure 4. Dual-Address Transfer Sequence

CHANNEL OPERATING MODES

There are three types of channel operations: 1) single block transfers, 2) continued operation, and 3) chained operations. The first two modes utilize on-chip registers; whereas, the last mode uses an on-chip address register to point to address and to count parameters stored in system memory.

When transferring single blocks of data, the memory address and device address registers are initialized by the user to specify the source and destination of the transfer. The memory transfer count register is also initialized to count the number of operands transferred in a block. Repeated transfers are possible with the continued mode of operation, where the memory address and transfer count registers are automatically loaded from internal registers upon completion of a block transfer (see Figure 5).

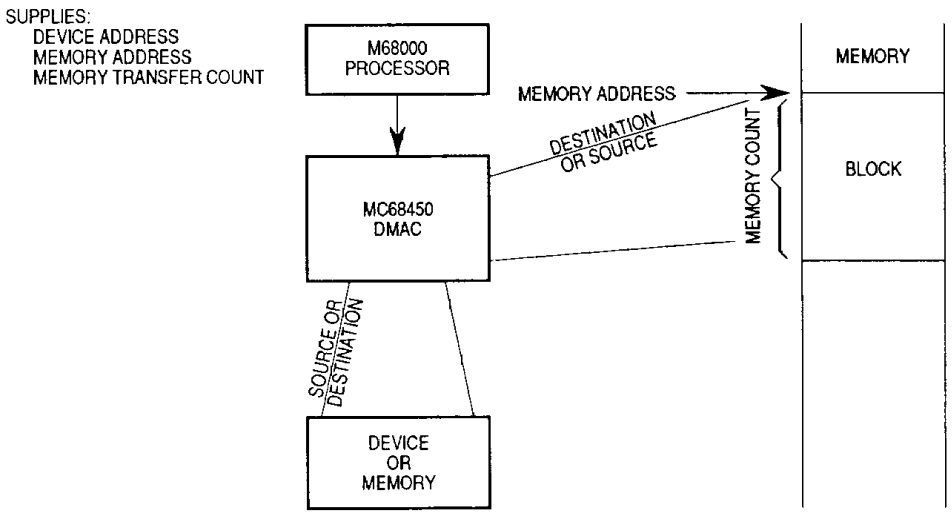


Figure 5. Single Block Transfer

The two chaining modes are array chaining and linked array chaining. The array chaining mode operates from a contiguous memory array consisting of memory addresses and transfer counts. The base address register and base transfer count register are initialized to point to the beginning address of the array and the number of array entries, respectively. As each block transfer is completed, the next entry is fetched from the array, the base transfer count is decremented, and the base address is incremented to point to the next array entry. When the base transfer count reaches zero, the entry just fetched is the last block transfer defined in the array (see Figure 6).

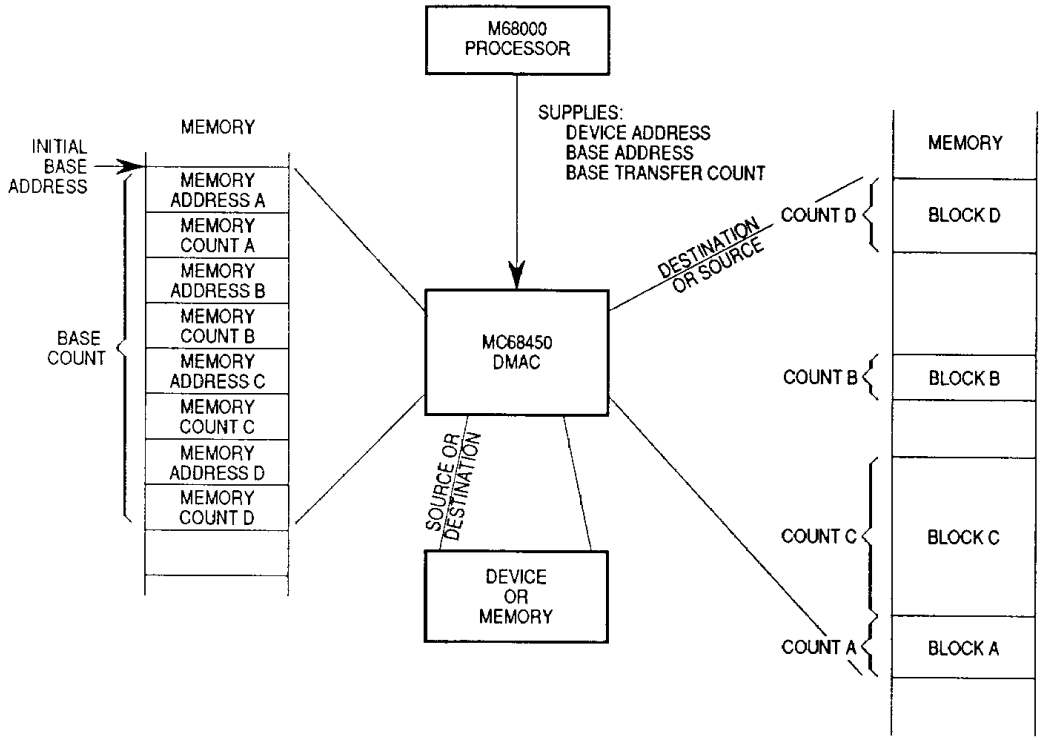


Figure 6. Array Chaining Transfer

The linked array chaining mode is similar to the array chaining mode, except that each entry in the memory array also contains a link address pointing to the next entry in the array. This method allows a noncontiguous memory array. The last entry contains a link address set to zero. No base transfer count register is needed in this mode. The base address register is initialized to the address of the first entry in the array. The link address is used to update the base address register at the beginning of each block transfer. This chaining mode allows array entries to be easily moved or inserted without reorganizing the array into sequential order. Also, the number of entries in the array need not be specified to the DMAC (see Figure 7).

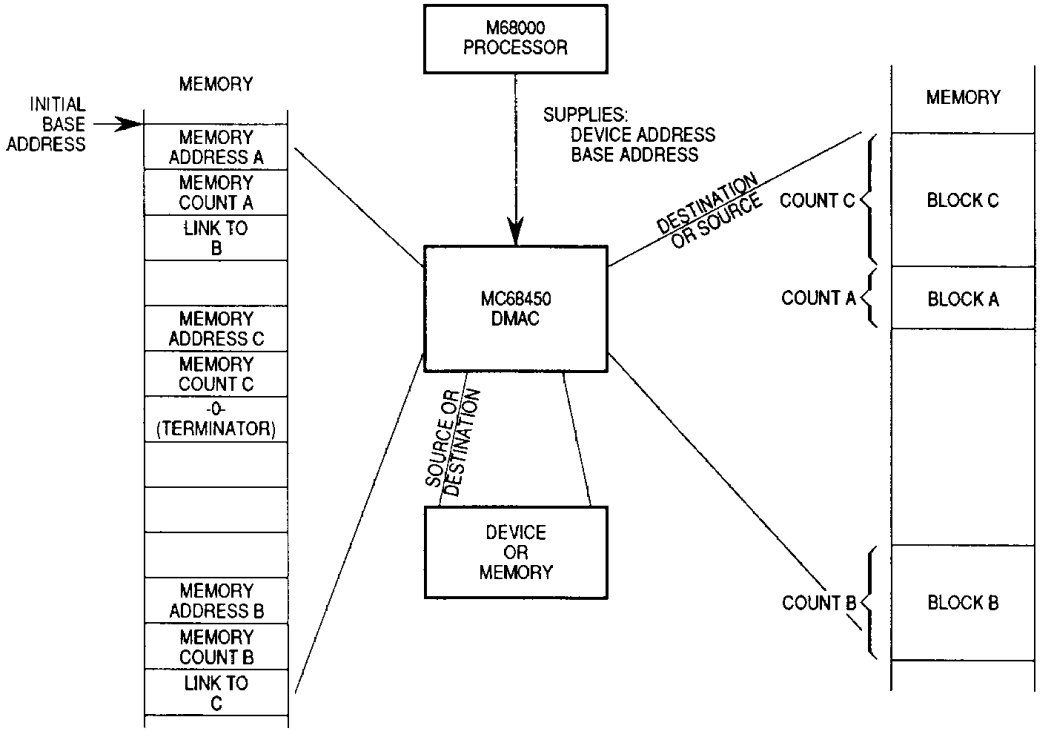


Figure 7. Linked Array Chaining Transfer

6

INTERRUPT OPERATION

The DMAC will interrupt the MPU for a number of occurrences, such as the completion of a DMA operation or at the request of a peripheral device using a peripheral control line. The user must write interrupt vectors into an on-chip vector register for use in the M68000 vectored interrupt structure. Two vector registers are available for each channel.

CHANNEL PRIORITY

Each channel may be given a priority level of 0, 1, 2, or 3. If several channel requests occur at the same priority level, a round-robin is entered automatically.

REQUEST MODES

Requests may be externally generated by a device or internally generated by the auto-request mechanism of the DMAC. Auto-requests may be generated either at the maximum rate, where the channel always has a request pending, or at a limited rate determined by selecting a portion of the bus bandwidth to be available for DMA activity. External requests can be either burst requests or cycle-steal requests generated by the request signal associated with each channel.

REGISTERS

The DMAC contains 17 registers for each of four channels plus one general control register, all of which are under complete software control. The user programmer's model of the registers is shown in Figure 8.

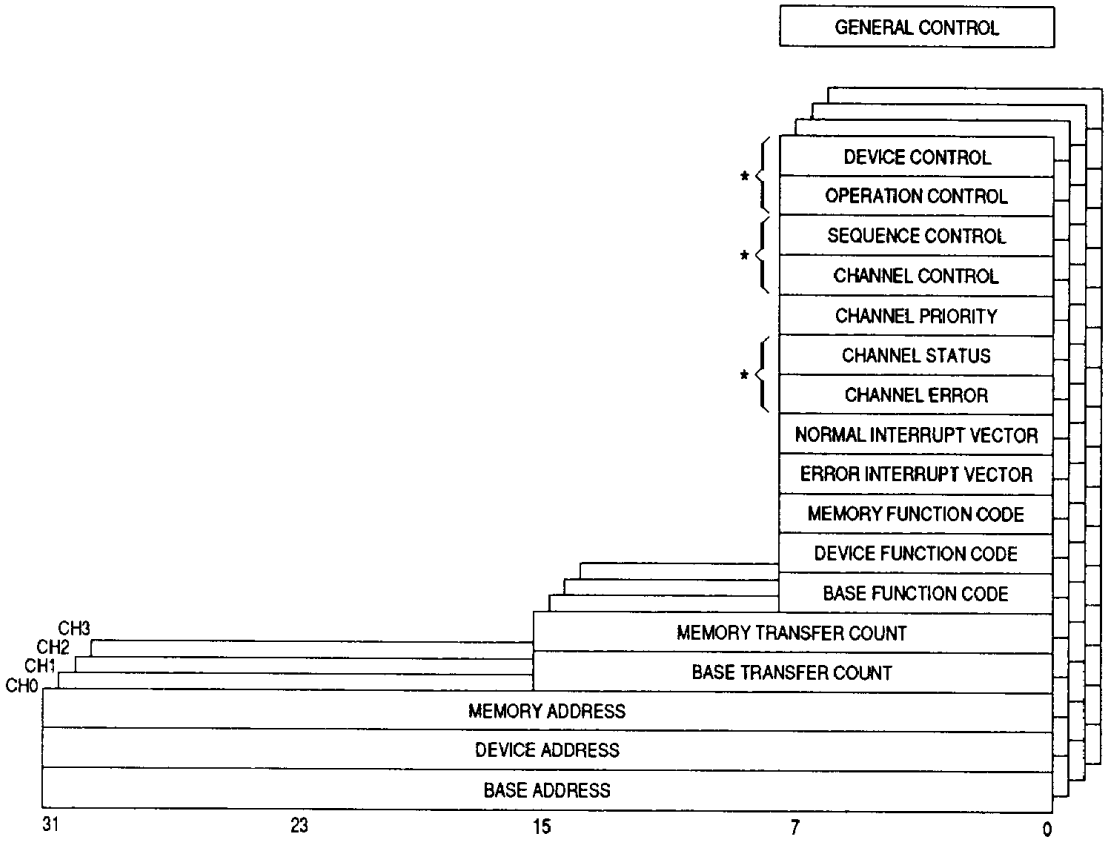


Figure 8. Programmer's Model

The DMAC registers contain information about the data transfers such as the source and destination address and function codes, transfer count, operand size, device port size, channel priority, continuation address and transfer count, and the function of the peripheral control line. One register also provides status and error information on channel activity, peripheral inputs, and various events that may have occurred during a DMA transfer. A general control register selects the bus utilization factor to be used in limited-rate auto-request DMA operations.

SIGNAL DESCRIPTION

The following paragraphs contain a brief description of the DMAC input and output signals. Included at the end of the functional description of the signals is a table describing the electrical characteristics of each pin (i.e., the type of driver used).

NOTE

The terms **assertion** and **negation** will be used extensively to avoid confusion when dealing with a mixture of active-low and active-high signals. The term **assert** or **assertion** is used to indicate that a signal is active or true, independent of whether that level is represented by a high or low voltage. The term **negate** or **negation** is used to indicate that a signal is inactive or false.

SIGNAL ORGANIZATION

The input and output signals can be functionally organized into the groups shown in Figure 9. The signal functions are discussed in the following paragraphs.

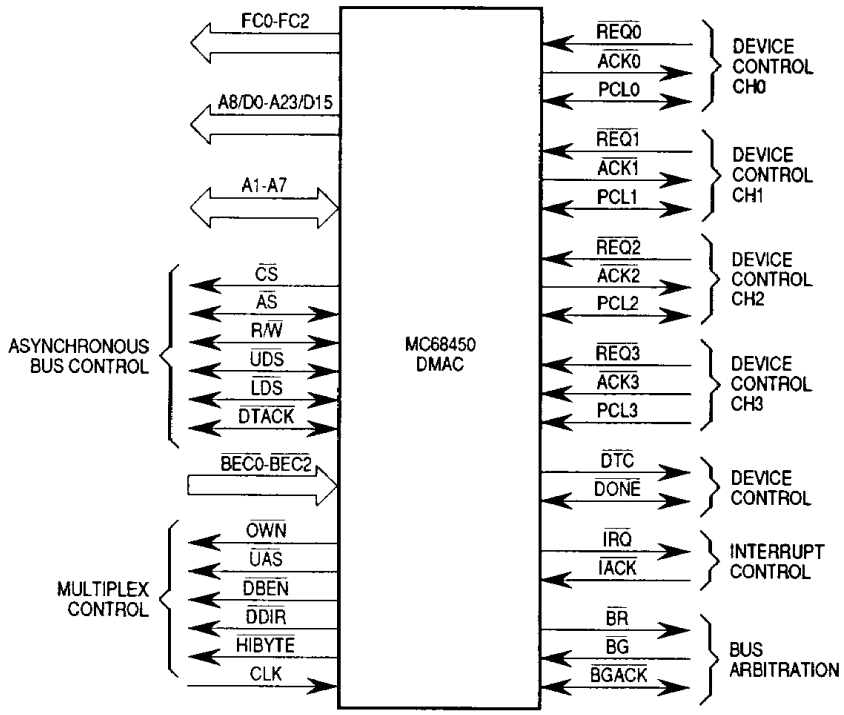


Figure 9. Functional Signal Organization

Address/Data Bus (A8/D0–A23/D15)

This 16-bit bus is time multiplexed to provide address outputs during the DMA mode of operation and is used as a bidirectional data bus to input data from an external device (during an MPU write or DMA read) or to output data to an external device (during an MPU read or a DMA write). This three-state bus is demultiplexed using external latches and buffers controlled by the multiplex control lines.

Lower Address Bus (A1–A7)

These bidirectional three-state signals are used to address the DMAC internal registers in the MPU mode and to provide the lower seven address outputs in the DMA mode.

Function Codes (FC0–FC2)

These three-state outputs are used in the DMA mode to further qualify the value on the address bus to provide eight separate address spaces that may be defined by the user. The value placed on these lines is taken from one of the internal function code registers, depending on the register that provides the address used during a DMA bus cycle.

Asynchronous Bus Control

Asynchronous data transfers are handled using the following control signals: chip select, address strobe, read/write, upper and lower data strobes, and data transfer acknowledge. These signals are described in the following paragraphs.

CHIP SELECT ($\overline{CS}$). This input selects the DMAC for an MPU bus cycle. When $\overline{CS}$ is asserted, the address on A1–A7 and the data strobes (or A when using an 8-bit bus) select the internal DMAC register that will be involved in the transfer. $\overline{CS}$ should be generated by qualifying an address decode signal with the address and data strobes.

ADDRESS STROBE ($\overline{AS}$). This bidirectional signal is an output in the DMA mode to indicate that a valid address is present on the address bus. In the MPU or IDLE modes, $\overline{AS}$ is an input to determine when the DMAC can take control of the bus (if the DMAC has requested and been granted use of the bus).

READ/WRITE (R/$\overline{W}$). This bidirectional signal indicates the direction of a data transfer during a bus cycle. In the MPU mode, a high level indicates that a transfer is from the DMAC to the data bus, and a low level indicates a transfer from the data bus to the DMAC. In the DMA mode, a high level indicates a transfer from the addressed memory or device to the data bus, and a low level indicates a transfer from the data bus to the addressed memory or device.

UPPER AND LOWER DATA STROBE ($\overline{UDS}$, $\overline{LDS}$). These bidirectional signals indicate when data is valid on the bus and what portions of the bus should be involved in the transfer.

DATA TRANSFER ACKNOWLEDGE ($\overline{DTACK}$). This bidirectional signal denotes that an asynchronous bus cycle may be terminated. In the MPU mode, this output indicates that the DMC has accepted data from the MPU or placed data on the bus for the MPU. In the DMA mode, this input is monitored by the DMAC to determine when to terminate a bus cycle. As long as $\overline{DTACK}$ remains negated, the DMAC will insert wait cycles into a bus cycle; when $\overline{DTACK}$ is asserted, the bus cycle will be terminated (except when PCL is used as a ready signal, in which case both signals must be asserted before the cycle is terminated).

BUS EXCEPTION CONTROL ($\overline{BEC0}$–$\overline{BEC2}$). These inputs provide an encoded signal that indicates an abnormal bus condition such as a bus error or reset.

Multiplex Control

These signals are used to control external multiplex/demultiplex devices to separate the address and data information on the A8/D0–A23/D15 lines and to transfer data between the upper and lower halves of the data bus during certain DMA bus cycles.

Figure 10 shows the five external devices needed to demultiplex the address/data pins and the interconnection of the multiplex control signals. The SN74LS245 that can connect the upper and lower halves of the data bus is needed only if an 8-bit device is used during single address transfers.

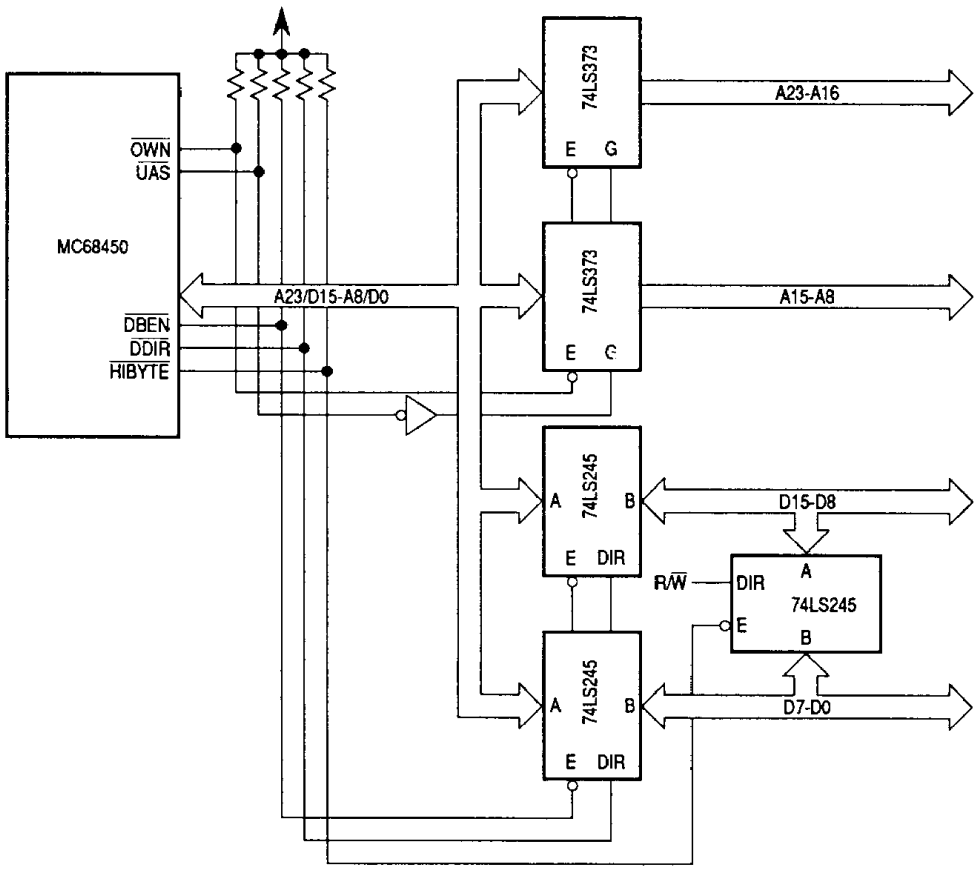


Figure 10. Demultiplex Logic

OWN ($\overline{\text{OWN}}$). This three-state output indicates that the DMAC is controlling the bus. It is used as the enable signal to turn on the external address drivers and control signal buffers.

UPPER ADDRESS STROBE ($\overline{\text{UAS}}$). This three-state output is used as the gate signal to the transparent latches that capture the value of A8–A23 on the multiplexed address/data bus.

DATA BUFFER ENABLE ($\overline{\text{DBEN}}$). This three-state output is used as the enable signal to the external bidirectional data buffers.

DATA DIRECTION ($\overline{\text{DDIR}}$). This three-state output controls the direction of the external bidirectional data buffers. If $\overline{\text{DDIR}}$ is high, the data transfer is from the DMAC to the data bus. If $\overline{\text{DDIR}}$ is low, the data transfer is from the data bus to the DMAC.

HIGH BYTE ($\overline{\text{HIBYTE}}$). This three-state output indicates that data present on data lines D8–D15 must be transferred to data lines D0–D7, or vice versa, through an external buffer during a single address transfer between an 8-bit device and memory.

Bus Arbitration Control

These three signals form a bus arbitration circuit used to determine which device in a system will be the current bus master.

BUS REQUEST ($\overline{\text{BR}}$). This output is asserted by the DMAC to request control of the bus.

BUS GRANT ($\overline{\text{BG}}$). This input is asserted by an external bus arbiter to inform the DMAC that it may assume bus mastership as soon as the current bus cycle is completed. The DMAC will not take control of the bus until $\overline{\text{CS}}$, $\overline{\text{IACK}}$, $\overline{\text{AS}}$, and $\overline{\text{BGACK}}$ are all negated.

BUS GRANT ACKNOWLEDGE ($\overline{\text{BGACK}}$). This bidirectional signal is asserted by the DMAC to indicate that it is the current bus master. $\overline{\text{BGACK}}$ is monitored as an input to determine when the DMAC can become bus mater and if a bus master other than the system MPU is a master during limited-rate auto-request operation.

Interrupt Control

These two signals form an interrupt request/acknowledge handshake circuit with an MPU.

INTERRUPT REQUEST ($\overline{\text{IRQ}}$). This output is asserted by the DMAC to request service from the MPU.

INTERRUPT ACKNOWLEDGE ($\overline{\text{IACK}}$). This input is asserted by the MPU to acknowledge that it has received an interrupt from the DMAC. In response to the assertion of $\overline{\text{IACK}}$, the DMAC will place a vector on D0–D7 that will be used by the MPU to fetch the address of the proper DMAC interrupt handler routine.

Device Control

These signals perform the interface between the DMAC and four peripheral devices. Four sets of three lines are dedicated to a single DMAC channel and its associated peripheral; the remaining two lines are global signals shared by all channels.

REQUEST ($\overline{\text{REQ0}}$–$\overline{\text{REQ3}}$). These inputs are asserted by a peripheral device to request an operand transfer between that peripheral device and memory. In the cycle-steal request generation mode, these inputs are edge sensitive; in burst mode, they are level sensitive.

ACKNOWLEDGE ($\overline{\text{ACK0}}$–$\overline{\text{ACK3}}$). These outputs are asserted by the DMAC to signal to a peripheral that an operand is being transferred in response to a previous transfer request.

PERIPHERAL CONTROL LINE (PCL0–PCL3). These bidirectional lines are multipurpose signals that may be programmed to function as ready, abort, reload, status, interrupt, or enable clock inputs or as start pulse outputs.

DATA TRANSFER COMPLETE ($\overline{\text{DTC}}$). This output is asserted by the DMAC during any DMAC bus cycle to indicate that data has been successfully transferred (i.e., the bus cycle was not terminated abnormally).

DONE ($\overline{\text{DONE}}$). This bidirectional signal is asserted by the DMAC or a peripheral device during any DMA bus cycle to indicate that the data being transferred is the last item in a block. The DMAC will assert this signal during a bus cycle when the memory transfer count register is decremented to zero. On a linked array chaining mode, the $\overline{\text{DONE}}$ signal is asserted after all the blocks have been transferred. Because this signal has two purposes, make sure that a pullup resistor (1k–2k Ω) is tied to this signal to ensure that no interchannel interactions occur.

Clock (CLK)

The clock input is a TTL-compatible signal that is internally buffered for development of the internal clocks needed by the DMAC. The clock input should not be gated off at any time, and the clock signal must conform to minimum and maximum pulse-width times.

SIGNAL SUMMARY

Table 1 is a summary of all signals discussed in the previous paragraphs.

Table 1. Signal Summary

Signal Name	Direction	Active State	Driver Type	Synchronizing Clock Edge
A8/D0–A23/D15	In/Out	High	Three State	Falling**
A1–A7	In/Out	High	Three State	Falling**
FC0–FC2	Out	High	Three State	—
$\overline{CS}$	In	Low	—	Falling
$\overline{AS}$	In/Out	Low	Three State*	Falling
R/$\overline{W}$	In/Out	High/Low	Three State*	Falling
$\overline{UDS}$	In/Out	Low/High	Three State*	Falling
$\overline{LDS}$	In/Out	Low	Three State*	Falling
$\overline{DTACK}$	In/Out	Low	Three State*	Rising
$\overline{OWN}$	Out	Low	Open Drain*	—
$\overline{UAS}$	Out	Low	Three State*	—
$\overline{DBEN}$	Out	Low	Three State*	—
$\overline{DDIR}$	Out	High/Low	Three State*	—
$\overline{HIBYTE}$	Out	Low	Three State*	—
$\overline{BEC0}$–$\overline{BEC2}$	In	Low	—	Rising
$\overline{BR}$	Out	Low	Open Drain	—
$\overline{BG}$	In	Low	—	Rising
$\overline{BGACK}$	In/Out	Low	Open Drain*	—
$\overline{IRQ}$	In	Low	—	—
$\overline{IACK}$	In	Low	—	Falling
$\overline{REQ0}$–$\overline{REQ3}$	In	Low	—	Rising
$\overline{ACK0}$–$\overline{ACK3}$	Out	Low	Always Driven	—
PCL0–PCL3	In/Out	Programmed	Three State	Rising
$\overline{DTC}$	Out	Low	Three State*	—
$\overline{DONE}$	In/Out	Low	Open Drain	Rising
CLK	In	—	—	—

*These signals require a pullup resistor to maintain a high voltage when in the high-impedance or negated state. When these signals go to the high-impedance or negated state, they momentarily drive the pin high to reduce the signal rise time.

**These signals are latched on a clock edge but are not synchronized (i.e., the latched value is used immediately rather than being delayed by one clock).

REGISTER DESCRIPTION

Figure 11 shows the memory-mapped locations of the registers for each channel. Figure 12, the register summary, can be used as a quick reference to the bit definitions within each register.

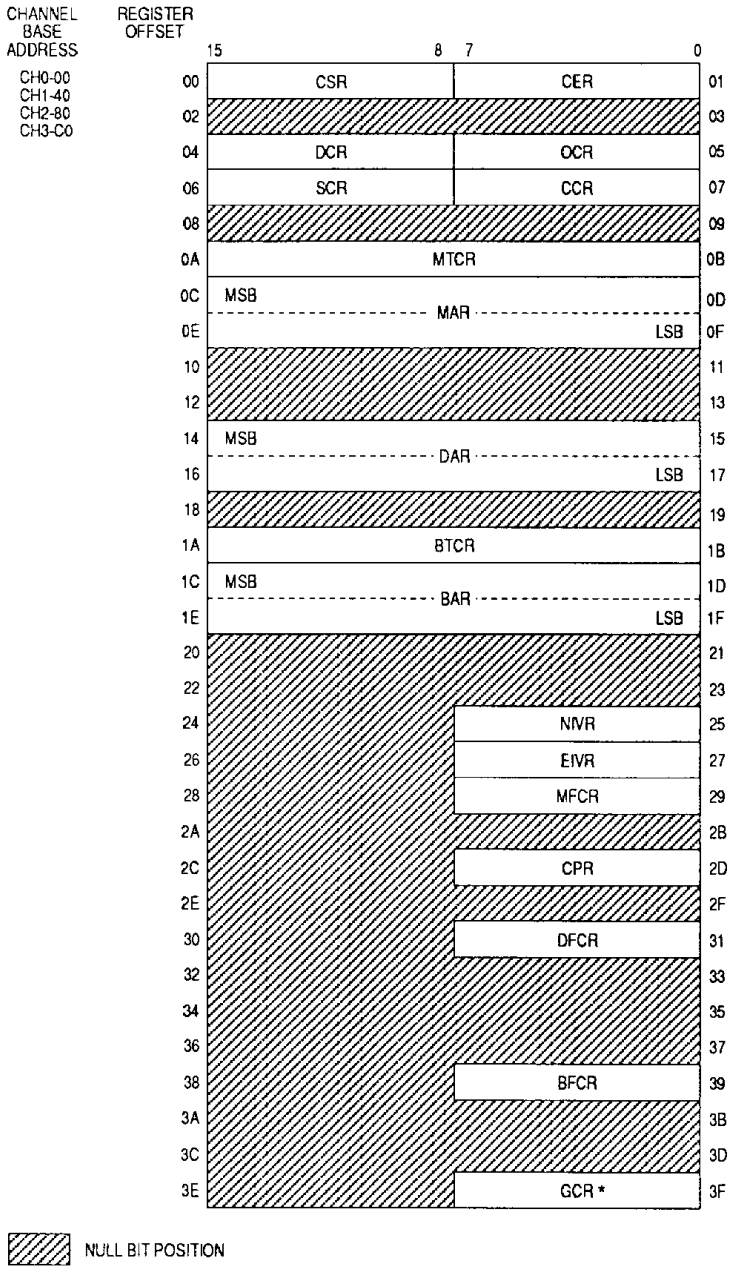


Figure 11. Register Memory Map

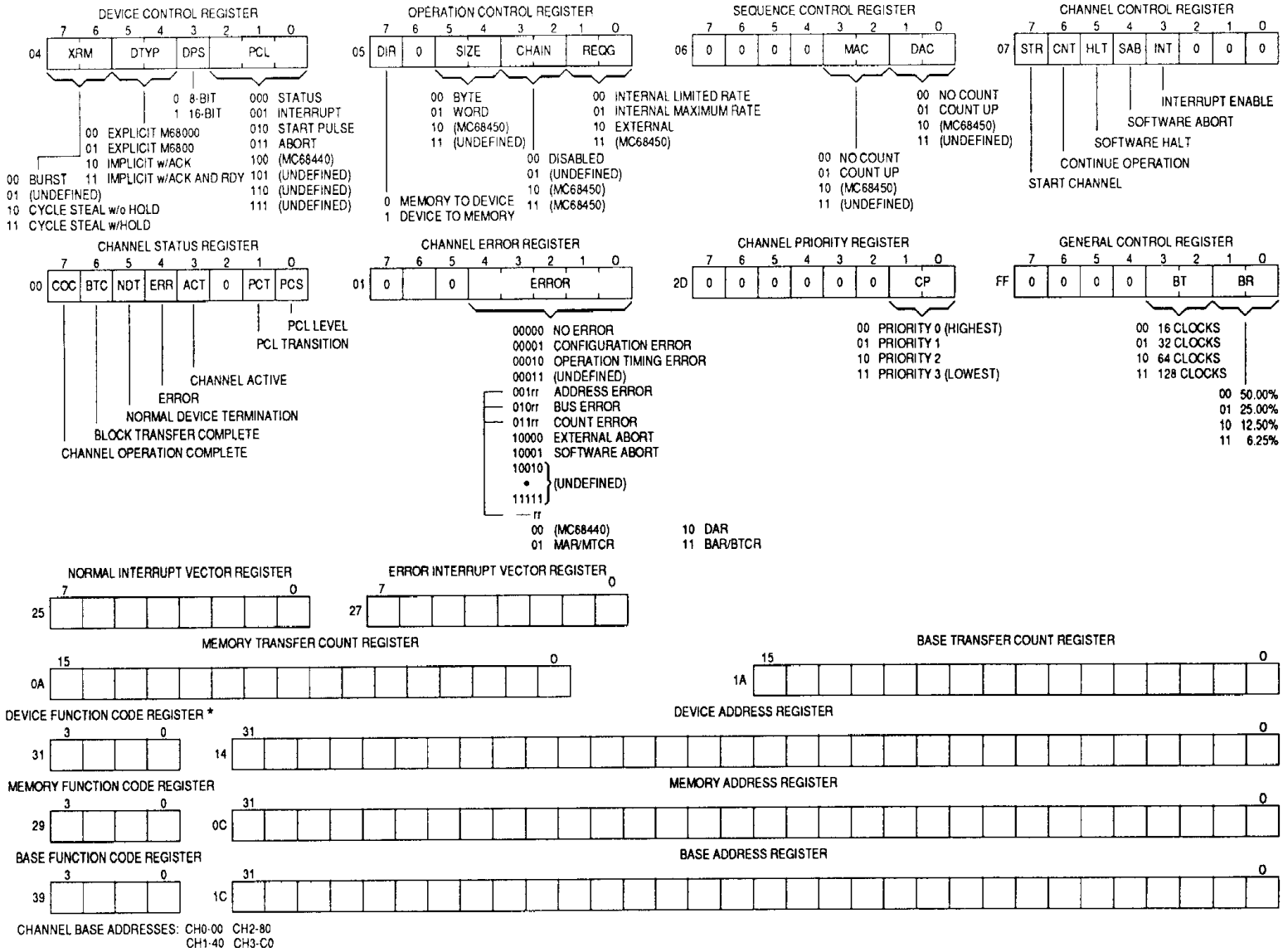


Figure 12. Register Summary

The register memory map for the MC68450 DMAC is identical to the register memory map for the MC68440 DDMA, including the individual bit assignments within the registers. However, not all functional options available on the DMAC are available on the DDMA and vice versa. If any programmable options labeled "MC68440 Reserved" or "Undefined, Reserved" are programmed into a DMAC channel, a configuration error occurs when the MPU attempts to start that channel.

All registers within the DMAC are always accessible as bytes or words by the MPU (assuming that the MPU can gain control of the DMA bus); however, some registers can not or should not be modified while a channel is actively transferring data. If a register can not be modified during operation and an attempt is made to write to it, an operation timing error is signaled and the channel operation is aborted.

RESET OPERATION RESULTS

When the DMAC is reset, either during a system power-up sequence or to reinitialize the DMAC, many of the registers will be affected and will be set to known values. Table 2 shows the hexadecimal value that will be placed in each register by a reset operation.

Table 2. Reset Operation Results

Register	Value	Comments
MARc	XXXXXXXX	Not Affected
DARc	XXXXXXXX	Not Affected
BARc	XXXXXXXX	Not Affected
MFCRc	X	Not Affected
DFCRc	X	Not Affected
BFCRc	X	Not Affected
MTCRc	XXXX	Not Affected
BRCRc	XXXX	Not Affected
NIVRc	0F	Uninitialized Vector
EIVRc	0F	Uninitialized Vector
CPRc	00	—
DCRc	00	—
OCRc	00	—
SCRc	00	—
CCRc	00	Channel Not Active, Interrupts Disabled
CSRc	00 or 01	(Depending on the Level of PCLc)
CERc	00	No Errors
GCR	00	—

X — Indicates an unknown value or the previous value of the register.
c — Indicates the channel number (i.e., 0, 1, 2, or 3).

ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either GND or V_{CC}).

THERMAL CHARACTERISTICS

Characteristic	Value		Rating
	θ_{JA}	θ_{JC}	
Thermal Resistance (Still Air)			°C/W
Ceramic (L/LC)	30	15*	
Plastic (P)	30	15*	
Pin Grid Array (R/RC)	30	15	

*Estimated

6

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{I/O}$
P_{INT} = $I_{CC} \times V_{CC}$, Watts—Chip Internal Power
$P_{I/O}$ = Power Dissipation on Input and Output Pins—User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

6

The total thermal resistance of a package (θ_{JA} can be separated into two components, θ_{JA} and θ_{CA}, representing the barrier to heat flow from the semiconductor junction to the package (case) surface (θ_{JC}) and from the case to the outside ambient (θ_{CA}). These terms are related by the equation:

$$\theta_{JA} = \theta_{JC} + \theta_{CA} \quad (4)$$

θ_{JC} is device related and cannot be influenced by the user. However, θ_{CA} is user dependent and can be minimized by such thermal management techniques as heat sinks, ambient air cooling, and thermal convection. Thus, good thermal management on the part of the user can significantly reduce θ_{CA} so that θ_{JA} aproximately equals θ_{JC}. Substitution of θ_{JC} for θ_{JA} in equation (1) will result in a lower semiconductor junction temperature.

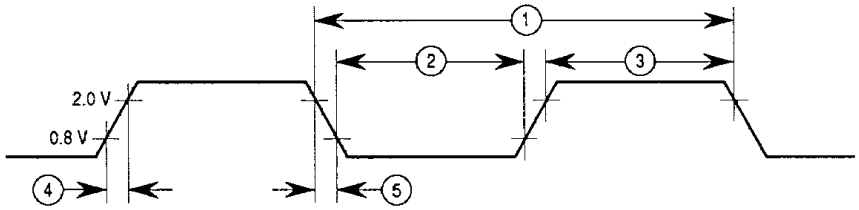
Values for thermal resistance presented in this document, unless estimated, were derived using the procedure described in Motorola Reliability Report 7843, "Thermal Resistance Measurement Method for MC68XX Microcomponent Devices," and are provided for design purposes only. Thermal measurements are complex and dependent on procedure and setup. User-derived values for thermal resistance may differ.

DC ELECTRICAL SPECIFICATIONS (T_A = 0°C to 70°C; V_{CC} = 5.0 V ±5%)

Characteristic	Symbol	Min	Max	Unit
Input High Voltage	V_{IH}	2.0	V_{CC}	V
Input Low Voltage	V_{IL}	GND −0.3	0.8	V
Input Leakage Current — $\overline{CS}$, $\overline{IACK}$, $\overline{BG}$, CLK, $\overline{BEC0}$–$\overline{BEC2}$, $\overline{REQ0}$–$\overline{REQ3}$	I_{in}	—	10	μA
Three-State (Off-State) Input Current — A1–A7, D0/A8–D15/A23, $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, R/$\overline{W}$, $\overline{UAS}$, $\overline{DTACK}$, $\overline{BGACK}$, $\overline{OWN}$, $\overline{DTC}$, $\overline{HIBYTE}$, $\overline{DDIR}$, $\overline{DBEN}$, FC0–FC2, PCL0–PCL3	I_{TSI}	—	20	μA
Input Capacitance (V_{in} = 0 V, T_A = 25°C, f = 1 MHz)	C_{in}	—	15	pF
Open-Drain (Off-State) Input Current — $\overline{IRQ}$, $\overline{DONE}$	I_{DD}	—	20	μA
Output High Voltage (I_{OH} = −400 μA) — A1–A7, D0/A8–D15/A23, $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, R/$\overline{W}$, $\overline{DTACK}$, $\overline{BGACK}$, $\overline{BR}$, $\overline{OWN}$, $\overline{DTC}$, $\overline{HIBYTE}$, $\overline{DDIR}$, $\overline{DBEN}$, $\overline{ACK0}$–$\overline{ACK3}$, PCL0–PCL3, FC0–FC2	V_{OH}	2.4	—	V
Output Low Voltage I_{OL} = 3.2 mA — A1–A7, FC0–FC2 I_{OL} = 5.3 mA — D0/A8–D15/A23, $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$, R/$\overline{W}$, $\overline{DTACK}$, $\overline{BR}$, $\overline{OWN}$, $\overline{DTC}$, $\overline{HIBYTE}$, $\overline{DDIR}$, $\overline{DBEN}$, $\overline{ACK0}$–$\overline{ACK3}$, $\overline{UAS}$, PCL0–PCL3, $\overline{BGACK}$ I_{OL} = 8.9 mA — $\overline{IRQ}$, $\overline{DONE}$	V_{OL}	—	0.5	V
Power Dissipation at 25°C (Frequency = 8 MHz)	P_D	—	2.0	W
Output Load Capacitance	C_L	—	130	pF

AC ELECTRICAL SPECIFICATIONS — CLOCK INPUT (see Figure 13)

Num.	Parameter	Symbol	8 MHz		10 MHz		Unit
			Min	Max	Min	Max	
	Frequency of Operation	f	2	8	2	10	MHz
1	Clock Time Period	t_{cyc}	125	500	100	500	ns
2,3	Clock Pulse Width	t_{CL}, t_{CH}	55	250	45	250	ns
4,5	Clock Rise and Fall Times	t_{Cf}, t_{Cr}	—	10	—	10	ns



NOTE: Timing measurements are referenced to and from a low voltage of 0.8 V and a high voltage of 2.0 V, unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall will be linear between 0.8 V and 2.0 V.

Figure 13. Clock Input Timing Diagram

AC ELECTRICAL SPECIFICATIONS — READ AND WRITE CYCLES

(V_{CC} = 15 V ± 5%, GND = 0 V, T_A = 0°C to 70°C, unless otherwise noted; see Figures 14–19)

Num.	Parameter	Symbol	8 MHz		10 MHz		Unit
			Min	Max	Min	Max	
6	Asynchronous Input Setup Time	t_{ASI}	20	—	15	—	ns
7	Data In to $\overline{DBEN}$ Low	t_{DIDBL}	0	—	0	—	ns
8	$\overline{DTACK}$ Low to Data In Invalid	t_{DTLDI}	0	—	0	—	ns
9	Address In to $\overline{AS}$ In Low	t_{AIASL}	0	—	0	—	ns
10	$\overline{AS}$ In High to Address In Invalid	t_{SIHAIV}	0	—	0	—	ns
11	Clock High to $\overline{DDIR}$ Low	t_{CHDRL}	—	70	—	60	ns
12	Clock High to $\overline{DDIR}$ High	t_{CHDRH}	—	70	—	60	ns
13	$\overline{DS}$ In High to $\overline{DDIR}$ High Impedance	t_{DSHDRZ}	—	120	—	110	ns
14	Clock Low to $\overline{DBEN}$ Low	t_{CLDBL}	—	70	—	60	ns
15	Clock Low to $\overline{DBEN}$ High	t_{CLDBH}	—	70	—	60	ns
16	$\overline{DS}$ In High to $\overline{DBEN}$ High Impedance	t_{DSHDBZ}	—	120	—	110	ns
17	Clock High to Data Out Valid (MPU Read)	t_{CHDVM}	—	180	—	160	ns
18	$\overline{DS}$ In High to Data Out Invalid	t_{DSHDZn}	0	—	0	—	ns
19	$\overline{DS}$ In High to Data High Impedance	t_{DSHDZ}	—	120	—	110	ns
20	Clock Low to $\overline{DTACK}$ Low	t_{CLDTL}	—	70	—	60	ns
21	$\overline{DS}$ In High to $\overline{DTACK}$ High	t_{DSHDTH}	—	110	—	110	ns
22	$\overline{DTACK}$ Width High	t_{DTH}	10	—	10	—	ns
23	$\overline{DS}$ In High to $\overline{DTACK}$ High Impedance	t_{DSHDTZ}	—	180	—	160	ns
24	$\overline{DTACK}$ Low to $\overline{DS}$ In High	t_{DTLDSH}	0	—	0	—	ns
25	$\overline{REQ}$ Width Low	t_{REQL}	2.0	—	2.0	—	Clk. Per.
26	$\overline{REQ}$ Low to $\overline{BR}$ Low	t_{RELBRL}	250	—	200	—	ns
27	Clock High to $\overline{BR}$ Low	t_{CHBRL}	—	70	—	60	ns
28	Clock High to $\overline{BR}$ High	t_{CHBRH}	—	70	—	60	ns
29	$\overline{BG}$ Low to $\overline{BGACK}$ Low	t_{BGLBL}	4.5	—	4.5	—	Clk. Per.
31	MPU Cycle End ($\overline{AS}$ In High) to $\overline{BGACK}$ Low	t_{ASHBL}	4.5	5.5	4.5	5.5	Clk. Per.
32	$\overline{REQ}$ Low to $\overline{BGACK}$ Low	t_{REQLBL}	12	—	12	—	Clk. Per.
33	Clock High to $\overline{BGACK}$ Low	t_{CHBL}	—	70	—	60	ns
34	Clock High to $\overline{BGACK}$ High	t_{CHBH}	—	70	—	60	ns
35	Clock Low to $\overline{BGACK}$ High Impedance	t_{CLBZ}	—	80	—	70	ns
36	Clock High to FC Valid	t_{CHFCV}	—	100	—	90	ns
37	Clock High to Address Valid	t_{CHAV}	—	120	—	110	ns
38	Clock High to Address/FC/Data High Impedance	t_{CHAZx}	—	100	—	100	ns
39	Clock High to Address/FC/Data Invalid	t_{CHAZn}	0	—	0	—	ns
40	Clock Low to Address High Impedance (Read)	t_{CLAZ}	—	100	—	90	ns

AC ELECTRICAL SPECIFICATIONS — READ AND WRITE CYCLES

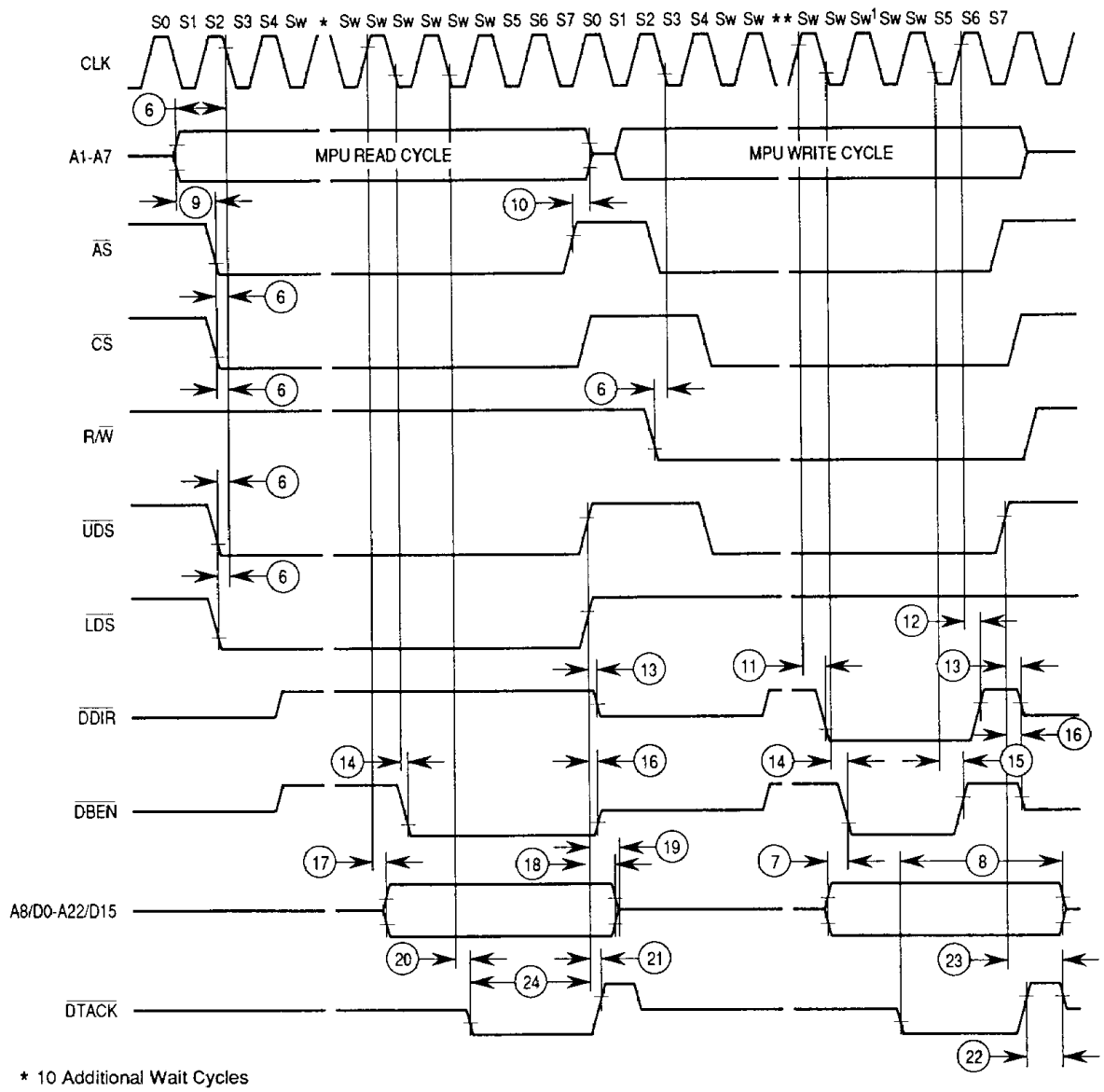
(Continued)

Num.	Parameter	Symbol	8 MHz		10 MHz		Unit
			Min	Max	Min	Max	
41	Clock High to $\overline{UAS}$ Low	t_{CHUL}	—	70	—	60	ns
42	Clock High to $\overline{UAS}$ High	t_{CHUH}	—	70	—	60	ns
43	Clock Low to $\overline{UAS}$ High Impedance	t_{CLUZ}	—	80	—	70	ns
44	$\overline{UAS}$ High to Address Invalid	t_{UHAI}	30	—	20	—	ns
45	Clock High to $\overline{AS}$, $\overline{DS}$ Low	t_{CHSL}	—	60	—	55	ns
46	Clock Low to $\overline{DS}$ Low (Write)	t_{CLDSL}	—	60	—	55	ns
47	Clock Low to $\overline{AS}$, $\overline{DS}$ High	t_{CLSH}	—	70	—	60	ns
48	Clock Low to $\overline{AS}$, $\overline{DS}$ High Impedance	t_{CLSZ}	—	80	—	70	ns
49	$\overline{AS}$ Width Low	t_{ASL}	255	—	195	—	ns
50	$\overline{DS}$ Width Low	t_{DSL}	255	—	190	—	ns
51	$\overline{AS}$, $\overline{DS}$ Width High	t_{SH}	150	—	105	—	ns
52	Address/FC Valid to $\overline{AS}$, $\overline{DS}$ Low (Read)	t_{AVSL}	30	—	20	—	ns
53	$\overline{AS}$, $\overline{DS}$ High to Address/FC/Data Invalid	t_{SHAZ}	30	—	20	—	ns
54	Clock High to R/$\overline{W}$ Low	t_{CHRL}	—	70	—	60	ns
55	Clock High to R/$\overline{W}$ High	t_{CHRH}	—	70	—	60	ns
56	Clock Low to R/$\overline{W}$ High Impedance	t_{CLRZ}	—	80	—	70	ns
57	Address/FC Valid to R/$\overline{W}$ Low	t_{AVRL}	20	—	10	—	ns
58	R/$\overline{W}$ Low to $\overline{DS}$ Low (Write)	t_{RLSL}	120	—	90	—	ns
59	$\overline{DS}$ High to R/$\overline{W}$ High	t_{SHRH}	40	—	20	—	ns
60	Clock Low to $\overline{OWN}$ Low	t_{CLOL}	—	70	—	60	ns
61	Clock Low to $\overline{OWN}$ High	t_{CLOH}	—	70	—	60	ns
62	Clock High to $\overline{OWN}$ High Impedance	t_{CHOZ}	—	80	—	70	ns
63	$\overline{OWN}$ Low to $\overline{BGACK}$ Low	t_{OLBL}	30	—	20	—	ns
64	$\overline{BGACK}$ High to $\overline{OWN}$ High	t_{BHOH}	30	—	20	—	ns
65	$\overline{OWN}$ Low to $\overline{UAS}$ Low	t_{OLUL}	30	—	20	—	ns
66	Clock High to $\overline{ACK}$ Low (Read)	t_{CHACL}	—	70	—	60	ns
67	Clock Low to $\overline{ACK}$ Low (Write)	t_{CLACL}	—	70	—	60	ns
68	Clock High to $\overline{ACK}$ High	t_{CHACH}	—	70	—	60	ns
69	$\overline{ACK}$ Low to $\overline{DS}$ Low	t_{ACLDSL}	100	—	80	—	ns
70	$\overline{DS}$ High to $\overline{ACK}$ High	t_{DSHACH}	30	—	20	—	ns
71	Clock High to $\overline{HIBYTE}$ Low	t_{CHHIL}	—	70	—	60	ns
72	Clock Low to $\overline{HIBYTE}$ Low	t_{CLHIL}	—	70	—	60	ns
73	Clock High to $\overline{HIBYTE}$ High	t_{CHHIH}	—	70	—	60	ns
74	Clock Low to $\overline{HIBYTE}$ High Impedance	t_{CLHIZ}	—	80	—	70	ns
75	Clock High to $\overline{DTC}$ Low	t_{CHDTL}	—	70	—	60	ns
76	Clock High to $\overline{DTC}$ High	t_{CHDTH}	—	70	—	60	ns
77	Clock Low to $\overline{DTC}$ High Impedance	t_{CLDTZ}	—	80	—	70	ns

AC ELECTRICAL SPECIFICATIONS — READ AND WRITE CYCLES
(Concluded)

Num.	Parameter	Symbol	8 MHz		10 MHz		Unit
			Min	Max	Min	Max	
78	$\overline{DTC}$ Width Low	t_{DTCL}	105	—	80	—	ns
79	$\overline{DTC}$ Low to $\overline{DS}$ High	t_{DTLDH}	30	—	20	—	ns
80	Clock High to $\overline{DONE}$ Low	t_{CHDOL}	—	70	—	60	ns
81	Clock Low to $\overline{DONE}$ Low	t_{CLDOL}	—	70	—	60	ns
82	Clock High to $\overline{DONE}$ High	t_{CHDOH}	—	130	—	120	ns
83	Clock High to $\overline{DDIR}$ High Impedance	t_{CLDRZ}	—	80	—	70	ns
84	Clock Low to $\overline{DBEN}$ High Impedance	t_{CLDBZ}	—	80	—	70	ns
85	$\overline{DDIR}$ Low to $\overline{DBEN}$ Low	t_{DRLDBL}	30	—	20	—	ns
86	$\overline{DBEN}$ High to $\overline{DDIR}$ High	t_{DBHDRH}	30	—	20	—	ns
87	$\overline{DBEN}$ Low to Address/Data High Impedance	t_{DBLAZ}	—	17	—	17	ns
88	Clock Low to PCL Low (1/8 Clock)	t_{CLPL}	—	70	—	60	ns
89	Clock Low to PCL High (1/8 Clock)	t_{CLPH}	—	70	—	60	ns
90	PCL Width Low (1/8 Clock)	t_{PCLL}	4.0	—	4.0	—	Clk. Per.
91	$\overline{DTACK}$ Low to Data In (Setup Time)	t_{DALDI}	—	150	—	115	ns
92	$\overline{DS}$ High to Data Invalid (Hold Time)	t_{SHDI}	0	—	0	—	ns
93	$\overline{DS}$ High to $\overline{DTACK}$ High	t_{SHDAH}	0	120	0	90	ns
94	Data Out Valid to $\overline{DS}$ Low	t_{DOSL}	0	—	0	—	ns
95	Data In to Clock Low (Setup Time)	t_{DICL}	15	—	15	—	ns
96	$\overline{BEC}$ Low to $\overline{DTACK}$ Low	t_{BECDAL}	50	—	50	—	ns
97	$\overline{BEC}$ Width Low	t_{BECL}	2.0	—	2.0	—	Clk. Per.
98	Clock High to $\overline{IRQ}$ Low	t_{CHIRL}	—	70	—	60	ns
99	Clock High to $\overline{IRQ}$ High Impedance	t_{CHIRH}	—	130	—	120	ns
100	PCL (as $\overline{READY}$) In to $\overline{DTC}$ Low (Read)	t_{RALDTL}	145	—	120	—	ns
101	PCL (as $\overline{READY}$) In to $\overline{DS}$ Low (Write)	t_{RALDSL}	205	—	170	—	ns
102	$\overline{DS}$ High to PCL (as $\overline{READY}$) High	t_{DSHRAH}	0	120	0	90	ns
103	$\overline{DONE}$ In Low to $\overline{DTACK}$ Low	t_{DOLDAL}	50	—	50	—	ns
104	$\overline{DS}$ High to $\overline{DONE}$ In High	t_{DSHDOH}	0	120	0	90	ns

6



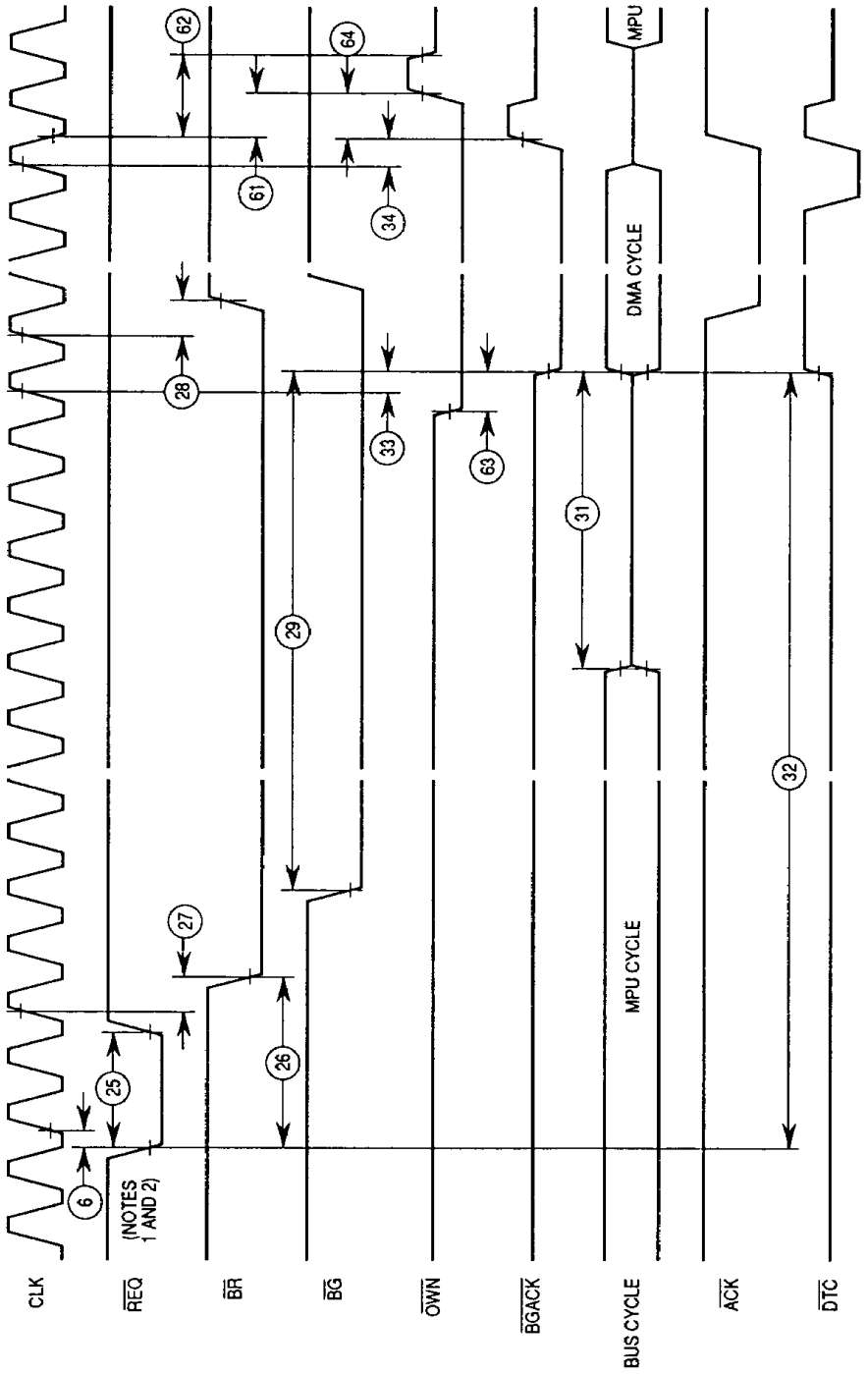
* 10 Additional Wait Cycles

** 8 Additional Wait Cycles

NOTES:

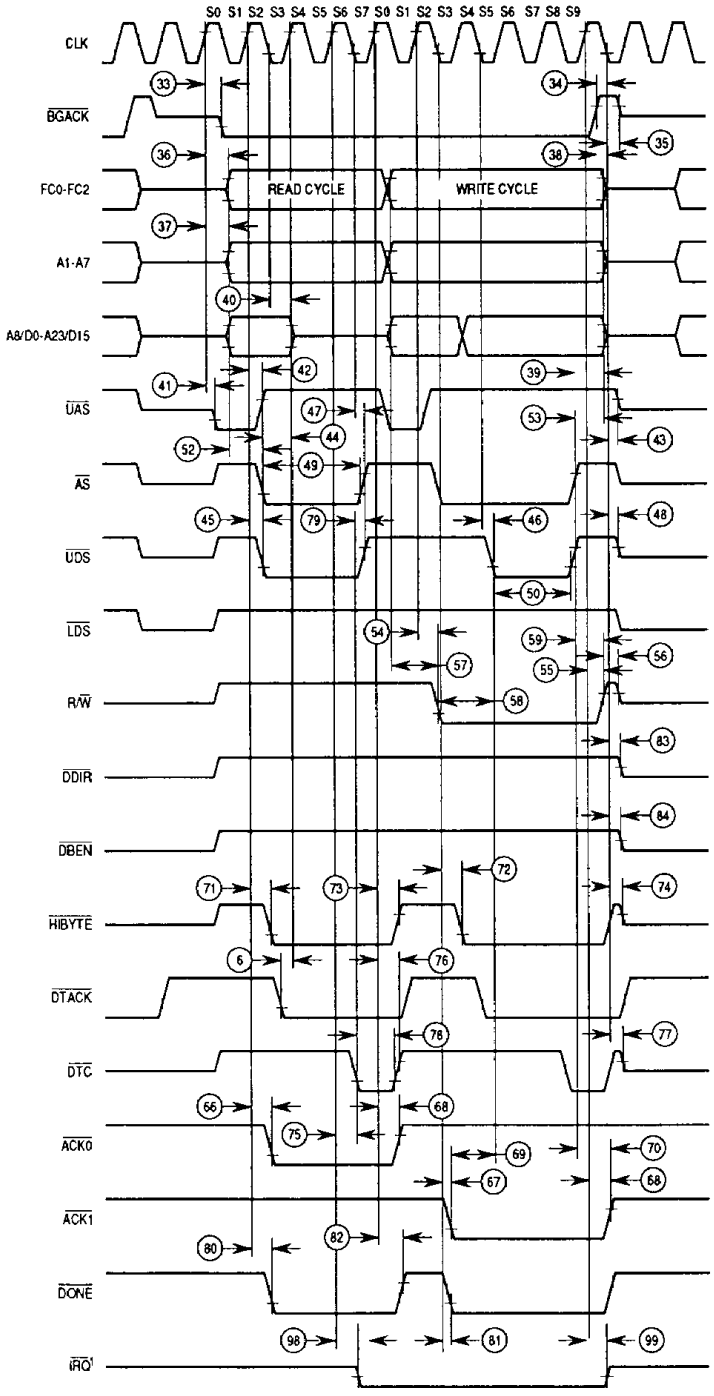
1. Data is latched at the end of this clock.
2. Timing measurements are referenced to and from a low voltage of 0.8 V and a high voltage of 2.0 V, unless otherwise noted.

Figure 14. MPU Read/Write Timing Diagram



NOTES:
1. $\overline{REQ}$ is sampled on the rising edge of CLK in cycle-steal and burst modes.
2. $\overline{BR}$ will not be asserted while any $\overline{BEC}$ exception condition exists or when $\overline{CS}$ or $\overline{IACK}$ is asserted.
3. Timing measurements are referenced to and from a low voltage of 0.8 V and a high voltage of 2.0 V, unless otherwise noted.

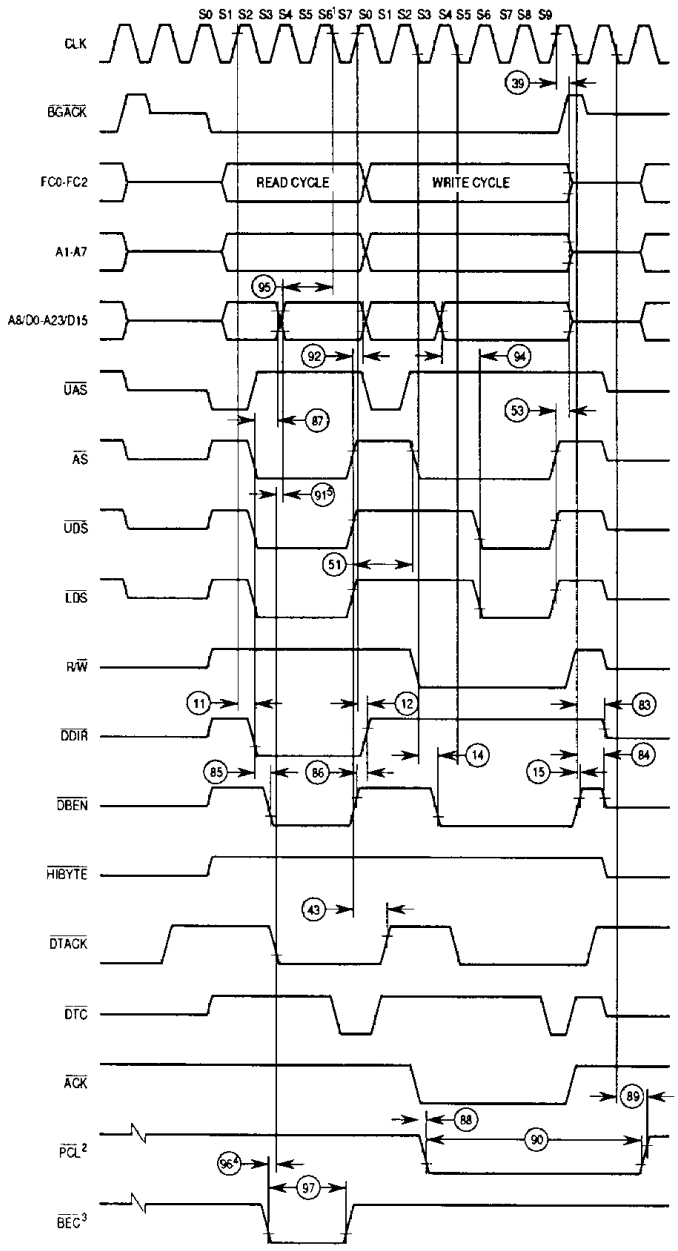
Figure 15. Bus Arbitration Timing Diagram



NOTES:

1. This timing is not directly related to the DMA read/write (single cycle) sequence.
2. Timing measurements are referenced to and from a low voltage of 0.8 V and a high voltage of 2.0 V, unless otherwise noted.

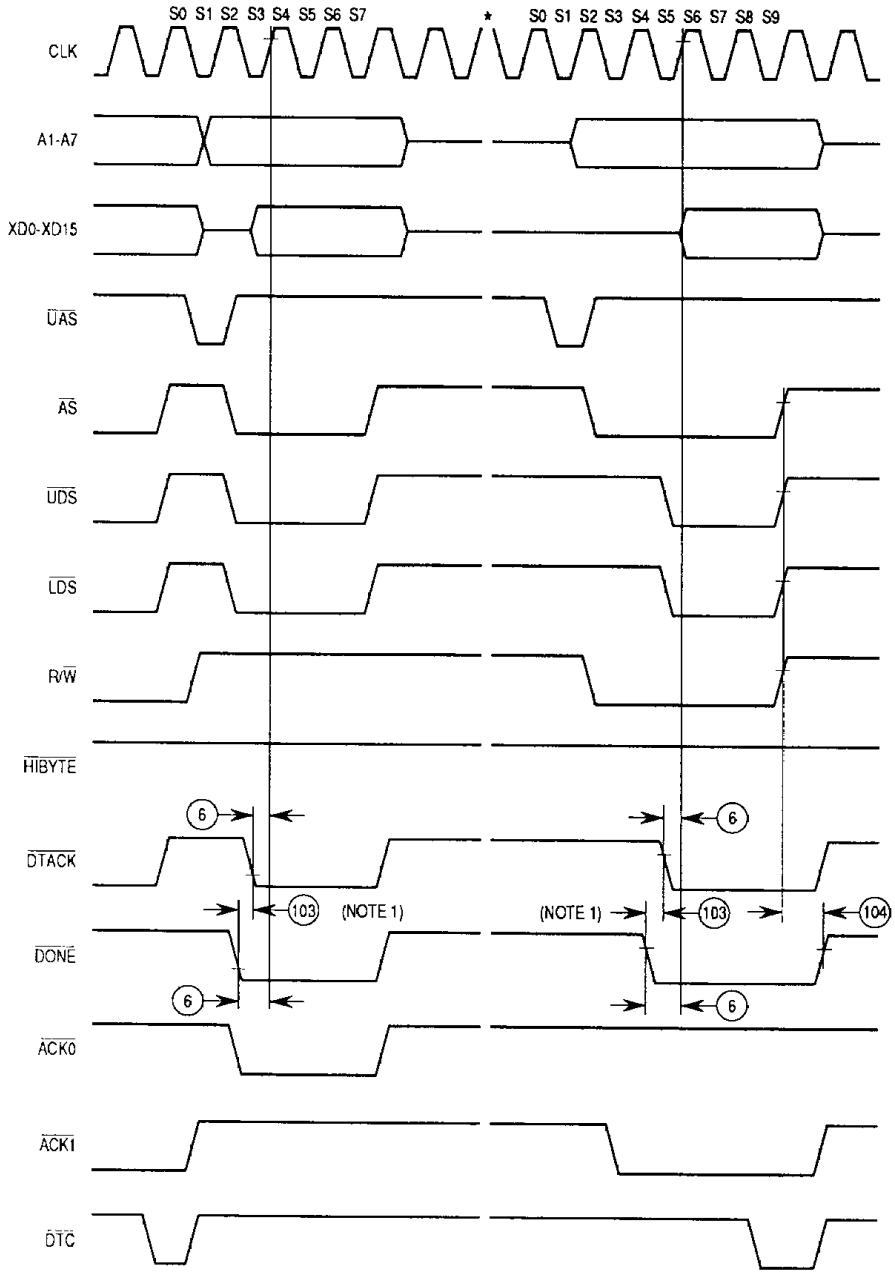
Figure 16. DMA Read/Write Timing Diagram (Single Cycle)



NOTES:

1. Data is latched at the end of clock S6.
2. Timing is not directly related to the DMA read/write (dual cycle) sequence and is only applicable when the start pulse mode is selected.
3. Timing is applicable when a bus exception occurs.
4. If specification number 6 is satisfied for both $\overline{DTACK}$ and $\overline{BEC}$, specification number 96 may be 0 ns.
5. If propagation delay of the external bidirectional buffer is less than 17 ns, a conflict may occur between the address output of the DMAC and the system data bus. In this case, the output of $\overline{DBEN}$ must be delayed externally.
6. Timing measurements are referenced to and from a low voltage of 0.8 V and a high voltage of 2.0 V, unless otherwise noted.

Figure 17. DMA Read/Write Timing Diagram (Dual Cycle)

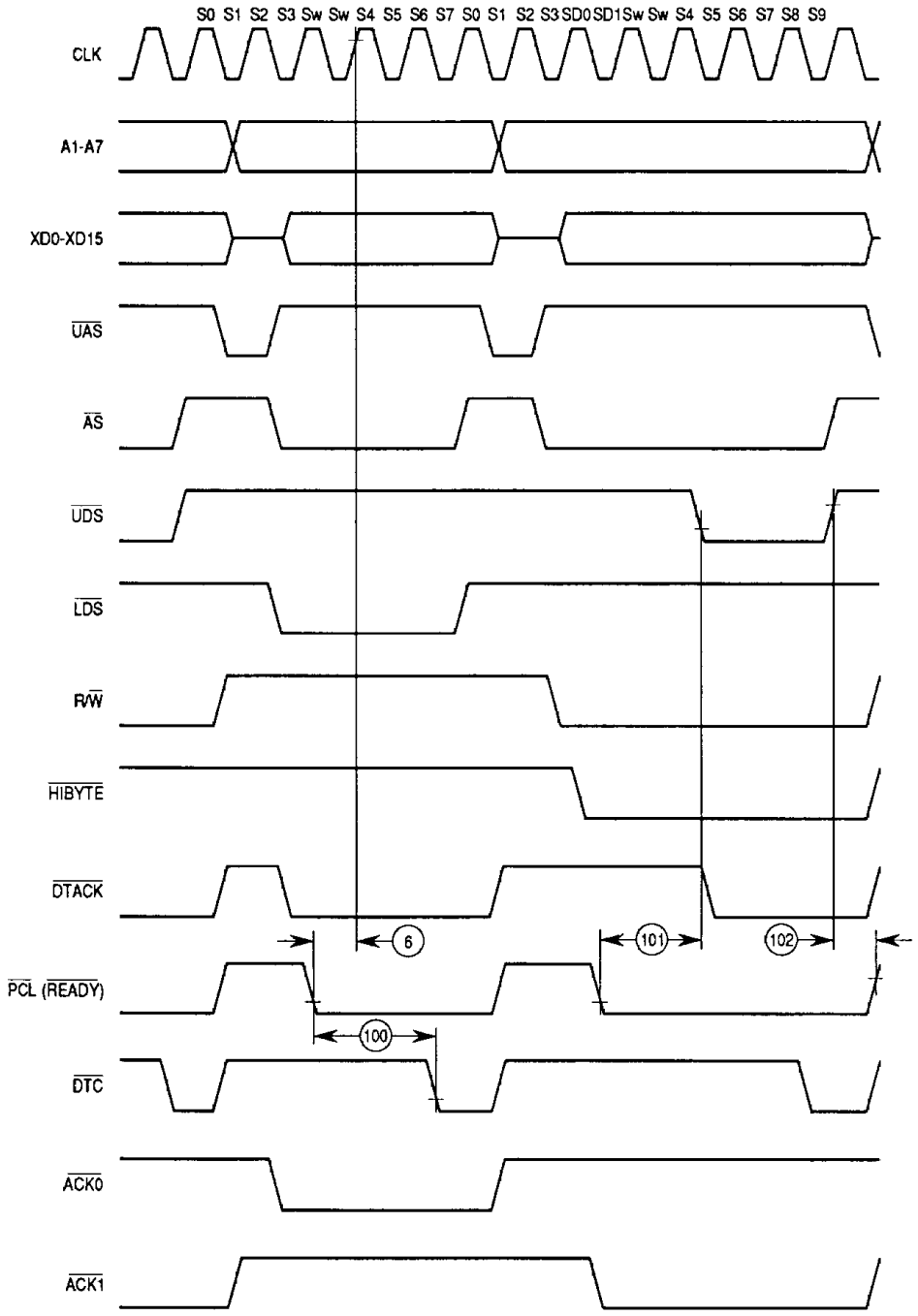


* 3 to 9 Additional Clocks

NOTES:

1. If specification number 6 is satisfied for both DTACK and DONE, specification number 103 may be 0 ns.
2. The setup time for asynchronous inputs BG, BGACK, CS, IACK, AS, UDS, LDS, and R/W guarantees their recognition at the next falling edge of the clock. The setup time for BEC0-BEC2, REQ0-REQ3, PCL0-PCL3, DTACK, and DONE guarantees their recognition at the next rising edge of the clock.
3. Timing measurements are referenced to and from a low voltage of 0.8 V and a high voltage of 2.0 V, unless otherwise noted.

Figure 18. DONE Input Timing Diagram

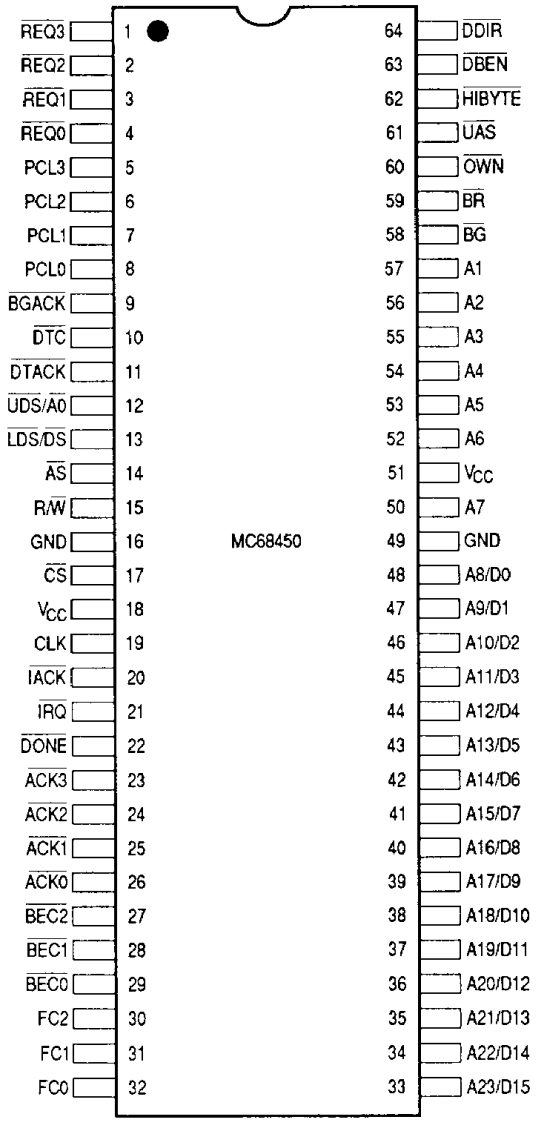


NOTE: Timing measurements are referenced to and from a low voltage of 0.8 V and a high voltage of 2.0 V, unless otherwise noted.

Figure 19. DMA Read/Write Timing Diagram (Single Cycle with PCL as READY)

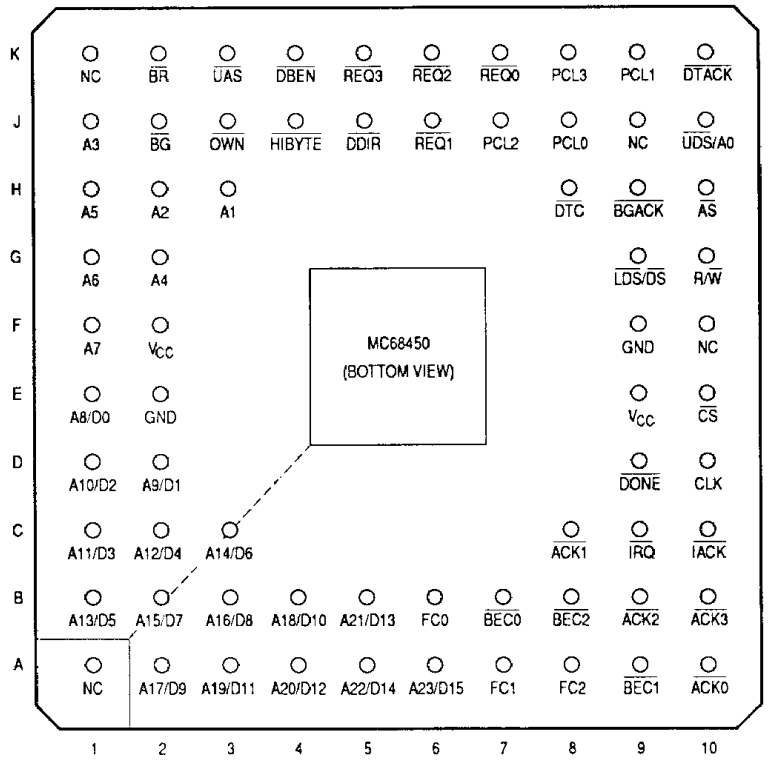
PIN ASSIGNMENTS

68-LEAD DUAL-IN-LINE PACKAGE



6

68-LEAD PIN GRID ARRAY



6