MOTOROLA
SEMICONDUCTORS
3501 ED BLUESTEIN BLVD., AUSTIN, TEXAS 78721

Advance Information

MC68451

MEMORY MANAGEMENT UNIT

APRIL, 1983

TABLE OF CONTENTS

Paragraph Number	Title	Page Number
	Section 1 Introduction	
1.1	Memory Segments	1-2
1.2	Function Codes and Address Spaces	1-2
1.3	Address Space Number	1-3
1.4	Descriptors	1-4
1.5	Translation	1-4
	Section 2 Signal Description	
2.1	V_{CC} and GND	2-1
2.2	Clock	2-2
2.3	Chip Select ($\overline{CS}$)	2-2
2.4	Register Selects (RS1-RS5)	2-2
2.5	Read/Write (R/$\overline{W}$)	2-2
2.6	$\overline{RESET}$	2-2
2.7	Data Transfer Acknowledge ($\overline{DTACK}$)	2-2
2.8	Upper and Lower Data Strobe ($\overline{UDS}$, $\overline{LDS}$)	2-2
2.9	Address Strobe ($\overline{AS}$)	2-3
2.10	Physical Address and Data (PAD0-PAD15)	2-3
2.11	Enable Data ($\overline{ED}$)	2-3
2.12	Hold Address ($\overline{HAD}$)	2-3
2.13	MODE	2-3
2.14	Interrupt Request ($\overline{IRQ}$)	2-3
2.15	Interrupt Acknowledge ($\overline{IACK}$)	2-4
2.16	$\overline{FAULT}$	2-4
2.17	Function Codes (FC0-FC3)	2-4
2.18	Address Bus (A8-A23)	2-4
2.19	Global Operation ($\overline{GO}$)	2-4
2.20	$\overline{ANY}$	2-4
2.21	ALL	2-5
2.22	Mapped Address Strobe ($\overline{MAS}$)	2-5
2.23	Write Inhibit ($\overline{WIN}$)	2-5
	Section 3 Register Description	
3.1	Descriptors	3-2
3.1.1	Logical Base Address (LBA)	3-2
3.1.2	Logical Address Mask (LAM)	3-2
3.1.3	Physical Base Address (PBA)	3-3

TABLE OF CONTENTS
(Continued)

Paragraph Number	Title	Page Number
3.1.4	Address Space Number (ASN)	3-3
3.1.5	Address Space Mask (ASM)	3-3
3.1.6	Segment Status Register (SSR)	3-3
3.2	System Registers	3-4
3.2.1	Address Space Table (AST)	3-4
3.2.2	Accumulator (AC0-AC8)	3-5
3.2.3	Global Status Register (GSR)	3-6
3.2.4	Local Status Register (LSR)	3-6
3.2.5	Descriptor Pointer (DP)	3-7
3.2.6	Result Descriptor Pointer (RDP)	3-8
3.2.7	Interrupt Descriptor Pointer (IDP)	3-8
3.2.8	Interrupt Vector Register (IVR)	3-9
	Section 4	
	MMU Functional Description	
4.1	Single-MMU Systems	4-1
4.2	Multiple-MMU Systems	4-1
4.3	MMU Timing	4-5
4.4	MMU Flow Diagram	4-5
4.5	MMU Functional States	4-5
4.5.1	The Reset State	4-10
4.5.2	The Idle State	4-11
4.5.3	Normal Address Translation	4-11
4.5.3.1	Matching	4-11
4.5.3.2	Successful Translation	4-11
4.5.3.3	Write Violation	4-12
4.5.3.4	No Address Match	4-12
4.5.3.5	External Translation	4-12
4.5.3.6	External Write Violation	4-13
4.5.3.7	Undefined Segment Access	4-14
4.6	MMU Operations	4-16
4.6.1	Operations Address Map	4-17
4.6.2	Local Operations	4-17
4.6.2.1	Interrupt Acknowledge	4-18
4.6.2.2	Read System Register	4-18
4.6.2.3	Transfer Descriptor	4-18
4.6.2.4	Write Segment Status Register	4-18
4.6.2.5	Null Operation	4-18
4.6.3	Global Operations	4-18
4.6.3.1	Write System Register	4-18
4.6.3.2	Load Descriptor Operation	4-23
4.6.3.3	Direct Translation	4-23
4.7	Operation Length Table	4-24

TABLE OF CONTENTS
(Concluded)

Paragraph Number	Title	Page Number
	Section 5 **Hardware Considerations**	
5.1	$\overline{\text{MAS}}$ Timing Modes and Physical Address	5-1
5.2	Physical Data Strobes	5-2
5.3	Interrupts	5-3
	Section 6 **Software Considerations**	
6.1	Segment Mapping Example	6-1
6.2	Segmentation	6-4
6.3	Virtual Memory Support	6-4
6.4	Initialization Software	6-4
6.5	Context Switching	6-5
	Section 7 **Electrical Specifications**	
7.1	Maximum Ratings	7-1
7.2	Thermal Characteristics	7-1
7.3	Power Considerations	7-1
7.4	AC Electrical Specifications	7-2
7.5	Electrical Characteristics	7-4
	Section 8 **Mechanical Data**	
8.1	Ordering Information	8-1
8.2	Pin Assignments	8-2
8.3	Package Dimensions	8-3

LIST OF ILLUSTRATIONS

Figure Number	Title	Page Number
1-1	Simplified Block Diagram of Single-MMU System	1-2
1-2	Address Space of MC68000	1-3
1-3	Memory Map of Typical Task Address Space	1-3
1-4	Schematic Diagram of Descriptor Mapping	1-4
1-5	Functional Block Diagram	1-5
3-1	Programmer's Model	3-1
3-2	Schematic Logic of Address Matching	3-2
3-3	Schematic Logic of Physical Address Generation	3-3
3-4	Address Space Table Organization	3-5
4-1	Single-MMU System with MC68000 or MC68010	4-2
4-2	Using the MC68008 with the MMU	4-3
4-3	Sample Circuit Diagram of a Two-MMU System with the MC68010	4-4
4-4	MMU Flow Diagram	4-6
4-5	Reset Flow Diagram	4-10
4-6	Normal Translation Timing in a Two-MMU System	4-13
4-7	Write Violation Timing	4-14
4-8	Undefined Segment Access Timing	4-15
4-9	Interrupt Acknowledge Timing	4-19
4-10	Read System Register Timing	4-20
4-11	Global Write Systems Register Timing	4-21
4-12	Successful Load Descriptor Operation in a Two-MMU System	4-22
4-13	Timing for Successful Direct Translation in a Two-MMU System	4-25
5-1	Physical Address Strobe Generation	5-2
5-2	Generation of Physical Read/Write and Physical Data Strobes	5-2
5-3	Circuit to Inhibit Translation during $\overline{IACK}$	5-3
6-1	Address Map Example	6-2
6-2	Translation Example	6-3
7-1	Test Loads	7-2
7-2	Normal Translation Timing	Foldout 1
7-3	Operations Timing	Foldout 2
7-4	Miscellaneous Signal Timing	Foldout 2

LIST OF TABLES

Table Number	Title	Page Number
3-1	Accumulator Assignments for Operations	3-5
4-1	Length of Operations	4-9
4-2	Summary of MMU Functions	4-16
4-3	Register/Operations Address Map	4-17

SECTION 1
INTRODUCTION

The MC68451 memory management unit (MMU) provides address translation and protection for the 16 megabyte addressing range of the MC68000 MPU. Each bus master (or processor) in the M68000 Family provides a function code and an address during each bus cycle. The function code specifies an address space and the address specifies a location within that address space. The function codes distinguish between user and supervisor spaces and, within these, between data and program spaces. This separation of address spaces provides the basis for memory management and protection by the operating system. Provision is also made for other bus masters, such as the MC68450 DMAC, to have separate address spaces for logical DMA. A multitasking operating system is simplified and reliability is enhanced through the use of the MMU.

Features of the MC68451 include:

- MC68451F — Faster Translation Times
- Compatible with MC68000 and MC68008
- Provides Virtual Memory Support for the MC68010
- Provides Efficient Memory Allocation
- Separates Address Spaces of System and User Resources
- Provides Write Protection
- Supports Paging and Segmentation
- 32 Segments of Variable Size with Each MMU
- Multiple MMU Capability to Expand to Any Number of Segments
- Allows Inter-Task Communication through Shared Segments
- Quick Context Switching to Cut Operating System Overhead
- Simplifies Programming Model of Address Space
- Increases System Reliability
- DMA Compatible

The MC68451 memory management unit (MMU) is the basic element of a memory management mechanism (MMM) in an M68000 Family system. The operating system is responsible for insuring the proper execution of user tasks in the system environment and memory management is basic to this responsibility. The MMM provides the operating system with the capability to allocate, control, and protect the system memory. A block diagram of a single-MMU system is shown in Figure 1-1.

An MMM, implemented with one or more MC68451 MMUs, can provide address translation, separation, and write protection for the system memory. The MMM can be programmed to cause an interrupt when a chosen section of memory is accessed and can directly translate a logical address into a physical address making it available to the MPU for use by the operating system. Using these features, the MMM can provide separation and security for user programs and allow the operating system to manage the memory in an efficient fashion for multitasking.

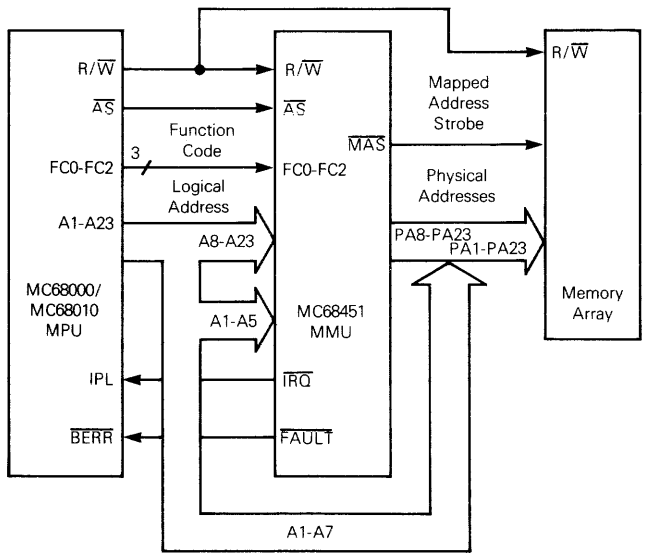


Figure 1-1. Simplified Block Diagram of Single-MMU System

1.1 MEMORY SEGMENTS

The MMM partitions the logical address space into contiguous pieces called segments. Each segment is a section of the logical address space of a task which is mapped via the MMM into the physical address space. Each task may have any number of segments. Segments may be defined as user or supervisor, data-only or program-only, or program and data. They may be accessed by only one task or shared between two or more tasks. In addition, any segment can be write protected to insure system integrity. A fault (MC68000 bus error) is generated by the MMM if an undefined segment is accessed.

1.2 FUNCTION CODES AND ADDRESS SPACES

Each bus master in the M68000 Family (including the MC68440 DMA controllers) provides a function code during each bus cycle to indicate the address space to be used for that cycle. The address bus then specifies a location within this address space for the operation taking place during that bus cycle.

The function codes appear on the FC0-FC2 lines of the MC68000 and divide the memory references into two logical address spaces — the supervisor and the user spaces. Each of these is further divided into program and data spaces. A separate address space is also provided for internal CPU-related activities such as interrupt acknowledge, giving a total of five defined function codes. The address space of the MC68000 is shown in Figure 1-2.

In addition to the 3-bit function code provided by the MC68000, the MC68451 MMU also allows a fourth bit (FC3) which provides for the possibility of another bus master in the system. In this case, FC3 would be a function of bus grant acknowledge ($\overline{\text{BGACK}}$) of the MC68000 to enable a second

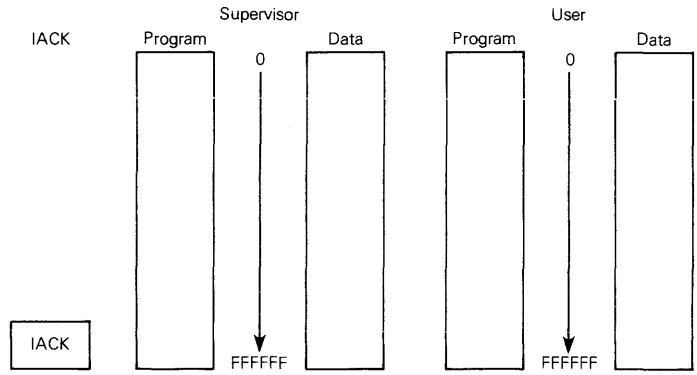


Figure 1-2. Address Space of MC68000

set of eight function codes. This raises the total number of possible function codes to sixteen. If there is only one bus master (the MPU), the FC3 pin on the MMU should be tied low and only eight address spaces can then be used.

1.3 ADDRESS SPACE NUMBER

Each task in a system has an address space which is comprised of all the segments defined for that task. This address space is assigned a number by programming all the address space number (ASN) fields in its descriptors with the same value. This value can be considered a task number. The currently active task's number is kept in the appropriate entry(s) in the address space table (AST).

The AST is a set of MMU registers which defines which task's segments are to be used in address translation for each cycle type (supervisor program, supervisor data, etc). The AST contains an 8-bit entry for each possible function code. Each entry is assigned an ASN (task number) and this is used to select which descriptors may be used for translation. The logical address is then translated by one of these to produce the physical address. Figure 1-3 is a typical memory map of a task's address space.

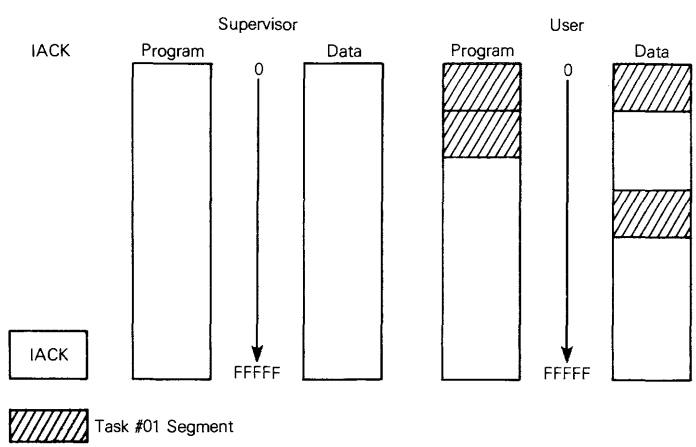


Figure 1-3. Memory Map of Typical Task Address Space

1.4 DESCRIPTORS

Address translation is done using descriptors. A descriptor is a set of six registers (nine bytes) which describe a memory segment and how that segment is to be mapped to the physical addresses. Each descriptor contains base addresses for the logical and physical spaces of each segment. These base addresses are then masked with the logical address masks. The size of the segment is then defined by "don't cares" in the masks. This method allows segment sizes from a minimum of 256 bytes to a maximum of 16 megabytes in binary increments (i.e., powers of two). This also forces both logical and physical addresses of segment boundaries to lie on a segment size boundary. That is, a segment can only start on an address which is a multiple of 2^K. The segments can be defined in such a way to allow them to be logically or physically shared between tasks. Descriptor mapping is shown schematically in Figure 1-4.

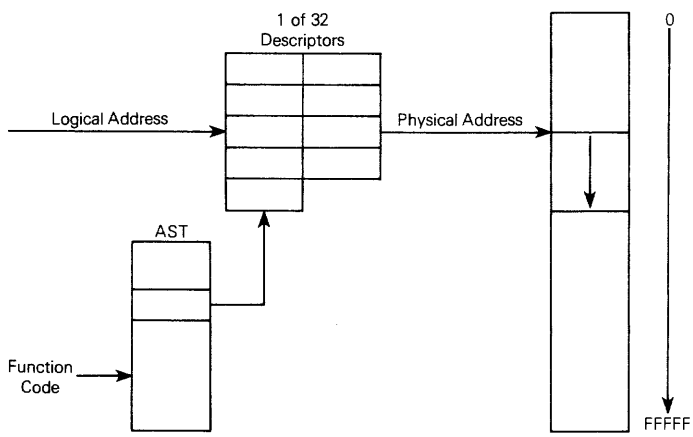


Figure 1-4. Schematic Diagram of Descriptor Mapping

1.5 TRANSLATION

During normal translation, the MMU translates the logical address provided by the MC68000 to produce a physical address which is then presented to the memory array. This is accomplished by matching the logical address with the information in the descriptors and then mapping it into the physical address space. A block diagram of the MC68451 MMU is shown in Figure 1-5.

Refer to Figure 1-1 for the following. The logical address is composed of address lines A1-A23. The upper 16 bits of this address (A8-A23) are translated by the MMU and mapped into a physical address (PA8-PA23). The lower seven bits of the logical address (A1-A7) bypass the MMU and become the low-order physical address bits (PA1-PA7). In addition, the data strobes ($\overline{UDS}$ and $\overline{LDS}$) remain unmapped to become the physical data strobes for a total of eight unmapped address lines.

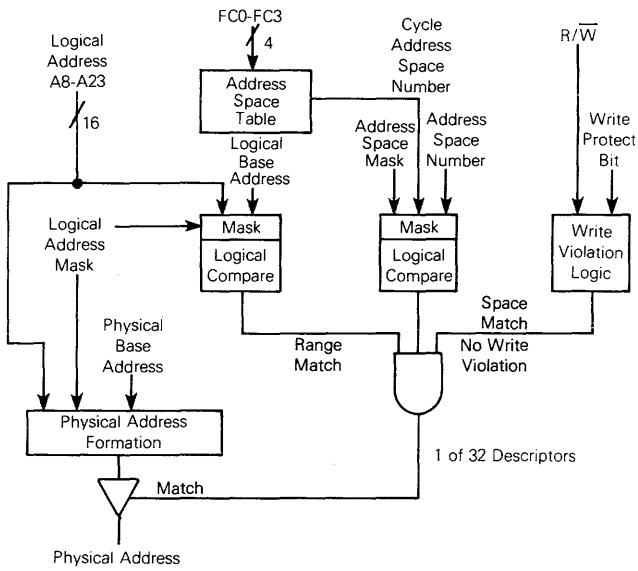


Figure 1-5. Functional Block Diagram

SECTION 2
SIGNAL DESCRIPTION

This section contains a brief description of the input and output signals.

NOTE

Throughout this document, active low signals are denoted by a superscript bar. This does not imply logical negation. To avoid confusion, a signal in its true state is said to be asserted whether that signal is active high or low. It is said to be negated when it is in its functionally inactive state. A signal which can be placed in the high-impedance state is said to be three-stated. A signal line which is first negated and then placed in the high-impedance state is said to be rescinded.

Some MMU signal lines are classed as input/output meaning that the bus buffers can be directed inward to input information into the MMU or outward to drive the bus. These signals must be either inputs or outputs at any given time, they may not be both. An example is the PAD port.

Still other types of signals can logically be both inputs or outputs at the same time. The internal signal controller may assert or negate a signal and read it at the same time. To distinguish between them, the suffix "in" will be used to denote the input signal and the suffix "out" will be used for the output signal.

Six pins on the MMU have this property – $\overline{IRQ}$, $\overline{FAULT}$, $\overline{MAS}$, $\overline{GO}$, $\overline{ANY}$, and ALL. In a multiple-MMU system, they would be wired in parallel to provide a single-signal level for the entire system. Of these, three pins – $\overline{IRQ}$, $\overline{FAULT}$, and $\overline{ANY}$ – are active low, wire-OR type signals. As such, asserting any one of the parallel pins will drive the line low (true). If any of these signals are asserted on any MMU in the system, they will be detected as asserted on the corresponding "in" pin of all MMUs in the system.

The ALL pin is an active-high open-drain gate and, as such, all pins are wire-ANDed and must be pulled high in order for the input to be high. Therefore, even if ALLout is asserted by an MMU, ALLin will not be detected true by any MMU unless all of the corresponding pins on all MMUs in the system are asserted.

The $\overline{GO}$ and $\overline{MAS}$ pins are not open drain but they can be put in the high-impedance state. They should each be wired in parallel on all MMUs in the system since only one MMU at any given time will assert these signals. A pullup resistor is required to hold the signal inactive when the pin is in the high-impedance state.

2.1 V_{CC} AND GND

These pins supply power to the MMU. The two V_{CC} pins are +5 volts ±5% and the two GND pins are ground.

2.2 CLOCK

This TTL input signal must be the MC68000 system clock and must not be gated off at any time.

2.3 CHIP SELECT ($\overline{CS}$)

$\overline{CS}$ is an input used to activate the MMU for accesses to the registers and other MMU operations. The assertion of $\overline{CS}$, in conjunction with the address of a global operation on pins RS1-RS5, selects the MMU to be a master for that operation. $\overline{CS}$ should be decoded from the physical address bus to protect the MMU registers from unauthorized access. See **4.6 MMU OPERATIONS.**

2.4 REGISTER SELECTS (RS1-RS5)

These five input pins should be the lower five bits of the physical address bus. When $\overline{CS}$ is asserted, these pins select the operation to be performed and the register involved (if any). See Table 4-3 for the operations address map.

2.5 READ/WRITE (R/$\overline{W}$)

The R/$\overline{W}$ input signal controls the direction of the data bus during an MMU operation. It is also used to compare against the matched descriptor to determine if a write violation has occurred during translation.

2.6 $\overline{RESET}$

Asserting the $\overline{RESET}$ input will reset the MMU regardless of what state it is in. The $\overline{RESET}$ pin must be held low for at least 16 clock cycles to reset the MMU. During power up, the $\overline{RESET}$ pin must be held low for at least 100 milliseconds after V_{CC} is established and the clock signal is present. See **4.5.1 The Reset State.**

2.7 DATA TRANSFER ACKNOWLEDGE ($\overline{DTACK}$)

The MMU uses this rescindable output to signal the completion of the operation phase of a bus cycle to the processor. If the bus cycle is a processor read, the MMU asserts $\overline{DTACK}$ to indicate that the information on the data bus is valid. If the bus cycle is a processor write to the MMU, $\overline{DTACK}$ is used to acknowledge acceptance of the data by the MMU. $\overline{DTACK}$ may be asserted only by an MMU that has $\overline{CS}$ or $\overline{IACK}$ asserted.

2.8 UPPER AND LOWER DATA STROBES ($\overline{UDS}$, $\overline{LDS}$)

The $\overline{UDS}$ and $\overline{LDS}$ inputs are used during MMU operation (processor access of MMU registers) to indicate which byte of the data bus is to be used. The assertion of the upper data strobe indicates that the operation is to be performed at an even address using the upper byte of the data bus. The assertion of lower data strobe indicates that the operation is to be performed at an odd address using the lower byte of the data bus. During a processor write operation, the data strobes indicate to the MMU that valid data is on the data bus.

2.9 ADDRESS STROBE ($\overline{AS}$)

This input signal indicates to the MMU that a bus cycle is in progress, and that there is a valid address on the logical address bus. The assertion of $\overline{AS}$ initiates the normal translation phase of the bus cycle.

2.10 PHYSICAL ADDRESS AND DATA (PAD0-PAD15)

During MMU operations, these 16 multiplexed input/output pins function as the data bus used to transfer data to and from the MMU. During normal translation, the physical address PA8-PA23 is gated out on this bus. External octal data transceivers are used to isolate the system data bus from the physical address bus. The MMU provides the enable data ($\overline{ED}$) signal to control these transceivers. See **2.11 ENABLE DATA** for more detail.

2.11 ENABLE DATA ($\overline{ED}$)

The enable data signal is a three-state output signal used to control the external bus transceivers on the PAD0-PAD15 bus. When $\overline{ED}$ is asserted, the transceivers should be enabled (i.e., they should drive the bus). When $\overline{ED}$ is negated, the transceivers should be in the high-impedance state. R/$\overline{W}$ is used to control the direction of the transceivers. Only the MMU with $\overline{CS}$ or $\overline{IACK}$ asserted will assert $\overline{ED}$.

NOTE

A pair of SN74LS245 type data transceivers may be used. $\overline{ED}$ will drive the output enable pin with no additional logic. See the circuit diagram in Section 4 (Figure 4-1).

2.12 HOLD ADDRESS ($\overline{HAD}$)

$\overline{HAD}$ is a rescindable output used to control an external latch on the physical address bus. After normal translation, $\overline{HAD}$ is asserted to hold the physical address stable. The latch should be of the transparent type such as an SN74LS373. $\overline{HAD}$ can directly interface with the enable pin of this type of latch. To provide address hold time, $\overline{HAD}$ is rescinded after $\overline{MAS}$ is rescinded.

2.13 MODE

MODE is a three-level input used to program the mode of operation of the $\overline{MAS}$ signal. There are three modes of operation: A, S1, and S2.

Mode A is selected by leaving the MODE pin unconnected. Mode S1 is selected by tying the MODE pin to V_{CC} and mode S2 is selected by tying the MODE pin to ground. For description of the different modes, see **SECTION 5 HARDWARE CONSIDERATIONS.**

2.14 INTERRUPT REQUEST ($\overline{IRQ}$)

$\overline{IRQ}$ is an open-drain input/output signal. $\overline{IRQ}$out is used to request an interrupt of the MPU. $\overline{IRQ}$out is asserted if a descriptor in which the I (interrupt) bit is set, is matched in normal translation, and the interrupt enable (IE) bit in the global status register (GSR) is set. Clearing all IP (interrupt pending) bits in all segment status registers or clearing IE in the GSR will cause $\overline{IRQ}$ to be negated. If $\overline{IRQ}$in and $\overline{IACK}$, and $\overline{UDS}$ and/or $\overline{LDS}$, are asserted, the MMU will perform an interrupt

acknowledge operation. See **4.6 MMU OPERATIONS.** The $\overline{IRQ}$ lines of all MMUs should be wire-ORed together and tied to V_{CC} through a pullup resistor. They should be isolated from the $\overline{IRQ}$ lines of other devices to prevent an MMU from detecting an erroneous interrupt.

2.15 INTERRUPT ACKNOWLEDGE ($\overline{IACK}$)

An MMU will begin the interrupt acknowledge operation if $\overline{IRQ}$in and $\overline{IACK}$ are both asserted. The interrupt vector supplied by the interrupt vector register (IVR) is placed on the data bus for the MPU. Only one MMU should have its $\overline{IACK}$ pin tied to the $\overline{IACK}$ circuitry from the processor; all other MMUs should have this pin tied high (inactive).

2.16 $\overline{FAULT}$

$\overline{FAULT}$ is an open-drain input/output signal. During normal translation, if an MMU detects a write violation or an undefined segment access, it asserts the $\overline{FAULT}$ line for five clock cycles or until $\overline{AS}$ becomes negated, whichever is longer. If an MMU detects $\overline{FAULT}$in asserted, it updates its global and local status registers to reflect this. The $\overline{FAULT}$ lines of all MMUs in the system should be wire-ORed and tied to V_{CC} through a pullup resistor. The $\overline{FAULT}$ signals can be connected directly to the MC68000 $\overline{BERR}$ pin but they should be isolated from any other bus error signals in the system to prevent the MMUs from detecting an erroneous $\overline{FAULT}$.

2.17 FUNCTION CODES (FC0-FC3)

The function code inputs specify the type of bus cycle being executed by the current bus master. The function code indicates which address space is to be used for that cycle and is used to index into the address space table for the cycle address space number used in descriptor matching. See **1.2 FUNCTION CODES AND ADDRESS SPACES.**

2.18 ADDRESS BUS (A8-A23)

These inputs are the upper 16 bits of the MPU address bus. They form the logical address which the MMU translates into 16 physical-address bits. The lower seven address lines (A1-A7) bypass the MMU.

2.19 GLOBAL OPERATION ($\overline{GO}$)

$\overline{GO}$ is a rescindable input/output inter-MMU signal used in multiple-MMU systems to indicate or detect global operations. If $\overline{CS}$ is asserted and the operation to be performed is global, the MMU is selected as the master MMU for that operation. $\overline{GO}$out is then asserted by the master MMU to signal other MMUs that they are slaves in a global operation. Thus, if $\overline{GO}$in is asserted while $\overline{CS}$ is negated, the MMU is selected as a slave for that operation. For a detailed description, see **4.6 MMU OPERATIONS.**

2.20 $\overline{ANY}$

The $\overline{ANY}$ signal is an open-drain input/output inter-MMU handshake signal for multiple-MMU systems. A slave MMU asserts $\overline{ANY}$out during a global operation if it has a local event to report which is significant if it occurs in one, but not necessarily all, MMUs. The $\overline{ANY}$ pins are wire-ORed and require a pullup resistor to V_{CC}.

2.21 ALL

The ALL pin is an open-drain input/output similar to $\overline{ANY}$ except that it reports events that are significant only when they occur in all MMUs. The ALL pins are wire-ANDed together and require a pullup resistor to V_{CC}.

2.22 MAPPED ADDRESS STROBE ($\overline{MAS}$)

$\overline{MAS}$ is a rescindable input/output signal. $\overline{MAS}$out is asserted by an MMU if an address match occurs during normal translation. This indicates that a valid physical address is present at the PAD0-PAD15 outputs. $\overline{MAS}$in is used by an MMU to detect a successful translation by another MMU. $\overline{MAS}$ can be programmed to operate in an asynchronous or synchronous mode by the MODE pin.

2.23 WRITE INHIBIT ($\overline{WIN}$)

$\overline{WIN}$ is a rescindable output provided to protect write-protected segments during read-modify-write bus cycles. Normally, write-protected segments are protected by the assertion of $\overline{FAULT}$ and the withholding of $\overline{MAS}$ upon detection of an attempted write to that segment. However, during read-modify-write bus cycles, $\overline{AS}$ remains asserted, making it difficult to prevent the write portion of the instruction from writing to the protected segment. To provide write protection during these instructions, $\overline{WIN}$ should be included in the decoding of the physical data strobes. See **5.2 PHYSICAL DATA STROBES.** $\overline{WIN}$ is asserted with $\overline{MAS}$ each time a write-protected segment is accessed, whether the access is a read or a write.

NOTE

In multiple-MMU systems, $\overline{MAS}$, $\overline{HAD}$, $\overline{WIN}$, $\overline{FAULT}$, $\overline{DTACK}$, $\overline{ED}$, and $\overline{GO}$ should be connected in parallel to their respective pins on all MMUs. Each should be tied to V_{CC} through a pullup resistor to insure that the signal is negated while the pin is in the high-impedance state.

SECTION 3
REGISTER DESCRIPTION

Figure 3-1 shows a programmer's model of the MMU. The MMU registers consist of two groups: the descriptors and the system registers. Each of the 32 descriptors is nine bytes long and defines one memory segment. See **3.1 DESCRIPTORS.**

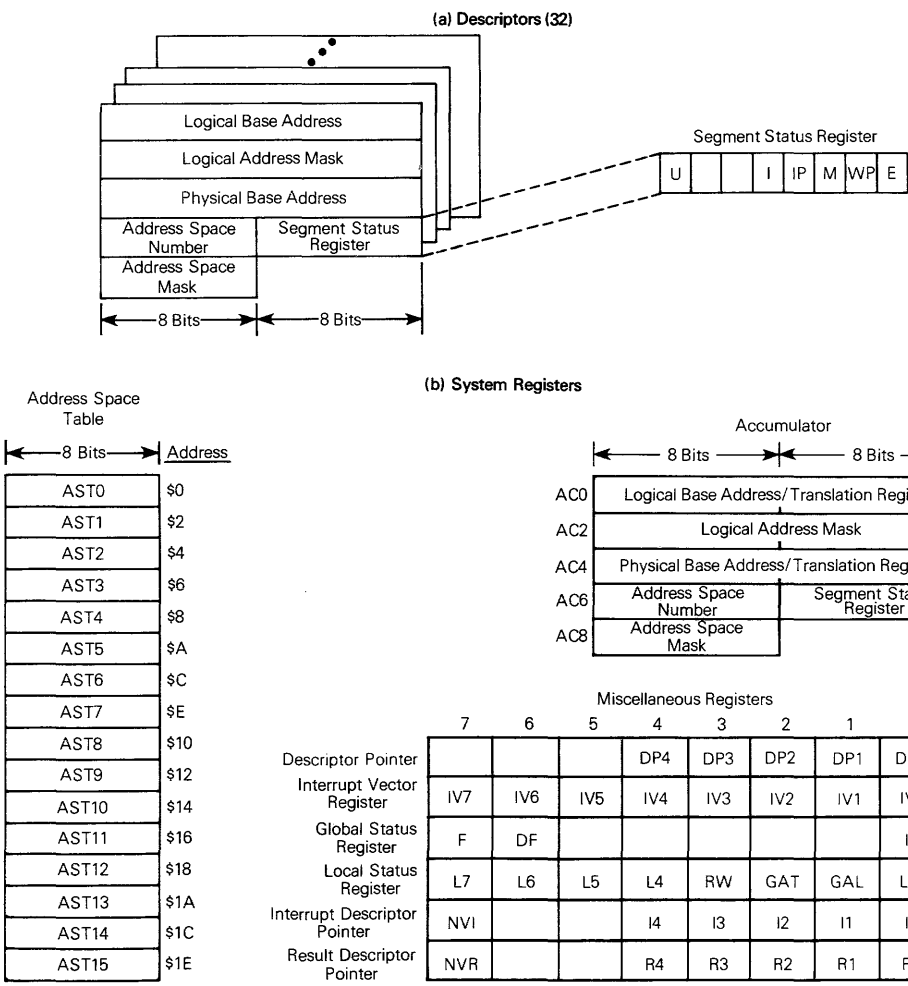


Figure 3-1. Programmer's Model

In the following discussion, a segment access is defined as a successful match occurring on a segment during normal translation.

The system registers contain both information local to the MMU and information global to the MMM. Each bit in the system registers and the segment status registers, except the address space table, is one of four types:

Control	Control bits can be set or cleared by the MPU to select MMU options. These are read/write bits.
Status Alterable	SA bits are set or cleared by the MMU to indicate status information. These are also read/write bits.
Status Unalterable	SU bits are set or cleared by the MMU to reflect status information. These bits cannot be written by the MPU.
Reserved	Reserved bits are reserved for future expansion. They cannot be written and are zero when read.

The system registers are all directly addressable from the physical-address space. Accessing registers causes certain operations to be performed. See **4.6.1 Operations Address Map** for the locations of system registers. The descriptors are not directly addressable, but are accessed using the descriptor pointer and the accumulator.

3.1 DESCRIPTORS

Each MMU contains 32 descriptors (0-31), each of which can define one memory segment. A descriptor is loaded by the MPU using the accumulator and descriptor pointer with a load descriptor operation. The segment status register (SSR) can be written to by the MPU indirectly using the descriptor pointer. Each descriptor consists of the following registers.

3.1.1 Logical Base Address (LBA)

The LBA is a 16-bit register which, together with the logical address mask (LAM), defines the logical addressing range of a segment. This is typically the first address in the segment, although it can be any address within the range defined by the LAM.

3.1.2 Logical Address Mask (LAM)

The LAM is a 16-bit mask which defines the bit positions in the LBA which are to be used for range matching. Ones, in the mask, mark significant bit positions while zeroes indicate "don't care" positions. A range match occurs if, in each bit position in the LAM which is set to one, the LBA matches the incoming logical address. The matching function is depicted schematically in Figure 3-2.

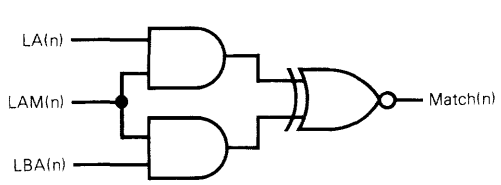


Figure 3-2. Schematic Logic of Address Matching

3.1.3 Physical Base Address (PBA)

The PBA is a 16-bit address which, with the LAM and the incoming logical address, is used to form the physical address. The logical address is passed through to the physical address in those bit positions in the LAM which contain zeroes (the "don't cares") and the PBA is gated out in those positions which contain ones. A schematic representation of the physical address generation mechanism is shown in Figure 3-3.

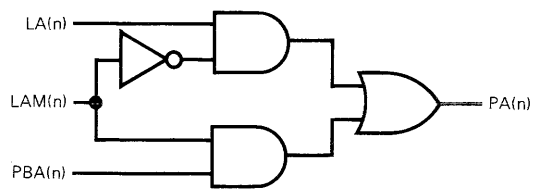


Figure 3-3. Schematic Logic of Physical Address Generation

3.1.4 Address Space Number (ASN)

The ASN is an 8-bit number which, together with the address space mask, is used in detecting a match with the cycle address space number.

3.1.5 Address Space Mask (ASM)

The ASM is an 8-bit mask which defines the significant bit positions in the ASN which are to be used in descriptor matching. As in the LAM, the bit positions which are set are used for matching and the bit positions that are clear are "don't cares". A space match occurs if, in the significant bit positions, the cycle address space number matches the ASN. Address space matching is schematically similar to logical address matching as shown in Figure 3-2.

3.1.6 Segment Status Register (SSR)

Each descriptor has an 8-bit SSR. The SSR can be written to in two ways: using the load descriptor operation or indirectly using the descriptor pointer in a write status register operation. Each bit is labeled as control or status alterable. Bits 5 and 6 are reserved for future use.

7	6	5	4	3	2	1	0
U			I	IP	M	WP	E

Addressed indirectly through the descriptor pointer

U U (used) is set by the MMU if the segment was accessed since it was defined. This bit is status alterable.

Set:
a) By a segment access (successful translation using the segment)
b) By an MPU write of "1"

Cleared:
a) Reset (in segment #0 of master)
b) MPU write of "0"

I If the I (interrupt) control bit is set, an interrupt is generated upon accessing the segment.

Set:
a) MPU writes a "1"

Cleared:
a) MPU writes a "0"
b) Reset (segment #0 of master)

IP IP (interrupt pending) is set if the I bit is set when the segment is accessed. $\overline{IRQ}$out is asserted if an IP bit, in one or more SSRs, is set and IE in the global status register (GSR) is set. $\overline{IRQ}$out is negated when all the IP bits in all SSRs are clear or IE is cleared. IP is status alterable and should be cleared by the interrupt service routine.

Set:
a) Segment access and I is set
b) MPU writes a "1"

Cleared:
a) MPU writes a "0"
b) Reset (in segment #0 of master)
c) E bit is a "0"

M The M (modified) bit is set by the MMU if the segment has been written to since it was defined. The M bit is status alterable.

Set:
a) Successful write to the segment
b) MPU writes a "1"

Cleared:
a) MPU writes a "0"
b) Reset (segment #0 in master)

WP If the WP (write protect) control bit is set, the segment is write protected. A write access to the segment with WP set will cause a write violation.

Set:
a) MPU writes a "1"

Cleared:
a) MPU writes a "0"
b) Reset (segment #0 in master)

E E (enable) is a control bit which, when set, enables the segment to participate in the matching process. E can be cleared (the segment disabled) by a write to the SSR, but a load descriptor operation must be performed to set it.

Set:
a) Load descriptor operation with AC7, bit #0 set
b) Reset (segment #0 in master)

Cleared:
a) MPU writes a "0"
b) Unsuccessful load descriptor operation on this descriptor
c) Load descriptor operation with AC7, bit #0 cleared

3.2 SYSTEM REGISTERS

The system registers consist of the address space table, accumulator, and miscellaneous registers which include global status register, local status register, descriptor pointer, result descriptor pointer, interrupt descriptor pointer, and interrupt vector register. Each of these registers is described below.

3.2.1 Address Space Table (AST)

Each MMU has a local copy of the AST. This table is organized as 16 8-bit, read/write registers located at even byte addresses starting at address $00. Each entry is programmed by the operating system with a unique address space number, each of which is associated with a task. During a memory access, the MMU receives a 4-bit function code (FC0-FC3) which is used to index into the AST to select the cycle address space number. This number is then used to check for a match with the ASN in each of the 32 descriptors within the MMU.

The MC68000 MPU and the MC68450 DMAC only provide a 3-bit function code, FC0-FC2. In a system with more than one bus master, the $\overline{\text{BGACK}}$ signal from the MPU could be inverted and used as FC3. This would result in the AST organization shown in Figure 3-4.

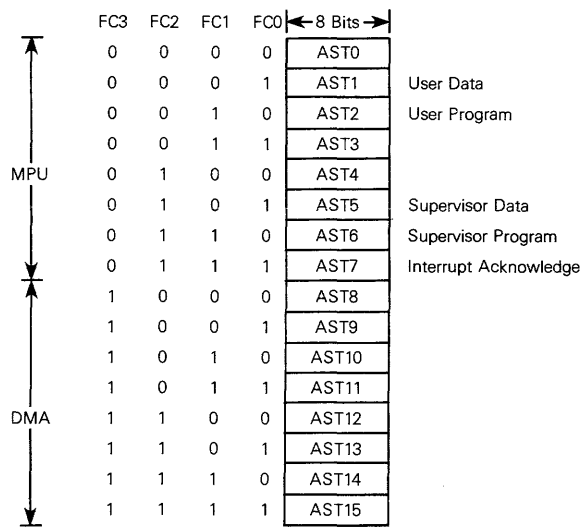


Figure 3-4. Address Space Table Organization

3.2.2 Accumulator (AC0-AC8)

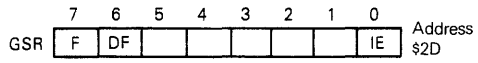
The accumulator (shown in Figure 3-1) is used to access the descriptors, perform direct translation, and latch information during a fault. The accumulator consists of nine 8-bit registers located at byte addresses from $20 to $28. The register assignments for each operation in which it participates is shown in Table 3-1.

Table 3-1. Accumulator Assignments for Operations

Register Assignment	Load/Read Descriptor	Direct Translation	Normal Translation (Fault)	Address
AC0	Logical Base Address (MSB)	Logical Translation Register (MSB)	Logical Address (MSB)	$20
AC1	Logical Base Address (LSB)	Logical Translation Register (LSB)	Logical Address (LSB)	$21
AC2	Logical Address Mask (MSB)			$22
AC3	Logical Address Mask (LSB)			$23
AC4	Physical Base Address (MSB)	Physical Translation Register (MSB)		$24
AC5	Physical Base Address (LSB)	Physical Translation Register (LSB)		$25
AC6	Address Space Number (ASN)	Address Space Number (ASN)	Cycle Address Space Number (CASN)	$26
AC7	Segment Status Register (SSR)			$27
AC8	Address Space Mask (ASM)			$28

3.2.3 Global Status Register (GSR)

The GSR is an 8-bit register used to reflect faults and to enable interrupts from an MMU. All MMUs maintain identical information in their GSRs. Bits 1, 2, 3, 4, and 5 are reserved for future use. The organization of the GSR is shown below. The GSR is located at address $2D.



F — F (fault) is a status alterable bit that is set by the MMU whenever $\overline{\text{FAULT}}$in is detected. Clearing the F bit automatically clears bits L4-L7 in the local status register.

Set:
a) Write violation detected in this MMU
b) $\overline{\text{FAULT}}$in detected
c) ALLin detected (undefined segment access)
d) MPU writes a "1"

Cleared:
a) Reset asserted
b) MPU writes a "0"

DF — DF (double fault) is set if a $\overline{\text{FAULT}}$in signal was detected with F set. DF is a status alterable bit.

Set:
a) Fault occurs and F was previously set
b) MPU writes a "1"

Cleared:
a) Reset
b) MPU writes a "0"

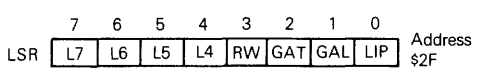
IE — If IE (interrupt enable) is set, the interrupt-request line is enabled. This is a read/write control bit.

Set:
a) MPU writes a "1"

Cleared:
a) Reset
b) MPU writes a "0"

3.2.4 Local Status Register (LSR)

The LSR is an 8-bit register which reflects information local to its MMU. The LSR can be globally written but the GAL, GAT, and LIP bits will not be affected. L4-L7 are cleared if F in the GSR is cleared on reset. The organization of the LSR is shown below. The LSR is located at address $2F.



RW — RW is a status alterable bit which reflects the state of the R/$\overline{\text{W}}$ pin at the time $\overline{\text{FAULT}}$in is asserted.

Set:
a) MPU writes a "1"
b) When fault occurs during read of segment

Cleared:
a) Reset
b) MPU writes a "0"
c) When fault occurs during write of segment

GAT — GAT (global accumulator for translate) is set by the MMU if AC0, AC1, and AC6 are globally consistent. See **4.5.3 Global Operations.**

Set:
a) If AC0, AC1, and AC6 are globally consistent (they were last modified as a result of a global write)

Cleared: a) Reset
b) If AC0, AC1, and AC6 are not globally consistent

GAL GAL (global accumulator for load) is set if AC0, AC1, AC2, AC3, AC6, and AC8 are globally consistent.
Set: a) If AC0, AC1, AC2, AC3, AC6, and AC8 are globally consistent
Cleared: a) Reset
b) If AC0, AC1, AC2, AC3, AC6, and AC8 are not globally consistent

LIP LIP (local interrupt pending) is set if one or more descriptors have IP set in their segment status registers.
Set: a) If IP is set in any descriptor
Cleared: a) Reset
b) If all IP bits are clear

L4-L7 The status information encoded in L4-L7 reflects the status of the MMU after the last event (an operation or fault). These bits are encoded and changed as a unit. They are cleared whenever the F bit in the GSR is cleared and are alterable by the MPU.

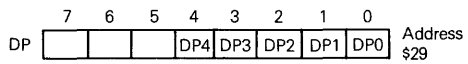
L7	L6	L5	L4		
0	0	0	0	NO	The MMU was not the source of the last event.
1	0	0	0	DT	A direct translation was locally successful. A match was found in one of the MMU descriptors.
1	0	0	1	LD	A load descriptor fault occurred. A previously defined descriptor conflicts with the descriptor being loaded.
1	0	1	0	USA	An undefined segment access was attempted. The logical address was not matched in any descriptor in the MMM.
1	1	0	0	WV	A write violation occurred. A segment defined in the MMU was write protected and a write to that memory segment was attempted. The NVR bit in the RDP will show whether the USA or WV occurred in the MMU. See **3.2.6 Result Descriptor Pointer (RDP).**

Set: a) Various bits set if DT, LD, USA, or WV occur
b) MPU writes a "1"
Cleared: a) Reset
b) MPU writes a "0"
c) When F bit in GSR is cleared
d) If MMU was not the source of the last event (NO)

3.2.5 Descriptor Pointer (DP)

The DP is an 8-bit read/write pointer register located at address $29. The five low-order bits identify the descriptor to be used in the load descriptor, read segment status (transfer descriptor), and write segment status operations. Bits 5, 6, and 7 are reserved.

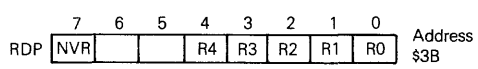
The DP is initialized to $00 on reset. It can be globally written by the MPU. The DP is loaded by the MMU with the number of the descriptor matched in a direct translation operation to allow a subsequent transfer descriptor operation to load the match descriptor into the accumulator. See **4.6.3 Global Operation** and **4.6.3.3 Direct Translation.**



3.2.6 Result Descriptor Pointer (RDP)

The RDP is an 8-bit, read-only register that identifies a descriptor involved in the following events: a write violation, a load descriptor failure, or a direct translation success. The RDP is loaded from a priority encoder which determines the highest priority descriptor involved. For example, in a load descriptor operation, more than one descriptor currently in the MMU may collide with the descriptor being loaded. Only the number of the highest priority descriptor will be loaded into the RDP. Descriptor 0 is considered to be the highest priority and 31 is the lowest.

The RDP bit assignments are shown below. Bits 5 and 6 are reserved. The RDP is initialized to $80 on reset. The RDP is located at address $3B.



NVR If no descriptor is selected by the priority encoder when the RDP is loaded, NVR (no valid result) is set; otherwise, it is cleared. This bit is status unalterable.

Set: a) Reset
b) No result from WV, LD, or DT

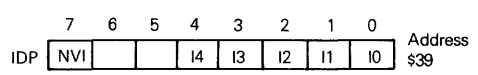
Cleared: a) AWV, LD failure of DT success in MMU

R0-R4 R0-R4 encode the number of the descriptor selected by the priority encoder.

3.2.7 Interrupt Descriptor Pointer (IDP)

The IDP is an 8-bit read-only register that is read to determine which descriptor caused an interrupt. Each time it is read, the IDP is loaded from the priority encoder with the highest-priority descriptor which has IP set in its SSR. If no descriptor has IP set, the no valid interrupt (NVI) bit is set. See **4.6.2.1 Interrupt Acknowledge.**

The bit IDP assignments are shown below. Bits 5 and 6 are reserved. The IDP is located at address $39.

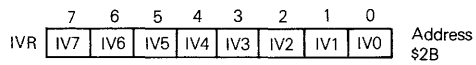


NVI NVI is set if no descriptor has IP set; otherwise, it is cleared.

I0-I4 These bits encode the number of the descriptor selected by the priority encoder.

3.2.8 Interrupt Vector Register (IVR)

The IVR is an 8-bit read/write register containing the interrupt vector. Its contents are placed on the data lines, D0-D7, during the interrupt acknowledge operation to provide the processor with a vector number. The IVR is initialized to $0F (the MC68000 uninitialized-device vector number) on reset. The IVR is located at address $2B.



SECTION 4
MMU FUNCTIONAL DESCRIPTION

4.1 SINGLE-MMU SYSTEMS

The memory management mechanism can be comprised of one or more memory management units. Implementing memory management using the MC68000 or MC68010 MPUs and a single MC68451 is straightforward. Two SN74LS245 type bidirectional buffers are needed to demultiplex the physical address/data port (PAD0-PAD15). These are directly controlled by the enable data ($\overline{ED}$) and read/write (R/$\overline{W}$) lines. Two SN74LS373 type transparent latches are used to latch the translated address when the PAD port is needed to transfer data to or from the MPU. These latches are controlled by the hold address ($\overline{HAD}$) line. A circuit diagram of a single-MMU system is shown in Figure 4-1.

The MMU may also be used with the MC68008 8-bit data bus version of the MC68000. This requires only a few gates to synthesize the upper and lower data strobe signals ($\overline{UDS}$ and $\overline{LDS}$) from address line A0 and data strobe ($\overline{DS}$) on the MC68008. Since there is only one data strobe in an MC68008 system, the physical data strobe generation is somewhat simpler; otherwise, the circuit is identical to that of the MC68000 system. A circuit diagram of a single-MMU system using the MC68008 MPU is shown in Figure 4-2.

4.2 MULTIPLE-MMU SYSTEMS

If more than 32 descriptors are needed for performance enhancement reasons, multiple MMUs may be configured in a system. The number of MMUs that can be used is not logically constrained. In practice, buffer drive limitations limit the number to six MMUs without using additional external buffering.

A circuit diagram showing two MC68451s in a system with the MC68000 or MC68010 is shown in Figure 4-3. Notice that the data buffers and address latches are shared by both MMUs and need not be duplicated. The $\overline{GO}$, $\overline{ANY}$, and ALL lines are bused together to provide the inter-MMU handshake necessary for the global operations as outlined below. Notice that only one of the MMUs is chip-selected on $\overline{RESET}$. This selects the master out of reset. For more information refer to **4.5.1 The Reset State.**

In order to perform its operations, some of the information in MMU registers must be global. That is, it must be duplicated in all the MMUs in the system. For example, the address space table must be global to insure that the address space numbers are common to all MMUs. To allow this, certain operations are defined as global. Any system register that can be written, is written globally. This includes the accumulator, the address space table, the descriptor pointer, the interrupt vector register, the global status register, and the local status register. The result descriptor pointer and the interrupt descriptor pointer are read only and, therefore, are local and not global.

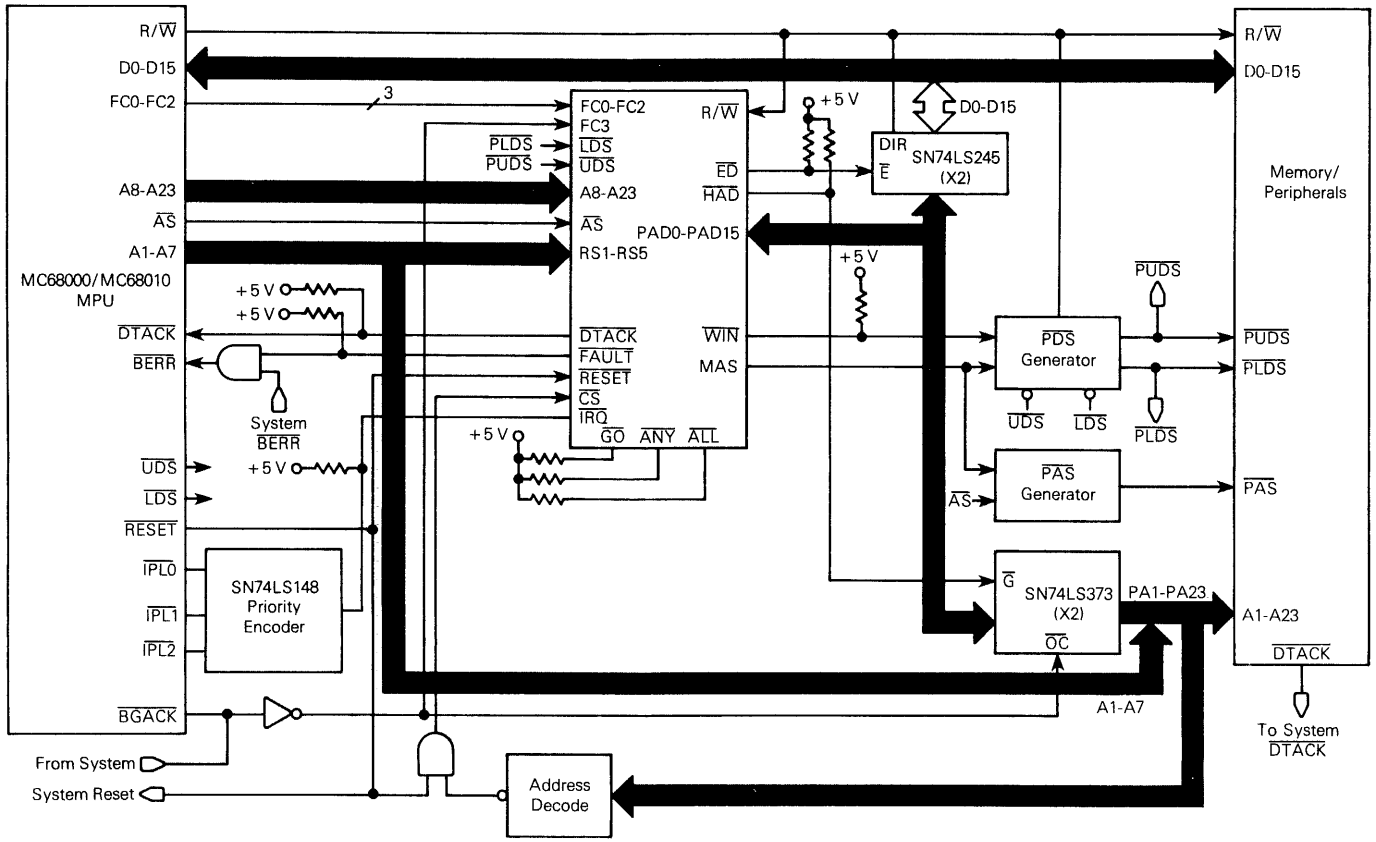


Figure 4-1. Single-MMU System with MC68000 or MC68010

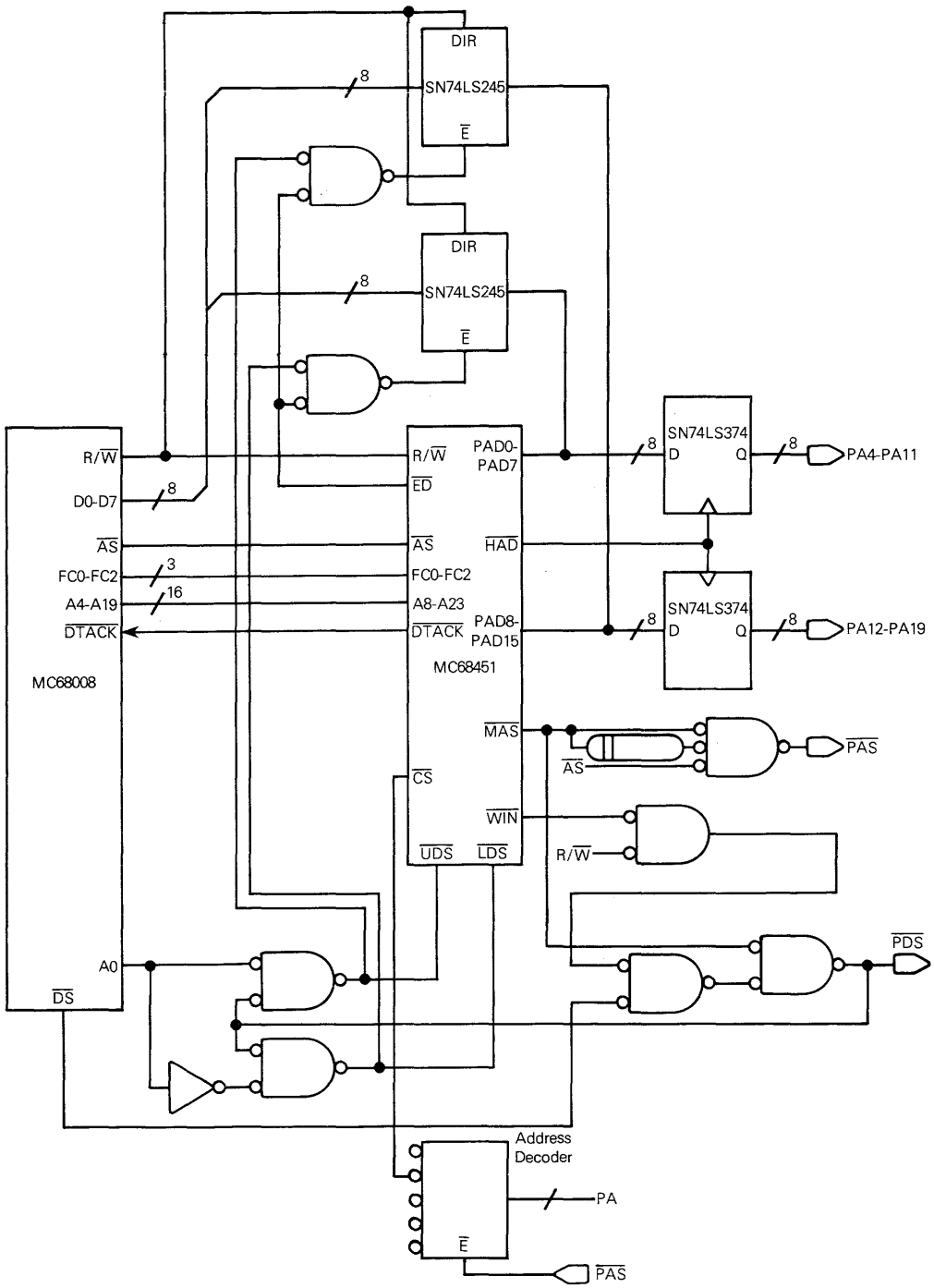


Figure 4-2. Using the MC68008 with the MMU

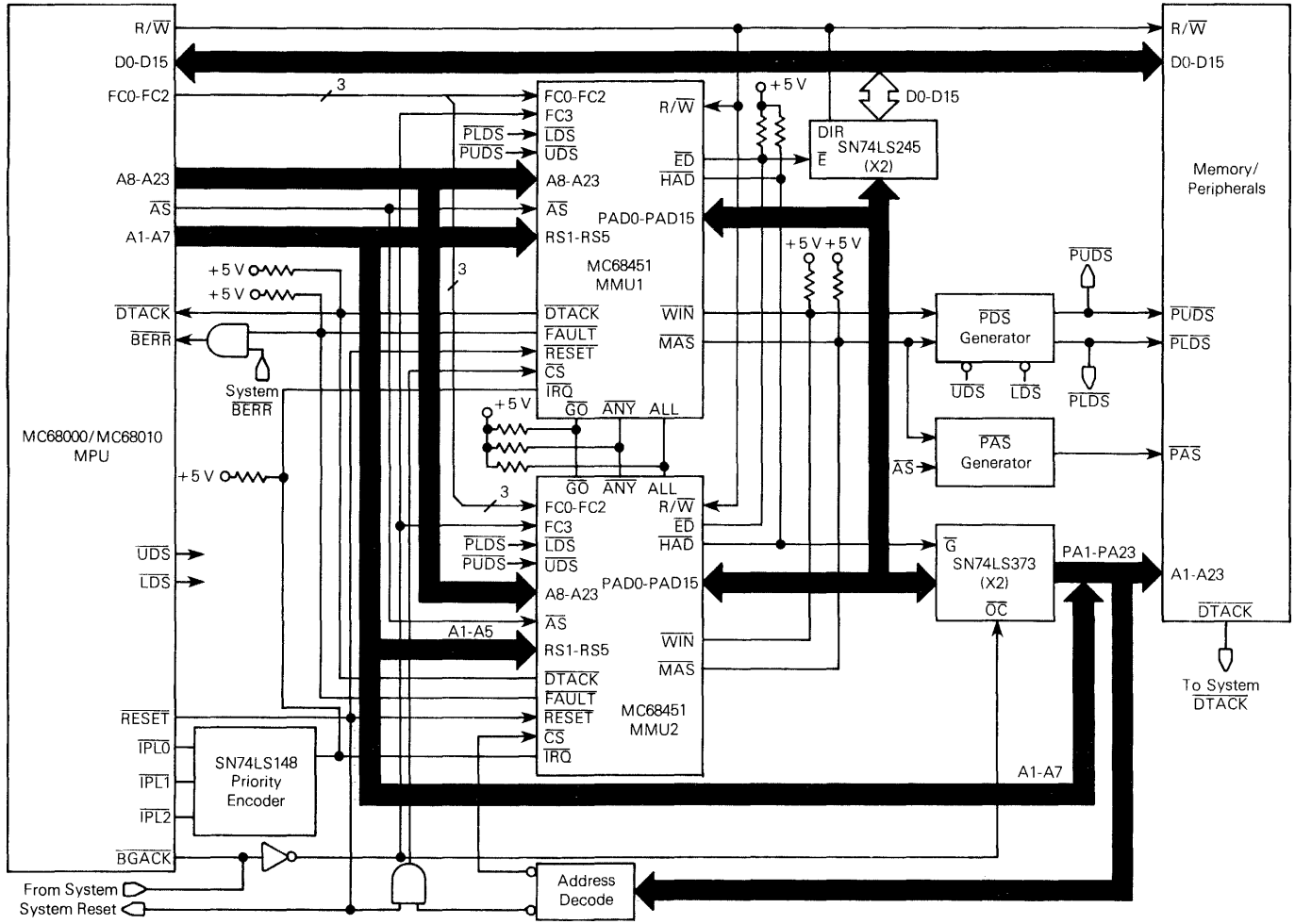


Figure 4-3. Sample Circuit Diagram of a Two-MMU System with the MC68000 or MC68010

The $\overline{ANY}$, ALL, and $\overline{GO}$ signal lines are used to connect multiple MMUs to form the MMM. The MMM uses these input/output signals to communicate information between MMUs and maintain functional unity. The $\overline{GO}$ (global operation) pin is used to establish the master-slave relationship between MMUs for a given operation. The $\overline{ANY}$ signal is detected as true if any MMU asserts it, allowing MMUs to report conditions that are important in even one device. The ALL signal is detected as true only if all MMUs assert it. It is used to verify that all MMUs in the system have performed some operation or are in the same state. A sample circuit diagram of a two-MMU system using the MC68000 or MC68010 is shown in Figure 4-3.

During each global operation, one MMU is specified as the master; all others are designated as slaves. The MMU which has its chip select ($\overline{CS}$) asserted becomes the master by asserting the $\overline{GO}$out signal. This signals the other MMUs that they are slaves for that operation. Note that all MMUs may be accessed and, therefore, any one may be the master for a given operation.

4.3 MMU TIMING

The MC68451 is a hybrid machine. The normal translation function is implemented in combinational logic for maximum speed. The operations (reading and writing registers, etc.) are implemented with a synchronous finite state machine. Due to the asynchronous nature of the signals used to communicate between MMUs, the timing of these operations will vary due to a number of factors, including the placement of $\overline{AS}$, the speed of translations, the relationship of $\overline{CS}$ to the internal clocks, and the need to internally synchronize the input signals. Because of this, timing diagrams are shown with breaks to indicate a variable number of clock periods between events.

4.4 MMU FLOW DIAGRAM

In order to give a functional description of the MMU, the flow diagrams of the device are given in Figure 4-4. Each box contains certain actions (loading registers, asserting signals, etc.) and the paths leading from it are labeled with a signal name or logical expression of signal names. If the signal or expression is true, that branch is taken. The only timing information given is the chronological order in which events occur. The length of each operation, given in system clock cycles, is shown in Table 4-1.

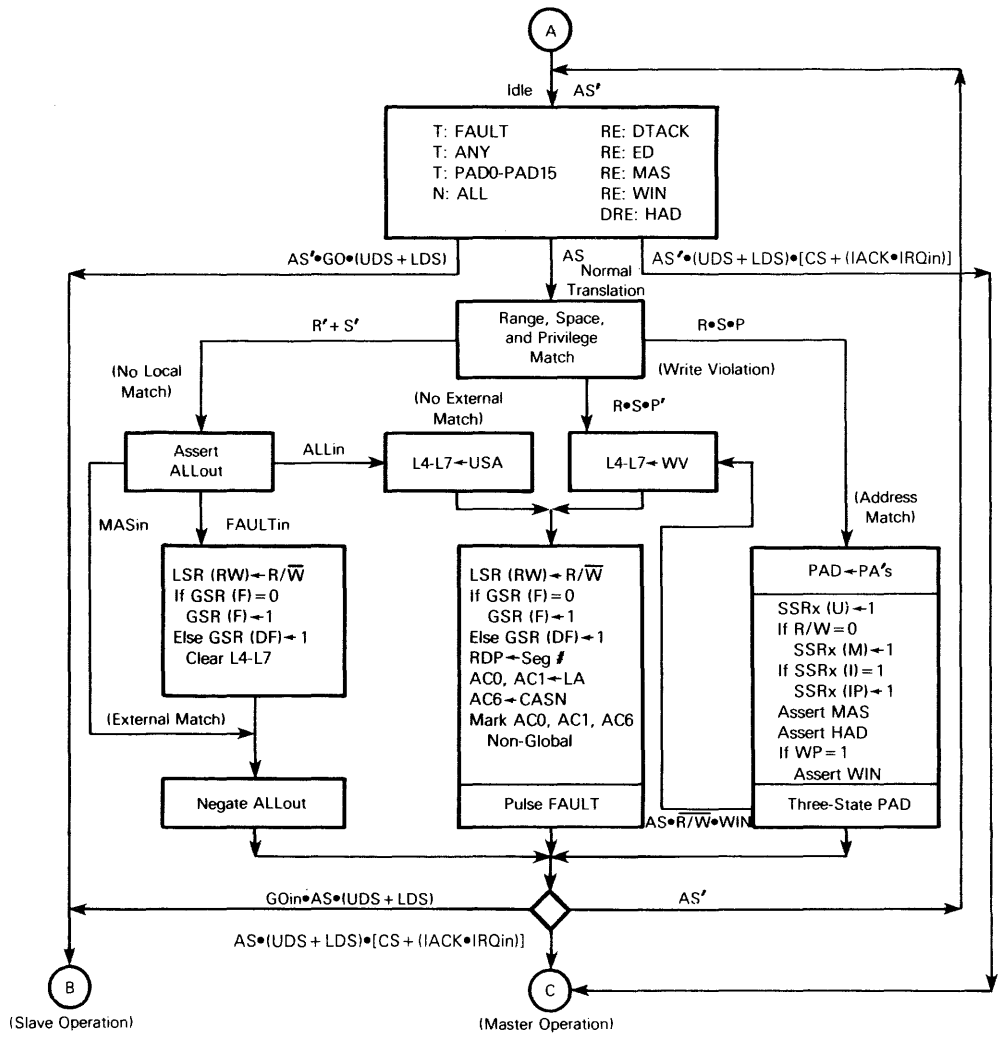
NOTE

In Figure 4-4, all signals are shown as active high regardless of the voltage level of the true state. Logical negation is denoted by the prime symbol (').

4.5 MMU FUNCTIONAL STATE

At any time an MMU may be in one of five states: reset, idle, normal translation, local operations, or global operations. In a global operation, an MMU may be a master (if the $\overline{CS}$ signal is asserted) or a slave (if $\overline{GO}$in is asserted). In addition, two actions can occur regardless of the current state:

1. If $\overline{RESET}$ is asserted, the reset operation begins. The MMM will remain in the reset state until $\overline{RESET}$ is negated.
2. $\overline{IRQ}$out is asserted if the local interrupt pending bit in the local status register and interrupt enable in the global status register are set; otherwise, it is placed in the high-impedance state and should be negated with a pullup resistor.



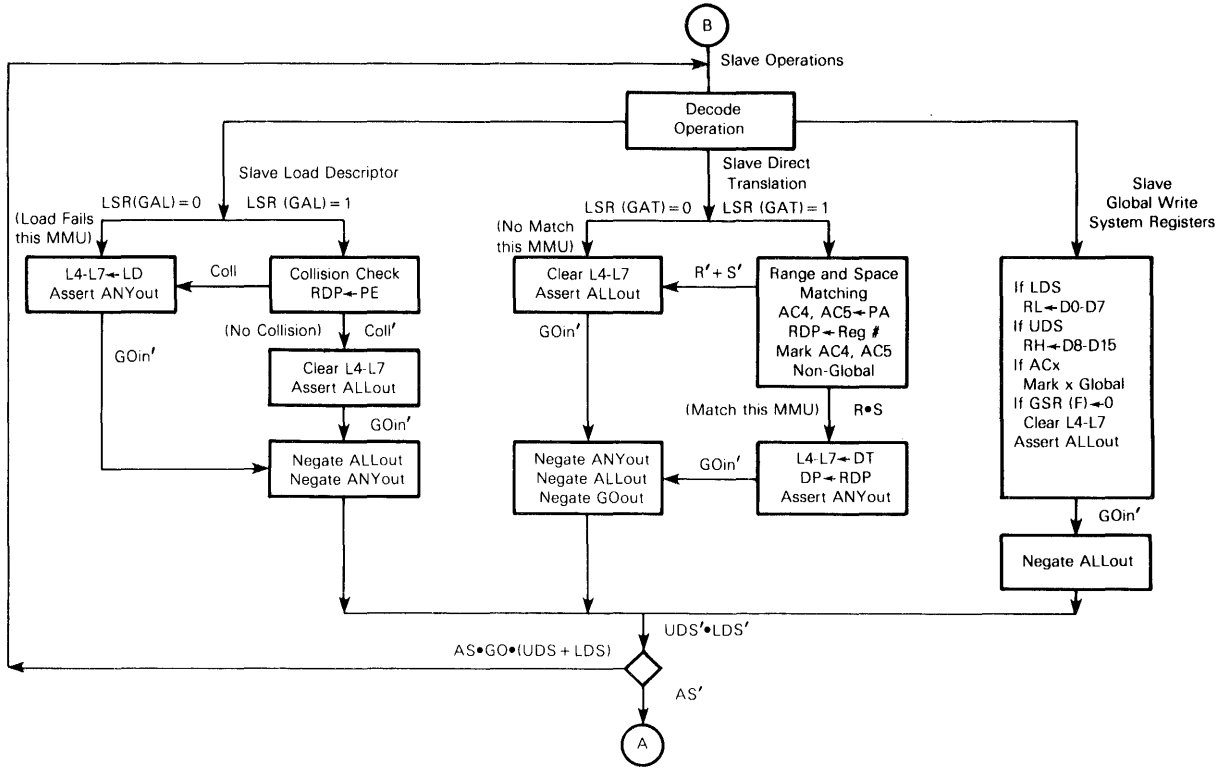
NOTES:

1. In this figure, all signals are shown as active high regardless of the voltage level of the true state. Logical negation is denoted by the prime symbol (').
2. Individual bits in registers are denoted as REG (bit); e.g., LSR (GAL) represents the GAL bit in the LSR.

LEGEND:

T	= Signal is three-stated.	N	= Signal is negated.	PE	= Priority Encoder
RE	= Rescind. Signal is driven high, then three-stated.	R	= Range Match	PA	= Physical Address
		S	= Space Match	COLL	= A collision occurred during descriptor operation.
DRE	= Delayed rescind. HAD is rescinded one-half clock period after MAS.	P	= Privilege Match (not a write with WP set)	RL	= Low-order byte of register
		Dx	= Descriptor #x (0-31)	RH	= High-order byte of register

Figure 4-4. MMU Flow Diagram (Sheet 1 of 3)

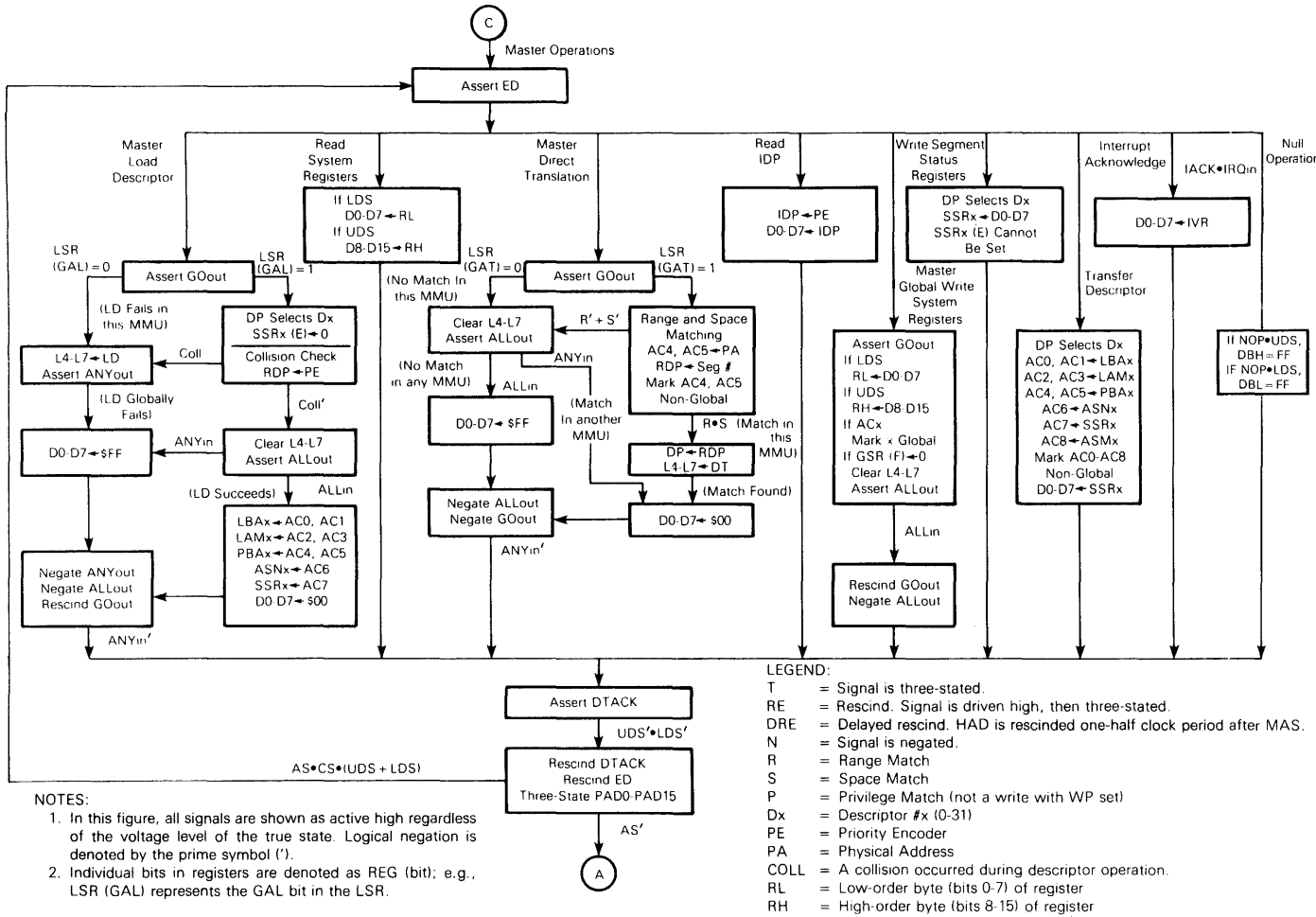


NOTES:
1. In this figure, all signals are shown as active high regardless of the voltage level of the true state. Logical negation is denoted by the prime symbol (').
2. Individual bits in registers are denoted as REG (bit); e.g., LSR (GAL) represents the GAL bit in the LSR.

LEGEND:

T	= Signal is three-stated.	DRE	= Delayed rescind. HAD is rescinded one-half clock period after MAS.	S	= Space Match	PA	= Physical Address
RE	= Rescind. Signal is driven high, then three-stated.	N	= Signal is negated.	P	= Privilege Match (not a write with WP set)	COLL	= A collision occurred during descriptor operation.
		R	= Range Match	Dx	= Descriptor #x (0-31)	RL	= Low-order byte of register
				PE	= Priority Encoder	RH	= High-order byte of register

Figure 4-4. MMU Flow Diagram (Sheet 2 of 3)



NOTES:

1. In this figure, all signals are shown as active high regardless of the voltage level of the true state. Logical negation is denoted by the prime symbol (′).
2. Individual bits in registers are denoted as REG (bit); e.g., LSR (GAL) represents the GAL bit in the LSR.

LEGEND:

T	=	Signal is three-stated.
RE	=	Rescind. Signal is driven high, then three-stated.
DRE	=	Delayed rescind. HAD is rescinded one-half clock period after MAS.
N	=	Signal is negated.
R	=	Range Match
S	=	Space Match
P	=	Privilege Match (not a write with WP set)
Dx	=	Descriptor #x (0-31)
PE	=	Priority Encoder
PA	=	Physical Address
COLL	=	A collision occurred during descriptor operation.
RL	=	Low-order byte (bits 0-7) of register
RH	=	High-order byte (bits 8-15) of register

Figure 4-4. MMU Flow Diagram (Sheet 3 of 3)

Table 4-1. Length of Operations

Operation		Length (Clock Periods)	
Single MMU			
		Min	**Max**
Read System Register (Except IDP)		13	15
Read Interrupt Descriptor Pointer		21	23
Write System Register (Except Segment Status)		17	19
Write to Segment Status Register		11	13
Load Descriptor	(No Error)	33	35
	(Collision Error)	27	29
	(No GAL)	21	23
Direct Translation	(No Error)	25	27
	(Undefined Segment)	27	29
	(No GAT)	19	21
Read SSR and Transfer Descriptor		17	19
Interrupt Acknowledge		13	15
Null Operation (Read)		11	13
Null Operation (Write)		9	11
Multiple MMU			
		Min	**Max**
Read System Register (Except IDP)		13	15
Read Interrupt Descriptor Pointer		21	23
Global Write System Register (Except Segment Status)		21	29
Write Segment Status		11	13
Load Descriptor	(No Error)	35	43
	(Match−{m}, No Err−{s})	27	29
	(Match−{m}, GAL−{s})	35	43
	(Match−{s}, No Err−{m})	39	51
	(No GAL−{m}, Match−{s})	21	23
	(No GAL−{m}, No Err−{s})	21	23
	(No GAL−{s}, No Err−{m})	31	43
	(No GAL−{s}, No GAL−{m})	27	31
Direct Translation	(Match−{m}, No Err−{s})	25	27
	(Match−{s}, No Err−{m})	39	51
	(Match−{m}, No GAT−{s})	25	27
	(Match−{s}, No GAT−{m})	39	51
	(No Match−{m, s})	31	39
	(No Match−{s}, No GAT−{m})	31	39
	(No Match−{m}, No GAT−{s})	27	33
	(No GAT−{m}, No GAT−{s})	23	31
Read SSR and Transfer Descriptor		17	19
Interrupt Acknowledge		13	15
Null Operation (Write)		9	11
Null Operation (Read)		11	13

NOTES:

1. Length of operations are given in system clock periods. The length of the operations is defined from the assertion of $\overline{CS}$ to the assertion of $\overline{DTACK}$ except in the case of interrupt acknowledge operations. In this case, the timing is given from $\overline{IACK}$ to $\overline{DTACK}$.
2. These operation lengths assume that all MMUs have a common system clock. The actual length of a given operation will depend upon when $\overline{CS}$ (or $\overline{IACK}$) is asserted at the MMU. In the case of multiple-MMU systems, the length will also depend on the rise and fall times of the inter-MMU handshake signals and their relationship to the internal clocks of the MMU.
3. In multiple MMU operations, the master is denoted by {m} and the slave is denoted by {s}. The notes 'no GAL' and 'no GAT' indicate that bit is cleared.
4. Immediately following an operation to an MMU, the time required for an MMU to re-enter the 'ready for translation' state is:
   $\overline{DTACK}$ to IDLE = 10 clock periods maximum
   $\overline{UDS}$ to $\overline{LDS}$ negated to IDLE = 8.5 clock periods maximum

4.5.1 The Reset State

Asserting $\overline{\text{RESET}}$ will initiate the reset sequence regardless of the state of the MMU. During reset, $\overline{\text{GO}}$, $\overline{\text{DTACK}}$, $\overline{\text{ED}}$, $\overline{\text{MAS}}$, $\overline{\text{HAD}}$, and $\overline{\text{WIN}}$ are rescinded. The PAD port, $\overline{\text{FAULT}}$, and $\overline{\text{ANY}}$ are placed in the high-impedance state. Pullup resistors on $\overline{\text{FAULT}}$ and $\overline{\text{ANY}}$ keep these signals negated. The ALL pin is driven low to negate it.

The GSR, LSR, DP, and the entire address space table are initialized to $00. The RDP is initialized to $80 and the IVR to $0F. All descriptors are disabled by clearing the enable bits in their segment status registers.

In order to allow the address bus to function before the operating system can initialize the MMM, one MMU is selected to have descriptor #0 initialized so that it maps any logical address unchanged to the physical address bus. The MMU is selected for this by having its chip-select line asserted during reset. This circuit is shown in Figure 4-3.

Descriptor 0 in the selected MMU will have had its LAM and ASN cleared to $00, its ASM set to $FF, and the enable bit set. Because of this, the logical address passes to the physical address bus (via descriptor #0) without alteration. The enable bits of descriptors 1-31 are cleared to disable them and their contents remain uninitialized. If the chip selected ($\overline{\text{CS}}$) is not selected during reset, the enable bits in all descriptors are cleared and no descriptor is initialized.

A flow diagram for the reset state is given in Figure 4-5.

NOTE

In Figure 4-5, all signals are shown as active high regardless of the voltage level of the true state. Logical negation is denoted by the prime symbol (').

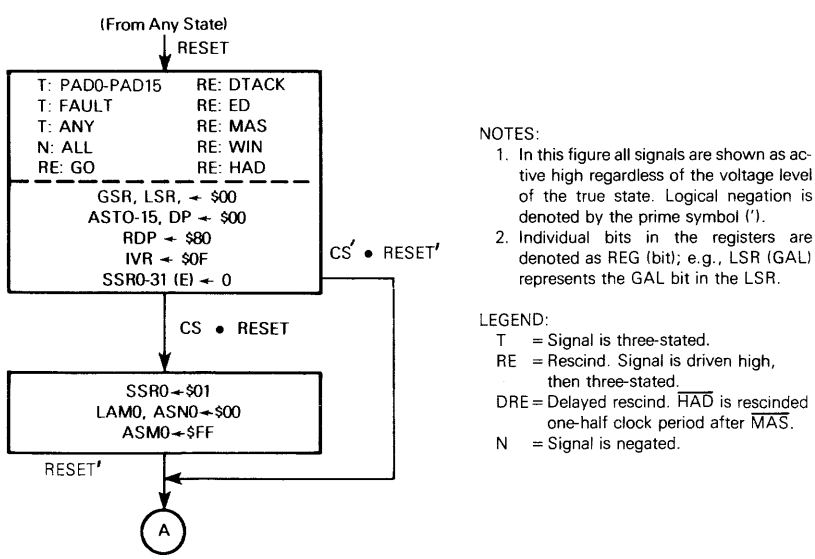


Figure 4-5. Reset Flow Diagram

4.5.2 The Idle State

The idle state is used to terminate bus accesses and prepare for new ones. The MMU is "backed-off" the bus; i.e., the data transceivers are placed in the high-impedance state and the address latches are put into the transparent mode. The outputs are driven to the same levels as in reset except that $\overline{HAD}$ is rescinded one-half clock after $\overline{MAS}$ to provide address hold time.

While in the idle state, the MMU uses the function code inputs to index into the AST to provide the cycle address space number. If $\overline{AS}$ is asserted, a normal translation is performed. If $\overline{AS}$ is negated and $\overline{CS}$, $\overline{IACK}$, $\overline{IRQ}$in, $\overline{GO}$, and the data strobes indicate an access from the physical bus, an operation is performed. For further information, see **4.6 MMU OPERATIONS.**

4.5.3 Normal Address Translation

At the start of a bus cycle, the processor presents the logical address, R/$\overline{W}$, and the function code to the MMM. The function code is used to index into the address space table to select the cycle address space number. When $\overline{AS}$ is asserted, the normal translation phase begins by sending the cycle address space number, the logical address, and R/$\overline{W}$ to each descriptor for matching.

NOTE

The function codes must be valid before $\overline{AS}$ is asserted to allow for the table lookup. Current versions of the MC68000 provide this setup time; however, early mask sets (R9M, T6E, BF4) do not. With these early mask sets, $\overline{AS}$ must be delayed to the MMU.

4.5.3.1 MATCHING. Matches can occur in two areas: range and space. A range match occurs if, in each bit position in the LAM which is set, the incoming logical address matches the LBA. A space match occurs if, in each bit position in the ASM which is set, the cycle address space number matches the ASN.

4.5.3.2 SUCCESSFUL TRANSLATION. An address match occurs if there is a range match and a space match. A write violation occurs if a write is attempted to a write-protected segment. If there is an address match in a descriptor and no write violation, the physical address is formed from the PBA of that descriptor and the logical address. The logical address is passed through in those bit positions in the LAM which are clear (the "don't cares"). In the other bit positions, the PBA is gated out to the physical address bus.

The U and, if the cycle was a write, the M bits in the segment status register are set. If the I bit is set, then the IP bit is set. $\overline{WIN}$ is asserted if the WP bit is set and the cycle was a read or a read-modify-write. If the cycle was a write, $\overline{MAS}$ is not asserted to prevent the write from modifying data.

After the physical address is stable, $\overline{MAS}$ is asserted to indicate a valid address is on the bus. $\overline{HAD}$ is asserted to hold the address stable on the latches and the PAD0-PAD15 lines are then placed in the high-impedance state. If $\overline{AS}$ is then negated, the cycle has terminated and the MMU returns to the idle state. A functional timing diagram for a normal translation in a two-MMU system is shown in Figure 4-6. If $\overline{AS}$ is not negated, the cycle can continue in three ways:

1. If $\overline{CS}$ or $\overline{IACK}$ and $\overline{IRQ}$in are asserted, the MMU will begin an operation as a master. See **4.6.3 Global Operations.**
2. If $\overline{GO}$in is detected by an MMU, it will begin a slave operation. See **4.6.3 Global Operations.**
3. If a high-to-low transition is detected on R/$\overline{W}$ (indicating a write) $\overline{AS}$ remains asserted, and the matched segment is write protected, a write violation occurs. The $\overline{WIN}$ signal is asserted to protect the memory. This would be the result of a read-modify-write bus cycle on a protected segment.

4.5.3.3 WRITE VIOLATION. If an address match occurs but the bus cycle was a write to write-protected segment, a write violation occurs. In this case, the RDP is loaded from the priority encoder, F is set in the GSR, and DF is set if F was previously set. The state of the R/$\overline{W}$ line is latched into the RW bit of the LSR and L4-L7 are encoded to indicate write violation (WV). The $\overline{FAULT}$out signal is then asserted for five clock cycles or until $\overline{AS}$ is negated, whichever is longer.

The logical address is latched into AC0 (MSB) and AC1 (LSB) of the accumulator (see Table 3-1). The cycle address space number is latched into AC6. These registers are marked as non-global with respect to the GAT and GAL bits. If the $\overline{FAULT}$ pin has been connected to the $\overline{BERR}$ pin on the MC68000, $\overline{AS}$ will be negated as the MPU begins the bus error exception processing. When $\overline{AS}$ is negated, the MMU will enter the idle state.

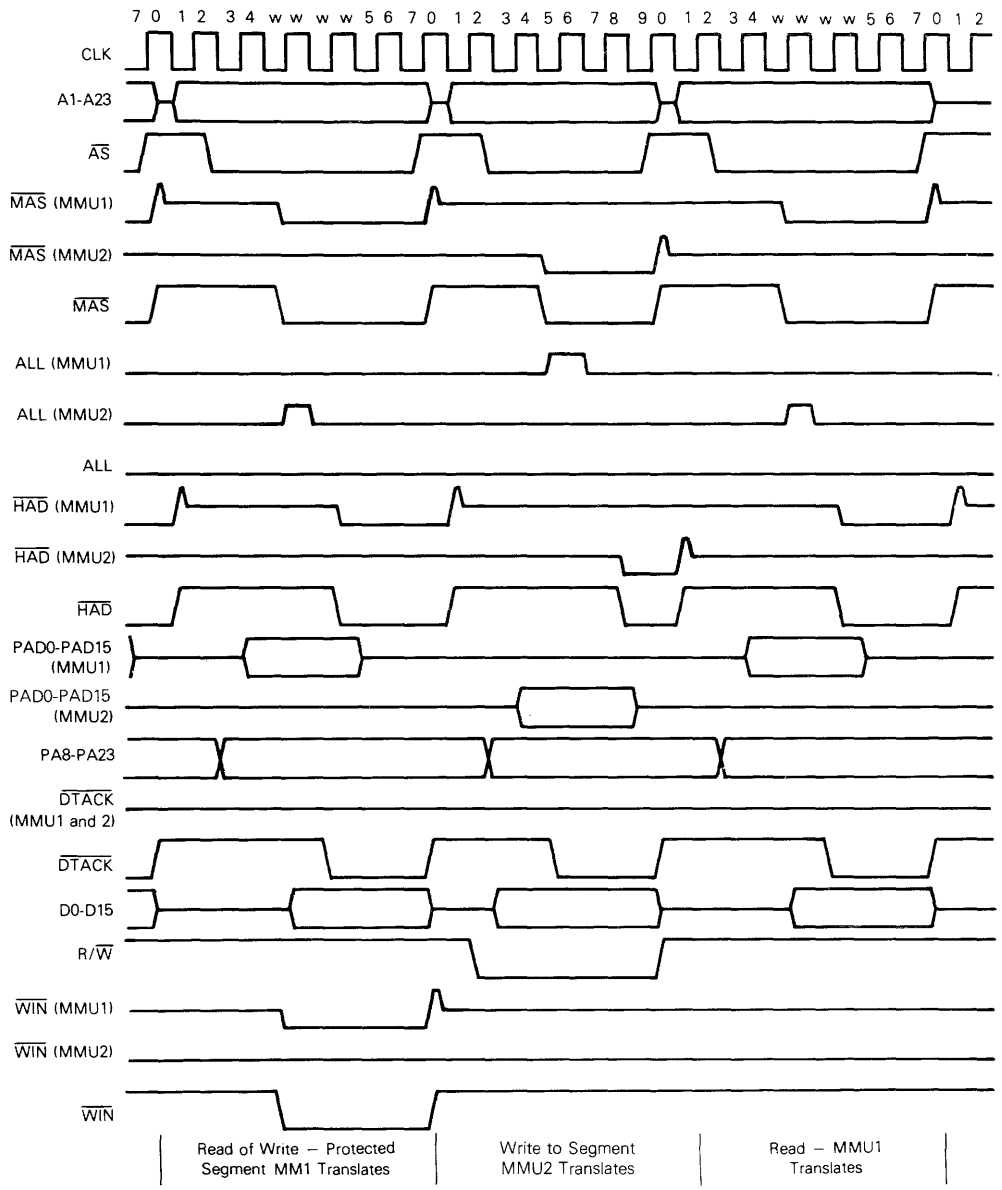
A functional timing diagram for a translation with a write violation is shown in Figure 4-7. The breaks in the diagram reflect the uncertainty due to the deskewing of the input signals from other MMUs.

4.5.3.4 NO ADDRESS MATCH. If none of the descriptors in an MMU has an address match, that MMU asserts ALLout, and monitors $\overline{MAS}$in, $\overline{FAULT}$in, and ALLin. There are then three possibilities:

1. The access was successfully translated in another MMU. See **4.5.3.5. EXTERNAL TRANSLATION.**
2. The access caused a write violation in another MMU. See **4.5.3.6 EXTERNAL WRITE VIOLATION.**
3. The access was to a globally undefined segment. See **4.5.3.7 UNDEFINED SEGMENT ACCESS.**

4.5.3.5 EXTERNAL TRANSLATION. If $\overline{MAS}$in becomes asserted, the access was successfully translated by another MMU. The MMU negates ALLout and prepares to end the normal translation phase. The cycle can then continue in one of three ways:

1. If $\overline{AS}$ becomes negated, the MMU returns to the idle state.
2. If $\overline{CS}$, or $\overline{IACK}$ and $\overline{IRQ}$in are asserted, the MMU begins an operation as a master.
3. If $\overline{GO}$in is detected true, the MMU begins an operation as a slave.



NOTE: Cycles denoted by "w" are MC68000 wait states.

Figure 4-6. Normal Translation Timing on a Two-MMU System

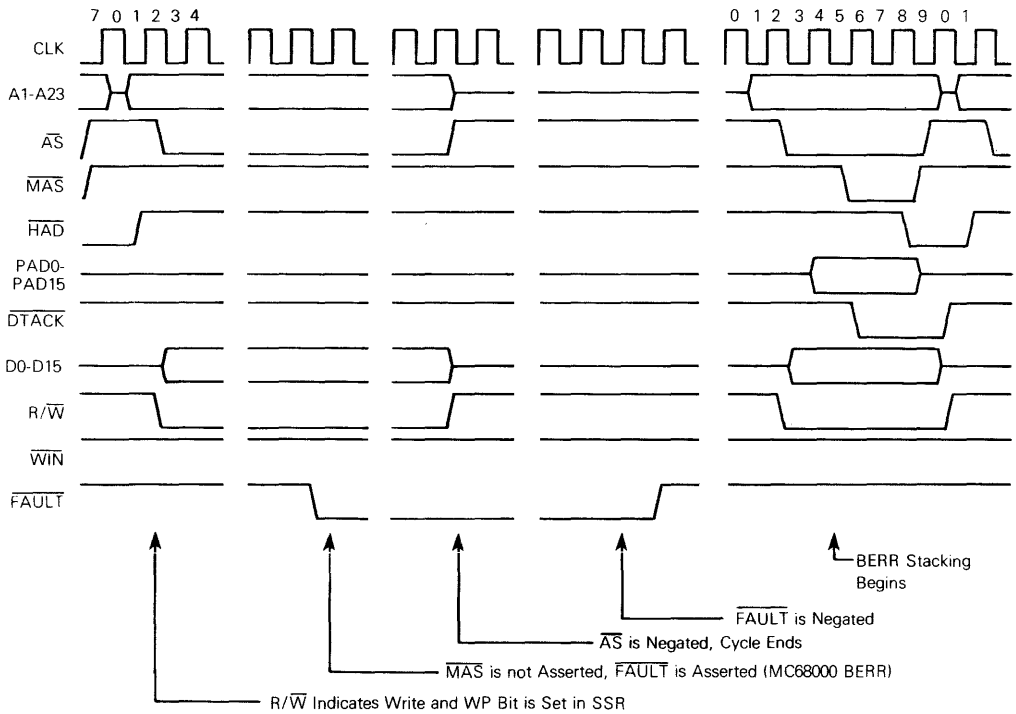


Figure 4-7. Write Violation Timing

4.5.3.6 EXTERNAL WRITE VIOLATION. If the FAULTin line is detected true (low), a write violation occurred in another MMU. The detecting MMU then sets the F bit in the GSR and the DF bit if the F bit was already set. R/W is latched into the RW bit of the LSR, and L4-L7 are cleared to show that the violation did not take place in the MMU. The cycle can then continue in one of the three ways described in **4.5.3.5 EXTERNAL TRANSLATION.**

4.5.3.7 UNDEFINED SEGMENT ACCESS. If ALLin is detected true (high), none of the other MMUs in the system obtained a match, indicating the segment is globally undefined. The MMU sets the F bit in the GSR and the DF bit if F was set previously. R/W is latched into the RW bit of the LSR and L4-L7 are encoded to show a USA.

The logical address (A8-A23) is latched into the accumulator, AC0 (MSB) and AC1 (LSB), and the cycle address space number is latched into AC6. These registers are marked non-global with respect to the GAL and GAT bits.

All MMUs assert the FAULT line for five clock periods or until AS is negated, whichever is longer. To assure the detection of ALLin by all MMUs, ALLout remains asserted for two clock cycles after ALLin is detected true. ALLout is negated before the beginning of the FAULT pulse. When AS is negated, the MMM returns to the idle state.

A functional timing diagram for an undefined segment access is given in Figure 4-8. The breaks in the diagram reflect the uncertainty due to the deskewing of the input signals from other MMUs.

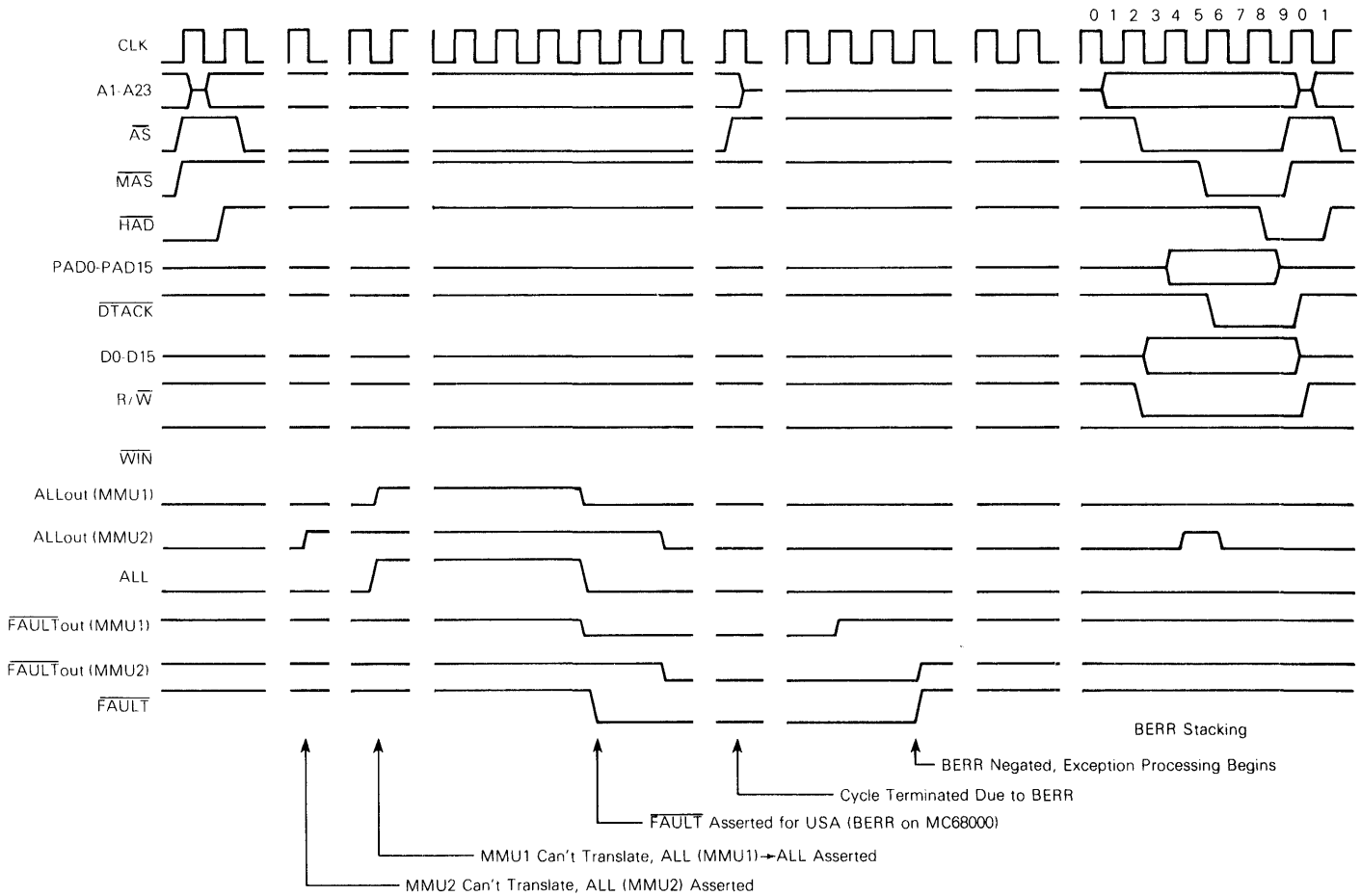


Figure 4-8. Undefined Segment Access Timing

4.6 MMU OPERATIONS

Table 4-2 shows the operations which can be performed. Each operation is initiated by the access of an address given on the register select lines RS1-RS5 and the upper and lower data strobes. The access can be from either the logical or physical address bus. In a multiprocessor system, an external processor could access the MMM from the physical address bus. If the access is from the logical address bus, and address translation is first performed. If the access is from the physical address bus, the operation state is entered directly from the idle state.

Table 4-2. Summary of MMU Functions

Function	Summary
Idle	The MMU backs off the bus to prepare for a new access.
Reset	The MMU is pre-emptively initialized.
Normal Translation	The MMU attempts to translate an access from the logical address bus.
Operations	The MMU is accessed from the logical or physical bus.
Write System Registers	An operation to globally write system registers.
Read System Registers	An operation to read the system registers.
Write Segment Status	The SSR of a descriptor can be quickly changed using this operation. The enable bit cannot be set using it, however.
Load Descriptor	With this operation, the contents of the accumulator are loaded into the descriptor pointed to by the descriptor pointer.
Transfer Descriptor	This operation transfers the contents of the slected descriptor into the accumulator.
Direct Translation	An operation to globally translate a logical address for the operating system.
Interrupt Acknowledge	An operation that supplies a vector number to the MPU in response to IACK.

The operation phase is always entered with PAD0-PAD15 in the high-impedance state and either (in the case of an operation following a normal translation) one MMU asserting $\overline{HAD}$ to hold the physical address, or (in the case of an access from the physical bus) the external processor holding the address. If both $\overline{CS}$ and either $\overline{UDS}$ or $\overline{LDS}$ are asserted or $\overline{IACK}$ and $\overline{IRQ}$in are asserted, the MMU asserts $\overline{ED}$ to enable the data transceivers.

If $\overline{IACK}$ and $\overline{IRQ}$in and $\overline{UDS}$ or $\overline{LDS}$ are asserted, an interrupt acknowledge operation is performed. If $\overline{CS}$ and $\overline{UDS}$ or $\overline{LDS}$ are asserted, the MMU determines which operation to perform by decoding RS1-RS5 and R/$\overline{W}$. These signals tell which register is associated with the operation, which operation to perform, and whether the operation is local or global.

After each operation, $\overline{DTACK}$ is asserted to indicate to the processor that the operation is finished. When the processor negates $\overline{UDS}$ and $\overline{LDS}$, $\overline{DTACK}$ and $\overline{ED}$ are rescinded and PAD0-PAD15 are placed in the high-impedance state. If $\overline{AS}$ is negated, or had been negated since the last normal translation, the MMU enters the idle state.

After the $\overline{DTACK}$ handshake, if $\overline{AS}$ remains asserted and $\overline{CS}$ and $\overline{UDS}$ or $\overline{LDS}$ are asserted, another master operation is performed. If $\overline{AS}$ remains asserted and $\overline{GO}$in and $\overline{UDS}$ or $\overline{LDS}$ are asserted, another slave operation is performed.

4.6.1 Operations Address Map

Table 4-3 shows the operations address map. Each system register has an address at which it can be read or written. In addition, some addresses do not correspond to a register, but rather designate an operation to be performed by reading that location.

The data strobes are logically separate and operations using both are independent. The operation ends when both data strobes are negated.

Some addresses are reserved for future expansion. Any access to an unused location will result in a null operation. If the access is a read, the appropriate byte of the data bus is driven high. If the access is a write, no side-effect occurs.

Table 4-3. Register/Operations Address Map

Address Binary RRRRRR SSSSSS 543210	Hex	Operation R	Operation W	Register or Operation
000000	00	L	G	AST 0 (Alternate, FC3=0)
000010	02	L	G	AST 1 (User Data)
000100	04	L	G	AST 2 (User Program)
000110	06	L	G	AST 3 (Alternate, FC3=0)
001000	08	L	G	AST 4 (Alternate, FC3=0)
001010	0A	L	G	AST 5 (Supervisor Data)
001100	0C	L	G	AST 6 (Supervisor Program)
001110	0E	L	G	AST 7 (Interrupt Acknowledge)
010000	10	L	G	AST 8 (Alternate, FC3=1)
010010	12	L	G	AST 9 (Alternate, FC3=1)
010100	14	L	G	AST 10 (Alternate, FC3=1)
010110	16	L	G	AST 11 (Alternate, FC3=1)
011000	18	L	G	AST 12 (Alternate, FC3=1)
011010	1A	L	G	AST 13 (Alternate, FC3=1)
011100	1C	L	G	AST 14 (Alternate, FC3=1)
011110	1E	L	G	AST 15 (Alternate, FC3=1)
100000	20	L	G	AC0 (LBA/Translation ADDR (MBS))
100001	21	L	G	AC1 (LAB/Translation ADDR (LSB))
100010	22	L	G	AC2 (LAM (MSB))
100011	23	L	G	AC3 (LAM (LSB))
100100	24	L	G	AC4 (PBA/Translated ADDR (MSB))
100101	25	L	G	AC5 (PBA/Translated ADDR (LSB))
100110	26	L	G	AC6 (Address Space Number)
100111	27	L	G	AC7 (Status Register)
101000	28	L	G	AC8 (Address Space Mask)
101001	29	L	G	DP Descriptor Pointer
101011	2B	L	G	IVR Interrupt Vector Register
101101	2D	L	G	GSR Global Status
101111	2F	L	G	LSR Local Status
110001	31	L	L	SSR Segment Status and Transfer Descriptor Operation
111001	39	L	LN	IDP Interrupt Description Pointer
111011	3B	L	LN	RDP Result Descriptor Pointer
111101	3D	G	LN	Direct Translation Operation
111111	3F	G	LN	Load Descriptor Operation
{Otherwise}		LN	LN	Null Operation

L: Local G: Global N: Null Operation

RS0 is an internal signal
If $\overline{UDS}$ = 0 and $\overline{LDS}$ = 1 → RS0 = 0
If $\overline{UDS}$ = 1 and $\overline{LDS}$ = 0 → RS0 = 1
If $\overline{UDS}$ = 0 and $\overline{LDS}$ = 0 → RSO = X
If $\overline{UDS}$ = 1 and $\overline{LDS}$ = 1 → RSO = X

4.6.2 Local Operations

Some operations, such as reading the status registers, affect only one MMU. These are called local operations. Local operations include: interrupt acknowledge, read system registers, transfer descriptor, and write segment status register.

4.6.2.1 INTERRUPT ACKNOWLEDGE. The interrupt acknowledge operation is performed if $\overline{IACK}$ and $\overline{IRQ}$in are asserted at the beginning of the operation phase. During interrupt acknowledge, the contents of the interrupt vector register are placed on D0-D7, to provide the MPU with a vector number. A timing diagram of the interrupt acknowledge operation is shown in Figure 4-9.

4.6.2.2 READ SYSTEM REGISTER. Each system register has an address at which it can be read. Each MMU should be chip selected at a different location to access the registers in each. During a processor read of the IDP, the IDP is first loaded from the priority encoder and then gated onto D0-D7. An example timing diagram of this operation is shown in Figure 4-10.

4.6.2.3 TRANSFER DESCRIPTOR. In order to read the contents of a descriptor, they must be transferred into the accumulator and read from there. The descriptor pointer is first written by the processor with the number of the descriptor desired. The transfer descriptor operation is then performed by reading from the SSR address ($31).

The contents of the selected descriptor are then transferred into the accumulator as sown in Table 4-3 and the contents of the SSR are gated onto D0-D7. The descriptor registers may then be read from the accumulator.

4.6.2.4 WRITE SEGMENT STATUS REGISTER. The SSR of any descriptor can be written using descriptor pointer (DP) as a pointer. Any bit may be written except the E bit. Enable may be cleared using this operation but it may not be set.

4.6.2.5 NULL OPERATION. A read or write of any location which is not a register will result in a null operation. In the case of a read, the data bus will read $FF and the $\overline{DTACK}$ signal will be asserted ending the bus cycle. In the case of a write, $\overline{DTACK}$ is asserted and no other effect occurs.

4.6.3 Global Operations

A global operation is one which is performed in parallel on all MMUs in the system. Global operations include all writes to system registers, the load descriptor operation, and direct translation. In global operation, one MMU is the master and the rest are slaves. The operation begins with $\overline{CS}$ and $\overline{UDS}$ or $\overline{LDS}$ asserted on one MMU. The MMU with $\overline{CS}$ asserted becomes the master for that operation. The master asserts $\overline{GO}$out and, upon detecting $\overline{GO}$in as true, the other MMUs become slaves in the operation.

The $\overline{ANY}$, ALL, and $\overline{GO}$ pins must have pullups to V_{CC}, even if there is only one MMU in the system. In the case of a single-MMU system, global operations then become local only.

4.6.3.1 WRITE SYSTEM REGISTER. Each system register that can be written to is written globally. This includes: AC0-AC8, AST0-AST15, DP, IVR, GSR, and LSR. The operation is performed by writing to the address of the desired register (see Figure 4-11).

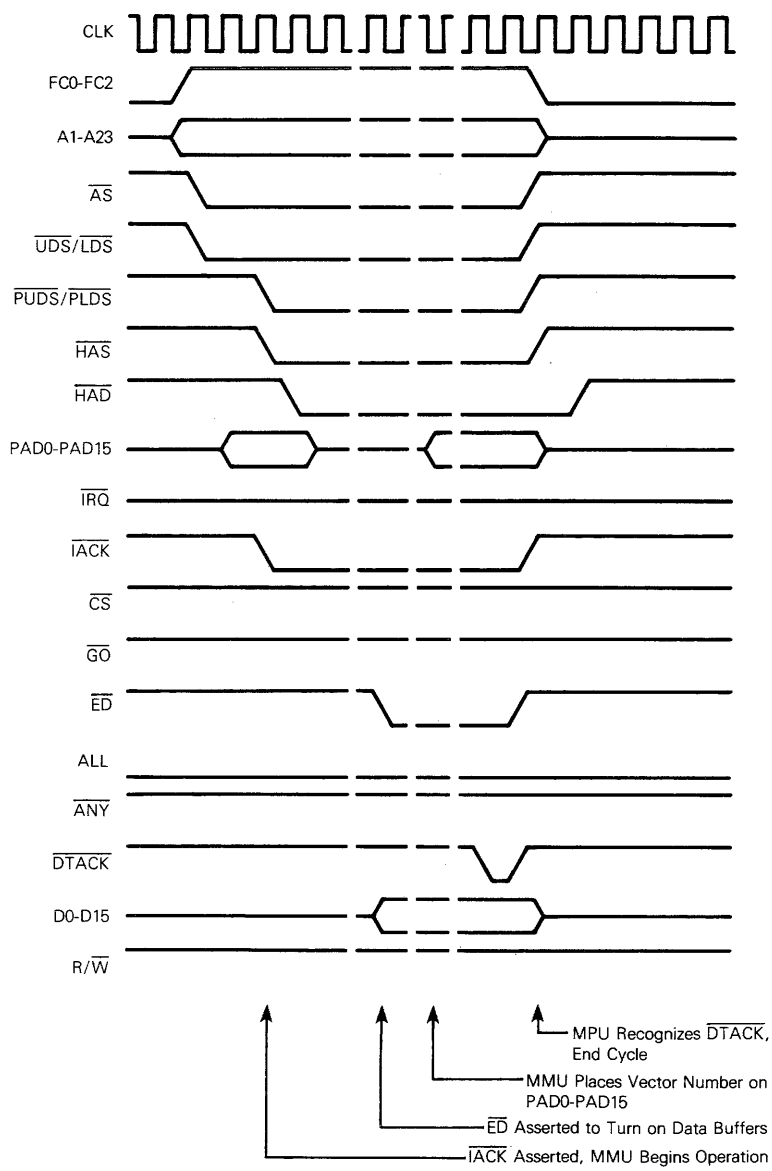


Figure 4-9. Interrupt Acknowledge Timing

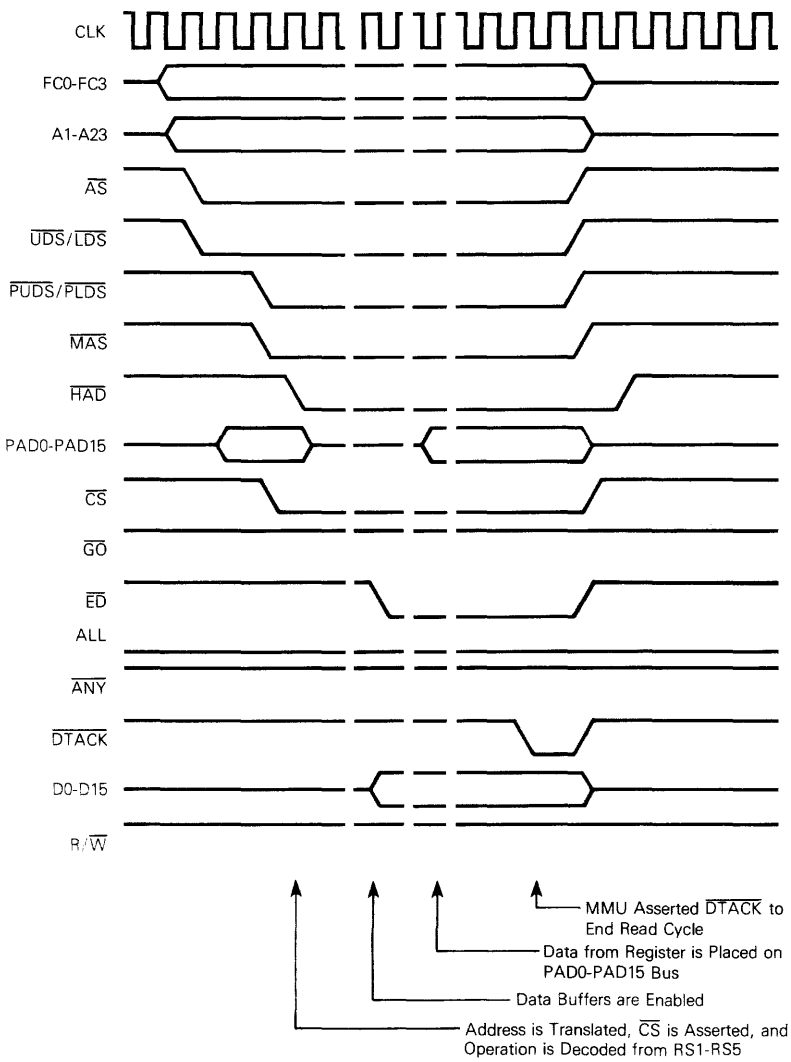


Figure 4-10. Read System Register Timing

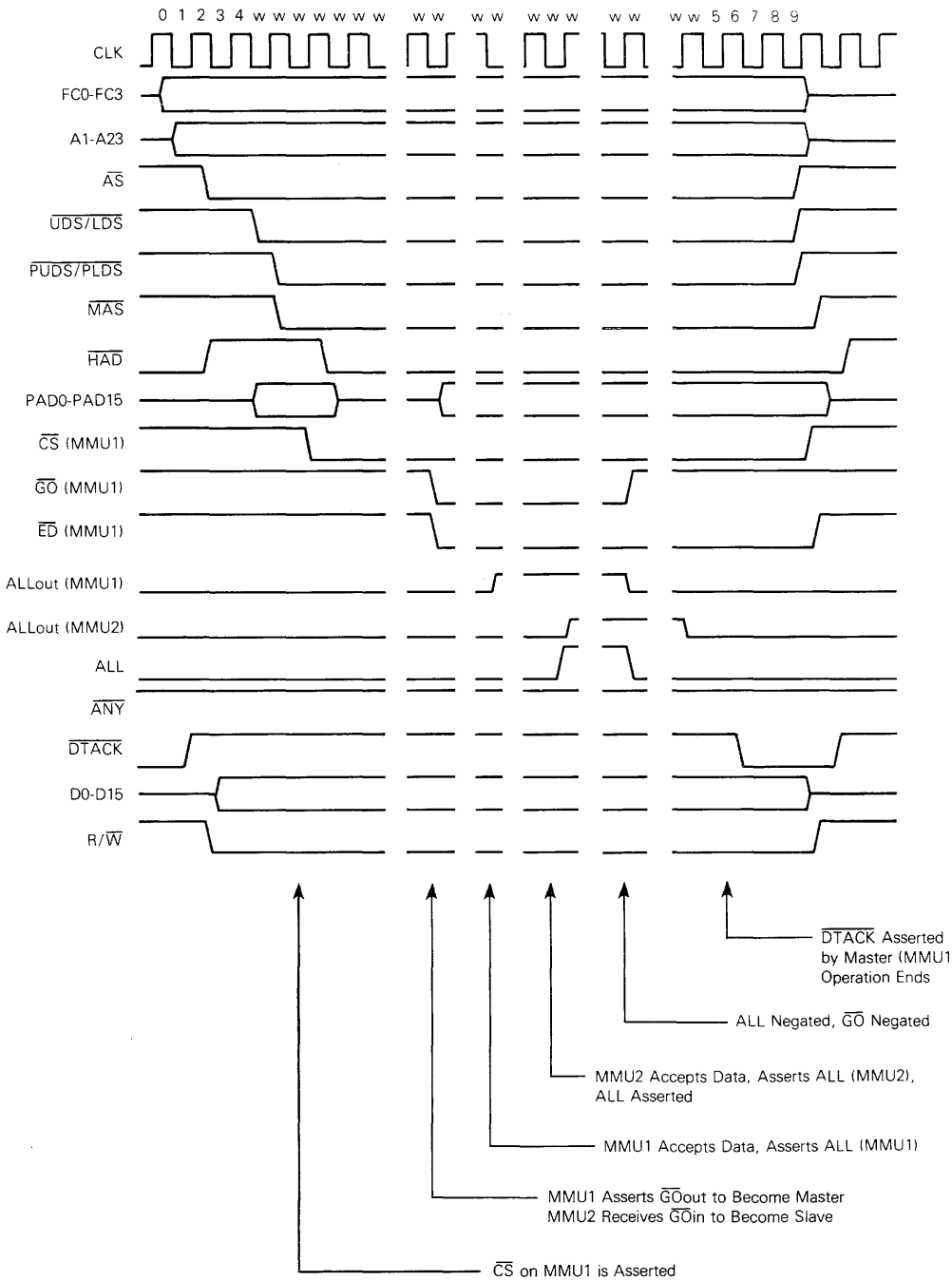


Figure 4-11. Global Write Systems Register Timing

The MMU which has $\overline{CS}$ asserted becomes the master by asserting $\overline{GO}$out. The other MMUs detect $\overline{GO}$in and become slaves. Each MMU transfers the data on the data bus to the selected register. If the write is to a byte of the accumulator, that register is marked as global. If F is cleared in the GSR, L4-L7 are also cleared.

When the transfer is completed in each MMU, each will assert ALLout. After all MMUs have asserted ALLout, ALLin will be true and, upon detecting ALLin, the master rescinds $\overline{GO}$.

4.6.3.2 LOAD DESCRIPTOR OPERATION. Descriptors are loaded by transferring the contents of the accumulator to the descriptor after performing global checks for collisions. A collision exists when two or more enabled descriptors are programmed to translate the same logical address.

To prepare for descriptor loading, the accumulator must be loaded globally with the LBA, LAM, ASN, and ASM as described in **3.2.2 Accumulator.** To make global collision checks, AC0, AC1, AC2, AC3, AC6, and AC8 must have been globally loaded. If they are, the global accumulator for load (GAL) bit in the LSR of each MMU is set. To initiate the operation, a read from address $3F is done. If the load is successful, the data bus will be set to $00. If a collision is found, the load is unsuccessful and the data bus is set to $FF.

During the load descriptor operation, the MMU with $\overline{CS}$ asserted becomes the master by asserting $\overline{GO}$out. The other MMUs detect $\overline{GO}$in and become slaves. The slave MMUs decode the operation from RS1-RS5, R/$\overline{W}$, and the data strobes ($\overline{UDS}$, $\overline{LDS}$). The descriptor whose number is in the descriptor pointer is disabled (its E bit is cleared) so that it cannot cause a collision.

If the GAL bit in the LSR of a slave is clear, L4-L7 is encoded to indicate LD and $\overline{ANY}$out are asserted. If GAL is set, the slave checks the enabled descriptors against its accumulator for collisions. If a conflict is found, the slave asserts $\overline{ANY}$out and loads its RDP with the number of the descriptor which caused the collision. If no collision is detected, L4-L7 are cleared. When $\overline{GO}$in is detected, ALLout, and $\overline{ANY}$out are negated and the operation ends.

The master aborts the transfer if there is a local descriptor conflict, if the GAL bit is clear, or if $\overline{ANY}$in is asserted. If the failure was not local, L4-L7 are cleared. Otherwise, L4-L7 are encoded with LD and $\overline{ANY}$out is asserted by the master. The master then puts $FF on D0-D7 to indicate failure to the MPU, negates ALLout and $\overline{ANY}$out, and rescinds $\overline{GO}$out. When $\overline{ANY}$in is negated, the operation is terminated.

If there were no local collisions, the GAL bit of the slave is set, and ALLin is asserted, the master completes the transfer and enables the loaded descriptor. It then puts $00 on D0-D7 to indicate success, clears L4-L7, negates ALLout, and rescinds $\overline{GO}$out. The example timing diagram in Figure 4-12 shows a successful load descriptor operation in a two-MMU system.

4.6.3.3 DIRECT TRANSLATION. The memory management mechanism can be used to directly translate the logical address into a physical address and make it available to the processor in the accumulator. The logical address to be translated is globally loaded into AC0-AC1 and the ASN to be used is loaded into AC6. Translation is initiated with a read from the address $3D.

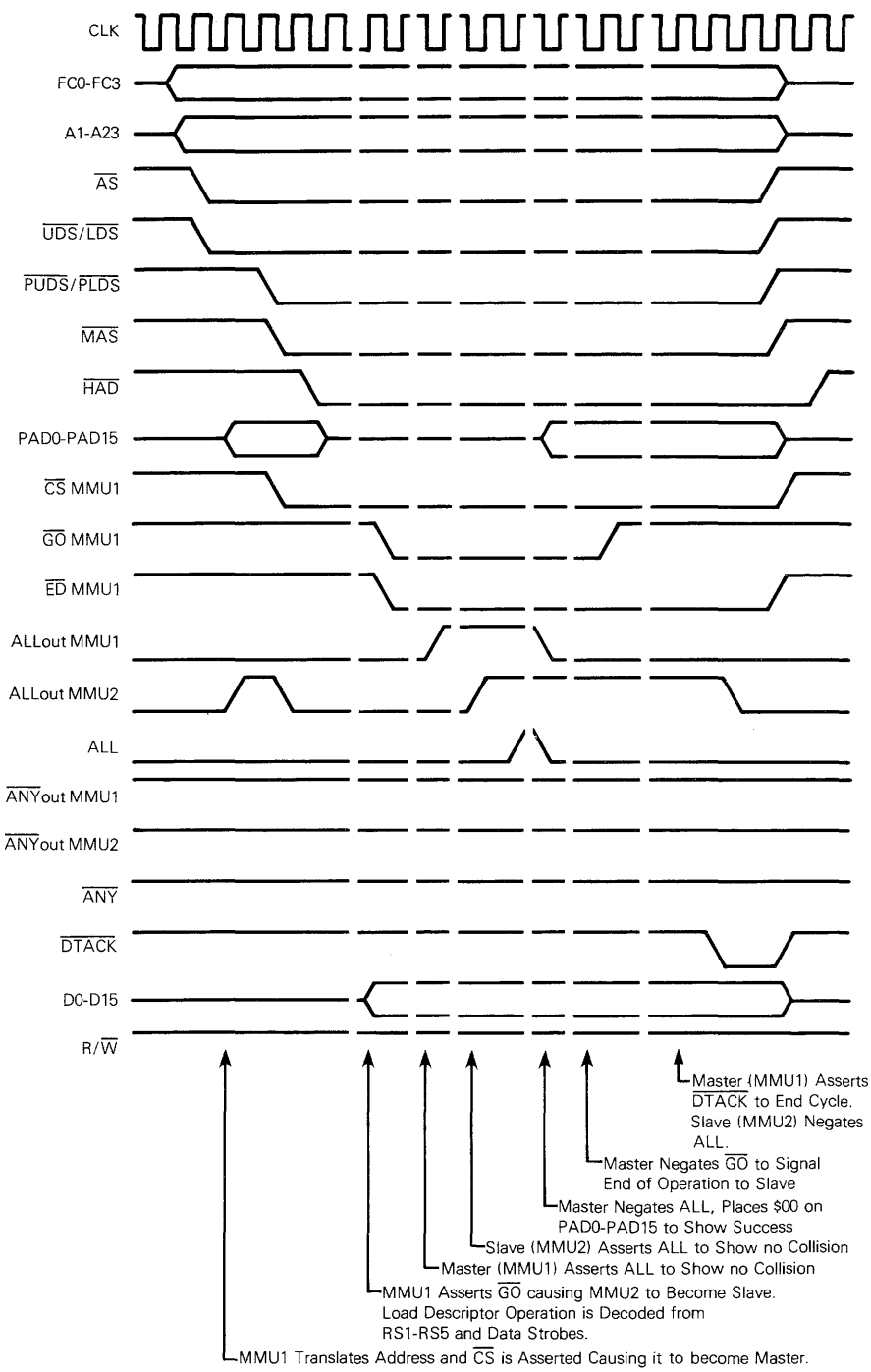


Figure 4-12. Successful Load Descriptor Operation in a Two—MMU System

If the translation is successful, the DP and RDP point to the descriptor which performed the translation and the physical address is loaded into AC4-AC5. The processor reads $00 from the data bus.

If the logical address could not be translated because it was globally undefined, the data bus is set to $FF to indicate the failure.

Using AC6 to supply the cycle address space number, each MMU attempts to match the logical address contained in AC0-AC1 with one of its enabled descriptors. Each MMU must have the same information in AC0, AC1, and AC6. The GAT (global accumulator for translation) bit in the LSR is set if these registers have each been globally loaded.

If a match is found, and GAT is set, the physical address is formed as in normal translation and put into AC4-AC5. The RDP and DP are loaded from the priority encoder and L4-L7 are encoded to indicate direct translation (DT). The master puts $00 on D0-D7 to signal that the translation was successful, and rescinds $\overline{GO}$ to terminate the operation.

If no match is found, or GAT is cleared, the MMU asserts ALLout and L4-L7 in the LSR are cleared. The master monitors $\overline{ANY}$in and ALLin.

If $\overline{ANY}$in becomes asserted, then another MMU performs the translation. The master puts $00 on D0-D7 to indicate success, negates ALLout, and rescinds $\overline{GO}$out. It waits until $\overline{ANY}$in is negated before terminating the operation.

If ALLin becomes asserted, then none of the MMUs performed the translation. The master puts $FF on D0-D7 to indicate failure, negates ALLout, and rescinds $\overline{GO}$out to terminate the operation. Each slave MMU negates $\overline{ANY}$out and ALLout when the master MMU rescinds GO at the end of the operation.

4.7 OPERATION LENGTH TABLE

The length of each operation, in system clock periods, is given in Table 4-1. The table includes operation lengths for both the single MMU case and for the multiple MMU situation. Both minimum and maximum numbers are given. Each operation is measured from the assertion of $\overline{CS}$ to the assertion of $\overline{DTACK}$. The minimum number is derived from "ideal" conditions. Each signal is assumed to appear at precisely the right time to be used internally for the optimum speed of the operation. Parametric timing diagrams are given in Figures 7-2, 7-3, and 7-4. These should be used for system hardware design considerations. The flow diagram (Figure 4-4) should be used for debugging multiple-MMU systems.

The example timing diagram in Figure 4-13 shows a successful direct translation operation in a two-MMU system.

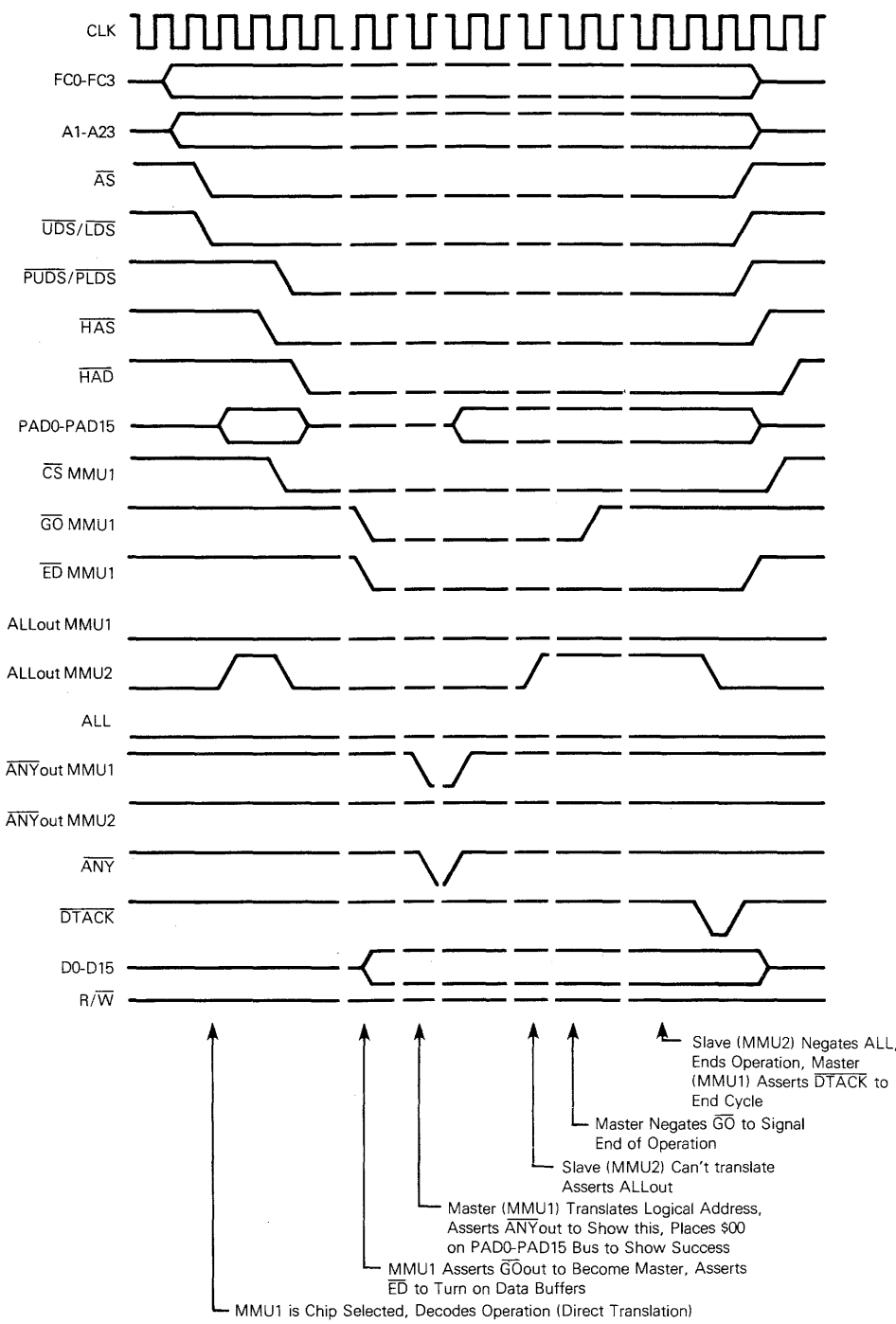


Figure 4-13. Timing for Successful Direct Translation in a Two-MMU System

SECTION 5
HARDWARE CONSIDERATIONS

5.1 $\overline{\text{MAS}}$ TIMING MODES AND PHYSICAL ADDRESS

The mapped address strobe ($\overline{\text{MAS}}$) signals that a valid translated address is present on the physical address bus. It should be included in the generation of the physical address strobe ($\overline{\text{PAS}}$) which is then used to gate the memory decode circuitry. $\overline{\text{AS}}$ is also included in the $\overline{\text{PAS}}$ to effect a quick release of the bus at the end of a cycle.

$\overline{\text{MAS}}$ can be programmed with the MODE pin in three modes:

Mode A $\overline{\text{MAS}}$ is asserted asynchronously. $\overline{\text{MAS}}$ is asserted as soon as a match for the function code and logical address is found, except in the case of a translation following an MMU operation. In this case, $\overline{\text{MAS}}$ will be asserted as described in **Mode S2** below. Two circuits for generating physical address strobe are shown in Figure 5-1(a) and Figure 5-1(b).

The circuit in Figure 5-1(a) uses a delay element to delay $\overline{\text{MAS}}$ to produce $\overline{\text{PAS}}$. The delay time should be equal to the delay from $\overline{\text{MAS}}$ to PAD0-PAD15 valid plus the desired setup time for physical addresses required by the user's bus. This circuit should be used when the setup time for physical addresses to $\overline{\text{PAS}}$ must be greater than parameter 13. Since the setup time from PAD0-PAD15 to $\overline{\text{MAS}}$ valid (parameter 13a) may be negative, the delay element must allow for this setup time as well as the setup required by the user's physical address bus.

The circuit shown in Figure 5-1(b) uses a delay element to delay $\overline{\text{AS}}$ to produce $\overline{\text{PAS}}$. The delay value must be equal to the $\overline{\text{AS}}$ to PAD0-PAD15 translation time plus the setup required for the user's bus. This circuit should be used when the setup time for physical addresses can be equal to or less than parameter 13. This is because the $\overline{\text{MAS}}$ strobe will be delayed until the next falling edge of the clock when the previous cycle was an MMU operation. This assures a positive address setup time equal to the setup time from PAD0-PAD15 to $\overline{\text{MAS}}$ asserted valid in mode S2 (parameter 13). This method is preferred if the setup time given in parameter 13 is sufficient for the user's bus since it does not require the additional delay of $\overline{\text{PAS}}$ due to the delay time of PAD0-PAD15 to $\overline{\text{MAS}}$ valid.

In either case, the translation time will be longer for translations following an MMU operation. See length of operations in Table 4-1 for more information.

Mode S1 $\overline{\text{MAS}}$ is asserted on the first rising edge of CLOCK after the physical address if valid. This mode was intended to allow the generation of $\overline{\text{PAS}}$ by ORing $\overline{\text{AS}}$ and $\overline{\text{MAS}}$. $\overline{\text{MAS}}$ asserts $\overline{\text{PAS}}$ and $\overline{\text{AS}}$ negates it to release the bus quickly at the end of the cycle. A circuit to generate $\overline{\text{PAS}}$ using $\overline{\text{MAS}}$ in mode S1 is shown in Figure 5-1(c).

Mode S2 $\overline{MAS}$ is asserted on the first falling edge of CLOCK after the address is valid. This mode was intended to allow the generation of $\overline{PAS}$ using AS and $\overline{MAS}$ only. A circuit to do this is shown in Figure 5-1(c).

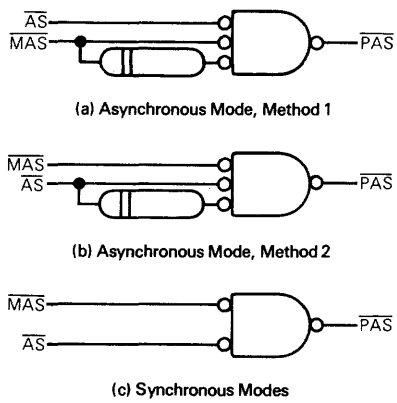


Figure 5-1. Physical Address Strobe Generation

5.2. PHYSICAL DATA STROBES

The physical data strobes, physical upper data strobe (PUDS), and physical lower data strobe (PLDS), should be generated using $\overline{UDS}$ or $\overline{LDS}$ gated with $\overline{MAS}$, $\overline{WIN}$, and R/$\overline{W}$. $\overline{WIN}$ and R/$\overline{W}$ will prevent the data strobes from being asserted during the write portion of and read-modify-write cycle on a write-protected segment. $\overline{MAS}$ is included to prevent data strobes from being asserted on a write cycle of a write-protected segment. A circuit to generate physical data strobes is shown in Figure 5-2. Three-state buffers are used to allow access from the physical bus by an external processor.

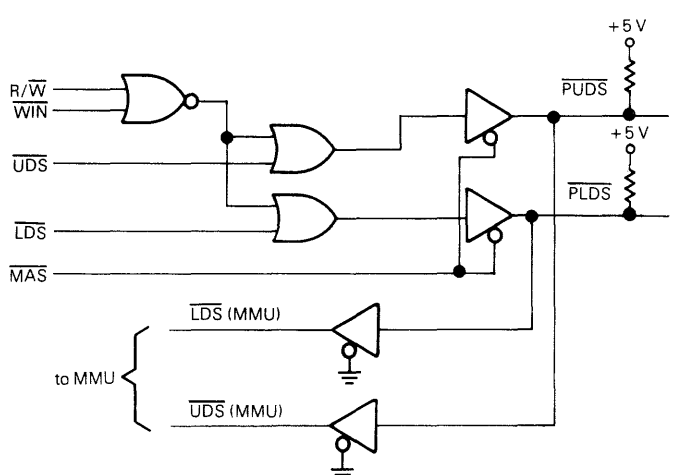


Figure 5-2. Generation of Physical Read/Write and Physical Data Strobes

5.3 INTERRUPTS

When the MC68000 responds to an interrupt, it places the interrupt acknowledge ($\overline{IACK}$) function code on FC0-FC3 and the level of the interrupt to which it is responding on the address lines A1-A3. This is not a true memory access, but the MMU must translate it since $\overline{AS}$ is asserted. To prevent the MMU from attempting to translate an address during an interrupt acknowledge cycle, the $\overline{MAS}$in signal is forced low. This informs the MMU that the cycle was translated by another MMU. However, the MMU will still assert ALLout to show that it could not translate the address. To prevent ALLin from being asserted and causing a fault, ALL must also be forced low. A circuit to do this is shown in Figure 5-3; however, it is acceptable to dedicate a descriptor to translate the $\overline{IACK}$ access ($FFFFFX), thus eliminating this external circuitry.

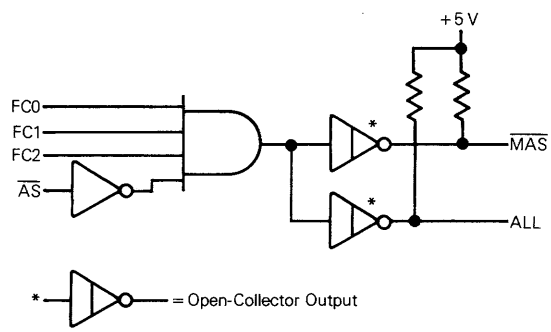


Figure 5-3. Circuit to Inhibit Translation During $\overline{IACK}$

SECTION 6
SOFTWARE CONSIDERATIONS

6.1 SEGMENT MAPPING EXAMPLE

In constructing segments, the size of a given segment is determined by the logical address mask. Although there are no constraints on which bits are significant, one approach is to allow only contiguous, low order zeroes ("don't cares"). With this constraint, if there are N zeroes, the size of the segment is $2^{(8+N)}$ bytes. Since the seven low-order address lines bypass the MMU, the smallest possible segment is 128 words (256 bytes).

In the logical address space, a segment defined this way extends from the address formed by the LBA with zeroes in the "don't care" positions in the LAM to the address formed by the LBA with ones in the "don't care" bit positions. In the physical address space, the segment extends from the address formed by the PBA with zeroes in the "don't care" bit positions in the LAM to the address formed with ones in the "don't care" positions.

This system is ideal for use with the binary buddy algorithm in which segments are all of 2^k multiple size.

Figure 6-1 shows an example memory map. In this example, the map has been divided between two users and the operating system. The user tasks each have three segments: A, B, and C. The operating system also has three segments: O, V, and R.

Segment R maps the logical addresses unchanged to the physical address space, but only for address space number $00. This segment is automatically generated in descriptor zero in the master MMU on reset. Segments O and V also belong to the operating system and are accessed only by ASNs with bit seven set. This is an arbitrary assignment and need not be followed.

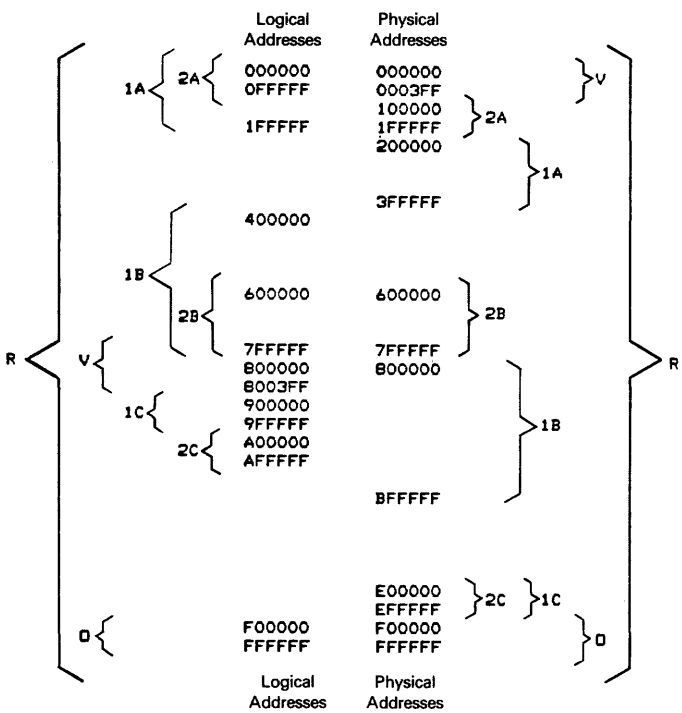
Segments 1A, 1B, and 1C belong to user number one, and are accessed by address space numbers $01 and $81. User one would be assigned ANS $01 and the operating system would use $81 to access those segments. A parallel situation exists with user two.

Note that segments 1A and 2A are isolated from each other even though they share the same logical addresses. User one is prevented from accessing the same memory as user two because his ASN does not match segment 2A. Segments can overlap in physical memory, however, as 1C and 2C do here.

Note the manner in which segments 1B and 2B are defined. Here the logical and physical base addresses are considered to be the top of the segment rather than the bottom. This is useful in describing push-down, pop-up stacks which grow towards low memory. The same segment can be

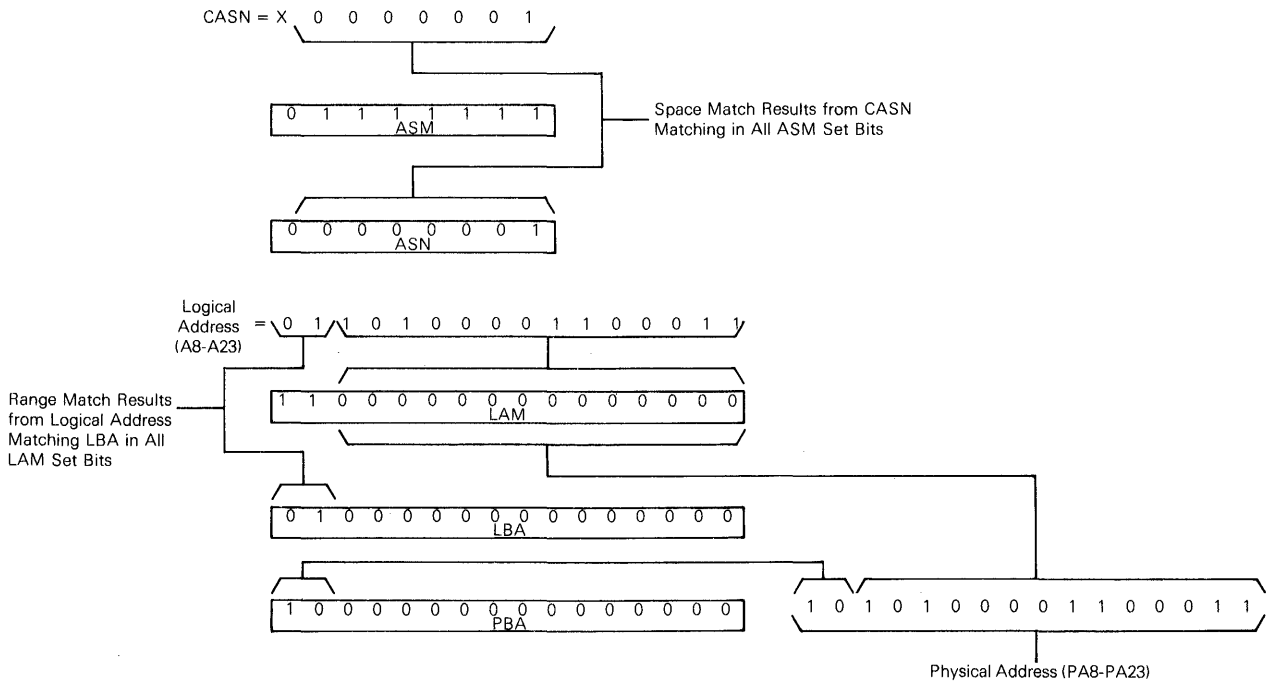
described in the other way, also. An alternate descriptor for segment 1B is given below and a translation example is given in Figure 6-2.

Descriptor	1B
Logical Base Address	$4000
Logical Address Mask	$C000
Physical Base Address	$8000
Address Space Number	$01
Address Space Mask	$7F



Segment	R	V	0	1A	1B	1C	2A	2B	2C
Logical Base Address (LBA)	0000	8000	F000	0000	7FFF	9000	0000	7FFF	A000
Logical Address Mask (LAM)	0000	FFFC	F000	E000	C000	F000	F000	E000	F000
Physical Base Address (PBA)	0000	0000	F000	2000	BFFF	E000	1000	7FFF	E000
Address Space Number (ASN)	00	80	80	01	01	01	02	02	02
Address Space Mask (ASM)	FF	80	80	7F	7F	7F	7F	7F	7F

Figure 6-1. Address Map Example



Physical address is formed by passing the logical address (A8-A23) in those positions which are clear ("don't care" bits) in the LAM. In the other LAM bit positions (set positions) the PBA is gated out to the physical address bus.

Figure 6-2. Translation Example

6.2 SEGMENTATION

Since segment sizes must be multiples of two, multiple descriptors can be used to map a segment of non-binary size. For example, a segment of 70K bytes could be constructed using two descriptors: one of 64K bytes and one of 8K bytes, losing 2K bytes to internal fragmentation. A purely binary system would allocate 128K bytes, wasting 58K bytes. For this reason, a modified binary buddy allocation algorithm should be used.

6.3 VIRTUAL MEMORY SUPPORT

The MC68451 MMU supports virtual memory systems as well. In a virtual system, the virtual address is larger than the physical RAM available and a combination of memory management hardware and software maintain a portion of the address space on a fast backing store such as a disk.

Virtual memory systems are usually implemented with a paging type of memory management system. In paging systems, the logical (virtual) address space is divided into equal length pages. A task is then allocated a certain number of these pages at initial load time and the memory management hardware is programmed to map these virtual pages to physical pages. This is equivalent to setting all descriptors to the same length. Thus, the MMU serves as a content-addressable translation buffer of 32 pages. In such a system, a page fault is caused when an access is attempted to a page that is not currently mapped by the MMU. This is an error in a non-virtual system but signals a page fault in a virtual system.

The other ingredient in a virtual system is a processor that is capable of aborting any instruction and finishing it a later time. This is necessary because any memory access could potentially be the cause of a page fault. When this occurs, the faulting bus cycle is suspended and the operating sytem takes corrective action. The bus cycle is then re-run and the instruction is continued.

The MC68000 does not save enough information when processing a bus error to return and finish the instruction that was being executed. Therefore, a true virtual system is not possible using a single MC68000 MPU. The MC68010 virtual microprocessor (VMPU) will allow a faulted bus cycle to be re-run and the instruction to be continued. This processor can, in conjunction with the MC68451 MMU, be used to implement a virtual memory system.

The MMU also provides hardware assistance for virtual paging systems. The logical address, the cycle address space number, and the R/$\overline{W}$ line are latched in the accumulator on a fault. This information is used by the page fault handler to fix the page fault. In addition, the used and modified bits in the segment status register allow the implementation of a variety of page replacement algorithms.

6.4 INITIALIZATION SOFTWARE

After a reset (power-on or processor initiated), the master MMU (the MMU for which $\overline{CS}$ was asserted during reset) will map the logical addresses unchanged into the physical address space using descriptor zero (see **2.6 RESET**). This will allow the processor to fetch its supervisor stack and program counter (if it was a power-on reset) and begin executing the operating sytem initialization routine. See the *MC68000 Data Sheet* for more information.

The operating system would then set up descriptors for itself and system resources (such as the MMU). To load a descriptor, the operating system loads the descriptor number in the DP register, and the LBA, LAM, PBA, ASN, and ASM into the accumulator as described in **4.6.3.2 LOAD DESCRIPTOR OPERATION.**

The processor then reads from the appropriate physical address to begin the loading operation. The MMM globally checks for conflicts and loads and enables the descriptor if none are found. As a result of the read, the processor gets a status byte in the low byte of the word. The status will read $00 if the load was successful and $FF if there was a conflict. If a conflict occurred, the RDP can be used to find the highest priority conflicting descriptor.

A descriptor can be quickly disabled by writing to its segment status register. The I and WP bits can be programmed and the U and M bits can be cleared but the E bit can be set only by a load descriptor operation.

Descriptors would then be set up for the user tasks and a task would be selected to execute. Address space table entries AST1 and AST2 would then be loaded with the address space number of the task to be run. These are the address spaces of user data and user program in the MC68000. The program counter and status register to be used by the task are then pushed onto the system stack. The processor then executes an RTE instruction which fetches the status register and program counter off of the stack. The status register should have had the supervisor bit cleared so that the processor will enter the user state and its acesses are then mapped through AST1 and AST2 to start the user task.

To return to the operating system from a user task, a watchdog timer could be used to interrupt the processor. The exception processing caused by this would switch the processor to the supervisor state and the supervisor address spaces would be mapped by the operating systems descriptors.

6.5 CONTEXT SWITCHING

Switching the MMU from one user task to the other is very efficient. Suppose two user tasks were present in memory and the processor had returned to the operating system as described above. To switch tasks, the operating system would change AST1 and AST2 to the ASN of the user task which it wished to execute. It would then push the new status register and program counter on the stack and execute an RTE.

Switching between two supervisor tasks is more complex. If AST5 and AST6 are changed while the processor is in the supervisor state, subsequent accesses are immediately mapped through the new address space. A move multiple (MOVEM) using the predecrement mode followed by an illegal instruction can be executed to perform the switch. The processor prefetch pipeline fetches the MOVEM and the illegal instruction, alters AST6 and AST5 (data entry last), then traps through the illegal instruction routine to the new supervisor task. A flag (possibly the illegal instruction opcode) is used to distinguish between normal illegal instructions and attempts to switch tasks in this manner.

Another method is to have a task in the user space perform the switch. The supervisor stack pointer is set up, the processor alters the status register to put itself in the user state, AST5 and AST6 are changed, and the processor traps to the supervisor task.

SECTION 7
ELECTRICAL SPECIFICATIONS

7.1 MAXIMUM RATINGS

Characteristics	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 V_{SS} or V_{CC}).

7.2 THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Rating
Thermal Resistance			
Ceramic		30	
Plastic	θ_{JA}	30	°C/W
Type B Chip Carrier		50	
Type C Chip Carrier		50	

7.3 POWER CONSIDERATIONS

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

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

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.

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

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

Solving equations 1 and 2 for K gives:

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

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.

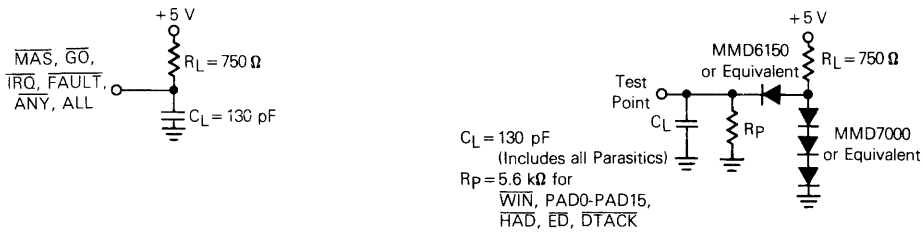


Figure 7-1. Test Loads

7.4 AC ELECTRICAL SPECIFICATIONS (V_{CC} = 5.0 Vdc ± 5%, GND = 0 Vdc, T_A = 0°C to 70°C, see Figures 7-2, 7-3, and 7-4)

Number	Characteristic	8 MHz		10 MHz		Unit
		Min	Max	Min	Max	
1	Clock Period	125	500	100	500	ns
2	Clock Width Low	55	250	45	250	ns
3	Clock Width High	55	250	45	250	ns
4	Clock Fall Time	–	10	–	10	ns
5	Clock Rise Time	–	10	–	10	ns
6	$\overline{AS}$ Width Asserted	200	–	180	–	ns
7	FC0-FC3 Valid to $\overline{AS}$ Asserted (FC Setup Time)	50	–	40	–	ns
8	A8-A23, FC0-FC3 Valid After $\overline{AS}$ Negated (Address, FC Hold Time)	0	–	0	–	ns
9	A8-A23 Valid to $\overline{AS}$ Asserted (Address Setup)	25	–	15	–	ns
10	$\overline{AS}$ Asserted to PAD0-PAD15 Valid	–	150	–	125	ns
11	$\overline{AS}$ Negated to $\overline{MAS}$ Negated	–	100	–	75	ns
12	$\overline{AS}$ Asserted to $\overline{MAS}$ Asserted (Asynchronous Mode)	–	150	–	125	ns
13	PAD0-PAD15 Valid to $\overline{MAS}$ Asserted (Mode S1 or S2)	30	–	25	–	ns
13a	PAD0-PAD15 Valid to $\overline{MAS}$ Asserted* (Mode A)	−20	–	−20	–	ns
14	Clock High to $\overline{MAS}$ Asserted (Mode S1)	–	80	–	60	ns
15	Clock Low to $\overline{MAS}$ Asserted (Mode S2)	–	80	–	60	ns
16	Clock High to $\overline{HAD}$ Asserted	–	100	–	80	ns
17	Clock High to $\overline{HAD}$ Negated	–	100	–	80	ns
18	PAD0-PAD15 Valid to $\overline{HAD}$ Asserted	40	–	30	–	ns
19	PAD0-PAD15 Valid after $\overline{HAD}$ Asserted	40	–	30	–	ns
20	$\overline{AS}$ Negated to $\overline{HAD}$ Negated	60	–	50	–	ns
21	$\overline{AS}$ Asserted to R/$\overline{W}$ Valid	–	20	–	20	ns
22	R/$\overline{W}$ High after $\overline{AS}$ Negated	40	–	40	–	ns
22a	R/$\overline{W}$ Low after $\overline{AS}$ Negated	0	–	0	–	ns
23	$\overline{MAS}$ Asserted to R/$\overline{W}$ Transition (Read-Modify-Write)	0	–	0	–	ns
24	$\overline{MAS}$ Asserted to $\overline{WIN}$ Asserted	–	20	–	20	ns
25	$\overline{MAS}$ Negated to $\overline{WIN}$ Negated	0	40	0	40	ns
26	$\overline{MAS}$ Asserted to $\overline{IRQ}$ Asserted	–	120	–	120	ns
27	Clock High to $\overline{IRQ}$ Negated	–	200	–	180	ns
28	$\overline{AS}$ Asserted to ALL Hi-Z (No Match)	–	200	–	160	ns
29	$\overline{MAS}$ Asserted to ALL Negated (External Match)	–	80	–	60	ns
30	$\overline{FAULT}$ Width Asserted	5	–	5	–	Clk. Per.
31	$\overline{AS}$ Asserted to $\overline{FAULT}$ Asserted (USA)**	8.5	12.5	8.5	12.5	Clk. Per.
32	$\overline{AS}$ Asserted to $\overline{FAULT}$ Asserted (WV)**	13.5	18.5	13.5	18.5	Clk. Per.
33	$\overline{AS}$ Negated to $\overline{FAULT}$ Hi-Z ($\overline{AS}$ Slower than Minimum Width)	–	100	–	90	ns
34	$\overline{CS}$ to $\overline{UDS}$, $\overline{LDS}$ Nonoverlap (to Avoid Operation)	0	–	0	–	ns
35	RS1-RS5, $\overline{CS}$, and $\overline{IACK}$ Valid to Clock Low	20	–	15	–	ns
36	RS1-RS5 Valid to $\overline{CS}$ Low	0	–	0	–	ns

* Except immediately following an operation. See Table 4-1 for more information.

** Assumes $\overline{AS}$ is not a limiting factor.

NOTES FOR FIGURES 7-2, 7-3, AND 7-4:

1. These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.
2. Waveform measurements for all inputs and outputs are specified at: logic high = 2.0 V, logic low = 0.8 V.

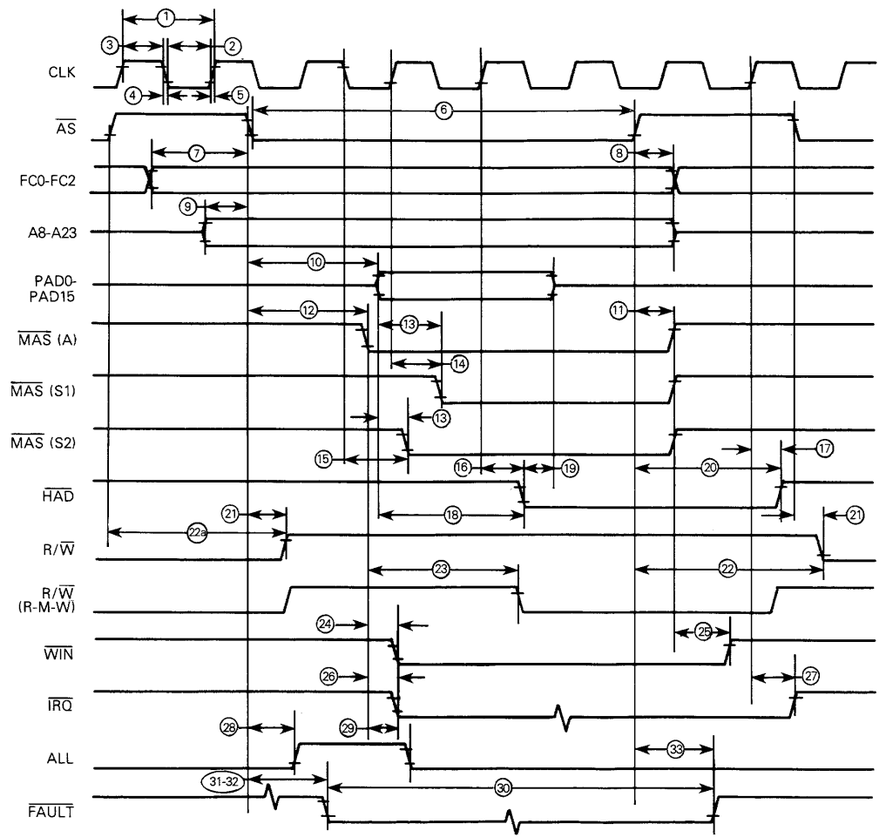


Figure 7-2. Normal Translation Timing

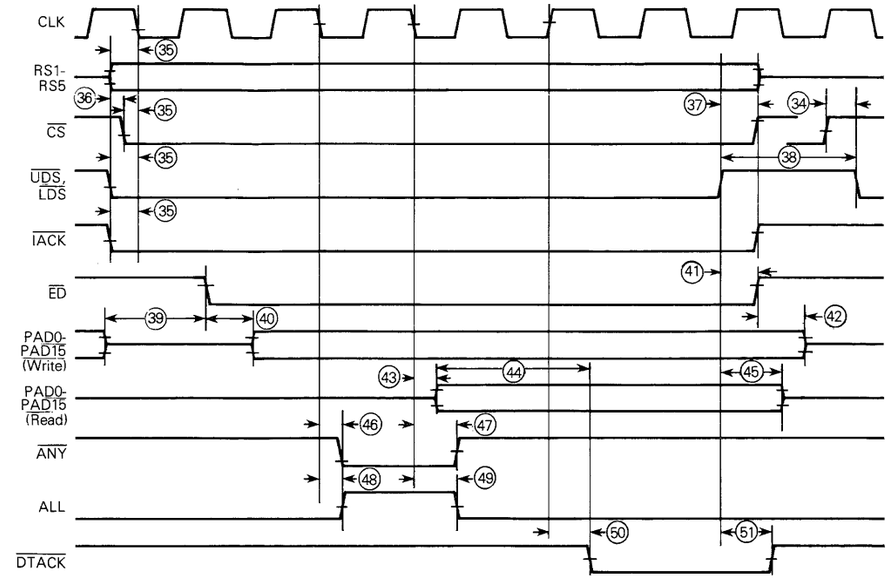


Figure 7-3. Operations Timing

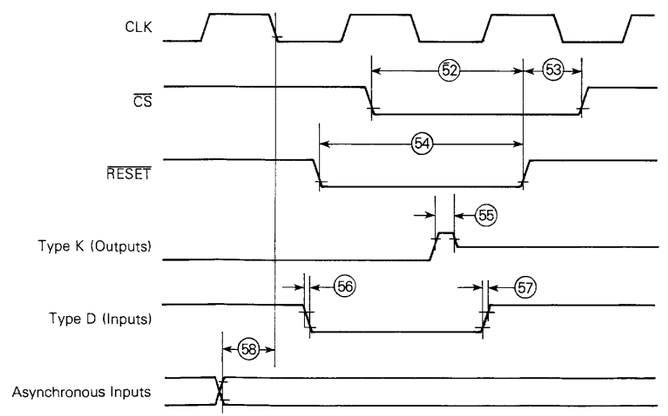


Figure 7-4. Miscellaneous Signal Timing

Timing Diagrams
(Timing tables located
on pages 7-2 and 7-3.)

7.4 AC ELECTRICAL SPECIFICATIONS (Continued)

(V_{CC} = 5.0 Vdc ± 5%, GND = 0 Vdc, T_A = 0°C to 70°C; see Figures 7-2, 7-3, and 7-4)

Number	Characteristic	8 MHz		10 MHz		Unit
		Min	Max	Min	Max	
37	Data Strobes Negated to RS1-RS5, $\overline{CS}$, and $\overline{IACK}$ Invalid	0	–	0	–	ns
38	$\overline{UDS}$, $\overline{LDS}$ Width Negated	140	–	100	–	ns
39	PAD0-PAD15 Hi-Z to $\overline{ED}$ Asserted	60	–	40	–	ns
40	$\overline{ED}$ Asserted to PAD0-PAD15 Valid (Write)	–	40	–	40	ns
41	$\overline{UDS}$, $\overline{LDS}$ Negated to $\overline{ED}$ Negated	–	100	–	50	ns
42	$\overline{ED}$ Negated to PAD0-PAD15 Hi-Z (Write)	–	60	–	60	ns
43	Clock Low to PAD0-PAD15 Valid (Read)	–	100	–	100	ns
44	PAD0-PAD15 Valid to $\overline{DTACK}$ Asserted (Read)	187	–	150	–	ns
45	$\overline{UDS}$, $\overline{LDS}$ Negated to PAD0-PAD15 Hi-Z (Read)	–	120	–	100	ns
46	Clock Low to $\overline{ANY}$ Asserted (Output)	–	125	–	100	ns
47	Clock Low to $\overline{ANY}$ Hi-Z (Output)	–	125	–	100	ns
48	Clock Low to ALL Hi-Z (Output)	–	125	–	100	ns
49	Clock Low to ALL Negated (Output)	–	125	–	100	ns
50	Clock High to $\overline{DTACK}$ Asserted	–	75	–	55	ns
51	$\overline{UDS}$, $\overline{LDS}$ Negated to $\overline{DTACK}$ Negated	–	100	–	80	ns
52	$\overline{CS}$ Valid to $\overline{RESET}$ Negated ($\overline{CS}$ Setup before $\overline{RESET}$)	5	–	5	–	Clk. Per.
53	$\overline{CS}$ Valid after $\overline{RESET}$ Negated ($\overline{CS}$ Hold Time after $\overline{RESET}$)	– 10	60	– 10	60	ns
54	$\overline{RESET}$ Width Asserted (See Note 1)	10	–	10	–	Clk. Per.
55	Type K Pins Logic High to Hi-Z (See Note 2)	–	100	–	100	ns
56	Type D Inputs Fall Time (See Note 3)	–	200	–	200	ns
57	Type D Inputs Rise Time	–	200	–	200	ns
58	Asynchronous Input Setup Time	30	–	20	–	ns

NOTES:
1. Initial power-on-reset pulse shall be ≥ 100 ms to allow for system clock stabilization.
2. Type K outputs are: $\overline{DTACK}$, $\overline{MAS}$, $\overline{WIN}$, $\overline{HAD}$, $\overline{ED}$, and GO.
3. Type D inputs are: $\overline{FAULT}$, $\overline{IRQ}$, $\overline{MAS}$, $\overline{GO}$, $\overline{ANY}$, and ALL.

MC68451F – The following parameters, for the MC68451F, differ from those given for the MC68451 while all others remain unchanged.

Number	Characteristic	8 MHz		10 MHz		Unit
		Min	Max	Min	Max	
10	$\overline{AS}$ Asserted to PAD0-PAD15 Valid	–	125	–	100	ns
12	$\overline{AS}$ Asserted to $\overline{MAS}$ Asserted (Asynchronous Mode)	–	125	–	100	ns
13	PAD0-PAD15 Valid to $\overline{MAS}$ Asserted (Mode S1 or S2)	40	–	40	–	ns
13a	PAD0-PAD15 Valid to $\overline{MAS}$ Asserted* (Mode A)	– 10	–	– 10	–	ns

*Except immediately following an operation. See Table 4-1 for more information.

Timing diagrams (Figures 7-2, 7-3, and 7-4) are located on a foldout page at the end of this document.

7.5 ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, GND = 0 Vdc, T_A = 0°C to 70°C)

Characteristics		Symbol	Min	Max	Unit
Input High Voltage	All Inputs Except MODE MODE	V_{IH}	V_{SS} + 2.0 V_{CC}	V_{CC} –	V
Input Low Voltage	All Inputs Except MODE MODE	V_{IL}	V_{SS} – 0.3 –	V_{SS} + 0.8 V_{SS}	V
Input Leakage Current @ 5.25 V		I_{in}	–	20	μA
Hi-Z (Off-State) Input Current (V_{CC} = Max) PAD0-PAD15, $\overline{DTACK}$, $\overline{MAS}$, $\overline{WIN}$, $\overline{ED}$, $\overline{GO}$		I_{TSI}	–	20	μA
Output High Voltage (I_{OH} = –400 μA, V_{OH} = 2.4 V) All Outputs Except $\overline{FAULT}$, $\overline{IRQ}$, $\overline{ANY}$, ALL		V_{OH}	V_{SS} + 2.4	–	V
Output Low Voltage (V_{CC} = Max) (I_{OL} = 5.3 mA) (I_{OL} = 10.7 mA)	 PAD0-PAD15, $\overline{DTACK}$, $\overline{MAS}$, $\overline{WIN}$, $\overline{HAD}$, $\overline{ED}$, $\overline{GO}$ $\overline{FAULT}$, $\overline{IRQ}$, $\overline{ANY}$, ALL	V_{OL}	 – –	 0.5 0.5	V
Power Dissipation		P_D	–	1.5	W
Capacitance (Package Type Dependent) (V_{in} = 0 V, T_A = 25°C, Frequency = 1 MHz)		C_{in}	–	20	pF

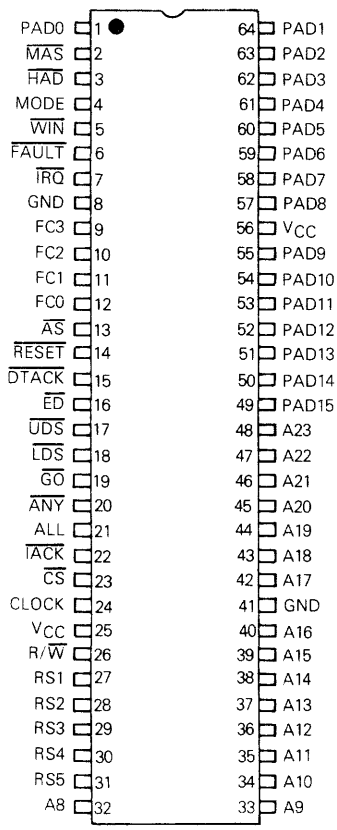
SECTION 8

8.1 ORDERING INFORMATION ($T_A = 0°C$ to $70°C$)

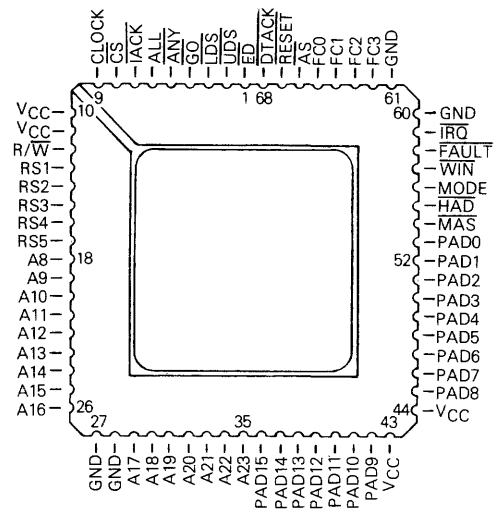
Package Type	Frequency	Order Number
Ceramic	8 MHz	MC68451L8
L Suffix		MC68451L8F
	10 MHz	MC68451L10
		MC68451L10F
Plastic	8 MHz	MC68451G8
G Suffix		MC68451G8F
	10 MHz	MC68451G10
		MC68451G10F
Type B Leadless	8 MHz	MC68451ZB8
Chip Carrier		MC68451ZB8F
ZB Suffix	10 MHz	MC68451ZB10
		MC68451ZB10F
Type C Leadless	8 MHz	MC68451ZC8
Chip Carrier		MC68451ZC8F
ZC Suffix	10 MHz	MC68451ZC10
		MC68451ZC10F
Pin Grid Array	8 MHz	MC68451R8
R Suffix		MC68451R8F
	10 MHz	MC68451R10
		MC68451R10F

8.2 PIN ASSIGNMENTS

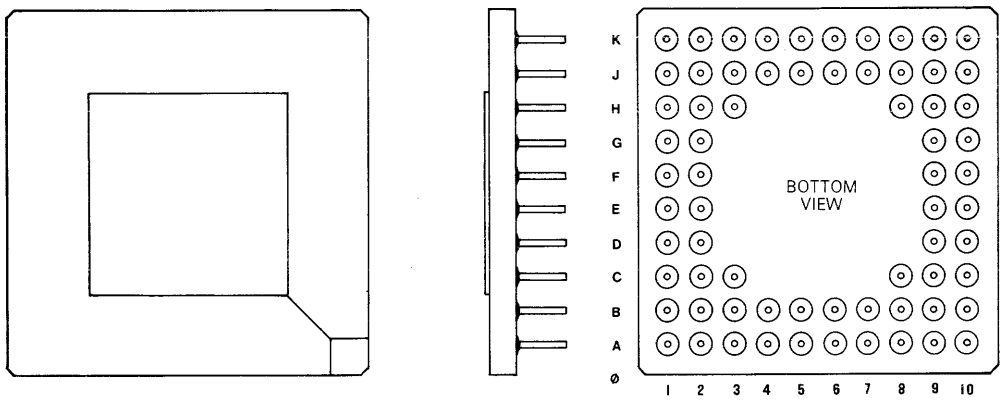
64-Pin Dual-in-line Package



68-Terminal Chip Carrier



68-Terminal Pin Grid Array

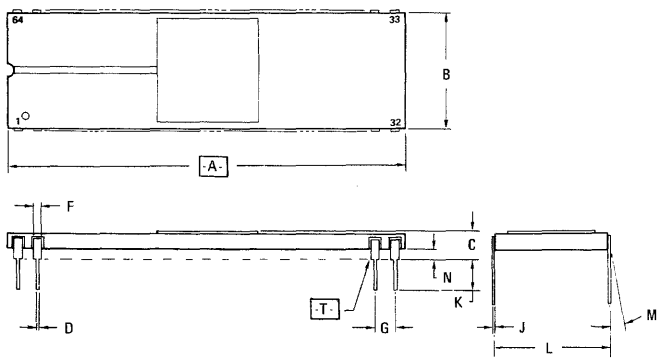


Pin Number	Function
A1	CLOCK
A2	ALL
A3	$\overline{GO}$
A4	$\overline{LDS}$
A5	$\overline{ED}$
A6	$\overline{DTACK}$
A7	$\overline{AS}$
A8	FC0
A9	FC2
A10	GND
B1	R/$\overline{W}$
B2	$\overline{CS}$
B3	$\overline{IACK}$
B4	$\overline{ANY}$
B5	$\overline{UDS}$
B6	$\overline{RESET}$
B7	FC1
B8	FC3
B9	$\overline{IRQ}$
B10	$\overline{FAULT}$
C1	RS2
C2	RS1
C3	V_{CC}
C8	GND
C9	MODE
C10	$\overline{HAD}$
D1	RS3
D2	RS4
D9	$\overline{WIN}$
D10	$\overline{MAS}$
E1	RS5
E2	A8
E9	PAD0
E10	PAD1

Pin Number	Function
F1	A9
F2	A10
F9	PAD3
F10	PAD2
G1	A11
G2	A13
G9	PAD6
G10	PAD4
H1	A12
H2	A15
H3	GND
H8	V_{CC}
H9	V_{CC}
H10	PAD5
J1	A14
J2	GND
J3	A17
J4	A19
J5	A21
J6	PAD15
J7	PAD12
J8	PAD10
J9	PAD9
J10	PAD7
K1	A16
K2	Do Not Connect
K3	A18
K4	A20
K5	A22
K6	A23
K7	PAD14
K8	PAD13
K9	PAD11
K10	PAD8

8.3 PACKAGE DIMENSIONS

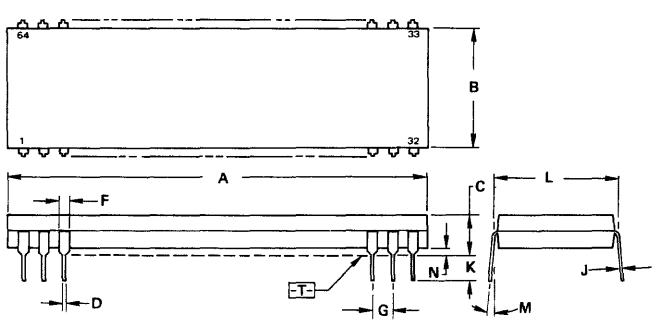
L SUFFIX
CERAMIC PACKAGE
CASE 746-01



NOTES:
1. DIMENSION -A- IS DATUM.
2. POSITIONAL TOLERANCE FOR LEADS: ⌖ 0.25 (0.010) Ⓜ T A Ⓜ
3. -T- IS SEATING PLANE.
4. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	80.52	82.04	3.170	3.230
B	22.25	22.96	0.876	0.904
C	3.05	4.32	0.120	0.170
D	0.38	0.53	0.015	0.021
F	0.76	1.40	0.030	0.055
G	2.54 BSC		0.100 BSC	
J	0.20	0.33	0.008	0.013
K	2.54	4.19	0.100	0.165
L	22.61	23.11	0.890	0.910
M	–	10°	–	10°
N	1.02	1.52	0.040	0.060

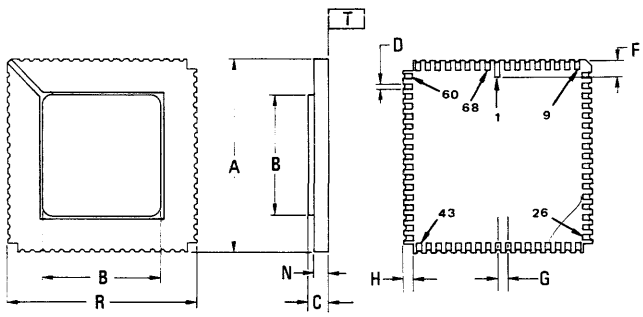
G SUFFIX
PLASTIC PACKAGE
CASE 754-01



NOTES:
1. DIMENSIONS A AND B ARE DATUMS.
2. -T- IS SEATING PLANE.
3. POSITIONAL TOLERANCE FOR LEADS (DIMENSION D): ⌖ Ø 0.25 (0.010) Ⓜ T A Ⓜ B Ⓜ
4. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSION B DOES NOT INCLUDE MOLD FLASH.
6. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	81.16	81.91	3.195	3.225
B	20.17	20.57	0.790	0.810
C	4.83	5.84	0.190	0.230
D	0.33	0.53	0.013	0.021
F	1.27	1.77	0.050	0.070
G	2.54 BSC		0.100 BSC	
J	0.20	0.38	0.008	0.015
K	3.05	3.55	0.120	0.140
L	22.86 BSC		0.900 BSC	
M	0°	15°	0°	15°
N	0.51	1.01	0.020	0.040

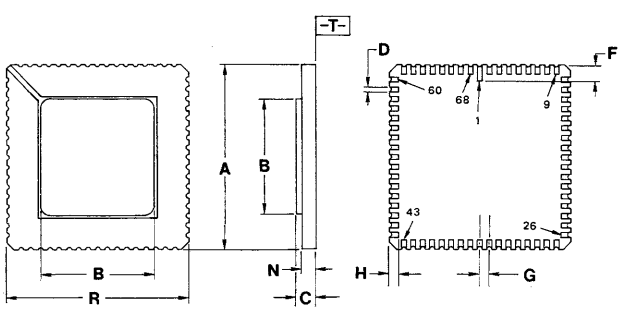
ZB SUFFIX
TYPE B LEADLESS
CHIP CARRIER
CASE 760A-01



NOTES:
1. DIMENSION A IS DATUM (2 PLACES).
2. -T- IS GAUGE PLANE.
3. POSITIONAL TOLERANCE FOR TERMINALS(D): 68 PLACES
   ⌖ 0.25 (0.010) Ⓜ T A Ⓢ R Ⓢ
4. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.
5. DIMENSION H PROVIDES THE SIZE FOR BOTH THE PAD LENGTH AND THE THREE CORNER NOTCHES.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	23.83	24.43	0.938	0.962
B	15.24	15.49	0.600	0.610
C	1.73	3.05	0.068	0.120
D	0.84	0.99	0.033	0.039
F	1.90	2.41	0.075	0.095
G	1.27 BSC		0.050 BSC	
H	1.02	1.52	0.040	0.060
N	1.14	2.24	0.045	0.088
R	23.83	24.43	0.938	0.962

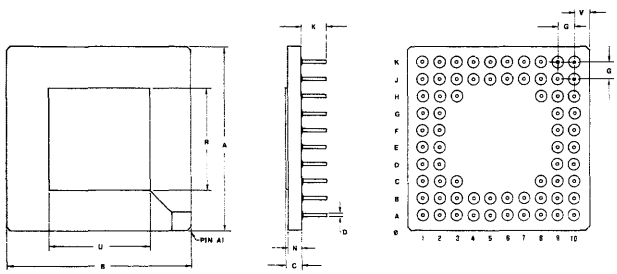
ZC SUFFIX
TYPE C LEADLESS
CHIP CARRIER
CASE 760-01



NOTES:
1. DIMENSION A IS DATUM (2 PLACES).
2. -T- IS GAUGE PLANE.
3. POSITIONAL TOLERANCE FOR TERMINALS(D): 68 PLACES
   ⌖ 0.25 (0.010) Ⓜ T A Ⓢ R Ⓢ
4. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	23.83	24.43	0.938	0.962
B	15.24	15.49	0.600	0.610
C	2.03	3.05	0.080	0.120
D	0.56	0.71	0.022	0.028
F	1.90	2.41	0.075	0.095
G	1.27 BSC		0.050 BSC	
H	1.02	1.52	0.040	0.060
N	1.78	2.29	0.070	0.090
R	23.83	24.43	0.938	0.962

R SUFFIX
PIN GRID ARRAY
CASE 765A-01



NOTES:
1. POSITIONAL TOLERANCE FO ! LEADS (68 PLACES):
   ⌖ Ø 0.13 (0.005) Ⓜ
2. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	26.41	27.43	1.040	1.080
B	26.41	27.43	1.040	1.080
C	2.10	2.59	0.083	0.102
D	0.51	0.60	0.020	0.024
G	2.54 BSC		0.100 BSC	
K	3.56	4.14	0.140	0.163
N	1.83	2.23	0.072	0.088
R	15.54	15.95	0.612	0.628
U	15.54	15.95	0.612	0.628
V	1.79	2.28	0.070	0.090

MOTOROLA *Semiconductor Products Inc.*

3501 ED BLUESTEIN BLVD., AUSTIN, TEXAS 78721 • A SUBSIDIARY OF MOTOROLA INC.

AI3277-11 PRINTED IN USA 6/90 IMPERIAL LITHO 072780 2,000 MPC YCABAA