MOTOROLA

MC68451L4
MC68451L6
MC68451L8

Advance Information

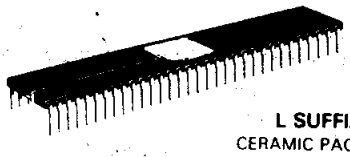
HMOS
(HIGH-DENSITY N-CHANNEL, SILICON-GATE)

MEMORY MANAGEMENT UNIT

MEMORY MANAGEMENT UNIT (MMU)

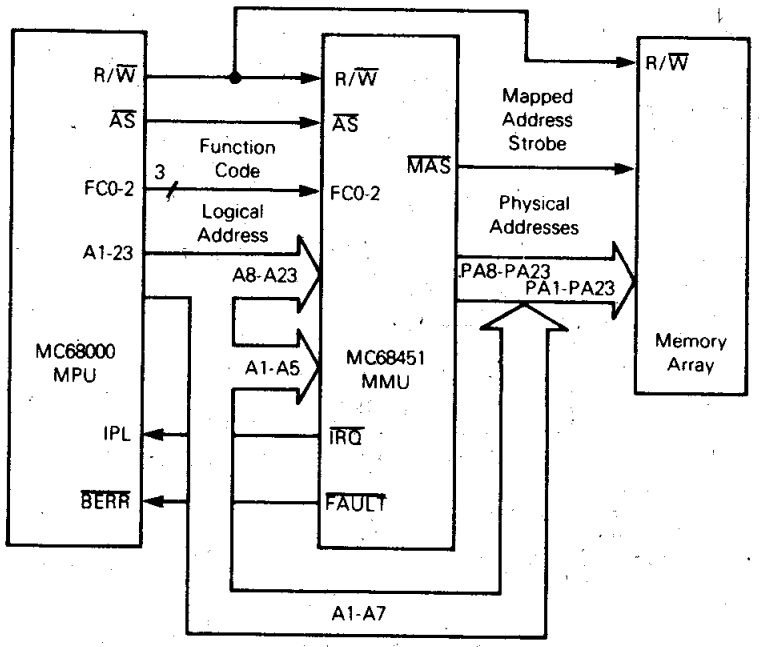
The MC68451 Memory Management Unit (MMU) provides address translation and protection for the 16 megabyte addressing range of the MC68000 MPU. Each bus master (or processor) in the M68000 Family provides a function code and an address during each bus cycle. The function code specifies an address space 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 efficient DMA. A multi-tasking operating system is simplified and reliability is enhanced through the use of the MMU.

- MC68000 Bus Compatible
- 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

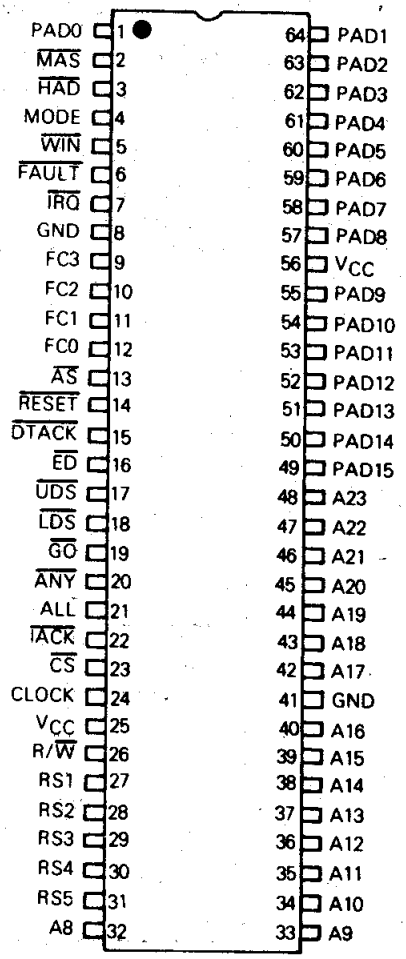


L SUFFIX
CERAMIC PACKAGE
CASE 746

FIGURE 1 — SIMPLIFIED BLOCK DIAGRAM OF SINGLE-MMU SYSTEM



PIN ASSIGNMENT



ADI-872

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

THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Rating
Thermal Resistance Ceramic	θ_{JA}	50	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precuations 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}).

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

$P_{I/O}$ = I/O Power Dissipation on Input and Output Pins, Watts — User Determined

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

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

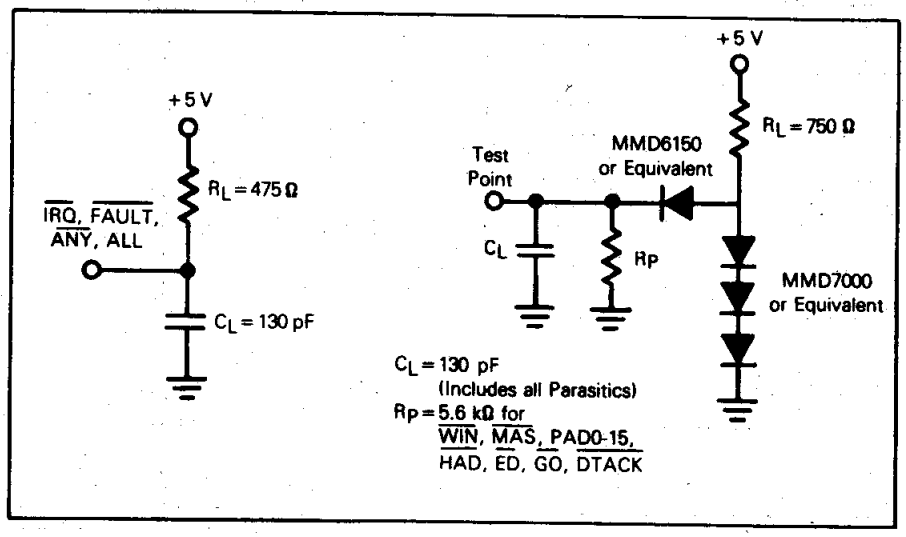
$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

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

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 2 — TEST LOADS



ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc; V_{SS} = 0 Vdc; T_A = 0°C to 70°C)

Characteristics		Symbol	Min	Max	Unit
Input High Voltage	All Inputs Except MODE	V_{IH}	V_{SS} + 2.0	V_{CC}	V
	MODE		V_{CC}	–	
Input Low Voltage	All Inputs Except MODE	V_{IL}	V_{SS} − 0.3	V_{SS} + 0.8	V
	MODE		–	V_{SS}	
Input Leakage Current		I_{in}	–	10	μA
Three-State (Off-State) Input Current (V_{CC} = Max) PAD0-15, $\overline{DTACK}$, $\overline{MAS}$, $\overline{WIN}$, $\overline{ED}$, $\overline{GO}$		I_{TSI}	–	10	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_{OL}			V
(I_{OL} = 5.3 mA)	PAD0-15, $\overline{DTACK}$, $\overline{MAS}$, $\overline{WIN}$, $\overline{HAD}$, $\overline{ED}$, $\overline{GO}$		V_{SS} + 0.4	–	
(I_{OL} = 10.7 mA)	$\overline{FAULT}$, $\overline{IRQ}$, $\overline{ANY}$, ALL		V_{SS} + 0.4	–	
Power Dissipation (Clock Frequency = 8 MHz)		P_D	–	1.0	W
Capacitance (Package Type Dependent) (V_{in} = 0 V, T_A = 25°C, Frequency = 1 MHz)		C_{in}	–	20	pF

AC ELECTRICAL SPECIFICATIONS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0 Vdc, T_A = 0°C to 70°C, See Figures 3, 4, and 5)

Number	Characteristic	4 MHz MC68451L4		6 MHz MC68451L6		8 MHz MC68451L8		Unit
		Min	Max	Min	Max	Min	Max	
1	Clock Period	250	500	167	500	125	500	ns
2	Clock Width Low	115	250	75	250	55	250	ns
3	Clock Width High	115	250	75	250	55	250	ns
4	Clock Fall Time	–	10	–	10	–	10	ns
5	Clock Rise Time	–	10	–	10	–	10	ns
6	$\overline{AS}$ Width Asserted	500	–	300	–	200	–	ns
7	FC0-FC3 Valid to $\overline{AS}$ Asserted (FC Setup Time)	60	–	50	–	40	–	ns
8	A8-A23, FC0-FC3 Valid After $\overline{AS}$ Negated (Address, FC Hold Time)	55	–	35	–	25	–	ns
9	A8-A23 Valid to $\overline{AS}$ Asserted (Address Setup)	45	–	30	–	25	–	ns
10	$\overline{AS}$ Asserted to PAD0-PAD15 Valid	–	300	–	250	–	187	ns
11	$\overline{AS}$ Negated to $\overline{MAS}$ Negated	–	125	–	125	–	125	ns
12	$\overline{AS}$ Asserted to $\overline{MAS}$ Asserted (Asynchronous Mode)	–	275	–	225	–	187	ns
13	PAD0-PAD15 Valid to $\overline{MAS}$ Asserted (Mode S1 or S2)	55	–	35	–	30	–	ns
14	Clock High to $\overline{MAS}$ Asserted (Mode S1)	–	80	–	80	–	80	ns
15	Clock Low to $\overline{MAS}$ Asserted (Mode S2)	–	80	–	80	–	80	ns
16	Clock High to $\overline{HAD}$ Asserted	–	100	–	100	–	100	ns
17	Clock High to $\overline{HAD}$ Negated	–	100	–	100	–	100	ns
18	PAD0-PAD15 Valid to $\overline{HAD}$ Asserted	40	–	40	–	40	–	ns
19	PAD0-PAD15 Valid after $\overline{HAD}$ Asserted	40	–	40	–	40	–	ns
20	$\overline{AS}$ Negated to $\overline{HAD}$ Negated	125	–	85	–	60	–	ns
21	$\overline{AS}$ Asserted to R/$\overline{W}$ Valid	20	–	20	–	20	–	ns

(Continued)

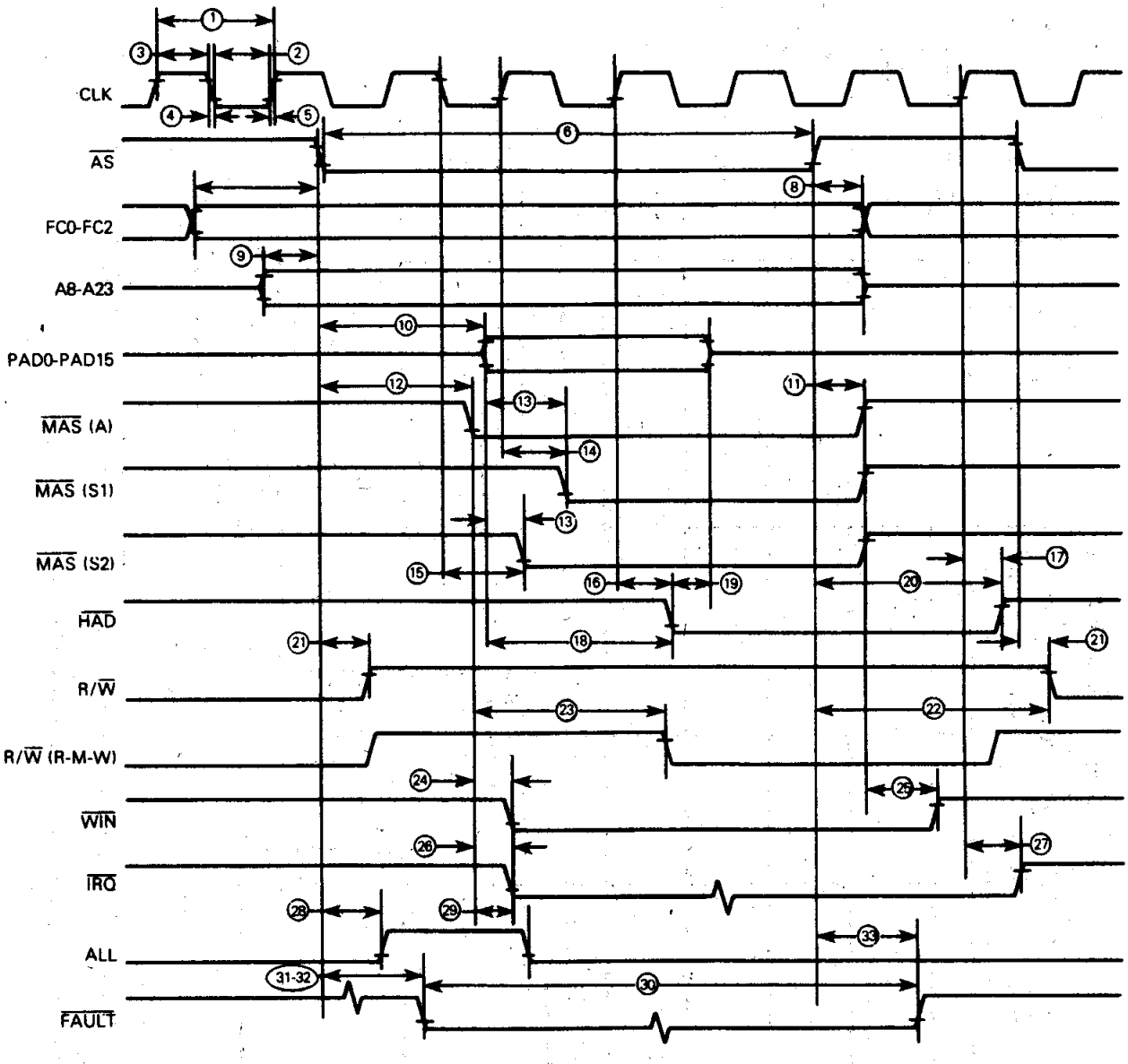
AC ELECTRICAL SPECIFICATIONS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0 Vdc, T_A = 0°C to 70°C, See Figures 3, 4, and 5)

Number	Characteristic	4 MHz MC68451L4		6 MHz MC68451L6		8 MHz MC68451L8		Unit
		Min	Max	Min	Max	Min	Max	
22	R/$\overline{W}$ Valid After $\overline{AS}$ Negated	20	–	20	–	20	–	ns
23	$\overline{MAS}$ Asserted to R/$\overline{W}$ Transition (Read-Modify-Write)	0	–	0	–	0	–	ns
24	$\overline{MAS}$ Asserted to $\overline{WIN}$ Asserted	–	40	–	30	–	20	ns
25	$\overline{MAS}$ Negated to $\overline{WIN}$ Negated	0	40	0	40	0	40	ns
26	$\overline{MAS}$ Asserted to $\overline{IRQ}$ Asserted	–	60	–	40	–	20	ns
27	Clock High to $\overline{IRQ}$ Negated	–	150	–	130	–	120	ns
28	$\overline{AS}$ Asserted to ALL High-Impedance (No Match)	–	250	–	225	–	200	ns
29	$\overline{MAS}$ Asserted to ALL Negated (External Match)	–	120	–	100	–	80	ns
30	$\overline{FAULT}$ Width Asserted	5	–	5	–	5	–	Clk Per
31	$\overline{AS}$ Asserted to $\overline{FAULT}$ Asserted (USA)*	6.5	9.5	6.5	10	6.5	10.5	Clk Per
32	$\overline{AS}$ Asserted to $\overline{FAULT}$ Asserted (WV)*	12.5	15.5	12.5	16	12.5	16.5	Clk Per
33	$\overline{AS}$ Negated to $\overline{FAULT}$ High-Impedance ($\overline{AS}$ Slower than Min Width)	–	100	–	75	–	60	ns
34	$\overline{CS}$ to $\overline{UDS}$, $\overline{LDS}$ Nonoverlap (to Avoid Operation)	80	–	70	–	60	–	ns
35	RS1-RS5, $\overline{CS}$, and $\overline{IACK}$ Valid to Clock Low	30	–	25	–	20	–	ns
36	RS1-RS5 Valid to $\overline{CS}$ Low	10	–	10	–	10	–	ns
37	Data Strobes Negated to RS1-RS5, $\overline{CS}$, and $\overline{IACK}$ Invalid	0	–	0	–	0	–	ns
38	$\overline{UDS}$, $\overline{LDS}$ Width Negated	275	–	210	–	140	–	ns
39	PAD0-PAD15 High-Impedance to $\overline{ED}$ Asserted	125	–	85	–	60	–	ns
40	$\overline{ED}$ Asserted to PAD0-PAD15 Valid (Write)	–	40	–	40	–	40	ns
41	$\overline{UDS}$, $\overline{LDS}$ Negated to $\overline{ED}$ Negated	–	125	–	110	–	100	ns
42	$\overline{ED}$ Negated to PAD0-PAD15 High-Impedance (Write)	–	60	–	60	–	60	ns
43	Clock Low to PAD0-PAD15 Valid (Read)	–	100	–	100	–	100	ns
44	PAD0-PAD15 Valid to $\overline{DTACK}$ Asserted (Read)	375	–	250	–	187	–	ns
45	$\overline{UDS}$, $\overline{LDS}$ Negated to PAD0-PAD15 High-Impedance (Read)	–	150	–	140	–	120	ns
46	Clock Low to $\overline{ANY}$ Asserted (Output)	–	250	–	167	–	125	ns
47	Clock Low to $\overline{ANY}$ High-Impedance (Output)	–	250	–	167	–	125	ns
48	Clock Low to ALL High-Impedance (Output)	–	250	–	167	–	125	ns
49	Clock Low to ALL Asserted (Output)	–	250	–	167	–	125	ns
50	Clock High to $\overline{DTACK}$ Asserted	–	150	–	110	–	75	ns
51	$\overline{UDS}$, $\overline{LDS}$ Negated to $\overline{DTACK}$ Negated	–	125	–	110	–	100	ns
52	$\overline{CS}$ Valid to $\overline{RESET}$ Negated ($\overline{CS}$ Setup Before $\overline{RESET}$)	5	–	5	–	5	–	Clk Per
53	$\overline{CS}$ Valid After $\overline{RESET}$ Negated ($\overline{CS}$ Hold Time After $\overline{RESET}$)	–10	60	–10	60	–10	60	ns
54	$\overline{RESET}$ Width Asserted (Note 1)	10	–	10	–	10	–	Clk Per
55	Type K Pins Logic High to High-Impedance (Note 2)	–	100	–	100	–	100	ns
56	Type D Inputs Fall Time (Note 3)	–	200	–	200	–	200	ns
57	Type D Inputs Rise Time	–	200	–	200	–	200	ns
58	Asynchronous Input Setup Time	30	–	30	–	30	–	ns

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

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 $\overline{GO}$.
3. Type D inputs are: $\overline{FAULT}$, $\overline{IRQ}$, $\overline{MAS}$, $\overline{GO}$, $\overline{ANY}$, and ALL.

FIGURE 3 — NORMAL TRANSLATION TIMING



NOTES:

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.

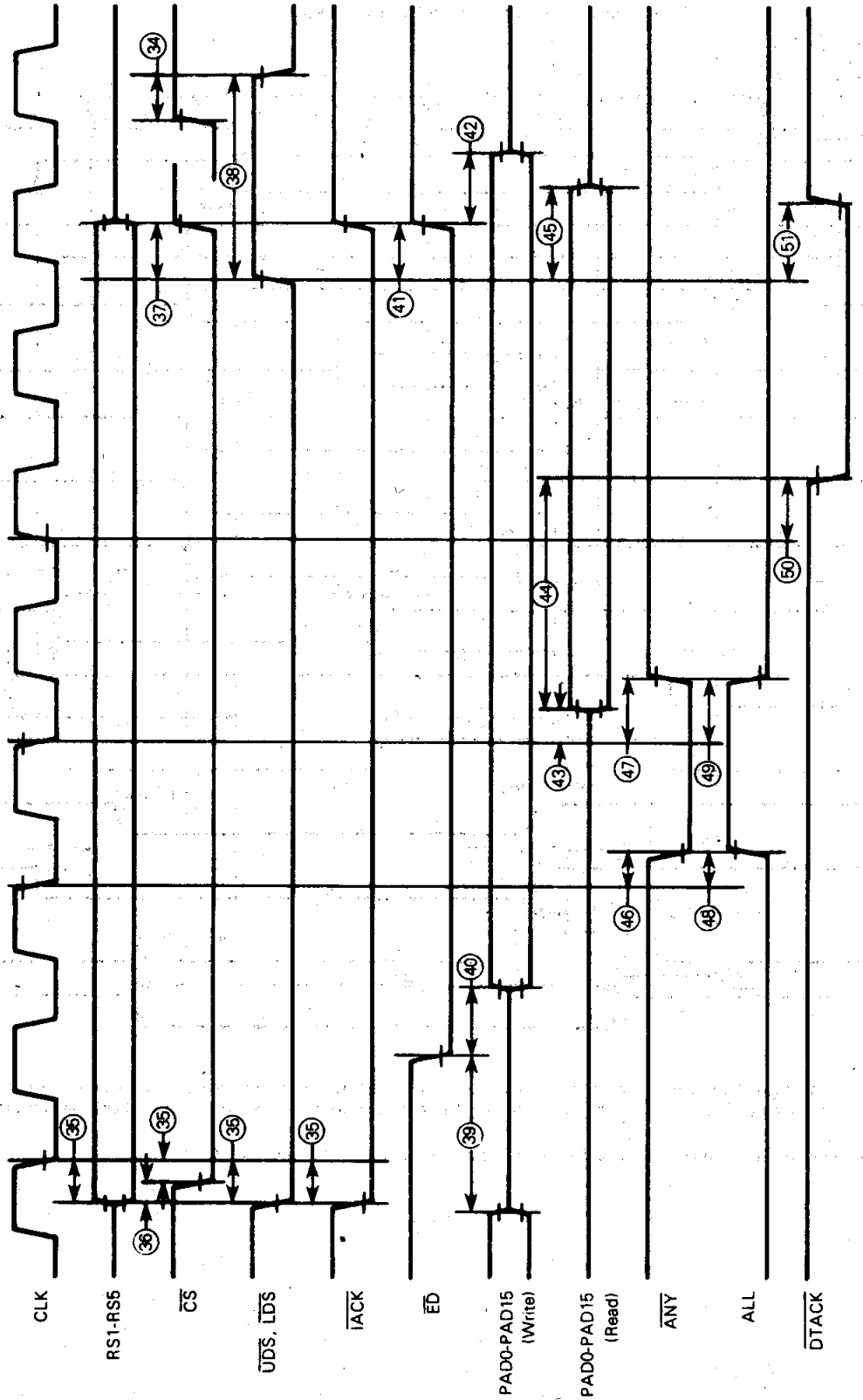
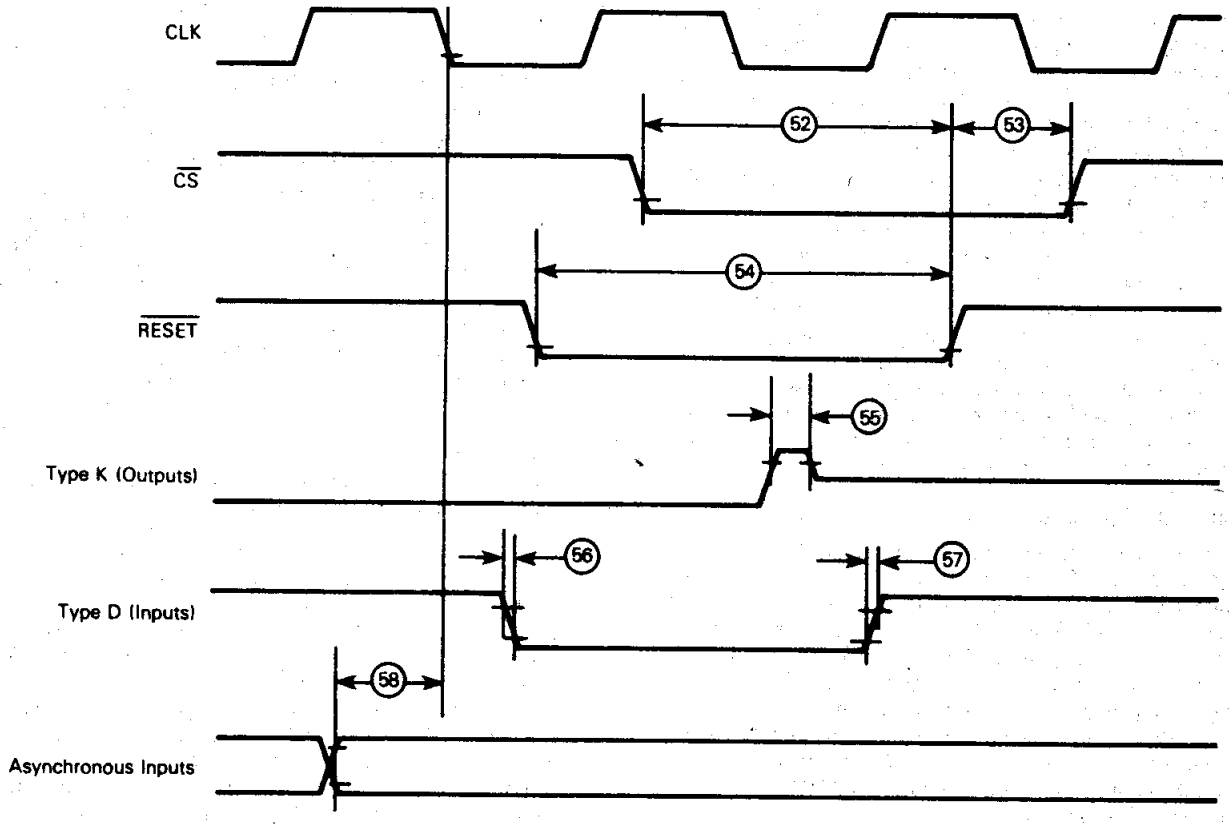


FIGURE 4 – OPERATIONS TIMING

NOTES:
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 5 — MISCELLANEOUS SIGNAL TIMING



NOTES:
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.

GENERAL DESCRIPTION

The MC68451 Memory Management Unit (MMU) is the basic element of a Memory Management Mechanism (MMM) in an MC68000-based 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.

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 multi-tasking.

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 $\overline{\text{FAULT}}$ (MC68000 Bus Error) is generated by the MMM if an undefined segment is accessed.

FUNCTION CODES AND ADDRESS SPACES

Each bus master in the M68000 family (including the MC68450 DMAC) 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 interrupt acknowledge bus cycles giving a total of five defined function codes.

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 tied to Bus Grant Acknowledge (BGACK) of the MC68000 to enable a second 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.

ADDRESS SPACE NUMBERS

To separate the address spaces of different tasks, each address space is given an identifying number. This should not be confused with the address space indicated by the function code. Each function code defines a unique address space and within each of these there can exist a number of different tasks. Each of these tasks need an Address Space Number (ASN) to distinguish it from the other tasks with which it may share an address space.

The Address Space Numbers are kept in the MMU in a set of registers called the Address Space Table (AST). The AST contains an 8-bit entry for each possible function code (16). Each entry can be assigned an ASN and, during a bus cycle, the function code is used to index into this table to select the Cycle Address Space Number. This number is then associatively compared with the Address Space Numbers in each Descriptor to attempt to find a match.

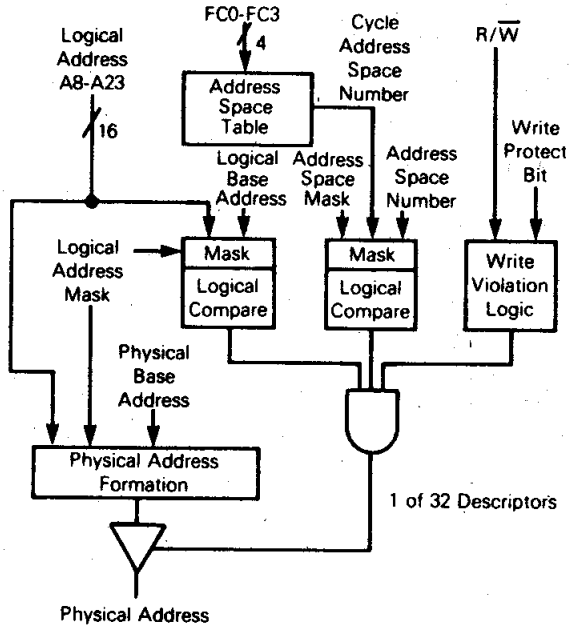
DESCRIPTORS

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 to segment boundaries to lie on a segment size boundary. That is, a segment can only start on an address which is multiple of 2^k. The segments can be defined in such a way to allow them to be logically or physically shared between tasks. A block diagram of the MC68451 MMU is shown in Figure 6.

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.

Refer to Figure 1 for the following. The logical address is composed of address lines A1-A23. The upper sixteen bits of this address (A8-A23) are translated by the MMU and mapped into a physical address (PA8-PA23). The lower 7 bits of the logical address (A1-A7) bypass the MMU and become the low-order physical address bits (PA1-PA7).

FIGURE 6 — MMU FUNCTIONAL BLOCK DIAGRAM



MMU PIN DESCRIPTION

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 driven high 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.

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

CLOCK — (TTL, input) — This signal must be the MC68000 system clock and must not be gated off at any time.

$\overline{CS}$ — (Chip-Select, input) — $\overline{CS}$ is 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 MMU OPERATIONS.

RS1-RS5 — (Register Selects, inputs) — These five 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 3 for the operations address map.

R/$\overline{W}$ — (Read/Write, input) — 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.

$\overline{RESET}$ — (Input) — 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 RESET STATE.

$\overline{\text{DTACK}}$ – (Data Transfer Acknowledge, output, rescindable) – The MMU uses this 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 $\overline{\text{DTACK}}$ to indicate that the information on the data bus is valid. If the bus cycle is a processor write to the MMU, $\overline{\text{DTACK}}$ is used to acknowledge acceptance of the data by the MMU. $\overline{\text{DTACK}}$ may be asserted only by an MMU that has $\overline{\text{CS}}$ or $\overline{\text{IACK}}$ asserted.

$\overline{\text{UDS}}$, $\overline{\text{LDS}}$ – (Upper Data Strobe, Lower Data Strobe, inputs) – $\overline{\text{UDS}}$ and $\overline{\text{LDS}}$ 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 the use of an odd address and 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.

$\overline{\text{AS}}$ – (Address Strobe, input) – This 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 $\overline{\text{AS}}$ initiates the normal translation phase of the bus cycle.

PAD0-PAD15 – (Physical Address and Data, multiplexed input/output, three-state) – during MMU operations, these 16 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 and the MMU provides the Enable Data ($\overline{\text{ED}}$) signal to control these transceivers. See the description of the $\overline{\text{ED}}$ signal for more detail.

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

NOTE

A pair of 74LS245 data transceivers may be used. $\overline{\text{ED}}$ will drive the Output Enable pin with no additional logic. See the circuit diagram in Figure 11.

$\overline{\text{HAD}}$ – (Hold Address, output, rescindable) – $\overline{\text{HAD}}$ is used to control an external latch on the physical address bus. After normal translation, $\overline{\text{HAD}}$ is asserted to hold the physical address stable. The latch should be of the transparent type such as a 74LS373. $\overline{\text{HAD}}$ can directly interface with the Enable pin of this type of latch. To provide address hold time, $\overline{\text{HAD}}$ is rescinded after $\overline{\text{MAS}}$ is rescinded.

MODE – (Input, three-level) – The MODE input is used to program the mode of operation of the $\overline{\text{MAS}}$ 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 MODE to V_{CC} and MODE S2 is selected by tying the MODE pin to ground. For a description of the different modes, see MAS TIMING MODES in the section on HARDWARE CONSIDERATIONS.

$\overline{\text{IRQ}}$ – (Interrupt Request, input/output, open drain) – $\overline{\text{IRQ}}$out is used to request an interrupt of the MPU. $\overline{\text{IRQ}}$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 $\overline{\text{IRQ}}$ to be negated. If $\overline{\text{IRQ}}$in and $\overline{\text{IACK}}$ and $\overline{\text{UDS}}$ or $\overline{\text{LDS}}$ are asserted, the MMU will perform the interrupt acknowledge operation. See MMU OPERATIONS. The $\overline{\text{IRQ}}$ lines of all MMUs should be wire-ORed together and tied to V_{CC} through a pullup resistor. They should be isolated from the $\overline{\text{IRQ}}$ lines of other devices to prevent an MMU from detecting an erroneous interrupt.

$\overline{\text{IACK}}$ – (Interrupt Acknowledge, input) – An MMU will begin the interrupt acknowledge operation if $\overline{\text{IRQ}}$in and $\overline{\text{IACK}}$ 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 $\overline{\text{IACK}}$ pin tied to the $\overline{\text{IACK}}$ circuitry from the processor; all other MMUs should have this pin tied high (inactive).

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

FC0-FC3 – (Function Code 0-3, inputs) – 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 FUNCTION CODES and ADDRESS SPACES. The MC68000 MPU and the MC68450 DMAC provide only three bits of the function code FC0-FC2. In a system with more than one bus master, the fourth pin could be tied to BGACK. If the system has only one bus master, FC3 should be tied low.

A8-A23 – (Inputs) – These are the upper sixteen bits of the MPU address bus. They form the logical address which the MMU translates into sixteen physical-address bits. The lower seven address lines bypass the MMU.

$\overline{\text{GO}}$ – (Global Operation, input/output, rescindable) – $\overline{\text{GO}}$ is an inter-MMU signal used in multiple-MMU systems to indicate or detect global operations. If $\overline{\text{CS}}$ is asserted and the operation to be performed is global, the MMU is selected as the master for that operation. $\overline{\text{GO}}$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 MMU OPERATIONS.

$\overline{ANY}$ – (Input/output, open drain) – The $\overline{ANY}$ signal is used in inter-MMU handshaking in 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}.

ALL – (Input/output, open drain) – The ALL pin is 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}.

$\overline{MAS}$ – (Mapped Address Strobe, input/output, rescindable) – $\overline{MAS}$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.

$\overline{WIN}$ – (Write Inhibit, output, rescindable) – $\overline{WIN}$ is 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 PHYSICAL DATA STROBES under HARDWARE CONSIDERATIONS. $\overline{WIN}$ 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.

MMU REGISTER DESCRIPTION

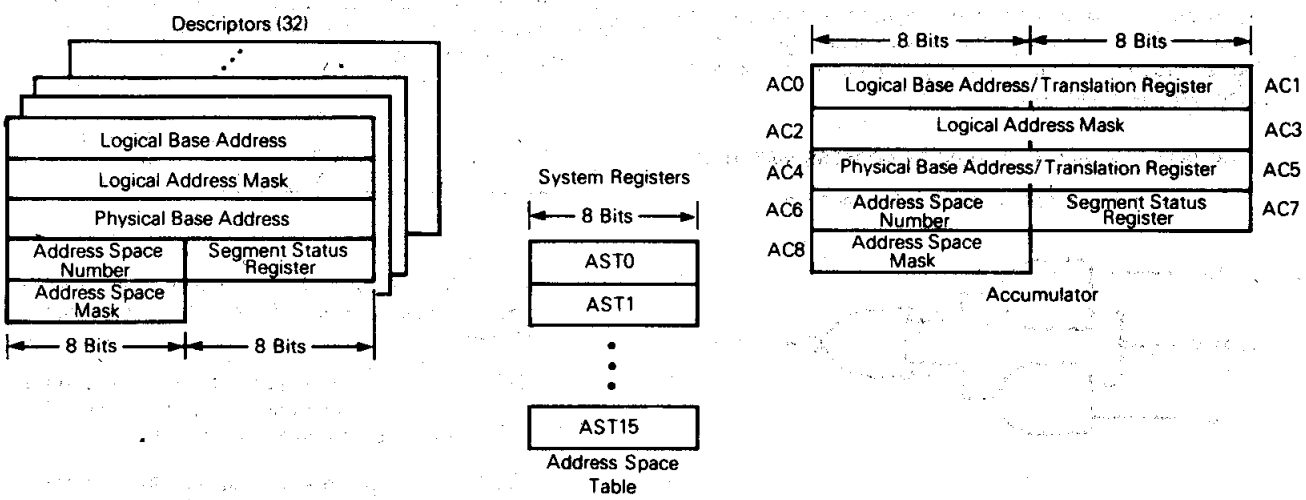
Figure 7 shows a programmer's model of the MMU. The MMU registers consist of two groups: the Descriptors and the System registers. Each of the 32 Descriptors is nine bytes long and defines one memory segment. See DESCRIPTORS below.

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.

FIGURE 7 – MMU PROGRAMMER'S MODEL



	7	6	5	4	3	2	1	0	Address
Descriptor Pointer				DP4	DP3	DP2	DP1	DP0	$29
Interrupt Vector Register				IV4	IV3	IV2	IV1	IV0	$2B
Global Status Register	F	DF						IE	$2D
Local Status Register	L7	L6	L5	L4	RW	GAT	GAL	LIP	$2F
Interrupt Descriptor Pointer	NVI			I4	I3	I2	I1	I0	$39
Result Descriptor Pointer	NVR			R4	R3	R2	R1	R0	$3B

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 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.

DESCRIPTORS

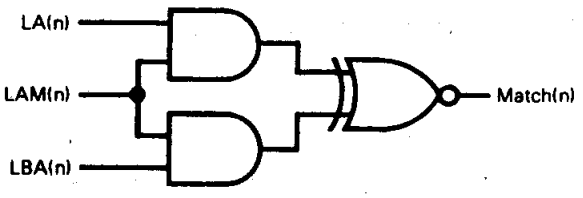
Each MMU contains 32 Descriptors (0-31), each of which can define one memory segment. A Descriptor is loaded by the MPU using the Accumulator and Discriptor Pointer with a Load Descriptor operation. The Segment Status Register (SSR) can be written to by the MPU indirectly using the Descriptor Pointer. Each Descriptor consists of the following registers.

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.

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 8.

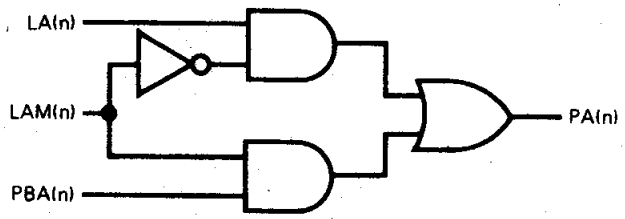
FIGURE 8 – SCHEMATIC LOGIC OF ADDRESS MATCHING

Note: LA(n) Indicates Bit N of Logical Address



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 9.

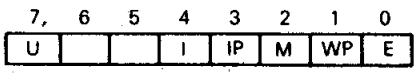
FIGURE 9 – SCHEMATIC LOGIC OF PHYSICAL ADDRESS GENERATION



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. See ADDRESS SPACE NUMBERS.

ADDRESS SPACE MASK (ASM) – The ASM is an 8-bit mask which defines the significant bit positions in the ASN 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 8.

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.



ADDRESS
Indirect through
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"

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

SET:	a) MPU writes "1"
CLEARED:	a) MPU writes "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 "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 with AC7, bit #0
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 clear

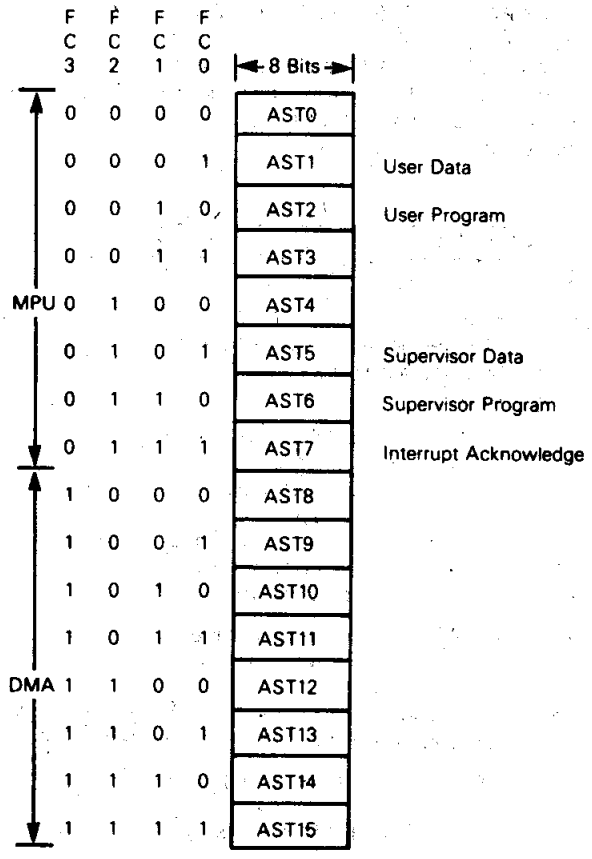
SYSTEM REGISTERS

The System Registers consist of the Address Space Table, Accumulator, 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.

ADDRESS SPACE TABLE (AST) — Each MMU has a local copy of the AST. This table is organized as sixteen 8-bit, read/write registers located 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 Segment Descriptors.

The MC68000 MPU and the MC68450 only provide a 3-bit function code, FC0-FC2. In a system with more than one bus master, the $\overline{BGACK}$ signal from the MPU could be inverted and used as FC3. This would result in the AST organization shown in Figure 10.

FIGURE 10 — ADDRESS SPACE TABLE ORGANIZATION



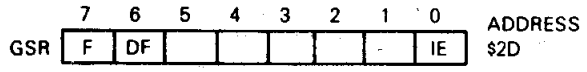
ACCUMULATOR (AC0-AC8) – The Accumulator (shown in Figure 7) is used to access the Descriptors, perform Direct Translation, and latch information during a Fault. The Accumulator consists of nine 8-bit registers located from $20 to $28. The register assignments for each operation in which it participates is shown in Table 1.

The contents of the Accumulator can be either local or global depending on the preceding operations. The GAL and GAT bits in the Local Status Register (LSR) indicate whether the information in the Accumulator is sufficiently global to perform a Load Descriptor or Direct Translation operation.

TABLE 1 – ACCUMULATOR ASSIGNMENTS FOR OPERATIONS

Register Assignment	Load/Read Descriptor	Direct Translation	Normal Translation (Fault)
AC0	Logical Base Address (MSB)	Logical Translation Register (MSB)	Logical Address (MSB)
AC1	Logical Base Address (LSB)	Logical Translation Register (LSB)	Logical Address (LSB)
AC2	Logical Address Mask (MSB)		
AC3	Logical Address Mask (LSB)		
AC4	Physical Base Address (MSB)	Physical Translation Register (MSB)	
AC5	Pysical Base Address (LSB)	Physical Translation Register (LSB)	
AC6	Address Space Number	Address Space Number	Cycle Address Space Number
AC7	Segment Status		
AC8	Address Space Mask		

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) Fault occurs and F was previously set
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"

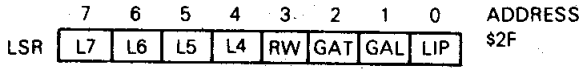
CLEARED:
a) Reset
b) MPU writes a "0"

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"

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. All bits in the LSR are 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 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 MMUs 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 this 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 this MMU. See RESULT DESCRIPTOR POINTER.

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)

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 matched Descriptor into the Accumulator. See GLOBAL OPERATIONS and DIRECT TRANSLATION.

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 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.

	7	6	5	4	3	2	1	0	
RDP	NVR			R4	R3	R2	R1	R0	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) A WV, LD failure of DT success in this MMU

R0-R4 R0-R4 encode the number of the Descriptor selected by the Priority Encoder.

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 INTERRUPT ACKNOWLEDGE.

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

	7	6	5	4	3	2	1	0	
IDP	NVI			I4	I3	I2	I1	I0	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.

INTERRUPT VECTOR REGISTER (IVR) – The IVR is an 8-bit read/write register containing the interrupt vector. Its contents are put 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.

MMU FUNCTIONAL DESCRIPTION

MULTIPLE MMU SYSTEMS

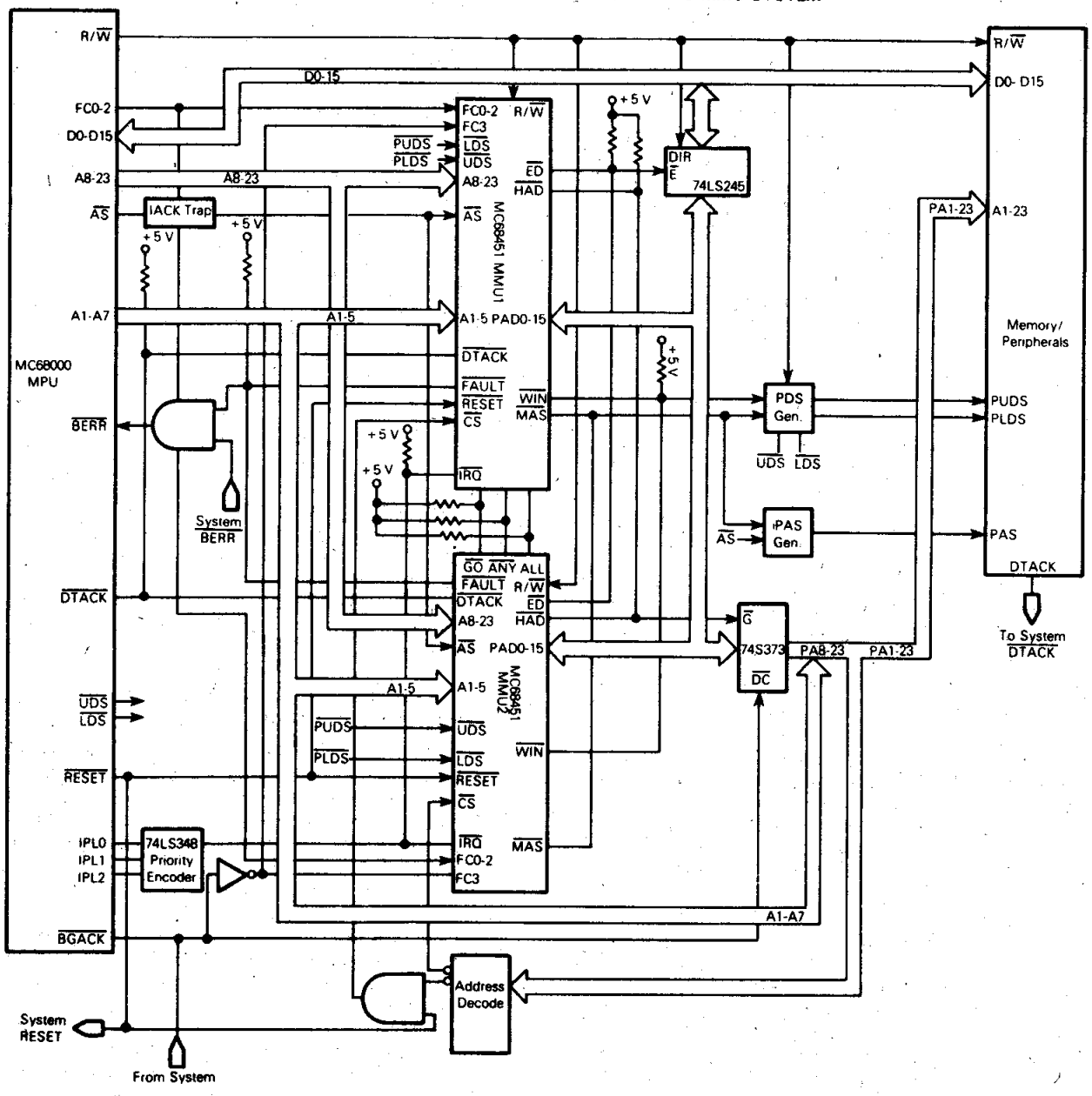
The Memory Management Mechanism is comprised of one or more Memory Management Units. Each MMU is capable of describing thirty-two segments. If more than thirty-two segments are required in the system, more MMUs can be added to increase the number in 32-segment increments.

In order to perform its operations, some of the information in the MMU's 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.

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 is shown in Figure 11.

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.

FIGURE 11 — SAMPLE CIRCUIT DIAGRAM OF A TWO-MMU SYSTEM



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, deterministic timing diagrams are not possible.

MMU FLOW DIAGRAM

In order to give a functional description of the MMU, the flow diagrams of the device are given in Figure 12. 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 2. See OPERATIONS LENGTH TABLE for more information.

NOTE

In Figure 12, all signals are shown as active high regardless of the voltage level of the true state. Logical negation is denoted by the prime symbol (').

MMU FUNCTIONAL STATES

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 LIP in the LSR and IE in the GSR are set; otherwise it is placed in the high-impedance state and should be negated with a pullup resistor.

THE RESET STATE – Asserting $\overline{RESET}$ will initiate the Reset sequence regardless of the state of the MMU. During Reset, $\overline{GO}$, $\overline{DTACK}$, $\overline{ED}$, $\overline{MAS}$, $\overline{HAD}$, and $\overline{WIN}$ are rescinded. The PAD port, $\overline{FAULT}$, and $\overline{ANY}$ are placed in the high-impedance state. Pullup resistors on $\overline{FAULT}$ and $\overline{ANY}$ 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 the diagram in Figure 11.

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 MMU is not chip 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 13.

NOTE

In Figure 13, all signals are shown as active high regardless of the voltage level of the true state. Logical negation is denoted by the prime symbol (').

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 tranceivers 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 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 MMU OPERATIONS.

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) do not. With these early mask sets, $\overline{AS}$ must be delayed to the MMU.

TABLE 2 — LENGTH OF OPERATIONS

Operation		Length (clock cycles)	
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)	17	19
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}, no GAL–{s})	35	43
	(match–{s}, no err–{m})	39	51
	(no GAL {m})	17	19
	(no GAL–{s}, no err–{m})	31	43
	(no GAL–{s}, no GAL–{m})	17	19
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

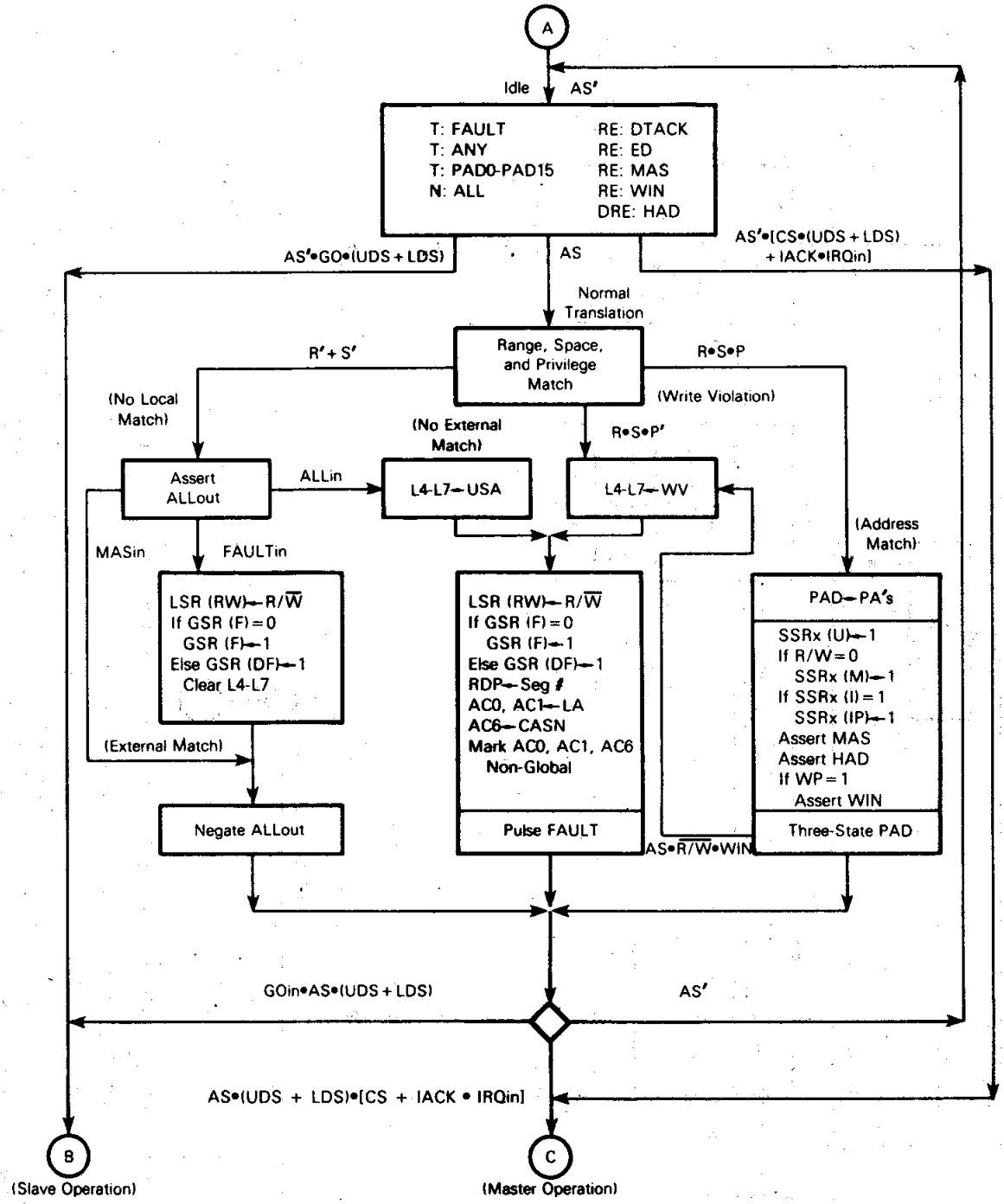
Notes:

1) Timings are given in system clock periods. The length of the operation 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 timings 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.

FIGURE 12 — MMU FLOW DIAGRAM (PAGE 1 OF 3)

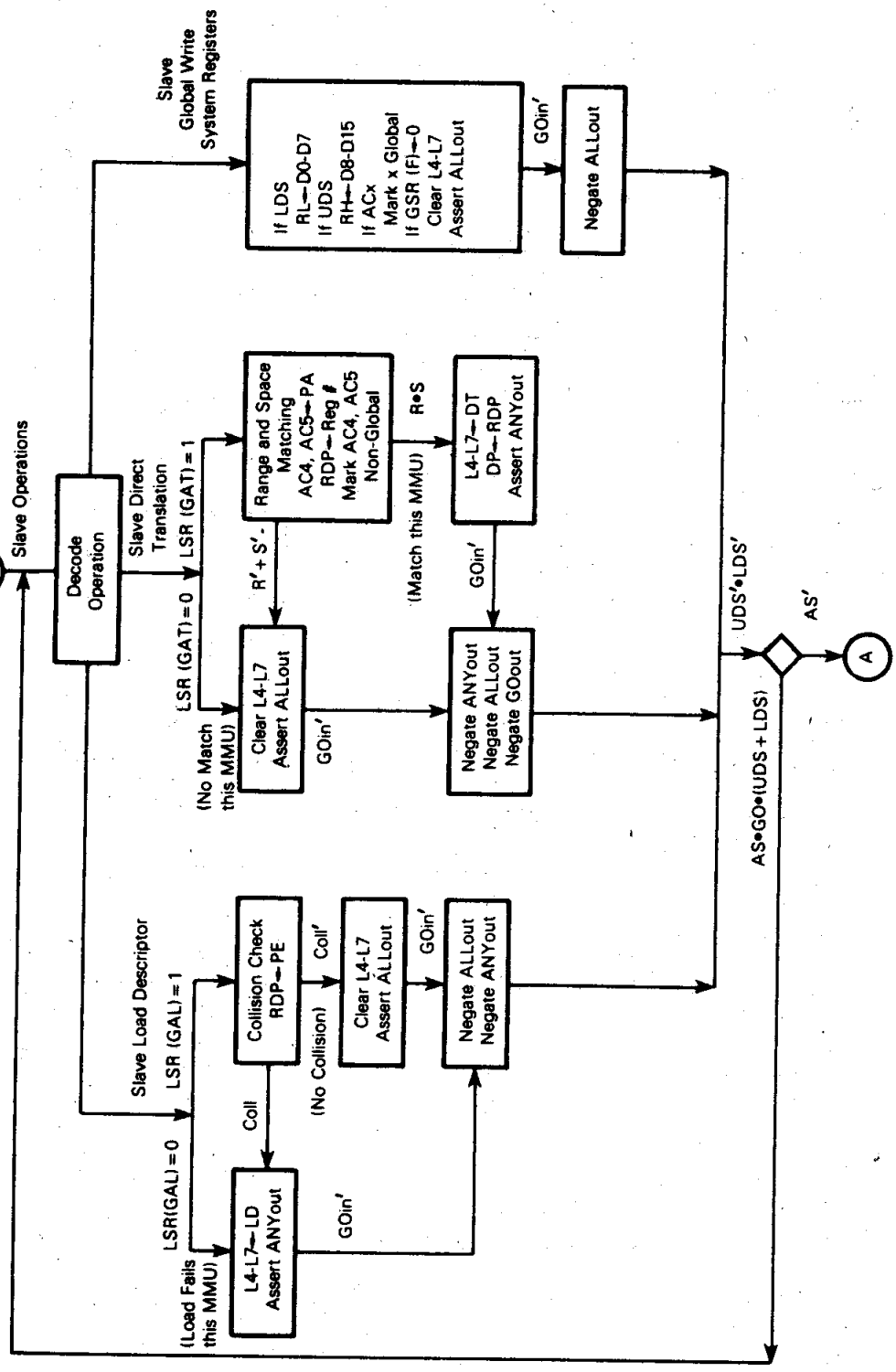


NOTES:
1. In this figure, all signals are shown as active high regardless of the voltage level of the true state. Logical negation is denoted by the prime symbol (').
2. Individual bits in registers are denoted as REG (Bit) e.g., LSR (GAL) represents the GAL bit in the LSR.

LEGEND:

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

FIGURE 12 — MMU FLOW DIAGRAM (PAGE 2 OF 3)

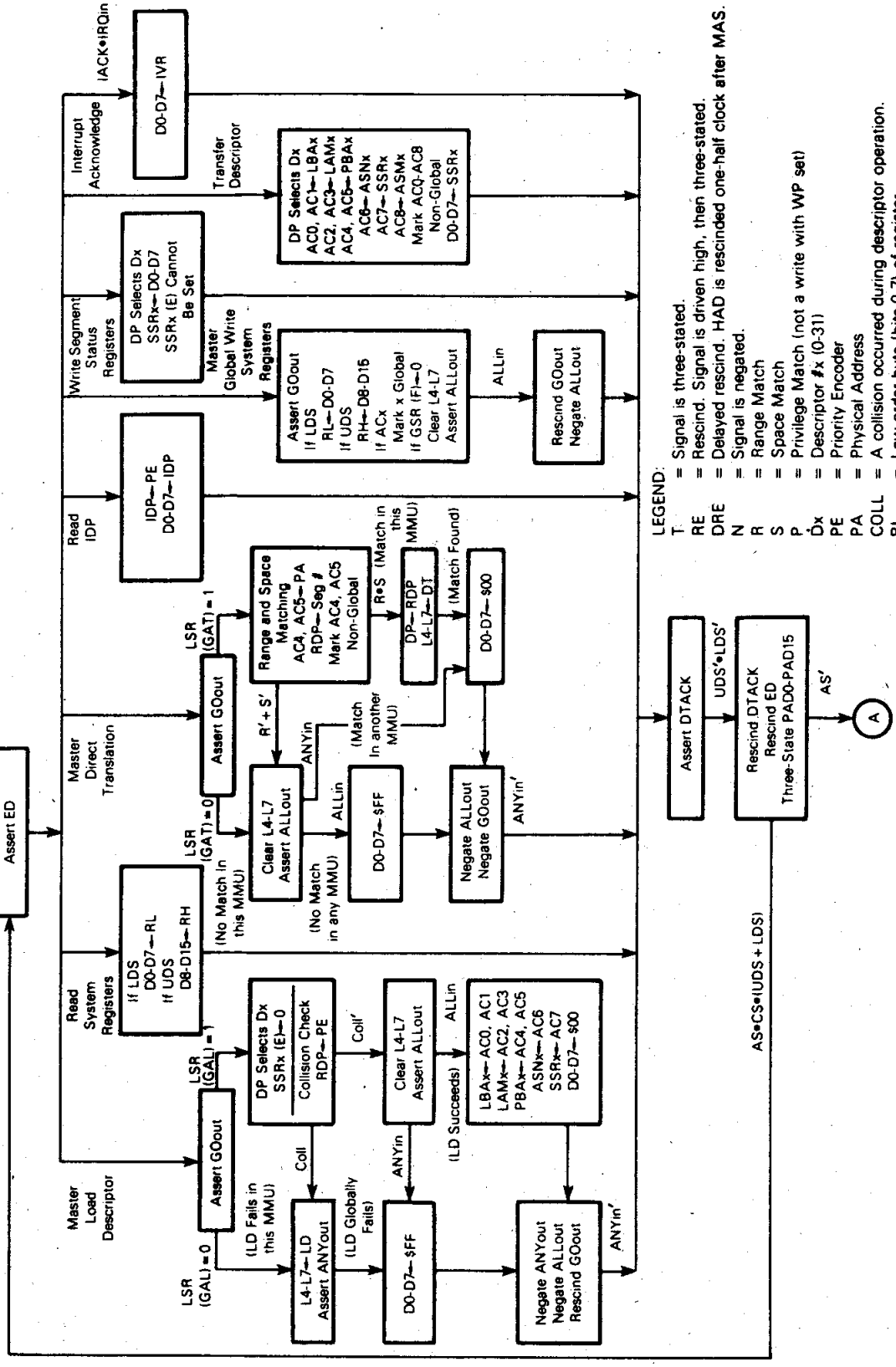


NOTES:
1. In this figure, all signals are shown as active high regardless of the voltage level of the true state. Logical negation is denoted by the prime symbol (').
2. Individual bits in registers are denoted as REG (Bit) e.g., LSR (GAL) represents the GAL bit in the LSR.

LEGEND:

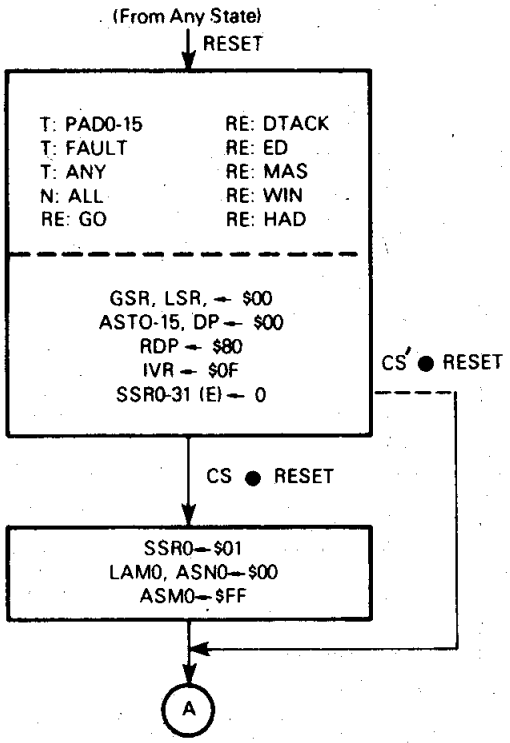
T	= Signal is three-stated.	R	= Range Match	PA	= Physical Address
RE	= Rescind. Signal is driven high, then three-stated.	S	= Space Match	COLL	= A collision occurred during descriptor operation.
DRE	= Delayed rescind. HAD is rescinded one-half clock after MAS.	P	= Privilege Match (not a write with WP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

FIGURE 12 — MMU FLOW DIAGRAM (PAGE 3 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.

FIGURE 13 — RESET FLOW DIAGRAM



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 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

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.

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. $\overline{\text{WIN}}$ 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, $\overline{\text{MAS}}$ 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 14. 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 GLOBAL OPERATIONS.
2. If $\overline{GO}$in is detected by an MMU it will begin a Slave operation. See 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. This would be the result of a read-modify-write bus cycle on a protected segment.

Write Violation – If an address match occurs but the bus cycle was a write to a 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 greater.

The logical address is latched into AC0 (MSB) and AC1 (LSB) of the Accumulator (see Table 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 15. The breaks in the diagram reflect the uncertainty due to the deskewing of the input signals from other MMUs.

No Address Match – If none of the Descriptors in a 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 External Translation.
2) The access caused a write violation in another MMU. See External Write Violation.
3) The access was to a globally undefined segment. See Undefined Segment Access.

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.

External Write Violation – If the $\overline{FAULT}$in 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/$\overline{W}$ is latched into the RW bit of the LSR, and L4-L7 are cleared to show that the violation did not take place in this MMU. The cycle can then continue in one of the three ways described above in External Translation.

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/$\overline{W}$ is latched into the RW bit of the LSR and L4-L7 are encoded to show a USA.

The logical address 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 $\overline{FAULT}$ line for five clock periods or until $\overline{AS}$ 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 $\overline{FAULT}$ pulse. When $\overline{AS}$ is negated, the MMM returns to the Idle state.

A functional timing diagram for an Undefined Segment Access is given in Figure 16. The breaks in the diagram reflect the uncertainty due to the deskewing of the input signals from other MMUs.

MMU OPERATIONS

Table 3 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 multi-processor system, an external processor could access the MMM from the physical address bus. If the access is from the logical address bus, an address translation is first performed. If the access is from the physical address bus, the operation state is entered directly from the Idle state.

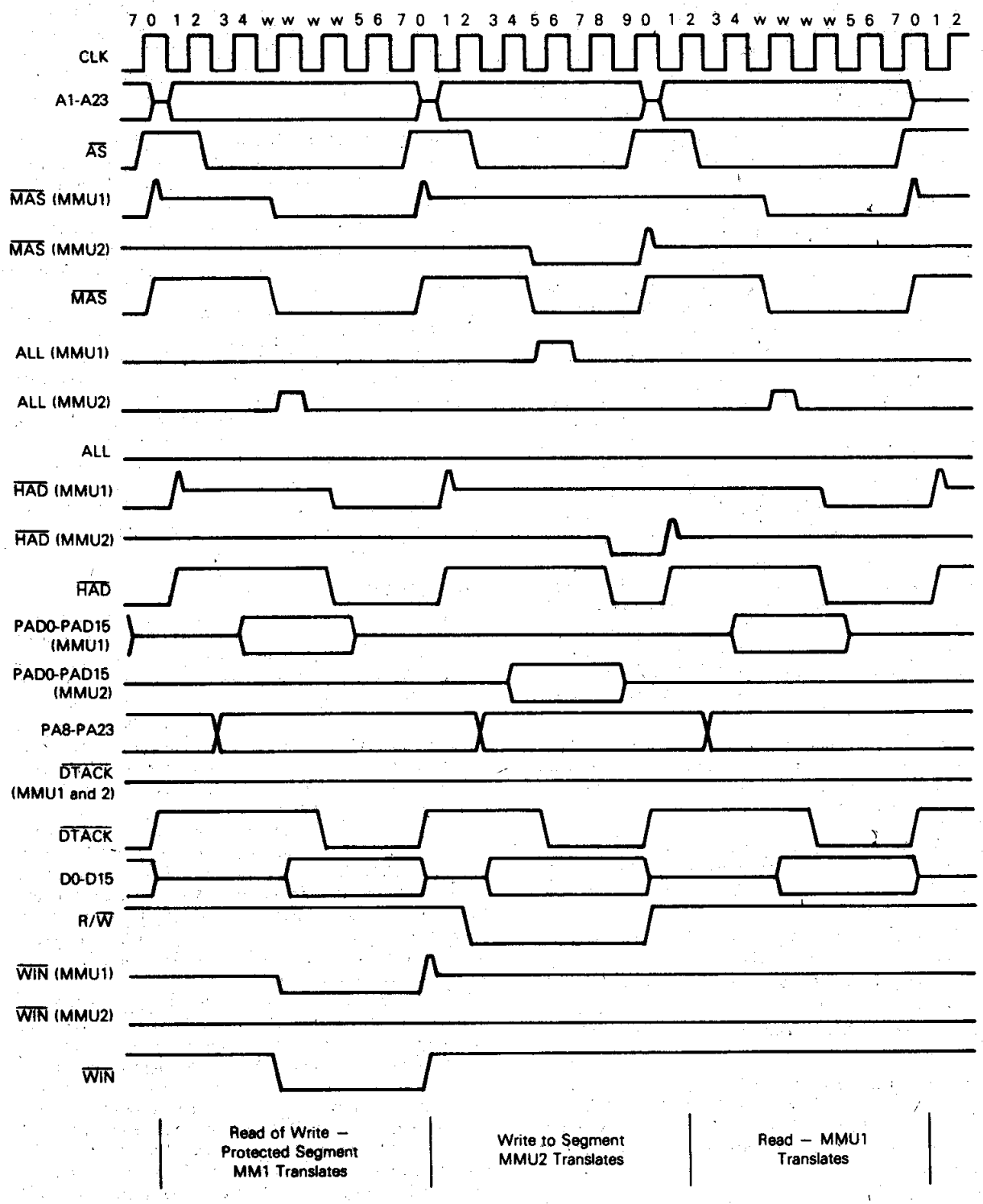
The operation phase is always entered with PAD0-PAD15 in the high-impedance state and either (in the case of an operation following a normal translation) one MMU asserting $\overline{HAD}$ to hold the physical address, or (in the case of an access from the physical bus) the external processor holding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

FIGURE 14 — NORMAL TRANSLATION TIMING IN A TWO-MMU SYSTEM

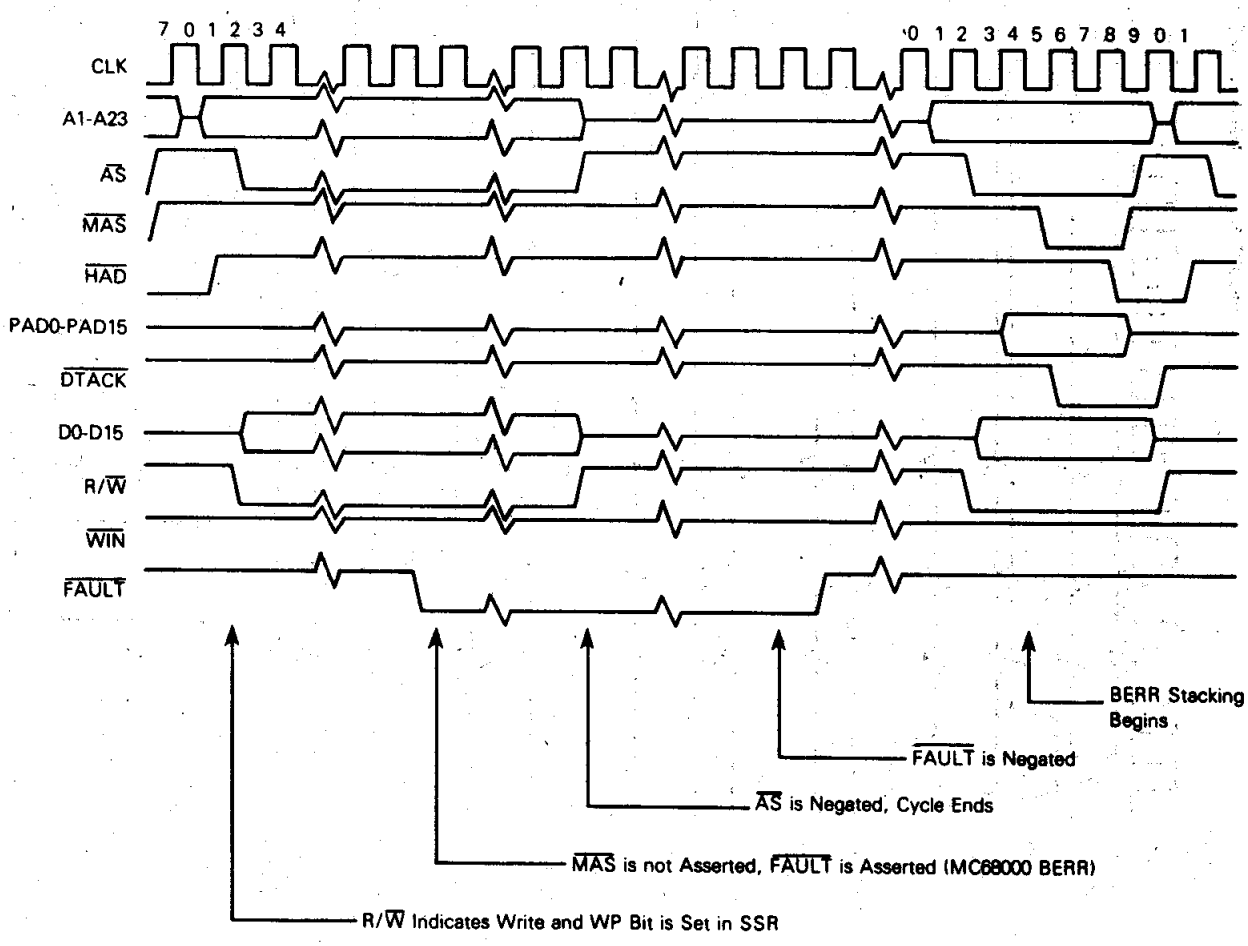


NOTE: Cycles denoted by "w" are MC68000 wait states.

TABLE 3 — 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 Sysem 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 selected 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.

FIGURE 15 — WRITE VIOLATION TIMING



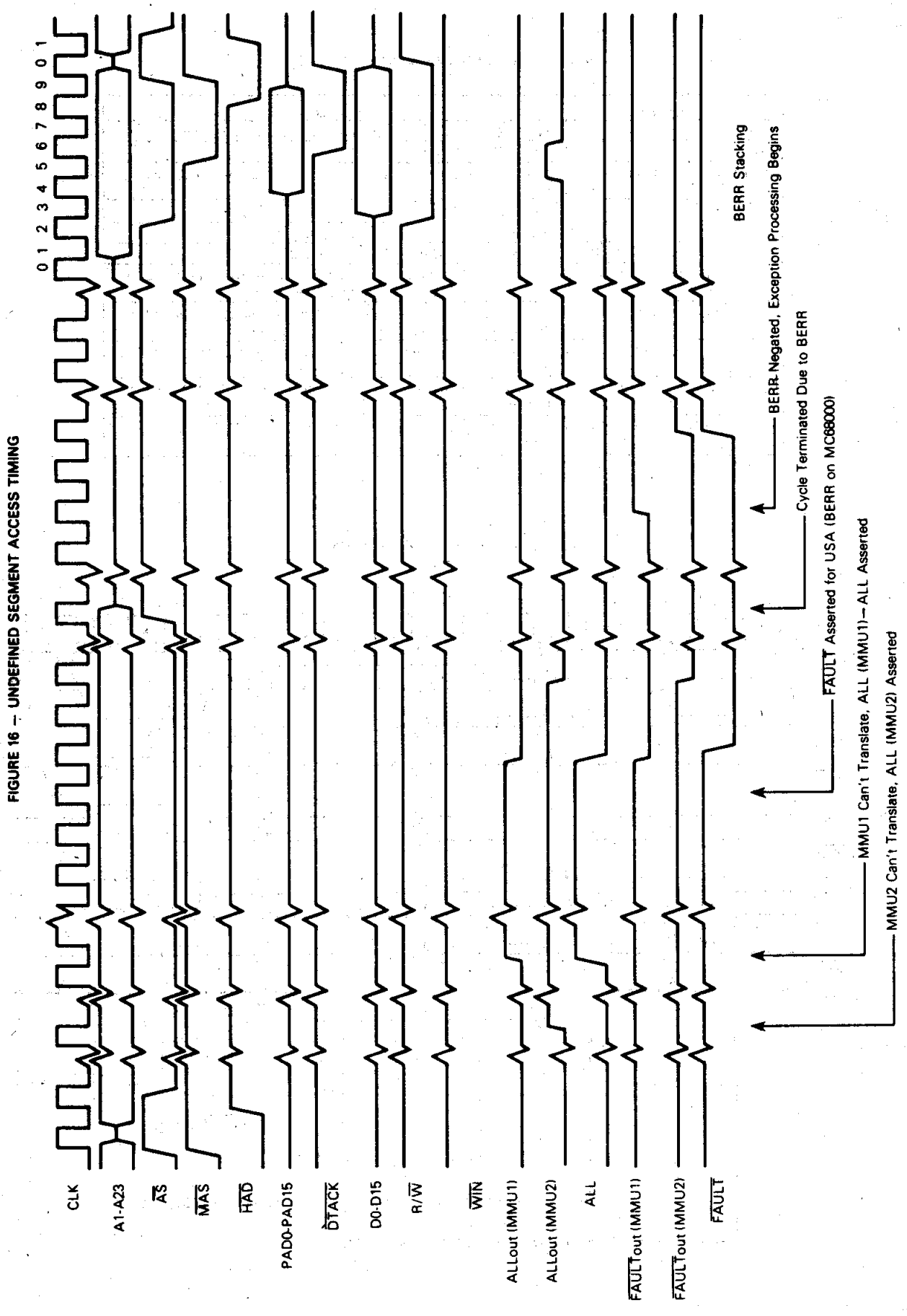


FIGURE 16 – UNDEFINED SEGMENT ACCESS TIMING

OPERATIONS ADDRESS MAP – Table 4 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.

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.

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.

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.

Transfer Descriptor – In order to read the contents of a Descriptor, it 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 is then transferred into the Accumulator as shown in Table 1 and the contents of the SSR are gated onto D0-D7. The Descriptor registers may then be read from the Accumulator.

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.

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. Global operations include: Global Write System Registers, Load Descriptor, and Direct Translation.

TABLE 4 – 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 (MSB))
100001	21	L	G	AC1 (LBA/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 → RS0=X
If $\overline{UDS}$=1 and $\overline{LDS}$=1 → RS0=X

If there is only one MMU present in the system, the $\overline{ANY}$, ALL, and $\overline{GO}$ pins must be tied to V_{CC} through pullup resistors. Global operations then become local only.

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 desired register's address (see Figure 17).

The MMU which has $\overline{CS}$ asserted becomes the Master by asserted $\overline{GO}$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 $\overline{GO}$.

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 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 the 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 GSR of a Slave is clear, LA4-LA7 is encoded to indicate LD and $\overline{ANY}$out is 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, its GAL bit was 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.

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.

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 clear, the MMU asserts ALLout and L4-L7 in the LSR are cleared. The Master monitors the $\overline{ANY}$in and ALLin inputs.

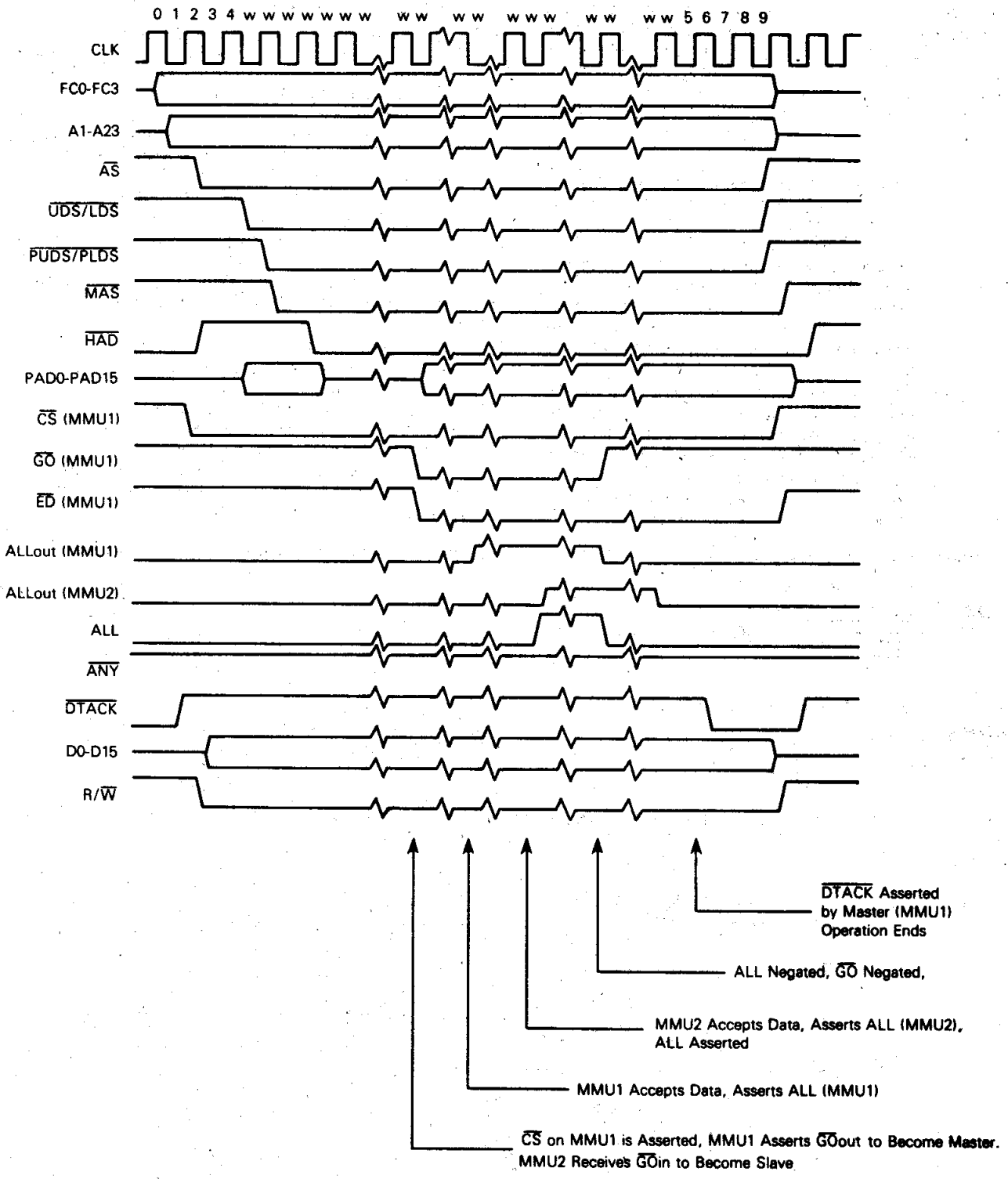
If $\overline{ANY}$in becomes asserted, then another MMU performs the translation. The Master puts $00 on D0-D7 to indicate success, negates ALLout and rescinds $\overline{GO}$out. It waits until $\overline{ANY}$in is negated before terminating the operation.

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 $\overline{GO}$ at the end of the operation.

OPERATION LENGTH TABLE

The length of each operation, in system clock cycles is given in Table 2. The table includes timings for both the single-MMU case and for the multiple MMU situation. In the case of more than one MMU, 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 3, 4, and 5 in the front of this document. These should be used for system hardware design considerations. The flow diagram (Figure 12) should be used for debugging of multiple MMU systems.

FIGURE 17 — GLOBAL WRITE SYSTEMS REGISTER TIMING



HARDWARE CONSIDERATIONS

$\overline{MAS}$ TIMING MODES AND PHYSICAL ADDRESS STROBES

The Mapped Address Strobe 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 (PAS) which is then used to gate the memory decode circuitry. $\overline{AS}$ is also included in the PAS to effect a quick release of the bus at the end of a cycle.

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

Mode A $\overline{MAS}$ is asserted asynchronously. $\overline{MAS}$ is asserted as soon as a match for the function code and logical address is found. A circuit to generate PAS with $\overline{MAS}$ in Mode A is shown in Figure 18a. The delay of $\overline{AS}$ should be the delay introduced by the MMU plus the desired setup time for the memory. This allows the user to fine-tune the bus speed for maximum efficiency. The delay can be implemented with delay lines or a clocked flip-flop.

Mode S1 $\overline{MAS}$ is asserted on the first rising edge of CLOCK after the physical address is valid. This mode was intended to allow the generation of PAS by ORing $\overline{AS}$ and $\overline{MAS}$. $\overline{MAS}$ asserts PAS and $\overline{AS}$ negates it to release the bus quickly at the end of the cycle. A circuit to generate PAS using $\overline{MAS}$ in mode S1 or S2 is shown in Figure 18b.

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 PAS using $\overline{AS}$ and $\overline{MAS}$ only. A circuit to do this is shown in Figure 18.

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 19. Three-state buffers are used to allow access from the physical bus by an external processor. If the $\overline{AS}$-defeat circuit described below is used, PLDS must be asserted to perform the Interrupt Acknowledge operation. The circuit shown, surrounded by the dotted line, provides for this.

INTERRUPTS

When the MC68000 responds to an interrupt, it places the Interrupt Acknowledge ($\overline{IACK}$) function code on FC0-FC2 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 match an address during an interrupt acknowledge cycle, $\overline{IACK}$ should be decoded from the function code lines and used to disable $\overline{AS}$ to the MMU. A circuit to accomplish this is shown in Figure 20.

FIGURE 18 — PHYSICAL ADDRESS STROBE GENERATION

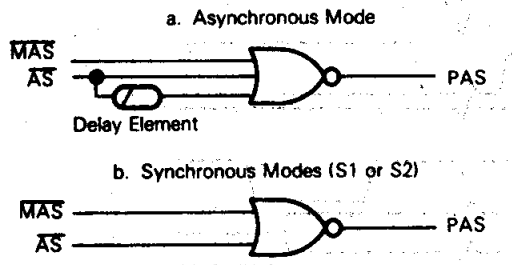


FIGURE 20 — CIRCUIT TO HOLD OFF $\overline{AS}$ DURING $\overline{IACK}$

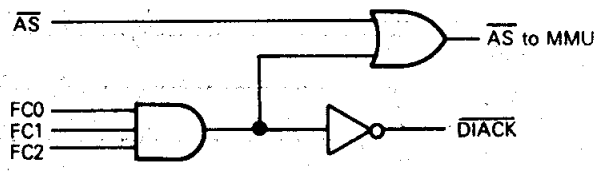
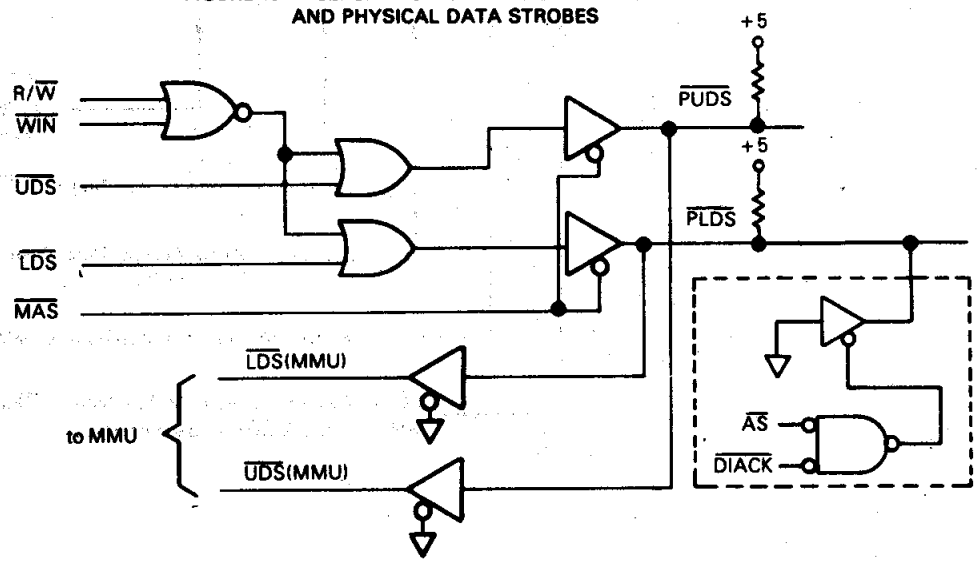


FIGURE 19 — GENERATION OF PHYSICAL READ/WRITE AND PHYSICAL DATA STROBES



NOTE: The circuit, surrounded by the dotted line, is necessary only if the $\overline{AS}$ defeat circuit in Figure 20 is used.

SOFTWARE CONSIDERATIONS

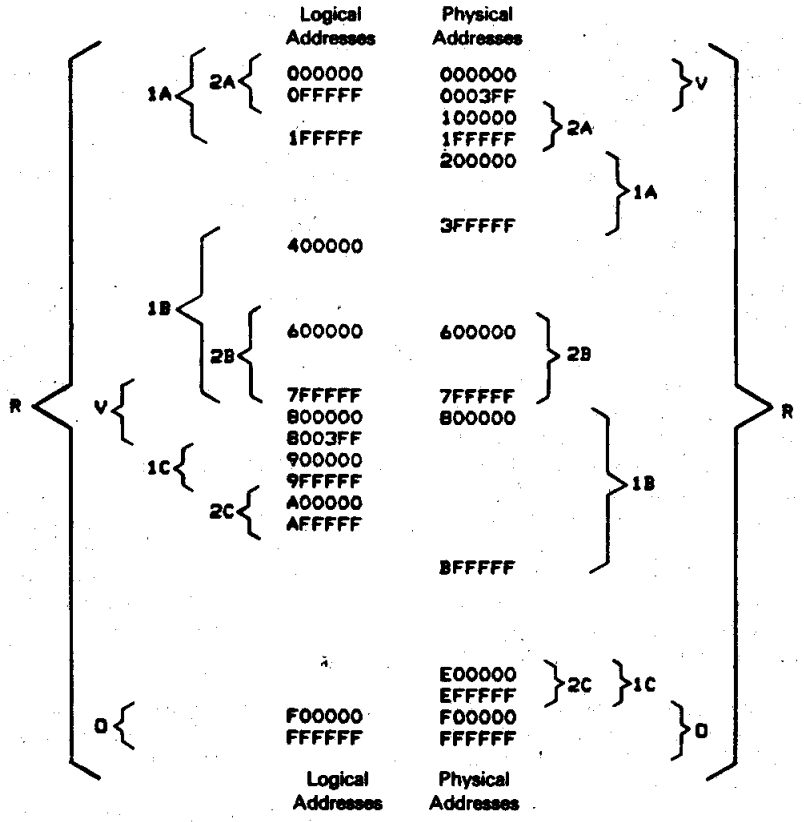
SEGMENT MAPPING EXAMPLE

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 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.

Figure 21 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.

FIGURE 21 — ADDRESS MAP EXAMPLE



Segment	R	V	O	1A	1B	2C	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)	FF	80	80	01	01	01	02	02	02
Address Space Mask (ASM)	FF	80	80	7F	7F	7F	7F	7F	7F

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 7 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 1 would be assigned ANS $01 and the operating system would use $81 to access those segments. A parallel situation exists with user 2.

Note that segments 1A and 2A are isolated from each other even though they share the same logical addresses. User 1 is prevented from accessing the same memory as user 2 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.

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

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.

PAGING

If each segment is the same size, a paged system could be implemented. The Used and Modified bits in the Segment Status Registers allow a variety of placement algorithms and the use of virtual memory.

The MMM supports virtual processing. The 16 bits of the logical address, the cycle address space number, and R/$\overline{W}$ are latched during a $\overline{FAULT}$ to provide enough information for an auxiliary processor to fix a page fault.

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 0 (see 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 system initialization routine. See the MC68000 Advance Information 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 LOAD DESCRIPTOR.

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 only be set 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 state bit cleared so that the processor will enter the user state and its accesses 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.

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 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 alerts the Status Register to put itself in the User state, AST5 and AST6 are changed, and the processor traps to the supervisor task.