

SEPTEMBER 1991

486SX / 486DX Single Chip AT
PRELIMINARY

FEATURES

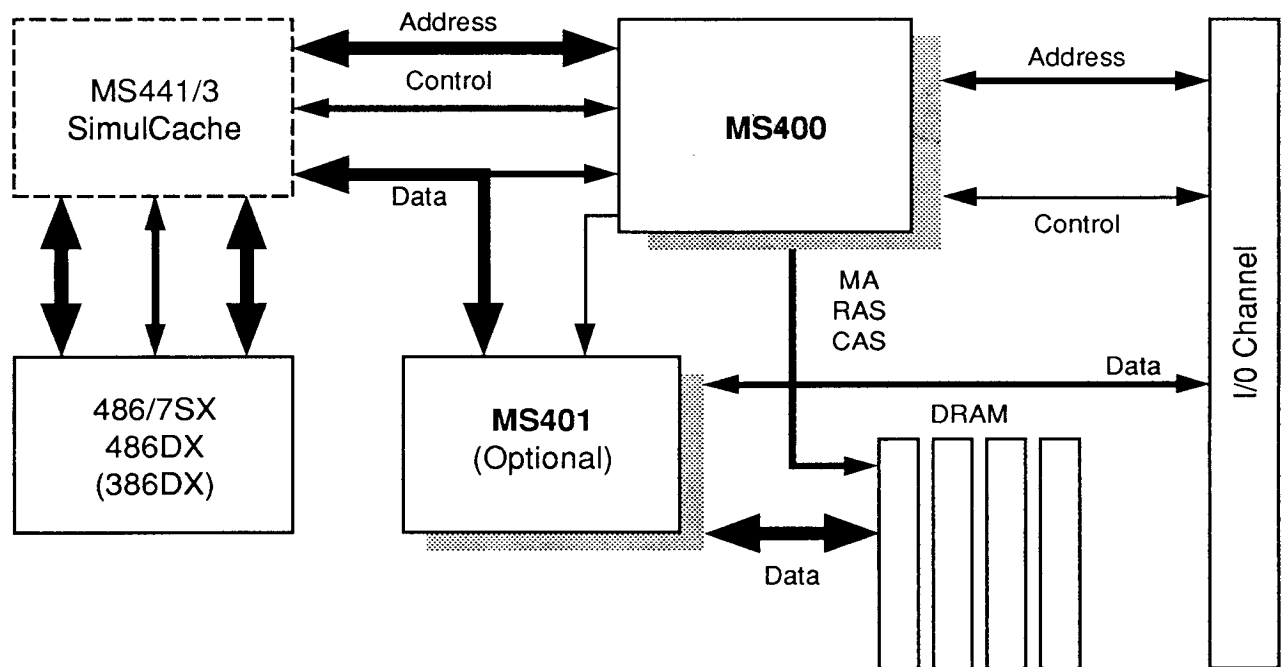
- Direct Interface to 486SX/487SX or 486DX at speeds from 20-33MHz
- 50MHz 486 supported with MOSEL SimulCache
- Fast Burst Mode Memory Manager — Burst Read and Burst Write Supported
- 64 Megabytes Main Memory in up to 4 Banks
- Memory Mixing (256K, 1M, 4M, x1 or x4)
- 32 or 64 bit Memory Data Path Supported
- Shadow RAM, Remap, Fast A20 Gate, Hot Reset
- 8 or 16 bit BIOS ROMs, System & Video BIOS may be combined
- Supports KEN# to control 486 internal cache
- Local Bus Peripherals Support (VGA and others)
- Synchronous I/O Clock Divide
- Discrete data buffers or optional MS401 Integrated Data Buffer
- 208 pin PQFP — Sub Micron CMOS Technology

DESCRIPTION

The MS400 is a highly integrated single chip AT optimized specifically for 486 CPUs. Emphasis has been placed on the cost reduction requirements of 486SX applications, though performance has not been compromised. Fast single clock bursting can be achieved with 60 or 80ns DRAMs. For peak performance at the higher clock speeds, the MS400 also has an optimized interface for the MS441/3 SimulCache subsystem.

The device integrates the DRAM controller, DMA controllers, interrupt controllers, memory mapper, timers and all other logic basic to an AT system. External logic required — RTC, keyboard controller, parity generators and data buffers. MOSEL offers an optional MS401 supporting 32 and 64 bit data paths to DRAM. This offers higher integration and better performance in non-cache modes. Low cost implementations with discrete buffers are also supported.

The MS400 has the ability to mix memory types and mix combinations of 32 and 64 bit memory data paths. This allows many different DRAM configurations up to 64 megabytes using 9 bit or 36 bit SIMMs.



MOSEL Corporation 914 West Maude Avenue, Sunnyvale, CA 94086 U.S.A. 408-733-4556

TABLE OF CONTENTS

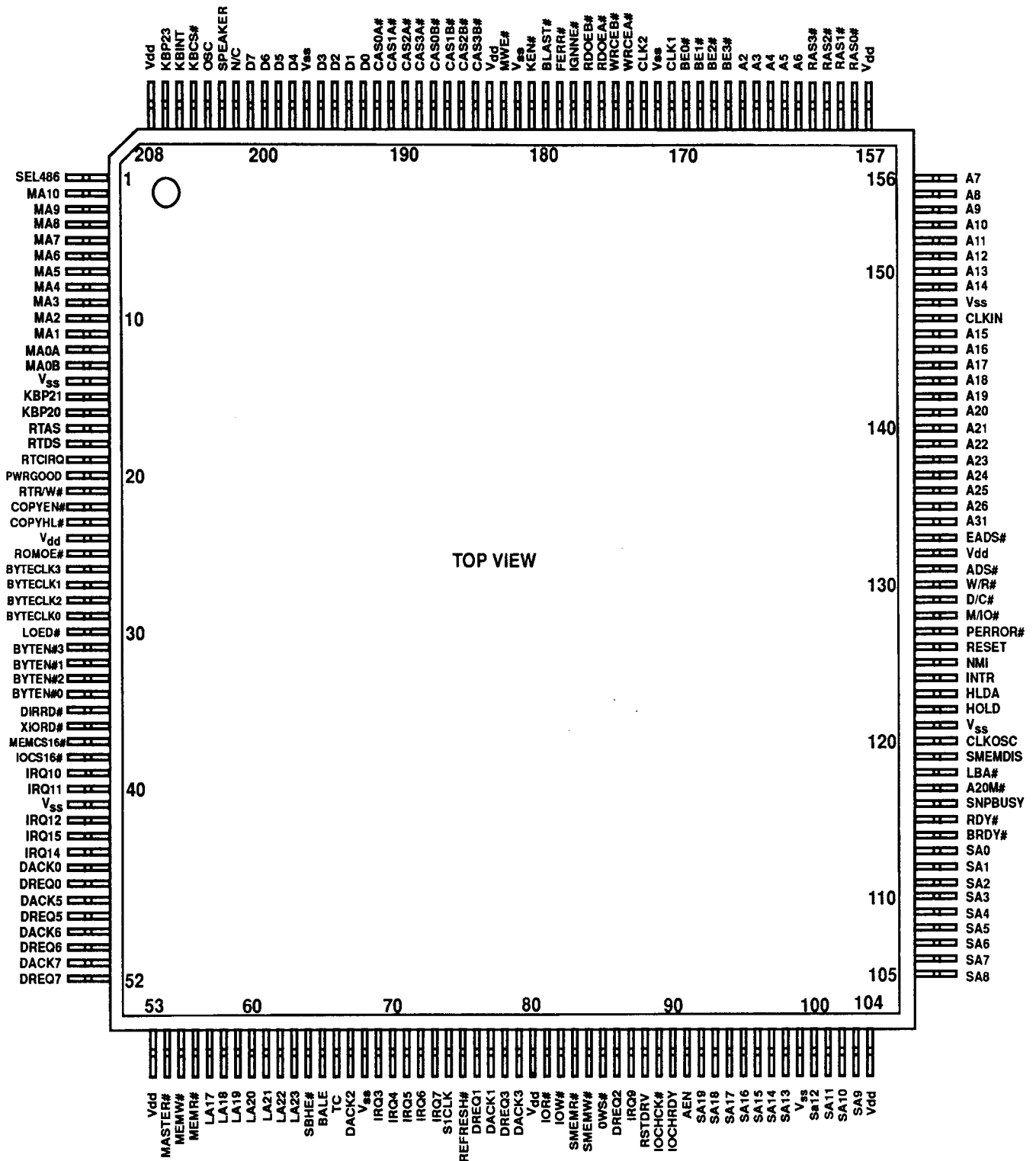
Section 1.	MS400 Pin-Out	4
Section 2.	MS400 Pin Descriptions	5
Section 3.	MS400 Product Overview	15
	3.1 Introduction	15
	3.2 Integration Of The MS400	15
	3.2.1 Address Buses	15
	3.2.2 Data Buses	16
	3.3 Theory Of Operations	17
	3.3.1 Bus Cycle Activity	17
	3.3.2 Burst/Non-burst Cycles	18
	3.4 Bus Cycle Response	18
	3.4.1 DRAM Memory Cycles	19
	3.4.2 AT Cycles	19
Section 4.	Programming Model & Register Set	20
	4.1 Programming And Reading Registers	20
	4.2 Register Bit Descriptions	20
	4.3 Default Register Values	25
Section 5.	Functional Description	26
	5.1 MS400 Clock Circuitry	26
	5.2 CPU/Local Bus Support	27
	5.2.1 CPU Support	27
	5.2.2 CPU Clock Switching	27
	5.2.3 Local Bus Access Support	28
	5.2.4 CPU/Simulcache Reset	29
	5.2.5 Cacheability Support	29
	5.3 DRAM Memory Controller	30
	5.3.1 DRAM sizes supported	30
	5.3.2 DRAM memory array mapping & organization	30
	5.3.3 DRAM Read Cycles	32
	5.4 AT Bus Interface	46
	5.4.1 AT Bus Overview	46
	5.4.2 Synchronous AT clock	46
	5.4.3 Basic AT Bus Cycle	47
	5.4.4 Aligned/mis-Aligned Transfers	47
	5.4.5 Byte Swapping	47
	5.4.6 Assembly on AT reads	47
	5.4.7 Dis-Assembly on AT Write Cycles	50
	5.4.8 AT DMA Operations	50

TABLE OF CONTENTS (continued)

5.4.9	ROM Accesses	52
5.4.10	Memory refresh cycle	52
5.4.11	AT Bus Owner Cycles	52
5.5	Interrupt Acknowledge Cycles	53
5.6	Halt/Shutdown Cycles	53
5.7	Flush Cycles	53
5.8	Misc Functional features	53
5.8.1	Memory Remap	53
5.8.2	Shadow RAM support	54
5.8.3	Turbo/Normal Operation	55
5.8.4	A20 Gate/Fast A20 Gate	55
5.8.5	Reset & Fast reset	55
5.8.6	Parity	55
5.8.7	BIOS Size and Location	56
5.8.8	Combined System/Video BIOS	56
Appendix A.	MS400 Standard Peripherals	57
Section 6.	DC Specifications	59
Section 7.	Thermal Specifications	59
Section 8.	AC Specifications & Test Measurement Points	60
Section 9.	Package Dimensions	69

MS400

SECTION 1. MS400 PIN-OUT



SECTION 2. MS400 PIN DESCRIPTIONS

LOCAL BUS INTERFACE PINS

The MS400 local bus consists of either the MOSEL Simulcache solution or the i486 CPU.

SYMBOL TYPE DESCRIPTION

A(31,26:2)	I/O	Address lines. These lines, together with the byte enables, define the physical area of memory or I/O space accessed. As inputs, these lines are driven by the CPU or cache controller. As outputs, the MS400 drives these lines during invalidate or snoop cycles.			
ADS#	I	Address Status. This input indicates to the MS400 that a valid bus cycle definition and address are available on its cycle definition lines and address bus.			
A20M#	O	Address bit 20 Mask. The MS400 asserts this signal to both the Simulcache and the i486 CPU. Assertion (low) of this signal masks address bit 20 for bus cycles from both the cache and CPU, and also masks A20 for invalidate/snoop cycles to the CPU/cache.			
BE(3:0)#	I/O	Byte Enables. As inputs, the byte enable signals indicate active bytes during read and write cycles. BE3# corresponds to D31:24, BE2# corresponds to D23:16, BE1# corresponds to D15:8, and BE0# corresponds to D7:0. As outputs, the MS400 drives these lines during snoop cycles to the CPU/cache.			
BRDY#	O	Burst Ready. Assertion of Burst Ready indicates the completion of each transfer to/from the MS400. BRDY# indicates that the MS400 system has presented valid data in response to a read or that the MS400 system has accepted data in response to a write cycle. BRDY# is never asserted in idle states or during T1 bus states.			
			BLAST#	I	Burst Last. This signal indicates to the MS400 whether the current transfer will be the final one for the cycle presently occurring. De-assertion (high) from either the cache or the CPU indicates to the MS400 that one or more additional transfers will be required in order to complete the cycle. The MS400 latches the BLAST# signal at the end of the T2 state.
			CLKIN	I	Clock Input. CLKIN is used as the clock signal for the internal MS400 state machines. CLKIN is always the same frequency as the system (i.e. CLKIN = 33 MHz for either 386 or 486 33 MHz systems). For most 486 systems, CLKIN should be sourced from CLK2, while 386 CPU systems will use CLK1 for this signal.
			CLKOSC	I	Clock Oscillator. This is the basic clock input to the MS400. This signal is driven from an external oscillator. If the MS400 is in 486 CPU mode, this signal will be a single frequency signal (i.e. 33 MHz CLKOSC input for a 33 MHz MS400 system). However, if the MS400 is configured in the 386 CPU mode, this input signal will be double frequency (66 MHz for a 33 MHz MS400 system).
			CLK1	O	Single-Frequency Clock output. This signal is always one-half the frequency of the CLKOSC input (i.e. 16.6 MHz for a 33.3 MHz 486 CPU system). For 386 CPU systems, CLK1 should connect to the MS400 CLKIN input.

The MS400 asserts BRDY# at the completion of all bus cycles. For burstable bus cycles, BRDY# will be asserted without the assertion of RDY#. The 486 CPU will then treat these cycles as burstable.

For 386 CPU systems, this signal should be connected to the CPU RDY# input.

The AT system bus clock S1CLK is derived from this signal.

MS400

		The functionality of the CLK1 signal allows split clock 486 CPU-based systems to be designed. Such systems operate with the CPU/cache subsystem at high frequency, while the chipset operates at half that frequency. To implement a split clock 486 CPU based system, CLK1 should be used for the MS400 CLKIN input instead of CLK2.	HOLD	O	Hold Request. Assertion of this signal allows the MS400 to gain complete control of the local bus. In response to the assertion of HOLD, HLDA will be returned by the cache or CPU at the end of the current bus cycle or series of locked cycles. The MS400 will continue to own the local bus until HOLD is de-asserted.
CLK2	O	Double-Frequency Clock Output. This signal is always the same frequency as the CLKOSC input. CLK2 is always connected to the CPU clock input. For most MS400/486 CPU systems, CLK2 is also connected to the MS400 CLKIN input. For system operation with split clocks, refer to the CLK1 description.	IGNNE#	O	Ignore Numerics Error. This signal is an output to the 486 CPU. Assertion (low) of IGNNE# informs the CPU that non-control floating point instructions may continue to be processed, even while an error condition is present in the CPU floating-point unit.
EADS#	O	External Address Strobe. Assertion of this signal (low) indicates that a valid address (A31, A26:2) and byte enables are being driven on the local bus. If no MOSEL cache is present in the system, an invalidate cycle will occur internally in the i486 CPU. If the MOSEL Simulcache is present, a snoop read or write cycle will be triggered in the cache, as well as an invalidate to the CPU for snoop write cycles.	INTR	O	Maskable Interrupt. The MS400 asserts this signal to the 386 or i486 CPU when an interrupt request is pending internally. In response, the CPU will generate two locked Interrupt Acknowledge cycles, in order to determine the interrupt vector.
FERR#	I	Floating-Point Error. This signal is an input from the 486 CPU. Assertion (low) of FERR# indicates to the MS400 that an error has occurred in the floating point unit of the CPU. If the 386 CPU is used in a system, the system should ensure that this signal is always driven to a logical '1'.	KEN#	O	Cache Enable. Activation of KEN# from the MS400 indicates that the current bus cycle is to a cacheable address. Assertion of KEN# will create an internal cache line fill in the i486 CPU cache, if the cache is enabled and the bus cycle is of the type that can be cached.
HLDA	I	Hold Acknowledge. In response to assertion of the HOLD pin, the cache or CPU will return HLDA to relinquish control of the MOSEL cache controller bus. HLDA is driven active in the same clock that the MS400 receives control of the local bus.	LBA#	I	Local Bus Interface. Assertion (low) of this signal indicates to the MS400 that the current bus cycle is a read or write to/from a resource which exists on the local CPU bus. In response, the MS400 will consider the cycle terminated. The local resource has ownership of the bus cycle and must return RDY# or BRDY# to complete the cycle.

M/IO#, D/C#, W/R# I **Memory/IO, Data/Code, and Write/Read.** These are the primary bus definition signals. These are driven valid as the ADS# signal is asserted. The combination of these signals, as shown below, defines the bus cycle type:

M/IO#	D/C#	W/R#	Type
0	0	0	Interrupt Ack
0	0	1	Special Cycle
0	1	0	I/O Read
0	1	1	I/O Write
1	0	0	Code Read
1	0	1	N/A
1	1	0	Memory Read
1	1	1	Memory Write

edge of SNPBUSY, the state of Memory Disable indicates to the MS400 whether the cache will supply the requested read data, or the MS400 will be required to supply the data.

While the Simulcache has an improved 486 CPU-style bus, SMEMDIS is one of three new signals that the Simulcache defines which the i486 CPU does not have. These three pins are used in order to support the enhanced snoop operations of the Simulcache.

For MS400 systems which do not include the Simulcache, SMEMDIS should be pulled inactive (low) to allow snoop cycles to occur correctly.

NMI O **Non-Maskable Interrupt.** The MS400 asserts this signal directly to the 386 or i486 CPU. Assertion of this signal indicates that a high-priority interrupt has occurred.

SNPBUSY I **Snoobusy.** The assertion (high) of this signal indicates to the MS400 that a snoop cycle is occurring in the MOSEL cache controller. After EADS# is asserted to the Simulcache, this signal is asserted in the following clock, and will stay asserted until the snoop completes from 2-12 CPU T-states later. The MS400 will not allow the current cycle to be completed until SNPBUSY has been sampled low again.

RDY# O **Ready.** Assertion of Ready indicates the completion of non-burstable transfers to/from the MS400. RDY# indicates that the MS400 system has presented valid data in response to a read or that the MS400 system has accepted data in response to a write cycle. RDY# is never asserted in T1 states.

While the Simulcache has an improved 486 CPU-style bus, SNPBUSY is one of three new signals that the Simulcache defines which the i486 CPU does not have.

SNPBUSY is also used to inform the MS400 of the presence or absence of the Simulcache in a system. On the falling edge of Reset, the MS400 will sample SNPBUSY. If the Simulcache is present, it will drive this signal to a logical '0'. If no Simulcache is present, the system should ensure that SNPBUSY is driven to a logical '1'.

For non-burstable cycles, the MS400 will assert both RDY# and BRDY# to complete the cycle. Since RDY# has higher priority than BRDY# internally in the 486 CPU, the CPU will treat such cycles as non-burstable cycles.

RESET O **Reset CPU.**

SMEMDIS I **System Memory Disable.** This signal is driven by the Simulcache cache controller in response to snoop read operations, to indicate whether the operation is a hit or miss in the cache. At the falling

MS400

PIN DESCRIPTIONS (Memory Bus Interface)

SYMBOL TYPE DESCRIPTION

CAS 3:0A#	O	<p>Set A Column Address Strobes 3:0. These signals are outputs from the MS400 to the control lines of the local DRAM array. CAS 3:0A# are used for set A of the DRAM's. The falling edge of CAS# latches the column address inside the DRAM's for a given read/write cycle.</p> <p>The assertion of each of the CAS# lines signifies the activation of the corresponding byte for that cycle. Read or write cycles to/from the DRAM array which have multiple transfers (BLAST# high) will force all CAS# signals for a bank to be low, returning four bytes of data. For operations which involve only a single transfer (BLAST# low), however, the CAS# signals for the accessed bank are derived from the cache/CPU byte enables.</p>	<p>MA 10:1</p> <p>O</p> <p>Memory Address Bits 10:1. These signals are outputs from the MS400 to the local DRAM memory array. The MS400 contains the necessary internal multiplexer logic in order to switch these bits between row and column addresses. If 4 Mbit DRAM's are used, all eleven MA bits will be required. If 1 Mbit DRAM's are in the system, only ten MA bits will be used, while 256 Kbit DRAM's will use only nine bits.</p>
CAS 3:0B#	O	<p>Set B Column Address Strobes 3:0. These signals are outputs from the MS400 to the control lines of the local DRAM array. CAS 3:0B# are used for set B of the DRAM's. The falling edge of CAS# latches the column address inside the DRAM's for a given read/write cycle.</p> <p>The assertion of each of the CAS# lines signifies the activation of the corresponding byte for that cycle. Read or write cycles to/from the DRAM array which have multiple transfers (BLAST# high) will force all CAS# signals for a bank to be low, returning four bytes of data. For operations which involve only a single transfer (BLAST# low), however, the CAS# signals for the accessed bank are derived from the cache/CPU byte enables.</p>	<p>MWE#</p> <p>O</p> <p>Memory Write Enable. This signal is connected to the Write Enable (WE#) control lines of the local DRAM memory array. MWE# is asserted (low) during write cycles in order to latch the write data inside the DRAM's. During reads, MWE# will be de-asserted.</p> <p>In addition, MWE# also controls the latches used for write cycles. MWE# should be connected to the output enable pin (G#) for the latches. During write cycles, the assertion of MWE# will enable the latches, gating data from the SPD bus onto the local memory MD bus.</p>
MA0A, MA0B	O	<p>Set A and B Memory Address 0. These signals are outputs from the MS400 to the local DRAM memory address bus. MA0A and</p>	<p>PERROR#</p> <p>I</p> <p>Parity Error. This signal is driven to the MS400 from the i486 CPU. Assertion (low) of PERROR# indicates the occurrence of a parity error on a read to the CPU. In response, the MS400 will set an internal latch inside one of its registers, and will assert NMI (Non-Maskable Interrupt) to the CPU.</p>

RAS 3:0#	O	<p>Bank Row Address Strobes 3:0. These signals are outputs from the MS400 to the control lines of the local DRAM memory array. Each bank of DRAM memory in an MS400 system will be driven by a separate RAS# signal. The falling edge of RAS# is used to latch the row address inside the DRAM's for a given read/write cycle.</p>	<p>PIN DESCRIPTION (AT Bus Interface)</p>	<p>SYMBOL TYPE DESCRIPTION</p>	
RDOEA#, RDOEB#	O	<p>Set A & B Read Output Enables. These signals control the latches which are used for read cycles. The RDOE# signals are connected to the output enables of the latches. When these signals are asserted (low), read data is gated from the memory data (MD) bus onto the SPD bus.</p>		AEN	O
WRCEA#, WRCEB#	O	<p>Set A & B Write Clock Enables. These signals control the latches which are used for local DRAM memory write cycles. The assertion (low) of these signals will, upon the next rising clock edge, latch data from the SPD bus onto the memory data (MD) bus.</p>	BALE	O	<p>Bus Address Latch Enable. This signal, when asserted (high), validates addresses on the I/O channel. BALE is driven high for the entire duration of DMA or Master cycles.</p>
			DACK 3:0#, DACK 7:5#	O	<p>DMA Acknowledge signals. These lines are asserted (low) in response to the corresponding DRQ signals. The falling edge of the appropriate DACK# line selects which add-on card will be part of the transfer cycle or will be granted bus ownership.</p>
			DREQ 3:0, DREQ 7:5	I	<p>DMA Requests. The assertion (high) of any of these indicates an I/O device request for a DMA transfer, or that a bus owner card wishes to take control of the bus. Once asserted, the DRQ line must remain asserted until the corresponding DACK# signal is activated.</p>
			IOCHCK#	I	<p>I/O Channel Check. The assertion (low) of this signal indicates an error condition on the I/O channel.</p>
			IOCHRDY	I	<p>I/O Channel Ready. A slow device on the I/O channel may drive this signal low to insert additional (wait) states for the current AT cycle. The cycle will be extended by the addition of wait states until IOCHRDY is sampled active (high).</p>
			IOCS16#	I	<p>16-bit I/O Chip Select. This signal asserted (low) from an I/O device indicates that it can support a 16-bit transfer. If not asserted, an 8-bit transfer is assumed.</p>

MS400

<p>IOR# I/O I/O Read. This signal is an output during CPU I/O read cycles, and an input during DMA and Master cycles.</p>	<p>snop operations. This connection indicates to the Simulcache whether the current snoop/invalidate cycle is a write or a read operation. High indicates a write operation, while low indicates a read. MEMR# will be asserted at the same time as EADS#.</p>
<p>IOW# I/O I/O Write. This signal is an output during CPU I/O write cycles, and an input during DMA and Master cycles.</p>	<p>Memory Write. The assertion of this signal indicates a memory write access to the AT bus. This signal has the same functionality as the SMEMW# signal, but indicates a write access within the entire 16 Mbyte AT memory space.</p>
<p>IRQ 7:3, IRQ 12:9, IRQ15:14 I Interrupt Requests. The IRQ pins are used to halt the current CPU task and transfer control in order to service some device. The assertion of any of these signals generates an INTR request to the CPU.</p>	<p>MEMW# I/O</p>
<p>LA 23:17 I/O Large Address bits 23:17. These are bi-directional address signals to/from the AT bus. These signals are outputs during CPU accesses, and are inputs during Master mode operation. However, in master refresh mode, these are output signals to the AT bus. During CPU accesses, LA23:17 are derived from the CPU/cache address pins and are internally latched inside the MS400 at the falling edge of the BALE signal.</p>	<p>OSC I Oscillator. This signal has a 14.31818 MHz frequency. This signal, divided by 12, is used as a clock for the 8254-compatible timer within the MS400.</p>
<p>MASTER# I Master. This signal, when asserted (low), indicates that another CPU or a DMA controller is residing on the I/O channel and controlling the system bus.</p>	<p>PWRGOOD I Powergood. This signal (active high) indicates that power to the MS400 system is at proper voltage levels and is stable.</p>
<p>MEMCS16# I 16-bit Memory Chip Select. This signal asserted (low) from a memory device indicates that it can support a 16-bit transfer. If not asserted, an 8-bit transfer is assumed.</p>	<p>REFRESH# I/O Refresh. As an input, this signal is driven by the I/O channel master during Master refresh cycles. As an output, this signal is driven during a normal AT refresh cycle.</p>
<p>MEMR# I/O Memory Read. The assertion of this signal indicates a memory read access to the AT bus. This signal has the same functionality as the SMEMR# signal, but indicates a read access within the entire 16 Mbyte AT memory space.</p> <p>If the Simulcache is included in an MS400 system, MEMR# should be connected to the Simulcache SMEMW/R# signal to support</p>	<p>RSTDRV O Reset Drive. This signal is a system reset generated from the POWERGOOD input. If the MOSEL Simulcache is included in a system, RSTDRV (not CPU Reset) should be used to reset the Simulcache. RSTDRV is synchronized with the AT bus clock S1CLK.</p>
<p></p>	<p>SA 19:0 I/O System Address 19:0. These are the AT bus address lines used to address memory or I/O devices within the system. On normal cycles, SA19:0 are driven to the system board on the falling edge of BALE. SA19:0 are valid throughout the bus command cycle.</p>

SBHE#	I/O	System Byte High Enable. This signal is driven by the current bus owner to indicate that valid data resides on the D15:8 signal lines. This signal, along with the SA0 signal, allows the current bus owner to indicate an 8- or 16-bit cycle.
SMEMR#	O	System Memory Read. The assertion of this signal (low) indicates an AT read cycle to the lower 1 Mbyte of memory space.
SMEMW#	O	System Memory Write. The assertion of this signal (low) indicates an AT write cycle to the lower 1 Mbyte of memory space.
S1CLK	O	System Clock. This is the basic clock frequency for the AT bus. This signal is generated synchronously from the CLKIN input to the MS400, and as such will be some fraction of CLKIN's frequency. S1CLK can be either 1/2, 1/2.5, 1/3, 1/4, 1/5, or 1/6 the frequency of CLKIN, as chosen by three bits in the control registers. In any configuration, S1CLK should not exceed 8.33 MHz in order to ensure compatibility.
TC	O	Timer Control. This signal is driven by the DMA controller to indicate that all data has been transferred. TC is a common signal which is shared by all DMA channels.
OWS#	I	0 Wait States. The assertion (low) of this signal allows the current bus cycle to terminate sooner than would a standard AT cycle.

PIN DESCRIPTION (Miscellaneous)

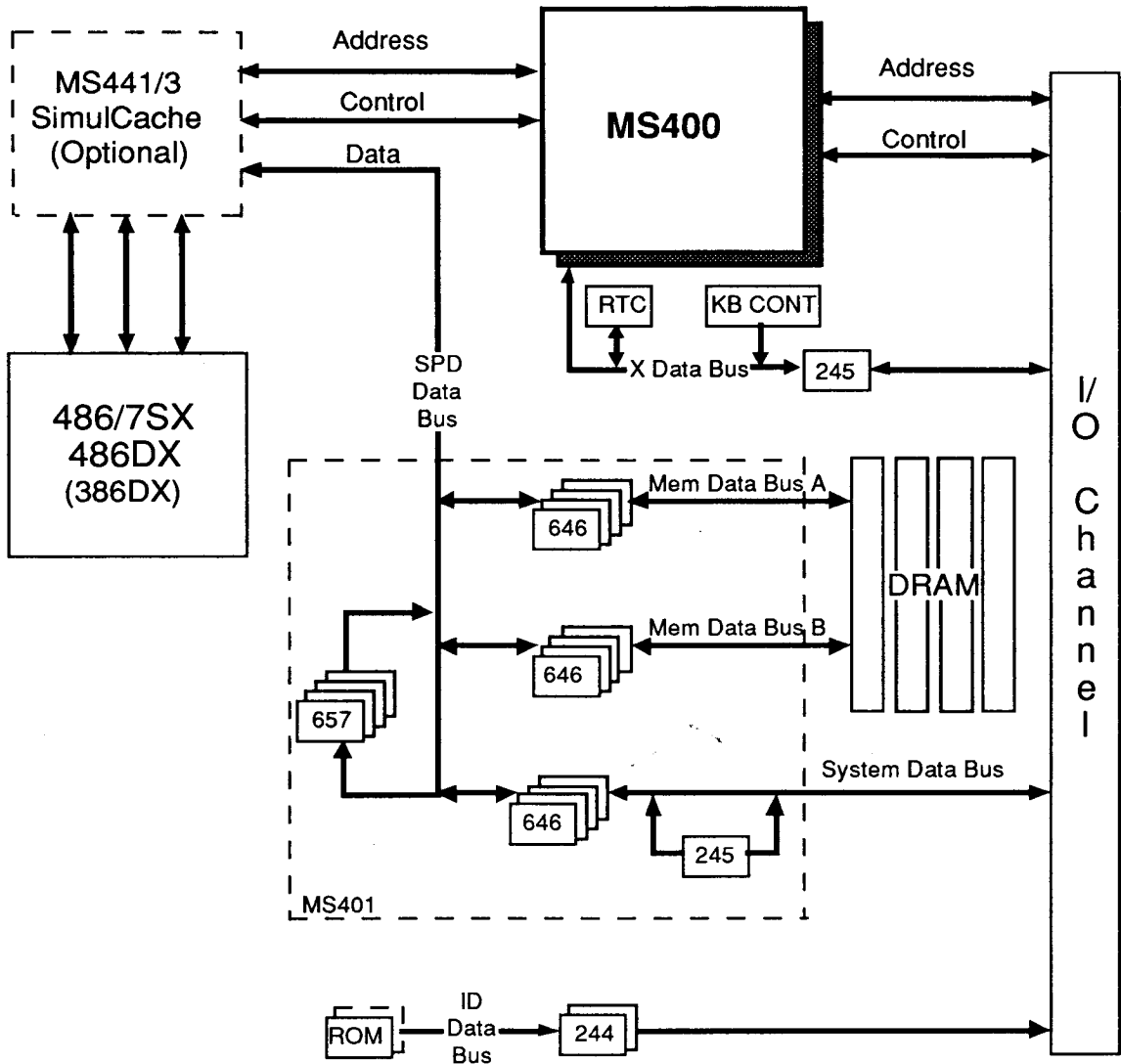
<u>SYMBOL</u>	<u>TYPE</u>	<u>DESCRIPTION</u>
BYTECLK3:0O		System Data Bus Byte CLK's. The de-activation (falling edge) of these signals latches read data from the SD (system data) bus onto the SPD bus, during AT read cycles. These signals are intended to be used with negative-edge latches. Each BYTECLK signal is connected to one latch device, each representing one byte of the SPD bus. Dword (32-bit) CPU reads are stored and assembled through multiple 8/16 bit accesses through use of these signals. The return of RDY# to the CPU, to complete such a cycle, is delayed until the fourth read operation is completed on the AT bus. Also, 16-bit read operations which are to 8-bit devices residing on the AT bus can be assembled through use of these devices. Since the SD data bus is only 16 bits wide while the SPD data bus is 32 bits wide, two of the latches are attached to the low byte of the SD bus (bytes 0 and 2 of the SPD bus), and two to the high byte of the SD bus (bytes 1 and 3 of the SPD bus).
BYTEN3:0#	O	Byte Enables 3:0. These signals control the output enables for the buffer devices which exist between the SD data bus and the SPD data bus. An active (low) assertion of any of these signals gates data from the SPD bus onto the SD bus. Since the SPD bus is 32 bits wide while the SD bus is only 16 bits wide, two of the buffers are connected to the low byte of the SD bus (SPD bytes 0 and 2), while two are connected to the high byte of the SD bus (SPD bytes 1 and 3).

MS400

<p>COPYEN# O</p>	<p>Copy Enable. This signal enables byte swapping (copying) on the AT bus. CPYEN# should be connected to the OE# (output enable) of an external 245 buffer chip. The activation of CPYEN# enables the outputs of the AT data byte swapper, gating the high data byte onto the lower byte or vice versa. CPYEN# is used in conjunction with CPYHL# (see below).</p>	<p>D7:0 I/O</p>	<p>level of DIRRD# will gate data from the SD data bus onto the SPD data bus, for latching onto the memory data bus.</p>
<p>COPYHL# O</p>	<p>Copy High to Low. This signal is used for byte swapping (copying) on the AT data bus. Along with the CPYEN# signal, COPYHL# can be used to control an external 245 buffer chip. CPYHL# should be connected to the T/R# (Transmit/Receive) pin.</p> <p>When CPYHL# is asserted (low), it indicates that the high data byte D15:8 should be copied to the low data byte D7:0. As such, the 'B' side pins (B8:1) should be connected to the high data byte, while A8:1 are connected to the low data byte.</p> <p>For reads from the AT bus which require byte swapping, COPYHL# will be de-asserted to perform a low-to-high byte copy. For writes to the AT bus which require swapping, COPYHL# will be asserted to copy from the high byte to the low byte.</p>	<p>KBCS# O</p> <p>KBINT I</p> <p>KBP20 I</p> <p>KBP21 I</p>	<p>Data Bits 7:0. The data bus is used for read/write operations to and from the MS400. Data passes through these pins for both MS400 configuration register operations, as well as read/writes to the integrated peripherals (8237's, 8254's, 8259's) contained within the MS400.</p> <p>Keyboard Chip Select. This signal enables the 8742 keyboard controller device.</p> <p>Keyboard Interrupt. This signal generates an interrupt to the CPU, with a vector of 1, in order to service the external 8042/8742 keyboard controller device.</p> <p>Keyboard Port 2 Bit 0. This is the Reset signal, and is an input from the 8042/8742 keyboard controller. This signal is used by software to perform a warm re-boot of the CPU (without affecting the system state), in order to perform the switch from Protected Virtual Address Mode to Real Mode.</p> <p>Keyboard Port 2 Bit 1. This signal is an input from the 8042/8742 keyboard controller. This is the GATEA20 signal. Disabling this signal is part of the procedure for switching from Protected Virtual Address Mode to Real Mode.</p>
<p>DIRRD# O</p>	<p>Direction Read. This signal is used to control the four buffer devices which exist between the SD data bus and the SPD data bus. This signal is used for writes from the SPD bus onto the SD bus. For such writes, DIRRD# is driven high to gate data onto the SD data bus, or driven low during DMA IOR#, MEMW#.</p> <p>DIRRD# is driven low during DMA operations, for reads from the I/O space of the AT bus which are write cycles to the DRAM memory array. For these cycles, the low</p>	<p>KBP23 I</p>	<p>Keyboard Port 2 Bit 3. This signal is an input from the 8042/8742 keyboard controller. This is the Turbo/Normal# signal, and is used to control the execution speed of the system. For compatibility, some programs require that the system perform operations no faster than the original IBM PC/AT.</p>

LOED#	O	Latch Output Enable. This signal controls the output enables for the four latches which exist between the SD data bus and the SPD data bus. The activation (low) of this signal enables data coming from the SD bus during AT bus reads onto the SPD bus. This signal is connected to the output enables for all four latch devices.	SEL486	I	SELECT 486. This signal informs the MS400 which CPU is operating in the system. When asserted (high), 486DX/SX CPU operation is chosen. If this signal is sampled low, 386 CPU operation is selected. SEL486 should be tied either high or low.
ROMOE#	O	ROM Output Enable. This signal enables the data from the BIOS ROM(s) onto the system BIOS ID data bus. In addition, ROMOE# should be connected to the output controls for the 244's, enabling the ROM data onto the AT system SD data bus.	SPEAKER	O	Speaker. This is the system speaker output.
			Vdd	I	Power Supply Pins. +5 Volts.
			Vss	I	Ground Pins.
			XIORD#	O	Peripheral I/O Read. This signal is used to control the direction of the standard 245 buffer device which exists between the SD (system data) bus and the XD peripheral data bus. As such, XIORD# should be connected to the T/R# pin of the buffer. Since the peripheral data bus is only eight bits wide, only one 245 device is required. The system data bus side of the 245 is always connected to the lower byte (D7:0) of the SD bus.
RTAS	O	Real-Time Clock Address Strobe. This is one of the interface signals between the MS400 and the system real-time clock chip. This signal is associated with port 70H.			
RTCIRQ	I	Real-Time Clock Interrupt Request. This is one of the interface signals between the MS400 and the system real-time clock chip. This signal generates an interrupt request, with a vector of 8, to service the external real-time clock.			
RTDS	O	Real-Time Clock Data Strobe. This is one of the interface signals between the MS400 and the system real-time clock chip. This signal is associated with port 71H.			
RTR/W#	O	Real-Time Clock Read/Write. This is one of the interface signals between the MS400 and the system real-time clock chip. This signal indicates a write or read to/from the RTC. When asserted high, a read is indicated, while a low on this signal indicates a write to the RTC.			

MS400 SYSTEM ARCHITECTURE



SECTION 3. MS400 PRODUCT OVERVIEW

3.1 Introduction

The MOSEL MS400 offers a high performance 25/33/40 MHz 386, 25/33/50 MHz i486DX, or 25/33 MHz i486SX CPU-based system. The MS400 is an integrated solution that offers 100% compatible IBM PC/AT system.

The MS400 chipset consists of the MS400 system chip, and the optional MS401 buffer device. The MS400 is designed for systems which include the MOSEL Simulcache integrated cache solutions. An i486DX CPU-based system is shown in Figure 1.

The MS400 system chip integrates an ISA system bus controller, integrated high-performance memory controller, and peripherals controller in a single VLSI chip. The MS400 supports a local CPU bus, compatible AT buses, and a memory bus. The memory bus is quite flexible, and can be optimized for cost efficiency or for high performance.

The MS400 operates at frequencies up to 50 MHz, supporting burst-mode memory accesses. It supports up to 64 Mbytes of DRAM's with either 256 Kbit, 1 Mbit, or 4 Mbit page mode DRAM's in either a one, two, three, or four bank configuration.

The MS400 can support either the 386, i486DX, or i486SX CPU's. For best performance, the MOSEL Simulcache should be included in a system design. The Simulcache supports all of the above CPU's. The Simulcache will also be offered at frequencies up to 50 MHz. For system designs not optimized for performance, the MS400 supports a direct connection to either the 386, i486SX, or i486DX CPU's.

For high integration and ease of design, use of the optional MS401 buffer device is recommended, which contains standard buffers for high integration, and specialized memory latches for high performance. Inclusion of the MS401 in a system greatly reduces system device count, while offering a 64-bit wide path to memory. The MS401 supports parity for all read operations from the DRAM memory banks.

The combination of the MS400 and MS401 in a system allows a high integration, 100% IBM PC/AT compatible design, while enjoying the advantage of high performance. An MS400/401 PC/AT system can be designed with as few as 11 IC components, excluding the DRAM array.

3.2 Integration Of The MS400

The MS400 retains 100% IBM compatibility, while offering system designers high performance and flexible system design alternatives. An MS400 system has 5 address and 6 data buses.

3.2.1 Address Buses

The address buses are as follows:

- 1) Local CPU Address Bus (A31,A26:2)
- 2) MOSEL Cache Address Bus (A31, A26:2)
- 3) DRAM Memory Address Bus (MA10:1, MA0A/B)
- 4) AT System Address Bus (SA19:0)
- 5) Latchable Address Bus (LA23:17)

Following is a description of each of the address buses listed above:

Local CPU Address Bus

The CPU (either 386, i486SX, or i486DX) and/or NPX (387, 487SX, 3167, or 4167) reside on this bus. In addition, the system designer may designate some other system resources to reside on this bus (such as a high-speed video controller, etc.). The control signals for this bus can be either 386 or i486 CPU style signals, depending on which CPU is operating within the system.

For simplicity, the i486SX and i486DX CPU's shall be notated as the i486 or 486 CPU's in this document. Any reference to 486 or i486 shall be understood to mean either of the CPU's. The exception is for numerics support, since the floating point unit inside the i486SX CPU is non-operational.

MOSEL Simulcache Address Bus

This is the address bus through which the MS400 communicates with the MOSEL Simulcache. This is a 486 CPU style bus, with some performance enhancements. The Simulcache drives the majority of cycles on this bus to the MS400. The MS400 supports burst reads to fill cache lines in both the MOSEL cache and the CPU. In addition, the MS400 can support burst writes from the Simulcache.

MS400

When DMA or other master cycles occur, the MS400 will pass such cycles on to the cache across this bus. Snoop writes will trigger an invalidate cycle to the internal 486 CPU cache if 486 CPU mode is chosen, and a possible update to the cache. Snoop read requests may be satisfied by the cache. Three new control pins not present in the 486 CPU bus definition are introduced on this bus to support snoop cycles.

The MS400 can operate in a system without the presence of the Simulcache. In such a system, the MS400 is connected directly to the 386 or i486 CPU address bus. Since the Simulcache bus is similar to the i486 CPU bus, a non-cache system can be built with minimum modification. For this type of system, the i486 CPU will drive most of the cycles occurring on the bus. However, the MS400 will drive invalidate cycles to the CPU when DMA write operations occur.

DRAM Memory Address Bus

This is the address bus which is presented to the DRAM array. This bus is output by the MS400. The MS400 inputs the addresses driven by the MOSEL cache or CPU, and generates MA10:1 and MA0A/B for local DRAM accesses. The MS400 contains multiplexer logic internally, in order to split the physical address into both row and column addresses, to be presented across this bus to the DRAM's.

Since the MS400 supports interleaved memory accesses, two address 0 bits (MA0A and MA0B) are generated. To support one clock burst operations, MA0A/B will each be presented to half of the array.

In addition, the MS400 contains a counter used to cycle through memory to support refresh operations.

AT System Address Bus

This is the standard PC/AT system bus. The MS400 provides all the address and control lines necessary to generate a 100% compatible bus. SA19:0 provide 1 Mbyte of addressability on this bus. The control lines support either 8- or 16-bit operations.

Latchable Address Bus

This address bus is used to generate the upper addresses on the AT bus. The LA bus can supply address bits 20 through 23, in order to provide a full 16 Mbytes of address space.

In addition, the LA bus is the output of the memory mapper contained inside the MS400 for DMA cycles.

3.2.2 Data Buses

The 6 data buses for an MS400 system are:

- 1) Local CPU Data Bus (D31:0, DP3:0)
- 2) MOSEL Cache Data Bus (SPD31:0, SDP3:0)
- 3) DRAM Memory Data Bus (MD31:0)
- 4) AT System Data Bus (SD15:0)
- 5) Peripheral Data Bus (XD7:0)
- 6) BIOS Data Bus (ID15:0)

Following is a description of each of the data buses:

Local CPU Data Bus

This is the data bus connecting the CPU, NPX, and the MOSEL cache. The cache supports one clock read bursts to the CPU, if the i486 CPU is operating in the system. This is a 32-bit bus.

MOSEL Cache Data Bus

The data bus between the cache and the MS400. For read miss operations, data supplied by the DRAM array or from the AT bus will be stored in the cache. Also, the cache burst-rams have a bypass mode in which the data is transferred to the CPU with no synchronization or clock delay. Like the local CPU data bus, this bus is also 32 bits wide.

DRAM Memory Data Bus

With one bank of DRAM's installed, this bus is optimized for cost efficiency and is 32 bits wide. If two or more banks are installed, interleaving is automatically employed and a 64-bit wide memory bus is effectively created. The memory bus is separated from the SPD bus by latches, which are utilized in both directions. One clock burst read operations from this bus are supported, in order to fill internal cache lines in the i486 CPU cache. In addition, if the MOSEL Simulcache is present in the system, write-backs of lines containing dirty data can be burst to this bus in one-clock transfers.

Standard RAS#, CAS#, and WE# signals are generated from the MS400 in order to transfer data in and out of the DRAM array, while the CAS# signals and RDOE# (Read Output Enable) signals are used to gate data during burst reads. For latching data during write or burst write operations, the CLK and WRCE# signals are used.

AT System Data Bus

This is the standard 16-bit data bus for PC/AT systems. AT read and write cycles will end up on this bus. In addition, all cycles controlled by the MS400 which are not to the local DRAM array will send data across this bus. Read/write operations to the MS400 registers and integrated peripherals, Real-Time clock cycles, BIOS cycles, and keyboard cycles will all use this bus to transfer data.

Byte swapping of mis-aligned data transfers will also be performed on this bus. Since the AT bus is 16 bits wide, but some peripheral devices are only 8 bits, it is necessary for the peripheral to be connected to both bytes of the AT bus. A standard 245 buffer is provided to connect the high byte of the AT bus to the low byte. The COPYEN# and COPYHL# signals are provided in order to control the swapping operations.

The AT system data (SD) bus is separated from the SPD bus by four latch devices. The latch functionality of these devices are used for transferring data from the SD bus to the SPD bus, while the latches are utilized in a transparent 'flow-through' mode for transfers in the opposite direction. The SPD is 32 bits, and as such two bytes of the SPD bus must be able to connect to each byte of the SD bus. The low byte of the SD bus connects to both byte 0 and byte 2 of the SPD bus, while the high byte of SD connects to bytes 1 and 3.

Peripheral Data Bus

This is an 8-bit bus. The data ports of the MS400 internal registers and integrated peripherals reside on this bus, as well as the Real-Time Clock (RTC) and keyboard BIOS. This is separated from the system data bus by a standard 245 buffer. For operations to/from this bus, the transceiver will be controlled by the MS400 AEN and XIORD# signals.

System BIOS Data Bus

This bus can be either 8- or 16-bits wide, depending if one or two EPROM devices are used in the MS400 system. Data from the outputs of the BIOS device(s) appears on this bus, where it is gated onto the system data bus. This bus is separated from the system data bus by two 244 type devices. Since the BIOS ROM's are read-only, the 244(s) gate data in only one direction. The ROMOE# signal from the MS400 controls the outputs of the ROM's, as well as opening the 244's.

3.3 Theory Of Operations

3.3.1 Bus Cycle Activity

With the MS400 system chip and the Simulcache, system designers can easily implement both 386 CPU and 486 CPU based systems. In fact, the CPU can become an upgradeable option, since moving from a 386 based system to a 486 CPU system is straightforward. This simple upgrade is possible due to the dual bus architecture of the Simulcache. The cache resides completely between the CPU and the system, isolating CPU operations and defining the MOSEL cache bus. The cache directly supports both the 386 and i486 CPU's, presenting a common interface to the MS400 regardless of which CPU is operating in the system. The MOSEL cache bus functionally performs as an enhanced i486 CPU bus.

For systems which do not include the Simulcache, the MS400 can operate directly with either the 386 or 486 CPU's. This is because the Simulcache bus definition is very similar to that of the 486 CPU. Since the 386 bus definition is essentially a non-burst 486 bus, a direct 386 CPU / MS400 connection is supported.

Regardless of which CPU is operating in a system, the local CPU bus will be the location of the highest bus activity and the starting point for most cycles, since the CPU generates the vast majority of cycles in a MS400 system. When the Simulcache is present in a system, many cycles will complete on this bus without propagating to the MS400 system bus. The Simulcache will classify about 95% of cycles as cache hits, and service them accordingly, without creating any system bus cycles. Since the Simulcache is a write-back solution, both read and write hit operations will not generate cycles to the MS400.

In addition, some bus cycles are non-cacheable, but designated as *local bus cycles* and will also not cause any system bus cycles. Examples of this type of cycle are accesses to the NPX device or a CPU write to an address reserved in the MOSEL cache as write-protected. All such bus cycles will terminate on the local bus without creating system bus cycles.

The system designer can use the LBA# (Local Bus Access) pin of the Simulcache in order to designate some address ranges as belonging to non-standard local bus resources. For cycles where LBA# is asserted to the Simulcache, it will not generate system cycles and will allow the local bus resource to complete the cycle by returning the RDY# pin.

3.3.2 Burst/Non-burst Cycles

Both burst and non-burst cycles will occur in an MS400 system. Bursting in an MS400 system is defined to be compatible with the burst definition of the i486 CPU bus. At the end of the first transfer of each cycle when BRDY# is returned, BLAST# will be monitored. If BLAST# is asserted (low), then the cycle is complete. If BLAST# is de-asserted, then bursting may occur for that cycle, depending on which type of cycle and which bus it is occurring on. Bursting indicates that one clock data transfers may occur for a cycle. The normal unit of data being burst will be cache lines, and will consist of 16 contiguous bytes. These bytes will be paragraph aligned, residing from XXXX0-XXXXFH.

If the Simulcache and i486 CPU are present in a system, the cache will support burst read operations to the CPU. Burst writes are not supported by the i486 CPU. If the cache is operating with the 386 CPU, no bursts will occur on this bus since the 386 CPU cannot support such bursts. On the system port side, the Simulcache appears as a CPU and can perform both read and write bursts.

The MS400 is ideally designed to interface with the Simulcache, and as such can support bursting for most cycles. Read/write operations to the local DRAM array can both be burst in MS400 systems. Local DRAM cycles will be the great majority of cycles. Cycles to other standard devices/peripherals in an MS400 system will not cause burst operations. Even when the i486 CPU indicates that bursting is possible by driving BLAST# high, (such as ROM reads for 16-byte code fetches), the MS400 will not support such bursts. However, local bus cycles may be burst, if the local resource is capable of such bursts.

For operations to the DRAM memory array, the MS400 can support burst reads to either the i486 CPU or the MOSEL cache. For an MS400 system which includes the cache, the MS400 can support burst writes as well. The MS400 monitors the BLAST# output of the cache/CPU, and will perform bursts from the DRAM array until BLAST# is asserted (low). The MS400 can support burst read and burst write operations of either one, two, three, or four transfers.

During DRAM burst read/write operations, the BLAST# signal being de-asserted (high) will force all CAS# signals to the DRAM's to be low, since bursts will transfer four Dwords of data.

3.4 Bus Cycle Response

The majority of bus cycles will be driven by the Simulcache to the MS400. Cacheable CPU cycles which are read misses will be driven by the Simulcache. In addition, write backs of lines of 'dirty' data will be driven after tag replacements have occurred in the cache. Non-cacheable cycles, such as INTA cycles, or addresses which are contained in the MOSEL NCA address registers will also generate bus cycles to the MS400. Also, I/O cycles will not cause cache operations inside the Simulcache, and will be passed on to the system bus. Some bus cycles in memory address space will be generated to the MS400, even if the cycle was a hit in the MOSEL cache. Such write cycles will occur as such if they are PWT writes, PCD writes, or locked writes.

The MS400 can interface to either the 386 or the i486 CPU's, or the MOSEL Simulcache. The ADS# signal from the CPU or the SADS# signal from the cache indicates the beginning of a new cycle. The MS400 samples ADS# on each rising CLKIN clock edge. When ADS# is sampled asserted, the MS400 determines the destination of the current bus cycle, as selected by the address and state of the M/IO, D/C, and W/R control signal lines. Possible destinations for the cycle include local DRAM cycles, AT cycles, MS400 internal cycles, keyboard controller cycles, RTC cycles, and system BIOS cycles.

For systems which do not use the Simulcache, the MS400 offers control of the i486 CPU internal cache. Such control is accomplished through use of the MS400 KEN# (cache enable) output signal. Control of KEN# prevents data in sensitive address ranges from being cached inside the i486 CPU cache. Since the 386 does not contain a cache, no such consideration is necessary for 386 systems. If the Simulcache is included in a system, the cache will control the CPU cacheability function.

When the Simulcache is operating in an MS400 system in write-back mode, the bus utilization that the MS400 sees will average only about 10-15%. This number is quite low, as normal i486 CPU bus utilization is about 50%, and 386 CPU bus utilization averages 85%. This means that sophisticated bus master cards may be added to the AT bus with no performance degradation due to bus saturation. Such master cards may access the system memory and I/O resources usually without colliding with the CPU. The Simulcache may continue to service CPU read/write hits while the master card accesses are occurring.

3.4.1 DRAM Memory Cycles

The majority of cycles will be destined for the DRAM array. The memory controller in the MS400 is very flexible, with a variety of RAS# and CAS# timings which are selected and controlled by bits in control registers. These variable timings allow optimization of the DRAM array, supporting either high speed DRAM's for performance or low speed DRAM's for cost-effectiveness. At the completion of each transfer to/from the DRAM's, SBRDY# will be asserted.

The MS400 allows a high-performance memory subsystem to be built. For best possible performance, this memory design supports bi-directional bursting (bursting of both read and write data). Page-mode DRAM's are supported by the MS400.

Memory for the MS400 data array are organized into 32-bit wide banks. The MS400 supports from one to four banks of memory in a system. If two or more banks are used, these banks are automatically interleaved in order to support bursting. Bursts consist of 16-byte quantities of data, or four transfers across the 32-bit local CPU bus. These four transfers correspond to the cache line size in both the i486 CPU and the Simulcache controller.

The MS400 supports the enhanced bus interface offered by the Mosel Simulcache solution. This bus is an enhanced i486 CPU bus, with extra performance and enhanced functionality. Due to the architecture of the Simulcache, which isolates the CPU from the remainder of the system, the system-side interface of the cache does not change, with use of either a 386 or an i486 CPU in the system. Even in the 386 CPU configuration, the MS400 and Simulcache will communicate with an i486 CPU style bus. Use of this bus adds additional performance to 386 CPU systems.

For best performance, the Simulcache bypasses data from the system to the CPU in only 5 nS asynchronously. The MS400 DRAM memory sub-system is built to operate with the additional data setup time required because of this bypass operation.

Since the i486 CPU and the MOSEL Simulcache can both be included in an MS400 system, the combination of both cache devices will drastically lower page hit percentages. Page mode systems employed in a 486 CPU system with external cache actually have a negative effect on performance due to the penalty imposed during page miss operations. Due to this nature of operation, the MS400 DRAM controller does not support full page-mode operation between cycles.

3.4.2 AT Cycles

For cycles which are to/from the AT bus, the MS400 will start its internal AT state machines.

The MS400 provides the state machines that control all bus accesses and procedures. The control signals interfacing with the AT bus are 100% compatible with an IBM PC/AT.

The AT state machine controls the AT bus cycle and generates all bus command. The AT state machine gains control of the buses when the current cycle is not a local cycle. The AT bus clock is programmable; the state machine controls the duration and associated wait states of AT bus command. Both 8- and 16-bit I/O devices are supported on the I/O channel.

The MS400 supports 8-, 16-, and 32-bit transfers between the CPU/cache and 8- or 16-bit memory and I/O peripherals which exist on the AT bus. Mis-aligned transfers are supported by the MS400. Byte swapping will be performed if the transfer requests data from a peripheral device which is not physically connected to the requested bytes. Since the AT bus is only 16 bits wide while the CPU and cache SPD data buses are 32-bits in width, Dword (32-bit) requests from the CPU will be generate multiple accesses to the AT bus. These multiple accesses will be held in latches and assembled by the MS400 before BRDY# is returned to the CPU. CPU Dword writes will be dis-assembled, creating multiple write cycles to the AT bus.

MS400

SECTION 4. PROGRAMMING MODEL & REGISTER SET

4.1 Programming And Reading Registers

The MS400 has two types of registers: those registers that are contained within the standard peripheral devices (8237, 8254, 8259), and non-standard configuration registers to support and control the unique functions of the MS400.

DESCRIPTION OF MS400 CONFIGURATION REGISTERS

The address range for the configuration registers of the MS400 is from 22H to 23H in the I/O address space. Within this two byte range, each register resides inside the MS400 at a unique index address.

Registers in the MS400 are accessed by a two-step indexing method. First, the index address corresponding to the register to be read or written is written to I/O port 22H. Next, the contents of that register can be read or written by performing a read or write cycle to I/O port 23H.

All other index addresses at ports 22H and 23H are reserved by MOSEL. Writing to undefined index addresses may result in unpredictable operation.

4.2 Register Bit Descriptions MS400 Configuration Registers

REGISTER AT INDEX 00H

BITS FUNCTION

- 7 Remap Enable—When set to '1', an additional 384k of DRAM memory appears at the top of physical memory.
- 6 BIOS Size—When set, 128 kBytes of BIOS is selected. If cleared, 64 kBytes is selected.
- 5 BIOS Location—When set, BIOS ROM appears just below the 16 MByte address limit. Otherwise, the BIOS ROM appears only at the top of the first MByte, and at the top of the 4 GByte address space.

- 4,3 DRBANKS—These bits indicate how many DRAM banks are installed, as shown below:
 - 00: 1 bank
 - 01: 2 banks
 - 10: 3 banks
 - 11: 4 banks
- 2-0 DRSIZE— These bits indicate the size of the installed DRAM modules.

	Banks 0,2	Banks 1,3
000:	256 kBytes	256 kBytes
001:	256 kBytes	1 MByte
011:	1 MByte	1 MByte
010:	256 kBytes	4 MBytes
110:	N/A	N/A
111:	N/A	N/A
101:	4 MBytes	4 MBytes
100:	1 MByte	4 MBytes

REGISTER AT INDEX 01H

BITS FUNCTION

- 7-3 Reserved
- 2 Short tRP—When set, RAS# precharge of two clocks is selected. When cleared, precharge is three clocks.
- 1,0 DRAM Timing—These two bits control the DRAM cycle RAS# and CAS# timing:
 - 00: tCAS=2, RAS=1.5
 - 01: tCAS=1.5, tRAS=1.5
 - 10: tCAS=1.5, tRAS=1
 - 11: tCAS=1, tRAS=1

The combination of these two bits controls the number of clocks in which DRAM cycles occur. The number shown above for the RAS# control bit signal indicates the number of CPU T-states which elapse between the assertion of the RAS# signal and the assertion of the CAS# signals (the RAS# to CAS# delay). The CAS# number shown above indicates the CAS# access time, or the number of CPU T-states which CAS# will stay asserted.

REGISTER AT INDEX 02H

BITS FUNCTION

7	SHCCWD—Enable shadow RAM in 0CC000h-0CFFFFh area. 1: Write disable 0: Write enable
6	SHC8WD—Enable shadow RAM in 0C8000h-0CBFFFh area. 1: Write disable 0: Write enable
5	SHC4WD—Enable shadow RAM in 0C4000h-0C7FFFh area. 1: Write disable 0: Write enable
4	SHC0WD—Enable shadow RAM in 0C0000h-0C3FFFh area. 1: Write disable 0: Write enable
3	SHCCRE—Enable shadow RAM in 0CC000h-0CFFFFh area. 1: Read enable 0: Read disable
2	SHC8RE—Enable shadow RAM in 0C8000h-0CbFFFh area. 1: Read enable 0: Read disable
1	SHC4RE - Enable shadow RAM in 0C4000h-0C7FFFh area. 1: Read enable 0: Read disable
0	SHC0RE—Enable shadow RAM in 0C0000h-0C3FFFh area. 1: Read enable 0: Read disable

REGISTER AT INDEX 03H

BITS FUNCTION

7	Reserved
6	SHFWD—Enable shadow RAM in 0F0000h-0FFFFFh area. 1: Write disable 0: Write enable
5	SHEWD—Enable shadow RAM in 0E0000h-0EFFFFh area. 1: Write disable 0: Write enable
4	SHDWD—Enable shadow RAM in 0D0000h-0DFFFFh area. 1: Write disable 0: Write enable
3	Reserved
2	SHFRE—Enable shadow RAM in 0F0000h-0FFFFFh area. 1: Read enable 0: Read disable
1	SHERE—Enable shadow RAM in 0E0000h-0EFFFFh area. 1: Read enable 0: Read disable
0	SHDRE—Enable shadow RAM in 0D0000h-0DFFFFh area. 1: Read enable 0: Read disable

<u>WD</u>	<u>RE</u>	<u>DESCRIPTION</u>
0	0	Writes go to RAM, reads to AT bus. This setting is used to copy from ROM to RAM.
0	1	Reads and writes both go to RAM. In this mode, the RAM is a scratch pad.
1	0	All reads and writes directed to AT bus.
1	1	Reads go to RAM, writes go to AT bus (which is used to disable writing). This setting is used after the ROM has been copied to RAM.

MS400

REGISTER AT INDEX 04H

BITS FUNCTION

- 7-4 Reserved
- 3 Fail-safe timer enable.
- 2-0 AT CLK select—These two bits control the AT bus clock S1CLK frequency:
 - 000: CLKIN/6
 - 001: CLKIN/5
 - 010: CLKIN/4
 - 011: CLKIN/3
 - 100: CLKIN/2.5
 - 101: CLKIN/2

REGISTER AT INDEX 05H

BITS FUNCTION

- 7-4 Reserved.
- 3 Turbo/Normal#. When this bit is cleared, it enables the operation of the second 8254-compatible timer. This timer is used to slow system operation by assertion of the HOLD input. When this bit is set to one, it disables the timer and forces system performance to turbo (full) speed. See section 5.8.3, Turbo/Normal Operation, for more discussion. This bit has a default setting of 1.
- 2-0 Reserved.

REGISTER AT INDEX 06H

BITS FUNCTION

- 7 Reserved
- 6 Fast Gate A20. When this bit is set to one, the address bit A20 from the CPU is passed through unchanged for DRAM and system cycles. When this bit is cleared to zero, A20 is forced to a value of zero. The default setting for this bit is high.
- 5-0 Reserved

REGISTER AT INDEX 07H

BITS FUNCTION

- 7 Fast Reset. When this bit is set, fast reset of the CPU will be utilized. When this is cleared, no fast reset will occur.
- 6-0 Reserved

REGISTER AT INDEX 08

FUNCTION -

This register is used to clear parity errors. The MS400 will assert PARERR# (parity error) low to indicate that an error has occurred. Usually an NMI is triggered in response to parity errors. As part of the BIOS routine to respond to such errors, the BIOS should clear the parity error latch within the MS400. Any write operation to this register will clear the parity error latch. Bit 7 of this register can be read to determine the status of the parity error latch.

REGISTER AT INDEX 09H

BITS FUNCTION

- 7-1 Reserved.
- 0 MS441 Present. This bit is set by the MS400 to signal the presence of the Simulcache. When cleared to '0', no Simulcache is present in the system. This bit is read-only. Any write to this bit will be ignored.

REGISTER AT INDEX 0AH

<u>BITS</u>	<u>FUNCTION</u>
7-3	Reserved.
2-0	Clock Frequency Select. These bits may be used in order to reduce the frequency of the CLK1 and CLK2 outputs from their normal frequencies.
	<u>BITS</u> <u>CLK1/CLK2</u>
000:	NORMAL
001:	NORMAL/2
010:	NORMAL/4
011:	NORMAL/8
100:	NORMAL/16

REGISTER AT INDEX 0BH

<u>BITS</u>	<u>FUNCTION</u>
7-6	Reserved.
5	LBADIS-When set to '1', the LBA# input is disabled, and has no effect upon MS400 operation. When cleared to '0', LBA# functionality is enabled.
4-0	Reserved.

REGISTER AT INDEX 0CH

<u>BITS</u>	<u>FUNCTION</u>
7-5	Reserved.
4	Combined ROM. When set to '1', 64k Bytes of BIOS and 64k Bytes of Video ROM are selected. If this bit is cleared to '0', 128k Bytes of system BIOS is selected.
3-0	Reserved.

REGISTER AT INDEX 0DH

<u>BITS</u>	<u>FUNCTION</u>
7	IOWAIT-When set to '1', two additional S1CLK periods will be inserted between two consecutive I/O accesses in order to provide sufficient device recovery time.
6-0	Reserved.

REGISTER AT INDEX 0EH

<u>BITS</u>	<u>FUNCTION</u>
7	Extended Cacheable. When set, memory appearing above 1M will not be cached by the 486 CPU.
6-0	Reserved.

MS400

AT Standard Peripheral Register Addresses & Functions

ADDRESS MAP FOR MS400 STANDARD PERIPHERAL REGISTERS

000-00FH	8237-1 (DMA Controller)
020-21H	8259-1 (Interrupt Controller)
024-3FH	8259-1 (Interrupt Controller)
040-043H	8254-X (Counter)
060H	Reset, IRQ1, AND IRQ12
061-06FH	Port B; Odd Only
070H	NMI Logic
070-071H	RTC Access Ports
080-08FH	DMA Page Registers
0A0-0BFH	8259-2 (Interrupt Controller)
0C0-0DEH	8237-2 (DMA Controller); Even Only
0F0-0FH	Coprocessor CS and Reset

8237 DMA CONTROLLER REGISTER ADDRESSES

<u>ADDRESS</u>	<u>REGISTER FUNCTION</u>
080	Reserved
081	DMA Channel 2 Low Page Register
082	DMA Channel 3 Low Page Register
083	DMA Channel 1 Low Page Register
084	Reserved
086	Reserved
087	DMA Channel 0 Low Page Register
088	Reserved
089	DMA Channel 6 Low Page Register
08A	DMA Channel 7 Low Page Register
08B	DMA Channel 5 Low Page Register
08C	Reserved
08D	Reserved
08E	Reserved
08F	Refresh Low Page Register

<u>DMA2</u>	<u>DMA1</u>	<u>REGISTER FUNCTION</u>
0C0	000	Ch 0 Base & Current Address Register
0C2	001	Ch 0 Base & Current Word Count Reg
0C4	002	Ch 1 Base & Current Address Register
0C6	003	Ch 1 Base & Current Word Count Reg
0C8	004	Ch 2 Base & Current Address Register
0CA	005	Ch 2 Base & Current Word Count Reg
0CC	006	Ch 3 Base & Current Address Register
0CE	007	Ch 3 Base & Current Word Count Reg
0D0	008	Read Status Reg/Write Command Reg
0D2	009	Write Request Register
0D4	00A	Write Single Mask Register bit
0D6	00B	Write Mode Register
0D8	00C	Clear Byte Pointer flip-flop
0DA	00D	Read Temp Reg / Write Master Clear
0DC	00E	Clear Mask Register
0DE	00F	Write All Mask Register Bits

8259 REGISTER WRITE OPERATIONS

<u>INT1</u>	<u>INT2</u>	<u>REGISTER FUNCTION</u>
020	0A0	Write ICW1
021	0A1	Write ICW2
021	0A1	Write ICW3
021	0A1	Write ICW4 (If Needed)
021	0A1	Write OCW1
020	0A0	Write OCW2
020	0A0	Write OCW3

8259 REGISTER READ OPERATIONS

<u>INT1</u>	<u>INT2</u>	<u>REGISTER FUNCTION</u>
020	0A0	Int Request Reg, ISR or Poll Command
021	0A1	Interrupt Mask Register

8254 COUNTER/TIMER REGISTER ADDRESSES

<u>ADDRESS</u>	<u>REGISTER FUNCTION</u>
040	System Clock (Timer 1 Counter 0)
041	Refresh Request (Timer 1 Counter 1)
042	Speaker Tone (Timer 1 Counter 2)
043	Timer 1 Control Word Register
048	Fail-Safe Timer (Timer 2 Counter 0)
049	Unused (Timer 2 Counter 1)
04A	Reserved (Timer 2 Counter 2)
04B	Timer 2 Control Word Register

4.3 Default Register Values

After a reset to the MS400, the following values will be contained in the MS400:

Register 00H (Default value = 0000 0000). This default value disables memory remap of the 384 kBytes. Also, 64k bytes of BIOS is assumed to be installed. The BIOS ROM is disabled in the 16th Mbyte of address space, and appears only at the top of the first Mbyte of memory and at the top of 4G memory. Also, the default indicates that 1 bank of DRAM's are assumed to be installed. The bank size is set to 256k DRAM's. With these two settings, the MS400 system will boot reliably regardless of how many banks are actually installed and the actual bank size. After memory sizing has occurred, the BIOS should re-configure these bits to indicate the actual system configuration.

Register 01H (Default value = xxxx x000). RAS# precharge is set to 3 CPU T-states. Also, the RAS# and CAS# bits which control the speed of DRAM accesses are set to perform the slowest possible accesses. With these assumptions, the MS400 system will boot with slow DRAM's. The BIOS should provide a configuration menu in which the user can indicate that fast DRAM devices are installed. In this method, the BIOS can re-configure this register to allow fast DRAM accesses to occur.

Register 02H (Default value = 0000 0000). This register controls the shadowing in the C segment (C0000-CFFFFH). The default setting will read from the AT bus for these addresses, but cycles to this address range will write to the DRAM array. This allows easy setup of shadowing.

Register 03H (Default value = x000 x000). This register controls the shadowing in the D, E, and F segments (C0000-FFFFFH). The default setting will read from the AT bus for these addresses, but cycles to this address range will write to the DRAM array. This allows easy setup of shadowing.

Register 04H (Default value = xxxx 0000). The default setting for this register leaves the fail-safe timer disabled after reset. In addition, the AT bus clock S1CLK is set to a setting of CPUOSC/5.

Register 05H (Default value = xxxx 1xxx). The default setting for this register indicates that turbo operation is selected.

Register 06H (Default value = x1xx xxxx). The default setting of this register enables the Fast Gate A20 feature of the MS400.

Register 07H (Default value = 0xxx xxxx). The default setting of this bit disables the Fast Reset feature of the MS400.

Register 08H. (Default value = 0xxx xxxx). The default setting of this bit indicates no parity error condition upon reset of the MS400.

Register 09H. There is no default setting for this register. Bit 0 will be either set or cleared, depending on whether or not a Simulcache is present in the MS400 system.

Register 0AH. (Default value = xxxx x000). The default setting for this register sets the CLK1 and CLK2 frequencies to their normal settings.

Register 0BH. (Default value = xx0x xxxx). The default setting for bit 5 of this register enables the local bus functionality of the MS400.

Register 0CH. (Default value = xxx0 xxxx). The default setting of bit 4 of this register disables the combined BIOS function.

Register 0DH. (Default value = 0xxx xxxx). The default setting of bit 7 of this register disables I/O recovery time

Register 0EH. (Default value = 0xxx xxxx). The default setting of bit 7 of this register allows extended memory to be cached.

MS400

SECTION 5. FUNCTIONAL DESCRIPTION

5.1 MS400 Clock Circuitry

There are several clock signals present in an MS400 system:

- 1) CLKOSC
- 2) CLK1
- 3) CLK2
- 4) CLKIN
- 5) S1CLK
- 6) OSC

CLKOSC (Clock Oscillator) is the basic clock input to the MS400. This signal is driven from an external oscillator. If the MS400 is in 486 CPU mode, this signal will be a single frequency signal (i.e. 33 MHz CLKOSC input for a 33 MHz MS400 system). However, if the MS400 is configured in the 386 CPU mode, this input signal will be double frequency (66 MHz for a 33 MHz MS400 system).

From CLKOSC, the MS400 generates CLK1, the Single-Frequency Clock output, and CLK2, the Double-Frequency clock output. CLK1 is always one-half the frequency of the CLKOSC input (i.e. 16.6 MHz for a 33.3 MHz 486 CPU system). For 386 CPU systems, CLK1 should be connected to the MS400 CLKIN input.

To conserve system power, control register 0AH is available in order to dynamically reduce the frequencies of CLK1 and CLK2. CLK1 and CLK2 may be reduced by a divisor which may be from 1 to 32.

The functionality of the CLK1 signal allows split clock 486 CPU-based systems to be designed. Such systems operate with the CPU/cache subsystem at high frequency, while the chipset operates at half that frequency. To implement a split clock 486 CPU based system, CLK1 should be used for the MS400 CLKIN input instead of CLK2.

CLK2 is always the same frequency as the CLKOSC input. CLK2 is usually connected to the CPU clock input. For most MS400/486 CPU systems, CLK2 is also connected to the MS400 CLKIN input. For system operation with split clocks, refer to the CLK1 description.

CLKIN is used as the clock signal for the internal MS400 state machines. CLKIN is always the same frequency as the system (i.e. CLKIN = 33 MHz for either 386 or 486 33 MHz systems). For most 486 systems, CLKIN should be sourced from CLK2, while

386 CPU systems will use CLK1 for this signal.

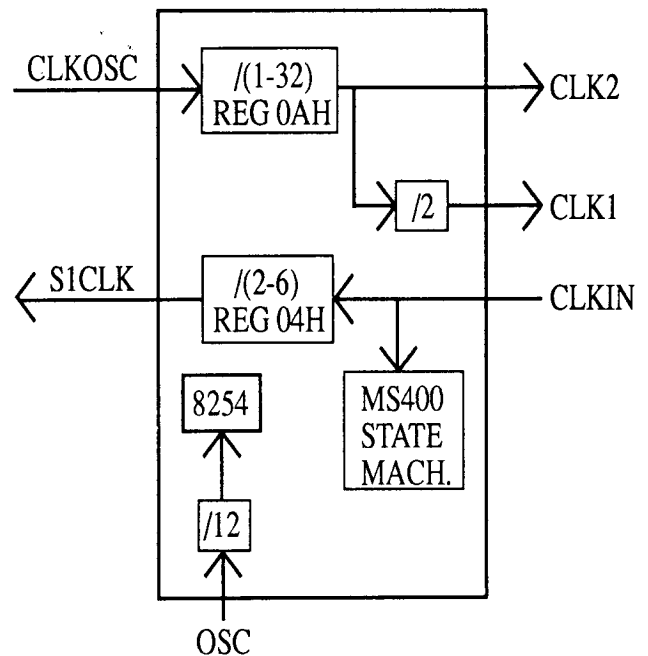
The AT system bus clock S1CLK is derived from this signal.

CLKOSC, CLK1, and CLK2 are not internally used by the MS400. If a system designer so chooses, CLKIN may be generated externally and input to the MS400. CLK1 and CLK2 are intended only as aids to the system designer, and CLKOSC, CLK1, and CLK2 may be unconnected signals without impairing system operation. CLK1 and CLK2 are internally matched inside the MS400 such that the skew is minimized as much as possible.

S1CLK (System Clock) is the basic clock frequency for the AT bus. This signal is generated synchronously from the CLKIN input to the MS400, and as such will be some fraction of CLKIN's frequency. S1CLK can be either 1/2, 1/2.5, 1/3, 1/4, 1/5, or 1/6 the frequency of CLKIN, as chosen by three bits in control register 04H. In any configuration, S1CLK should not exceed 8.33 MHz for compatibility.

The Oscillator signal is input to the MS400 from an external oscillator. This signal has a frequency of 14.31818 MHz. The Oscillator input is internally divided by 12 inside the MS400, and is used as an input to the 8254 compatible timer/counter peripherals within the MS400.

CLK2 is basically a buffered version of CLKOSC, and will retain the same duty cycle. CLK1 will have a duty cycle of 50%, as will S1CLK.



5.2 CPU/Local Bus Support

The MS400 will support either the 386 CPU or the i486 CPU. For both 386 and i486 CPU systems, the MS400 is designed to work either with or without the Simulcache.

5.2.1 CPU Support

BUS CYCLE DEFINITIONS

M/IO#, D/C#, and W/R# from the CPU/cache are the primary bus cycle definition signals for the MS400. These signals are always driven valid in the T1 bus state, as ADS# is asserted. M/IO# distinguishes between memory and I/O cycles. D/C# distinguishes between data or code accesses. W/R# indicates write or read operations.

Note that 386 bus cycle encodings are somewhat different than those of the 486 CPU. The MS400 is quite flexible, and will interpret bus cycle encodings to be that of the CPU operating in the system. The interpretation of 486 CPU encodings is shown below:

M/IO#	D/C#	W/R#	Meaning
0	0	0	Interrupt Acknowledge
0	0	1	Special Cycles
0	1	0	I/O Read
0	1	1	I/O Write
1	0	0	Memory Code Read
1	0	1	Reserved
1	1	0	Memory Data Read
1	1	1	Memory Data Write

486 SPECIAL CYCLE DEFINITIONS

In addition to the M/IO#, D/C#, and W/R# signals, the MS400 supports the special cycles of the i486 CPU. These special cycles indicate that certain conditions have occurred internally or certain instructions have been executed. As shown above, the special cycles are indicated by the pattern of M/IO# = 0, D/C# = 0, and W/R# = 1. When a special cycle is indicated, the MS400 uses the byte enable signals to determine which special cycle is occurring:

BE3#	BE2#	BE1#	BE0#	Cycles
1	1	1	0	Shutdown
1	1	0	1	Flush
1	0	1	1	Halt
0	1	1	1	Write Back

The SEL486 input of the MS400 is used to select the CPU operating in the system. SEL486 should be strapped low for 386 CPU mode, and high for i486 CPU operation.

5.2.2 CPU Clock Switching

For power-sensitive systems, the MS400 adds support for power efficiency. When a system is powered on but not currently in use, power may be conserved by reducing the operating frequency of the CPU.

Three bits in index register 0AH are used to reduce the CLK1 and CLK2 signal frequency. These bits may be dynamically toggled during normal system operation.

BITS	FUNCTIONALITY
000:	CLKOSC/1
001:	CLKOSC/2
010:	CLKOSC/4
011:	CLKOSC/8
100:	CLKOSC/16

Note that by reducing CLK1 and CLK2, the AT bus clock S1CLK signal frequency will be accordingly reduced by the same ratio as above. Very slow AT clock frequency may occur as a result of the above clock divider and the AT clock divider.

If the 486 CPU is operating in a system, care must be taken when manipulating these bits. Large frequency changes should be induced in the smallest possible increments. For example, switching from 33 MHz to 8.25 MHz should occur in two steps, the first reducing the frequency to 16.5 MHz and the second to fix the frequency at 8.25 MHz. This precaution is to avoid any mis-operation of the internal phase-locked loop clock circuitry internal to the 486 CPU.

MS400

5.2.3 Local Bus Access Support

Designers may wish to customize systems by incorporating non-standard I/O resources on the local CPU bus. Such resources would then operate at the higher CPU frequency for optimized performance, rather than at the system bus frequency. The MS400 can support such customizations.

The LBA# (Local Bus Access) input of the MS400 is used to support such resources. The attached resource should monitor CPU bus cycles on a per-cycle basis, dynamically informing the MS400 that a given cycle is intended for the local resource.

The MS400 ignores the LBA# input in the T1 state. The MS400 will assume that all driven cycles are normal system cycles, and will begin to perform the indicated cycle normally. At the middle of the first T2 state (upon the falling edge of the CLKIN signal), LBA# will be sampled. A high assertion will indicate that the cycle will also complete normally. However, if LBA# is sampled low in the middle of the first T2 state, a local cycle is indicated. The assertion of LBA# must be synchronized to the MS400 CLKIN signal, meeting setup and hold times. In response to this indication, the MS400 will immediately abort the current cycle, resetting its state machines to idle states and leaving all output buffers disabled.

LBA# will only be sampled for cycles not to the local DRAM array. Assertion of LBA# during DRAM read/writes will be ignored, with the cycle being completed normally by the MS400. Local bus resources can not exist in the DRAM address space.

After LBA# is signalled to the MS400, the local resource owns the bus cycle and is responsible for completing the cycle. The MS400 will immediately float both its RDY# and BRDY# outputs upon sampling LBA# asserted. The MS400 will continue to

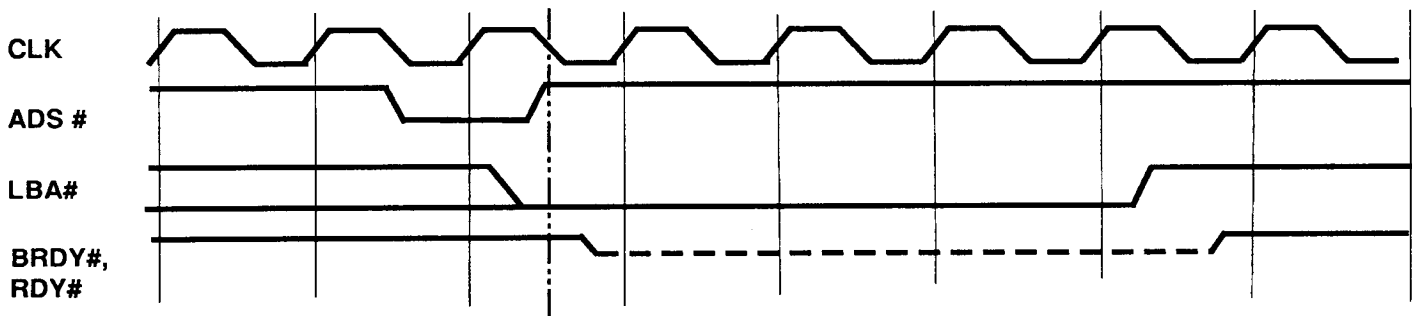
float these signals in order to allow the local bus resource to complete the cycle. RDY# or BRDY# must be returned by the local resource in order to complete the cycle. Pull-ups for both RDY# and BRDY# should be included in a system design, to prevent any cycle from terminating prematurely.

The middle of the first T2 state is the only window available to inform the MS400 of the occurrence of a local bus cycle. If LBA# is not sampled low at this point, the cycle will complete normally. In addition, the LBA# function may be disabled if no local resources are present in a system. Bit 5 of the register at index 0BH may be set to '1' in order to disable LBA# functionality. If no resource is available to drive LBA#, this input should be pulled inactive to prevent spurious activity.

Once LBA# has been driven active in order to designate a local bus cycle to the MS400, it must remain asserted for the remainder of the cycle. In the clock after RDY# or BRDY# is asserted, LBA# may then be de-asserted.

RDY# and BRDY# will remain floated as long as LBA# continues to be asserted. When LBA# is de-asserted to the MS400, it will asynchronously begin to drive RDY# and BRDY#.

When the Simulcache is operating in an MS400 system, any local bus resources will normally appear on the cache host port bus; such resources will be supported by the Simulcache, and will be invisible to the MS400. In these cases, the LBA# input of the MS400 should be disabled as described above. However, local bus resources may be designed to appear on the system side of the cache, or two levels of resources may exist in a system. In such a case, the MS400 LBA# input is used as described above.



LBA# FUNCTIONALITY

5.2.4 CPU/Simulcache Reset

Note that two Reset signals are present in an MS400 system. Reset should be used for only the CPU. For downward software compatibility with the 80286, some software requires the CPU to be reset independently from the remainder of the system. Reset supports this functionality. RSTDRV, however, derived exclusively from the MS400 PWRGOOD input signal and is driven to the entire system.

Note that if the MOSEL Simulcache is present in an MS400 system, the Simulcache reset input should be connected to the RSTDRV signal and not to the CPU reset. Since the Simulcache is a write-back cache and will contain dirty data during normal operation, it should not be reset each and every time that a CPU reset occurs.

5.2.5 Cacheability Support

The MS400 controls the internal cache of the 486 CPU on a cycle-by cycle basis. Standard configurations for most systems dictate that DRAM memory space is cacheable inside the 486 CPU, while reserved AT memory space is not cacheable. The MS400 offers support for the internal cache in a straightforward manner. Since the 386 CPU contains no internal cache, no considerations are necessary for MS400/386 CPU systems.

As the address for each cycle is driven out from the CPU, the MS400 will decode the address. Any cycle which is to/from the local DRAM's will cause the MS400 to indicate a cacheable cycle to the CPU; otherwise, non-cacheable cycles will occur.

As an example, assume that four Mbytes of memory are installed in a system. Any address from 0-640K range, or from 1Mbyte-4Mbytes will occur as a DRAM cycle and thus will be deemed cacheable by the MS400. Any address from 640K-1M, or above 4Mbytes will be deemed non-cacheable, and so signalled to the CPU.

The KEN# (cache enable) signal is the output of the MS400 cacheability logic, and should be connected to the CPU. KEN# will become valid to the CPU late in the T1 state, or early in the first T2 state, depending on the system operating frequency.

Note that KEN# is an unconditional and unqualified decode of the CPU address lines, and indicates that cacheability is possible, and not that a cache line fill must necessarily occur. For instance, KEN# will be driven during I/O cycles, write cycles, idle cycles, PCD cycles, and cycles when the internal 486 cache

is software-disabled. In these cases (and others), the state of KEN# will be ignored by the CPU.

KEN# will always be asserted as described above for addresses in the 0-640 kByte address range. There is no MS400 control bit to disable caching in this range. To disable caching for this address range, the CD and NW bits present in Control Register 0 of the 486 CPU may be used.

Addresses above 1 MByte may be designated as non-cacheable addresses. This is accomplished by setting bit 7 in control register 0EH in the MS400. If this bit is set, addresses above 1 MByte will never activate KEN#.

Inclusion of the Simulcache in an MS400 system will obviate the need for the MS400 KEN# signal. In such cases, the Simulcache will supply the CPU KEN# signal. The Simulcache cacheability logic is quite flexible, offering four address regions which may be declared cacheable, non-cacheable, or write-protected.

The system designer may decide to add additional cacheability logic if the Simulcache is not present. Such logic should intercept the KEN# output, and modify it accordingly. Such modifications should be performed with the assurance that KEN# can be driven valid one clock before the first RDY# or BRDY# is returned, due to the CPU KEN# specification. Even at high system frequencies, such a modification can be supported with medium-speed logic devices.

5.3 DRAM Memory Controller

The MS400 features an optimized DRAM memory controller, offering the highest performance for both 386 and i486 CPU-based systems, while still retaining flexibility to allow the user to tailor the system to meet his/her needs.

The MS400 memory controller features bi-directional bursting, with optimized write posting logic. Up to 8 bytes of write data can be externally latched and posted by the MS400. To support write posting and read bursts, the MS400 will utilize external standard latches.

For additional performance, the MS400 memory controller contains two-phase logic. To fine-tune system performance to the speed of the installed DRAM's, the memory control signals may be activated during both phase 1 and phase 2 of clock periods.

Many memory configurations are available with the MS400. These configurations are accomplished through use of the programmable register set which was previously described. Slower DRAM devices can be accommodated through configuring the RAS# and CAS# access clocks. The number of clocks from assertion of RAS# to assertion of CAS# can be set to either 1.0 or 1.5 CPU T-states, while the number of clocks for CAS# access time can be either 1.0, 1.5, or 2.0 T-states. In addition, the RAS# precharge time tRP is programmable to either two or three clocks.

The MS400 supports up to 64 Mbytes of memory, with use of 4 Mbit DRAM's. 256 Kbit and 1 Mbit DRAM's are also supported, once again through use of the programmable registers.

For cost efficiency, a one bank 32-bit wide memory system can also be supported. Through use of this interface, some DRAM components as well as TTL can be saved.

5.3.1 DRAM sizes supported

For system design flexibility, the MS400 can support either 256 K, 1 M, or 4 M DRAM's. The amount of DRAM installed, and the size is indicated to the MS400 by the use of register bits. Bits 4 and 3 at the index register 00H are used to indicate installed banks of DRAM, while bits 2 through 0 of the same register indicate the size of the DRAM's. From the settings of these bits, the MS400 internally calculates the bank address mappings and the top of physical memory. Possible DRAM bank configurations are shown below:

MS400 DRAM CONFIGURATIONS

<u>BK 0</u>	<u>BK 1</u>	<u>BK 2</u>	<u>BK 3</u>	<u>TOTAL</u>
256K				1 Mbyte
256K		256K		2 Mbyte
256K	256K	256K		3 Mbyte
256K	256K	256K	256K	4 Mbyte
256K	256K	1M		6 Mbyte
256K	256K	1M	1M	10 Mbyte
256K	256K	4M		18 Mbyte
256K	256K	4M	4M	34 Mbyte
1M				4 Mbyte
1M	1M			8 Mbyte
1M	1M	1M		12 Mbyte
1M	1M	1M	1M	16 Mbyte
1M	1M	4M		24 Mbyte
1M	1M	4M	4M	40 Mbyte
4M				16 Mbyte
4M	4M			32 Mbyte
4M	4M	4M		48 Mbyte
4M	4M	4M	4M	64 Mbyte

5.3.2 DRAM memory array mapping & organization

As previously described, the MS400 can support from one to four 32-bit banks of memory. Organization of the memory array for a four-bank subsystem is shown in Figure 2. Each bank is driven by a separate RAS# signal. Within the bank, nine bits each represent byte 0, 1, 2, and 3.

When two or more banks of memory are installed in a system, interleaving will be employed by the MS400 to support burst operations. Burst accesses will transfer 16 bytes of data (four Dwords), which will all reside within an aligned 16-byte boundary. For these four transfers, data will access two of the banks.

The CAS# signals are used to individually enable a byte during each transfer of a cycle. For single transfer cycles (indicated by BLAST# low for the first transfer), the CAS# signals for writes will be derived from the byte enables of the CPU/cache, while reads will force all CAS#'s low. However, for multiple transfer cycles, all CAS# signals will be forced low, regardless of the state of the byte enables.

Figure 3 shows how the memory map is supported by memory banks for one-bank to four-bank configurations.

01C	Bank 0	Dword 3
018	Bank 0	Dword 2
014	Bank 0	Dword 1
010	Bank 0	Dword 0
00C	Bank 0	Dword 3
008	Bank 0	Dword 2
004	Bank 0	Dword 1
000	Bank 0	Dword 0

One Bank

01C	Bank 2	Dword 3
018	Bank 0	Dword 2
014	Bank 2	Dword 1
010	Bank 0	Dword 0
00C	Bank 2	Dword 3
008	Bank 0	Dword 2
004	Bank 2	Dword 1
000	Bank 0	Dword 0

Two Banks

20000C	Bank 1	Dword 3
200008	Bank 1	Dword 2
200004	Bank 1	Dword 1
200000	Bank 1	Dword 0
1FFFFC	Bank 2	Dword 3
1FFFF8	Bank 0	Dword 2
1FFFF4	Bank 2	Dword 1
1FFFF0	Bank 0	Dword 0

Three Banks

01C	Bank 3	Dword 3
018	Bank 1	Dword 2
014	Bank 3	Dword 1
010	Bank 1	Dword 0
00C	Bank 2	Dword 3
008	Bank 0	Dword 2
004	Bank 2	Dword 1
000	Bank 0	Dword 0

Four Banks

MS400 BANK MEMORY MAPPING

One Bank Organization

One bank operation is supported by the MS400. With this setting, one clock burst accesses from the DRAM array are not possible. This is the default setting for the Bank Installed bits.

Two Bank Organization

When two banks of DRAM's are installed in an MS400 system, the banks are interleaved by doublewords (Dwords), or 4-byte quantities of data. If two banks are utilized, banks 0 and 2 should be installed. Two bank operation will offer a very high level of performance, as the two interleaved banks can support one-clock read and write bursts (with use of fast DRAM devices).

Three Bank Organization

DRAM memory organization for a three-bank MS400 system is also shown in Figure 3. Three banks of 256 Kbit DRAM's are installed, for a total of 3 Mbytes of memory. Banks 0 and 2 appear as a normal 2-bank system, are interleaved together by Dwords, and are mapped into the lower portion of memory, in the address space from 0 to 2 Mbytes. Bank 1 is logically mapped above the interleaved banks, and appears in the 2-3 Mbyte address range.

Cycles to the lower 2 Mbytes of memory occur exactly as previously described for a two-bank system. Cycles to the top 1 Mbyte of memory will occur as one-bank accesses.

This is the most efficient optimization for three banks. The interleaving of banks 0 and 2 allows them to support read and write burst operations. Moreover, placing the interleaved banks in the lower address space, which is more frequently used than higher memory, will allow these bursts to occur as often as possible.

Four Bank Organization

The four banks of the MS400 memory system are further organized into two sets. Set A corresponds to banks 0 and 1, while set B corresponds to banks 2 and 3. Set A maps into Dwords 0 and 2 of a line fill, while set B maps into Dwords 1 and 3. A burst transfer will always access one bank from set A and one bank from set B. The address bit 4 is used to select between which banks are accessed. If A4 is low, banks 0 and 2 will supply data, while A4 high selects banks 1 and 3.

MS400

5.3.3 DRAM Cycles

Reads

The MS400 supports a variety of read operations from the DRAM array. The basic RAS# and CAS# timings are controlled by register bits contained inside the MS400.

For DRAM read operations, the MS400 will latch the BLAST# input on the rising edge at the end of the first data transfer. The MS400 will always return four bytes of data for all transfers of the cycle, regardless of the byte enables. This is in order to comply with the i486 CPU specification, since byte enables may be invalid for the first transfer of a line fill (byte enables from the Simulcache are always valid). The cache/CPU will disregard any returned bytes for which the byte enable is not asserted.

If the Simulcache is present in a system, data returned for most read cycles will be bypassed on to the CPU in the same clock, as well as being latched inside the cache. This Bypass path will take about 5 ns. The MS400 system logic has taken this bypass into consideration, by returning data early enough to allow it to be passed on to the CPU in the same clock without violating the CPU read data setup time specification. In addition, BRDY# or RDY# from the MS400 will be passed on asynchronously through the Simulcache and on to the CPU. This bypass propagation delay has also been accounted for by the MS400, by presenting a valid BRDY#/RDY# signal early enough to be bypassed by the Simulcache.

Read Timing Control Bits

Bits 0 and 1 of the register at index 01H control the RAS# and CAS# access times, and bit 2 controls the RAS# precharge time. At reset, the MS400 assumes only one bank of 256K DRAM's is installed, and that the bank is of slow DRAM devices. These assumptions allow the MS400 to boot regardless of how much DRAM is actually installed in the system.

It is the responsibility of the system BIOS (usually through a setup program) to actually determine how much DRAM is installed, and of what speed the DRAM is. From this, the setup program can program the correct settings for the DRAM control bits. Note that incorrect settings for these bits can cause errors for DRAM operations, resulting in system failure.

RAS# and CAS# Access Timing Control Bits

The RAS# bit controls the RAS#-to-CAS# assertion delay (in CPU clocks). For DRAM cycles, increasing this variable will effectively increase the number of clocks available to meet the RAS# access time t_{RAC} for the DRAM array. RAS#-to-CAS# delays of either 1.0 or 1.5 T-states are available with the MS400.

The CAS# bit controls the amount of time (in CPU T-states) from the assertion of the CAS# signals to the completion of the transfer. The CAS# access time controlled by this bit satisfies the t_{CAC} access spec of DRAM devices. Essentially, this bit enables or disables one clock DRAM burst transfers. CAS# access times of 1.0, 1.5, or 2.0 CPU T-states are possible.

The RAS# and CAS# bits can be encoded in four possible configurations for the MS400. The number of clocks for single and burst read operations to occur is a direct result of these configurations. These encodings are interpreted by the MS400 as follows:

<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>
0:0	2,1,5	6-1-2-1	6-3-3-3
0:1	1,5,1,5	5-1-1-1	5-2-2-2
1:0	1,5,1	5-1-1-1	5-2-2-2
1:1	1,1	4-1-1-1	4-2-2-2

- A) BITS(1:0)
- B) t_{CAS#}, t_{RAS#}
- C) BURST READ TO TWO BANKS
- D) BURST READ TO ONE BANK

Note that the resulting right columns are listed as the fastest burst cycle achievable. Some burst cycles may operate slower (more wait states) than the listed numbers; for example, if the RAS# pre-charge time must be absorbed before an access to a bank can occur. However, DRAM cycles will never occur faster than the given numbers.

The above encodings give the basic available amount of clocks in which to satisfy RAS# and CAS# access specifications. The actual amount of time must take into consideration the operating frequency, RAS# and CAS# valid delays from the MS400, buffer delays, and data setup times.

Accounting for these factors, the following chart can be constructed which shows the speed of DRAM's necessary to operate for each frequency and RAS#/CAS# configuration of the MS400. To use the chart, simply cross-index the register bit configuration with the appropriate frequency to find the necessary

DRAM speed. Note that equivalent speed or faster DRAM's will allow operation, while slower DRAM's may cause system failure. This chart assumes standard buffer devices and four banks of installed DRAM's:

FREQUENCY (MHz)

<u>BITS</u>	<u>(1:0)</u>	<u>20</u>	<u>25</u>	<u>33</u>	<u>40</u>	<u>50</u>
1	1	80	70	60	NA	NA
1	0	100	80	70	60	NA
0	1	100	80	70	60	NA
0	0	120	100	80	70	60

RAS PRECHARGE CONTROL BIT

Similar to the RAS# and CAS# control bits just described, the MS400 operation of the RAS# precharge time is programmable. Bit 2 in the register at index 01H determines the number of T-states allotted to meet the tRP pre-charge specification of DRAM's. When this bit is set to one, two clocks will be absorbed for the pre-charge. However, at a high frequency of operation or when slow DRAM's are being used, an additional T-state may be required for pre-charge. By clearing this bit to zero, an additional (third) clock will be used for the pre-charge.

Single Read

Figure 4, below, details a single read transfer from the DRAM memory array. The MS400 monitors the ADS# (Address Status) output from the CPU/cache. The assertion (low) of ADS# on a rising CPU clock edge indicates the beginning of a bus cycle, and the occurrence of the CPU T1 bus state. By decoding the address signals and the M/IO# (Memory/IO) control line, the MS400 can determine that a DRAM cycle is indicated by the CPU. In addition, the control signals are examined to determine whether a read or write cycle is occurring. The byte enables are sampled in order to determine which bytes will be transferred for a cycle.

The assertion of BLAST# low from the CPU/cache indicates that only one transfer will occur for this cycle. The MS400 latches the state of BLAST# at the end of the first data transfer in order to determine how many bytes will be transferred for the current cycle. BLAST# low indicates the completion of the transfer, while BLAST# being sampled high informs the MS400 that 16 bytes will be transferred in a burst of four Dword transfers.

The MS400 has an internal address multiplexer (mux) in order to generate both the row and column addresses to the DRAM array. The row address consists of the high order address bits, while the column address is the lower order bits. At the beginning of each cycle, the mux is internally configured such that the row address from the CPU/cache flows through the mux and onto the MA memory address pins.

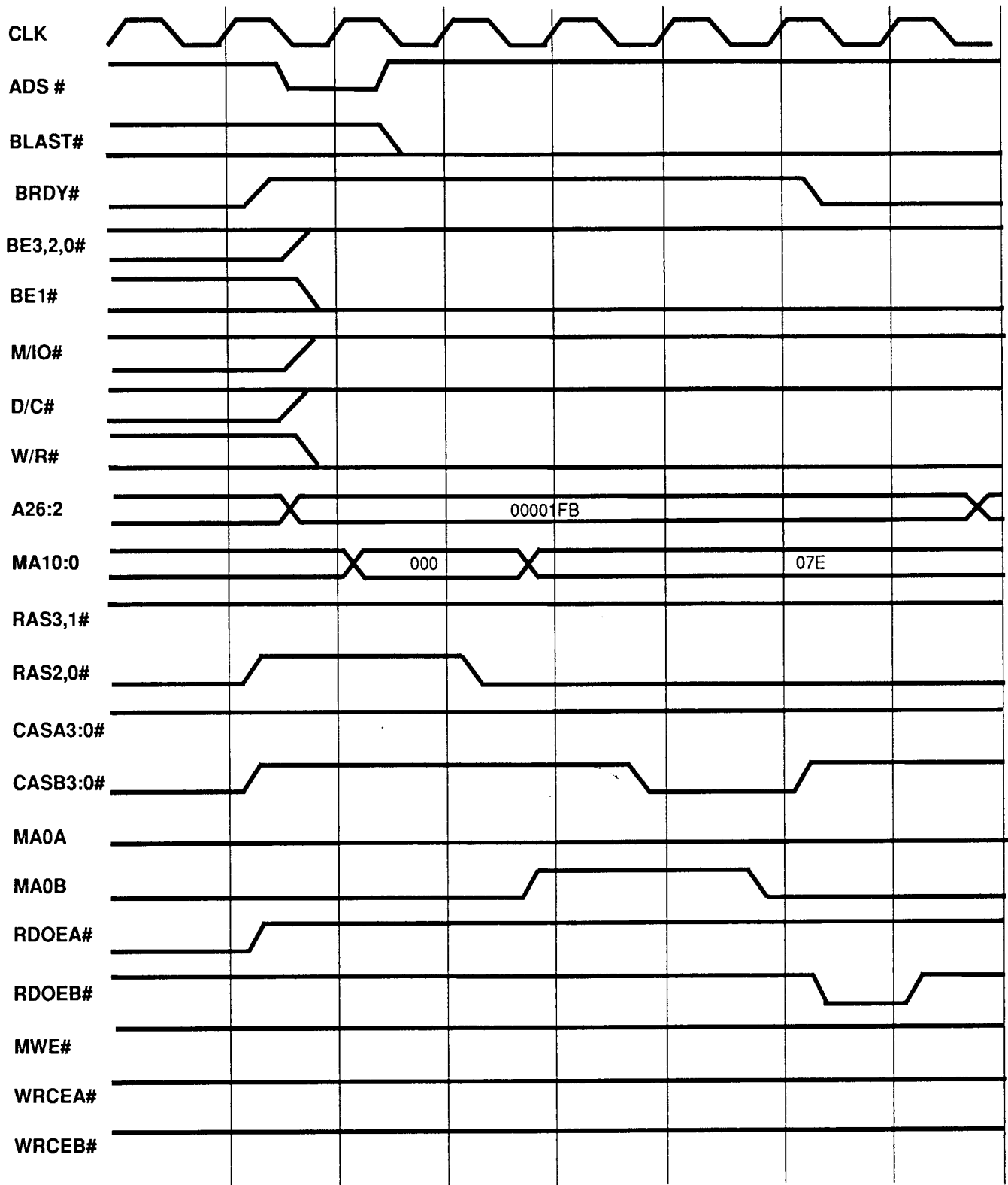
The flow-through described above allows the earliest possible assertion of the RAS# (row address strobe) signals. For some cycles, the MS400 will delay the beginning of the DRAM access (by withholding assertion of RAS#) to ensure that the RAS# pre-charge from the previous cycle has been met. If a cycle has previously just completed, this pre-charge time may not yet have been satisfied. The number of CPU T-states for RAS# pre-charge is selected by bit 2 in the register at index 01H. If the RAS# pre-charge has been satisfied, the MS400 is free to immediately begin the DRAM access by asserting RAS#.

Only one 32-bit bank will be accessed for single transfers, regardless of how many DRAM banks are installed in a system. RAS# is asserted to latch the row address appearing on the MA lines into the DRAM's. For all DRAM accesses, two of the four RAS# signals are always asserted, regardless of how many banks are installed in the system. The MWE# (Memory Write Enable) signal is connected to the Write Enables (WE#) inputs of standard DRAM's, to indicate read or write cycles to the DRAM devices. MWE# is derived from the CPU/cache W/R# control line. For read cycles, MWE# is held de-asserted (high).

After the assertion of RAS#, the MS400 switches control of its internal address mux in order to present the column address to the DRAM array. Once the column address has stabilized and the RAS#-to-CAS# delay has passed (as indicated by bits 0 and 1 from index register 01H), then CAS# is asserted to the DRAM array in order to latch the column address inside the DRAM's. For single read (and write) transfers, the CAS# signals asserted are derived directly from the byte enable controls of the CPU/cache. For example, BE0# (byte enable 0) activated low from the CPU/cache will cause the assertion of CAS0#.

After the CAS# access spec has been satisfied, the CAS# signals will be de-asserted by the MS400. As with the activation of the CAS# signals, the length of time (in CPU clocks) that CAS# will remain asserted is determined by the two control bits in register 01H. At the completion of the CAS# access time, valid read

MS400



SINGLE DRAM MEMORY READ

data is available at the DRAM outputs.

In the final clock of the DRAM access, the MS400 initiates three actions to terminate the cycle and return the read data to the CPU. First, the rising edge of the CAS# signals latches the data read into the latch devices. To accomplish this, the CAS# signal for each byte should be connected to the clock input of the appropriate latch device. The rising edge of CAS# will disable the DRAM output buffers at the same time that the read data is being latched.

Next, BRDY# is asserted low by the MS400 in order to terminate the cycle. Returning BRDY# instead of RDY# causes no complications, since the cache/CPU has indicated no burst is possible by asserting BLAST# low. BRDY# is driven active early enough in the T-state in order to be bypassed by the Simulcache to the CPU, if the cache is present in the system. Finally, either RDOEA# or RDOEB# is asserted low in order to gate the read data from the latches. The RDOE# signals should be connected to the output enables of the latches.

Burst Read Operations

However, for most cycles, the CPU/cache will desire to read multiple Dwords per bus cycle, in order to fill internal (16-byte) cache lines. Such burst read operations are also supported by the MS400. Burst reads can either be to the i486 CPU or the Simulcache. The 386 CPU does not support burst read operations.

During the first transfer of a burst read operation, the CPU/cache will de-assert BLAST# (high) in order to indicate that more transfers are necessary to complete the bus cycle. For such cycles to the DRAM array, the MS400 will support burst read accesses. Once the MS400 samples BLAST# low in T2, it will begin supplying four bytes of read data per clock until BLAST# is sampled low. Burst cycles of two, three, or four transfers can be supported by the MS400.

Bursting can only be supported by the MS400 if two or more banks are installed in a system. One bank systems can never support one-clock burst reads from the DRAM array. For systems which contain either two or four banks or DRAM's, accesses to any DRAM address can support bursting. For systems which contain three banks of DRAM's, most accesses will allow bursting from the memory array; however, bursting will not be allowed from some addresses. See section 5.3.2, DRAM Memory Array Organization, for more description of which addresses will allow burst accesses.

For DRAM read cycles which the MS400 can support with bursting, two banks of DRAM's will be used to supply the burst reads. Two banks will be utilized to provide the burst data, regardless of whether two, three, or four banks are contained in the system.

Figure 5 details a burst read access. From either the i486 CPU or the Simulcache a burst read operation begins like a single transfer as previously described by assertion of the ADS# output of either the CPU or the cache. The MS400 qualifies this signal as valid on the next rising clock edge (the first T2 state). At this point, the MS400 begins the DRAM cycle by driving the appropriate RAS# signals (either RAS0# and RAS2# or RAS1# and RAS3#). Note that the first transfer of a burst read can begin with either pair of banks, without incurring any system delay. The first transfer occurs as previously described for single transfer cycles.

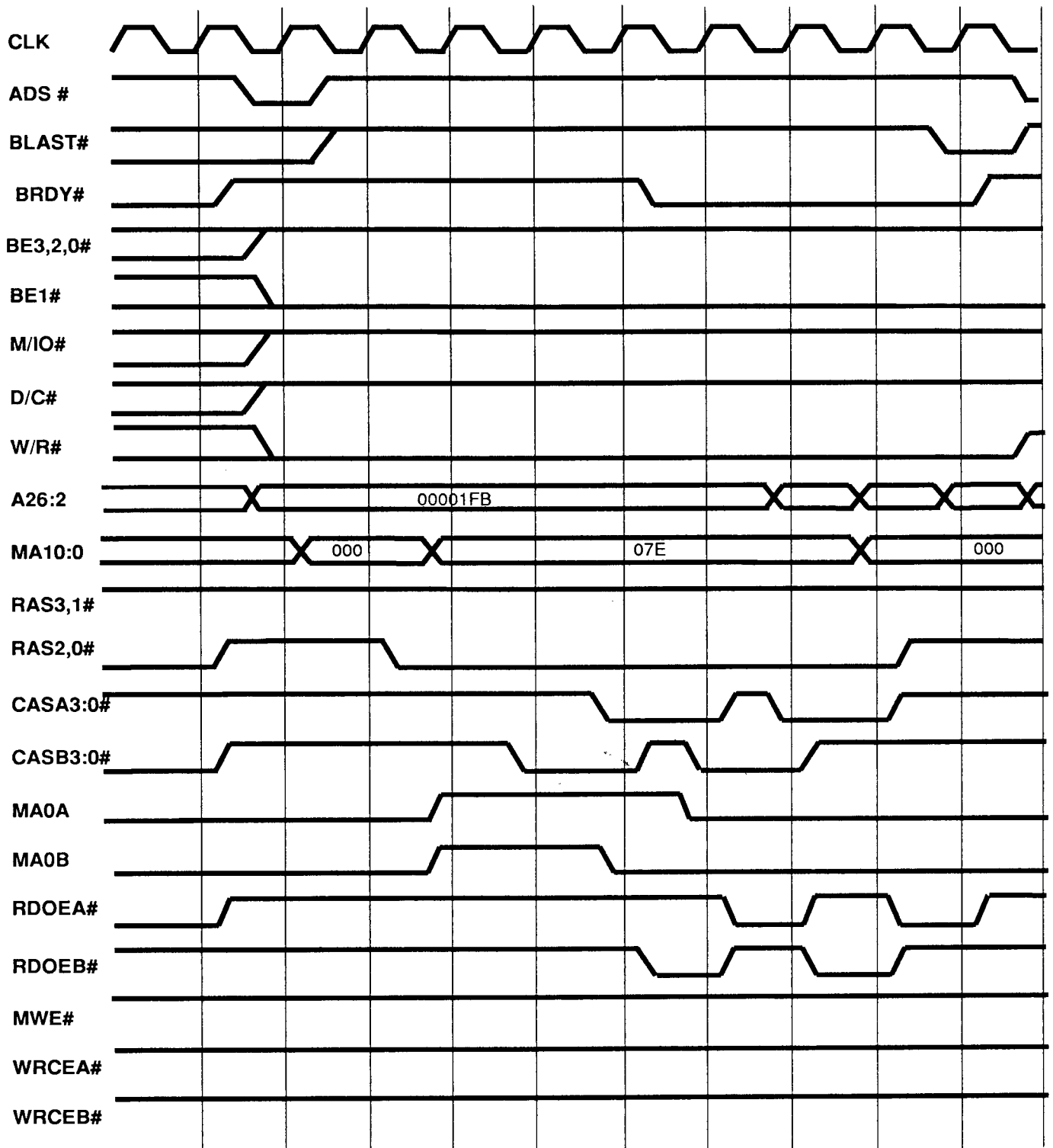
If BLAST# is sampled high, the MS400 will prepare for a burst read operation by accessing data for the second transfer as the first transfer is being performed. The interleaved architecture of the MS400 memory system effectively allows two DRAM accesses to occur simultaneously. During the first transfer of a burst read, both banks are actually accessed by assertion of two RAS# signals. As previously described, each bank has separate RAS# controls. With this method, data is essentially available from the second bank at the same time (or 1 clock later) that it is from the first.

After the completion of the first transfer, the second transfer occurs in only one clock by disabling the first bank and enabling the second. In the example shown, the first read transfer comes from bank A. The interleaving of two banks allows one-clock bursts to occur by hiding the CAS# pre-charge and CAS# access time away from the CPU.

Similar to the single read transfer described above, the assertion of the CAS# signals is controlled by the bits from index register 01H. Note that the RAS#-to-CAS# delay control parameter only applies for the first transfer. Assertion of the CAS# signals for the second Dword transfer will follow the first CAS# signal by one CPU T-state.

At completion of the transfer of the first Dword from each bank, CAS# and OE# are de-asserted in order to prepare the bank for the second Dword access that the bank will supply data for. In the example, the first and third transfers come from bank A, while the second and fourth transfers are from bank B. With one clock bursting, this gives each bank two clock

MS400



BURST DRAM MEMORY READ

periods in order to meet the CAS# specifications tCP (pre-charge) and tCAC (access time) in order to supply its second Dword.

After the first access from each bank occurs, the CAS# pre-charge time for the bank is satisfied (by pulling the corresponding CAS# control high) in order to ready the bank for the second access from the same bank. In this method, memory operations alternate, first coming from one bank and then the next. This continues until all four transfers are completed, and the cache line has been read. The MS400 will also support short bursts (less than 4 transfers) from either the CPU or the cache. The CPU/cache BLAST# signal being sampled asserted (low) when BRDY# is returned by the MS400 will cause immediate termination of the burst cycle.

With use of the interleaved memory system previously described, one clock bursting can be supported at 25 MHz with 80 nS DRAM's. This arrangement allows a cache line transfer in 320 nS, or 8 CPU T-states (in a 5-1-1-1 sequence).

The interleaving of memory banks to support burst accesses and simultaneous operations is the reason for the existence of the MA0A and MA0B address lines. During the second transfer of a burst cycle, the two banks being accessed will need to see different addresses. This is because the bank which is to supply the third Dword of data is being prepared (de-asserting the CAS# signals), while the other bank is supplying data for the second transfer.

Even when two banks are being utilized to support bursting in an MS400 system, full one-clock read bursts can only be supported if the CAS# access time is 1.5 CPU T-states or less. This is because each bank must supply data every other clock in order to provide one-clock bursts to the CPU. Operation of the MS400 with the CAS# access time set to 2 CPU T-states will result in a x-1-2-1 type read cycle. This is because the second read transfer from each bank can only occur three clocks after the first, instead of two clocks as in normal operation.

During burst read cycles, data appearing from the DRAM CPU outputs is latched into the 823 latches. This data will be returned to the CPU one clock later.

Each Dword transfer of the burst is completed by assertion of both the CAS# and OE# control lines for the bank being accessed. The rising edge of the CAS# signals latches read data from the DRAM into the latches; the latched data will be read out by asserting the OE# signals.

Multiple Transfer Read Cycles From One Bank

In some cases, the CPU/cache will desire to perform a burst operation to an area of memory which is supported by only one bank of DRAM's. This will either occur when only one bank is installed in a system, or a memory access in a three bank system is to the bank which is not interleaved.

In such accesses, it is not possible for one-clock bursting to occur. This is because the CAS# pre-charge and CAS# access times can not be hidden from the CPU. At the end of each transfer, the CAS# signals are de-asserted high to satisfy the CAS# pre-charge, followed by re-assertion one-half T-state later. This will necessarily incur wait states for the CPU/cache.

Multiple transfers from the same bank will usually require two clocks for each transfer that is performed. For example, a full cache line (16-byte) burst will occur as x-2-2-2 from a single bank. If the CAS# access parameter is set to 2 CPU T-states, such a cycle will occur as x-3-3-3 cycle.

Writes

Approximately 75-80% of the bus cycles from the i486 CPU are write cycles. The MS400 has been especially optimized for this statistic. For best system performance, as many writes as possible should be performed in zero wait states. With a normal DRAM memory system, no writes could be performed in zero wait states, due to the long access times for DRAM's. However, the MS400 design allows for latching of writes into external latches, thus allowing completion of the CPU cycle in zero wait states. The vast majority of writes will then complete in zero wait states. This external latching is called write posting. Up to eight bytes of write data can be posted by the MS400 before the external latches become filled.

Write Timing Control Bits

Similar to the previous discussion on DRAM read cycles, the timing of DRAM write burst operations can also be controlled through use of the programmable RAS# and CAS# register bits of the MS400.

The RAS# and CAS# bits can be encoded in four possible configurations for the MS400. Each of these configurations results in a unique number of clocks for burst read accesses to occur. These encodings are interpreted by the MS400 as follows:

MS400

A	B	C	D
0:	2,1.5	x-1-2-1	x-3-3-3
0:1	1.5,1.5	x-1-1-1	x-2-2-2
1:0	1.5,1	x-1-1-1	x-2-2-2
1:1	1,1	x-1-1-1	x-2-2-2

- A) BITS (1:0)
- B) tCAS#, tRAS#
- C) BURST READ
- D) BURST READ TO ONE BANK

As before in the read case, the resulting right column is the fastest burst cycle achievable. Some burst cycles may operate slower (more wait states) than the listed numbers; for example, if the RAS# pre-charge time must be absorbed before an access to a bank can occur. However, DRAM cycles will never occur faster than the given numbers.

In addition, the numbers listed above are from the point of view of the DRAM array. In other words, the number of T-states necessary to complete the burst write operations into the DRAM devices.

However, from the CPU perspective almost all writes will complete in zero wait states. The latch devices will be utilized in order to latch all write data from the CPU/cache, allowing zero wait state write operations. In effect, even burst writes from the cache will be posted by the MS400 and completed in zero wait states.

Single Writes

A single write operation is shown in Figure 6. Single write cycles will be the only type of writes which occur for systems which use the 386 or i486 CPU without the Simulcache. As previously described for read cycles, the assertion of ADS# by the CPU/cache indicates the beginning of a cycle. In this case, the MS400 samples the W/R# output high, indicating that a write cycle is occurring.

To begin the DRAM write access, the MS400 will immediately assert the RAS# signal(s) to the DRAM's. At the beginning of each cycle, the internal address mux of the MS400 is controlled to allow row addresses to flow through the MS400 and onto the DRAM MA memory address bus. In this manner, row addresses will become valid to the DRAM's in the first (T1) bus state.

In a few cases, the write cycle is necessarily delayed some number of clocks. For example, a previous bus cycle having just occurred may mean that the RAS# precharge spec is not yet satisfied. If the following write cycle is to the same bank, the write may need to be 'frozen' until the RAS# precharge is completed.

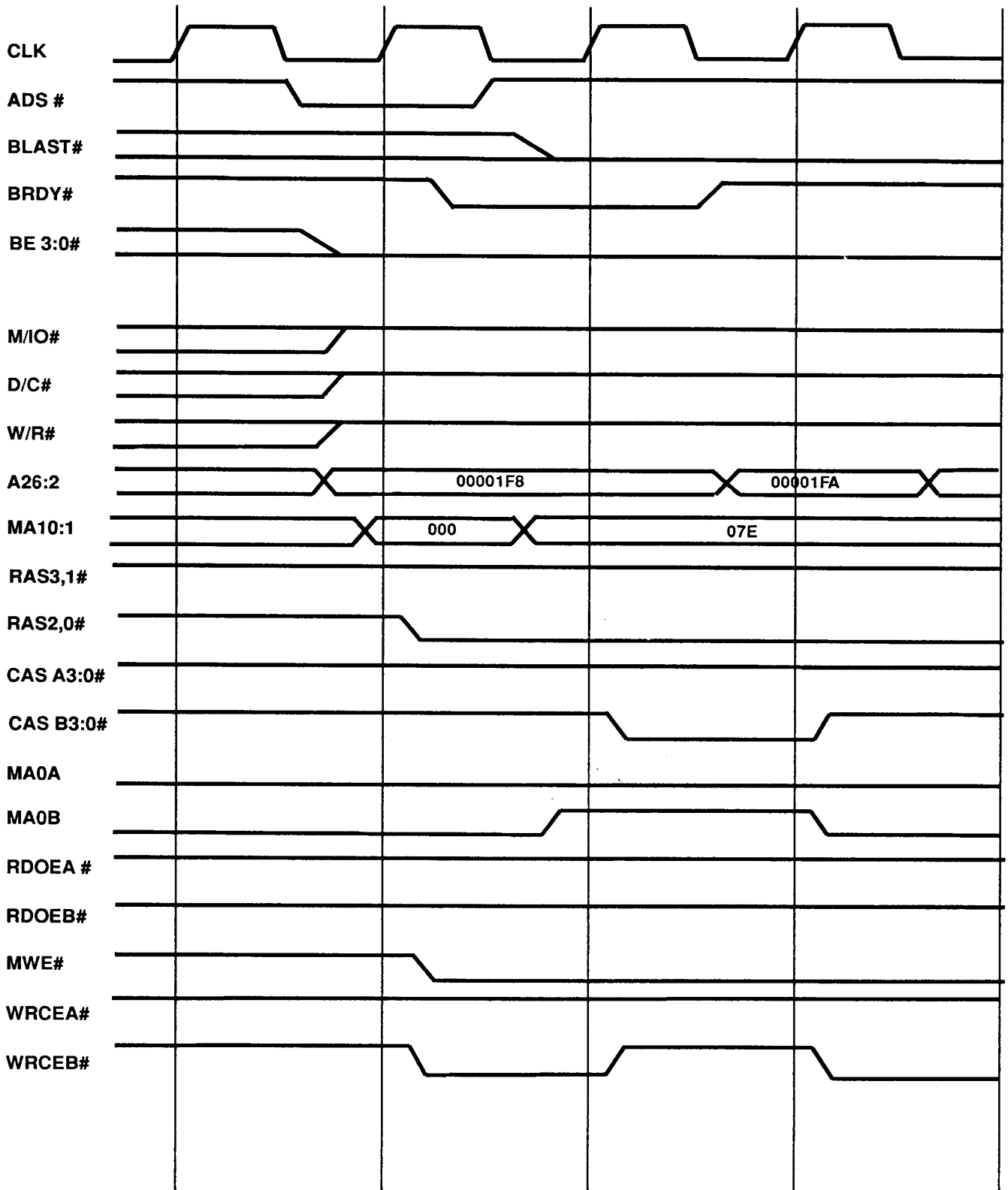
Assuming that external latches are clear and can accept write data, the MS400 will also assert one of the write clock enables (either WRCEA# or WRCEB#) to the latches, and BRDY# to the CPU. These two signals will allow latching of the write data, and termination of the CPU transfer.

The assertion of WRCEA# or WRCEB# will cause the latches to latch the write data on the next rising clock edge. The latches are clocked by the same clock signal that the CPU sees. At the same time that the latches are latching the data, the CPU cycle is being terminated by assertion of BRDY#.

Since posting is occurring, the DRAM cycle will complete later. Once RAS# has been asserted, the MS400 will switch control of the address mux in order to provide the column address to the DRAM address inputs. When the column address is stable and the RAS#-to-CAS# delay spec has been satisfied, the MS400 will assert the CAS# signals. For write cycles, MWE# (memory write enable) input of the DRAM's will be asserted low to indicate a write to the DRAM's.

For write operations which involve only single transfers, the MS400 will sample the byte enables provided by the cache/CPU. The CAS# signals will be derived from the appropriate byte enables. This will prevent data from being written into the array for bytes which the cache/CPU has declared invalid. This is in contrast to read operations, where the MS400 will always drive all CAS# signals low.

The MS400 will employ early write cycles to the DRAM's. This means that after RAS# is asserted, the data to be written is presented on the data inputs of the DRAM's. This data is then internally latched into the DRAM's by the assertion (falling edge) of the CAS# signal.



SINGLE WRITE TO DRAM

MS400

Burst Write Cycles

Burst writes are also supported by the MS400. Although the i486 CPU does not burst writes, inclusion of the Simulcache solution in a system allows burst writes to occur, for both 386 and i486 CPU systems. The Simulcache performs burst writes of cache lines (four Dword quantities) which are 'dirty' back to the DRAM memory, if the system memory controller can support such bursts. With the MS400, burst writes will occur in zero wait states (2-1-1-1 clocks). At 25 MHz, this burst transfer will take 200nS for the four transfers to occur.

A burst write operation is shown in Figure 7. Essentially, the MS400 has control lines which are capable of controlling latches for up to 8 bytes of data, or one 32 bit write for each set. The MS400 pipelines burst write operations, completing the write into one bank while the CPU begins the next write operation. As in the read example, the first and third of the four transfers involves set A. The rising edge of the CLK signal latches write data from the cache into the latches when WRCEA# or WRCEB# are asserted. At the same time, BRDY# is returned to the cache, terminating the Dword transfer. However, it can be seen that each Dword write operation requires one additional clock period to complete, which occurs when the corresponding CAS# signal is de-asserted.

As before, assertion of ADS# from the cache will indicate the beginning of a write. For burst write cycles, BLAST# will be asserted high to indicate that a burst will occur.

The MS400 will latch the state of BLAST# at the end of the first data transfer. If BLAST# is de-asserted high, the MS400 will force all four CAS# signals low for each transfer and will write four bytes of data for each transfer until BLAST# is sampled asserted (low).

The first transfer of a burst will occur as previously described for a single write transfer. The first transfer can access either set A or set B with no performance penalty. The second write transfer will be to the set of banks not used in the first transfer. For example, if the first transfer accesses set A, the second will access set B. All write transfers will occur in the same method as described, with write data being latched into the appropriate latch devices.

Burst Write To A Single Bank

Some write cycles will access an area of memory supported by only one DRAM bank. For cycles to these areas, one-clock write bursts can not be

supported by the MS400. Since the data is being written to only one bank of DRAM's, the CAS# precharge and CAS# access times can not be hidden from the CPU. Such cycles will either be performed as x-2-2-2 cycles, unless the CAS# access time is 2 CPU T-states, in which case x-3-3-3 cycles will occur.

Multiple Write-Backs

The MS400 supports optimization for the MOSEL Simulcache. These optimizations are for the MS400 DRAM memory subsystem. As previously described, burst write-backs from the Simulcache are supported, while page mode operations are supported to supply burst read data to the cache. As an additional optimization, the MS400 supports multiple write-back cycles.

The Simulcache contains two lines of data for each tag that it has. When a tag is replaced in the internal Simulcache tag directory, any dirty data that exists in either line must be written back to memory. Since dirty data can exist in either line, some situations will occur where both lines contain such dirty data. At the end of the first write-back cycle, the Simulcache controller asserts the MWB (Multiple Write Back) line for one clock, which signals the MS443 burst-rams to prepare to perform a write-back of the second line of data associated with the tag replaced. See the MS441 cache controller datasheet for a complete description of write-back cycles.

When the MS400 system has four banks of memory installed, it is capable of posting eight doublewords (32 bytes) of data. This optimization will be particularly effective supporting the Simulcache, although non-cache 486 CPU systems will also benefit from the posting of writes.

The MS400 bank organization with four banks is designed such that each burst write-back from the Simulcache will access a different set in the memory array. Regardless of which set the first write-back accesses, the second (multiple) write-back will be to the other memory set, allowing both burst writes to occur in zero wait states.

Combinations

Back-to-Back Write Cycles

As described previously, the MS400 performs burst write operations by latching the write data into latch devices and later completing the write cycles to the DRAM array. However, the CPU/cache may perform

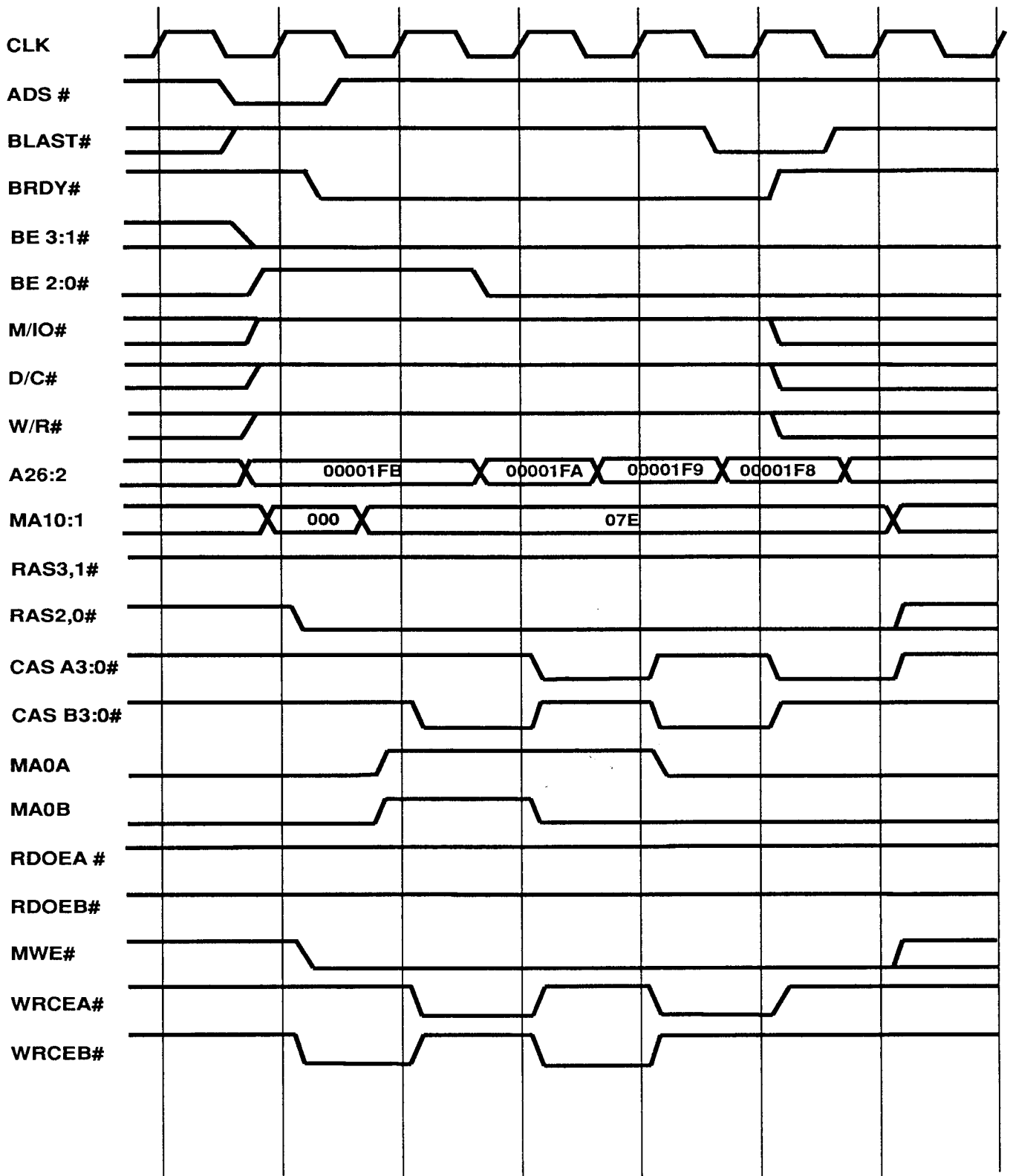


FIGURE X - 2,1,1,1 BURST WRITE CYCLE

a second write cycle immediately after the first. In some cases, the first cycle will not yet have completed when the second occurs. In such a situation, the second write cycle is 'frozen' by the MS400 until the first cycle has completed.

Continuous single writes to the same DRAM bank will fill up the latches. Once the latches have been filled, subsequent CPU write cycles will be delayed (frozen) until the latches can be cleared by completing previous write cycles to the DRAM array.

Two consecutive burst writes are shown in Figure 8. Note that the final write of the first burst access does not complete until frame 7, while the final BRDY# for the first access was returned to the CPU in frame 5. The beginning of the second burst access to the DRAM's is delayed until frame 10.

Note that system operation with four banks installed will reduce the occurrence of this 'freeze'. If the second cycle is to a different set of banks than was the first, the second may occur as normally with no performance degradation.

Write Cycle Followed By Read

Write cycles which are followed by read cycles will also induce 'freezes' of the read until the write cycle can complete. Once again, this is due to the fact that write cycles will complete some number of clocks after BRDY# has been returned to the CPU. A read cycle occurring immediately after the write will be delayed until the write has completed to the DRAM array.

Read Cycle Followed By Write

In contrast to the Write-read combination just described above, a read followed by a write cycle never incur delays or 'freezes'. This is due to the presence of the latches used for writes. At the end of the read cycle, RAS# will be de-asserted to the DRAM's in order to prepare for the next cycle. Even as the DRAM's are being pre-charged, a write cycle which occurs immediately after the read will simply be latched into the latches, allowing posted write zero wait-state CPU operations.

Figure 9 details such a read-write combination. A burst read which transfers a full line of data to the CPU/cache from banks 0 and 2. The burst read executes in nine clocks, as a 6-1-1-1 cycle. This burst read is immediately followed by a burst write. The burst write also transfers four Dwords of data.

Note that even though this write accesses the same two banks of data, it completes in zero wait states, executing as a 2-1-1-1 cycle.

Bank Combinations

When only one bank of memory is installed in an MS400 system, some modifications to memory cycles are necessary. One clock bursting cannot be supported, since the CAS# precharge time necessary to complete successive accesses cannot be hidden from the CPU. Two clock bursts are the fastest allowable accesses in this case. Figure 10 shows a four-transfer read cycle to one bank of DRAM's. At the end of each transfer, CAS# is de-activated.

Operations for systems with four banks have already been described. The MS400 architecture provides that operation with only two banks will also allow one clock bursting to occur. Since burst cycles of even four transfers will access only two banks, there is very little difference in operations between two-bank systems and four-bank systems other than the memory addressing map. This means that even systems with only two installed banks may enjoy very high performance.

DRAM Refresh Operations

Due to the dynamic storage architecture of DRAM's, periodic refresh operations are necessary to prevent the corruption of data stored within the DRAM's. A refresh request is generated within the MS400 every 15.6 microseconds. This request will hold the CPU/cache, and the MS400 will output the refresh (row) address onto both the DRAM MA10:0 bits and the SA bus.

The MS400 contains an 11-bit counter, which generates the refresh addresses and is incremented every refresh cycle. This ensures that the full DRAM array is periodically refreshed.

Inclusion of the Simulcache in a system will bring an additional performance benefit. Because of the dual port architecture of the Simulcache the system side interface of the cache will grant HLDA to the MS400, while the host interface of the cache will continue to service CPU bus cycles. Both read and write hits may continue to operate on the CPU local bus while the refresh is accomplished. Thus, refresh operations to memory can be transparent to the CPU.

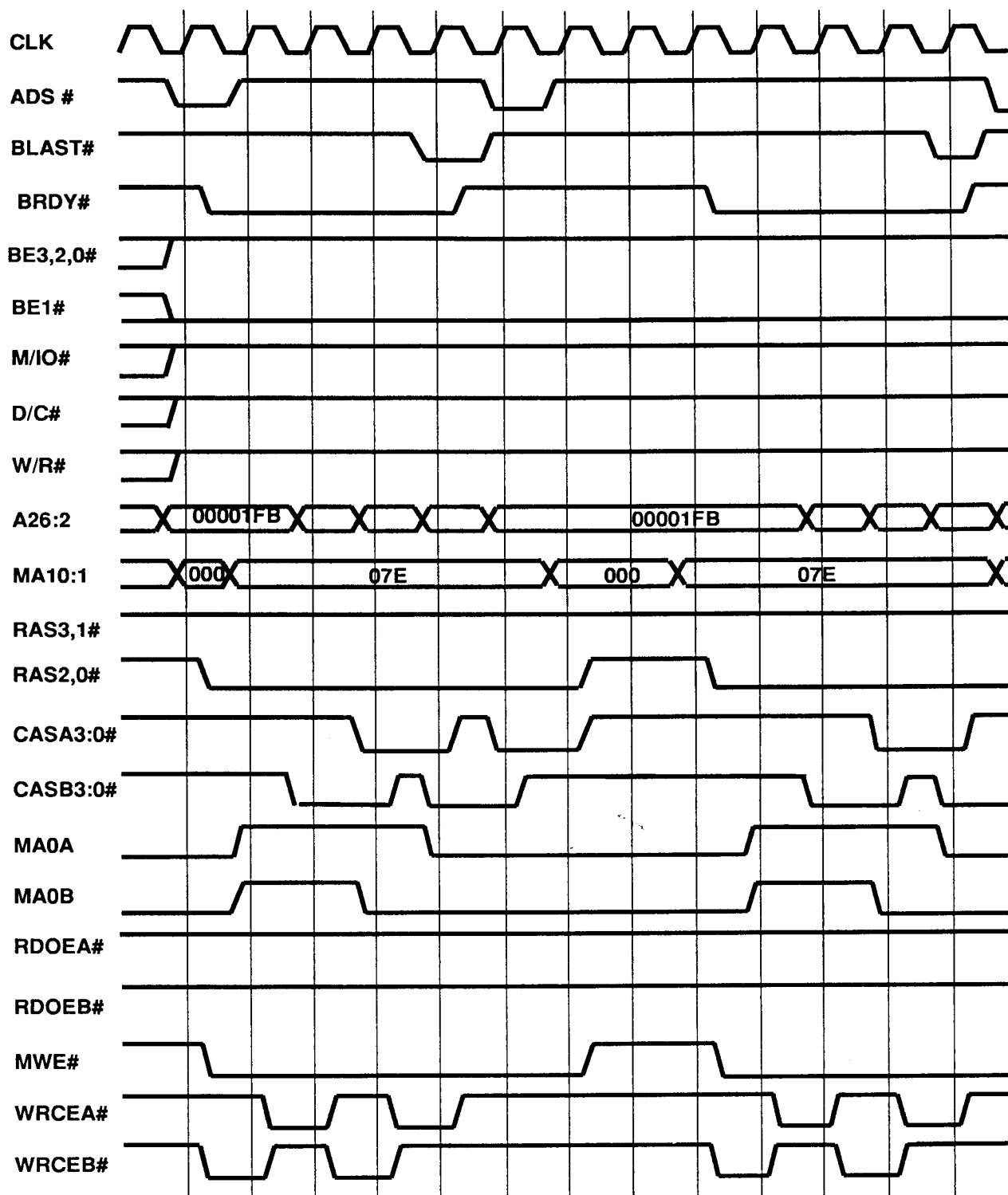


FIGURE - CONSECUTIVE BURST WRITES

MS400

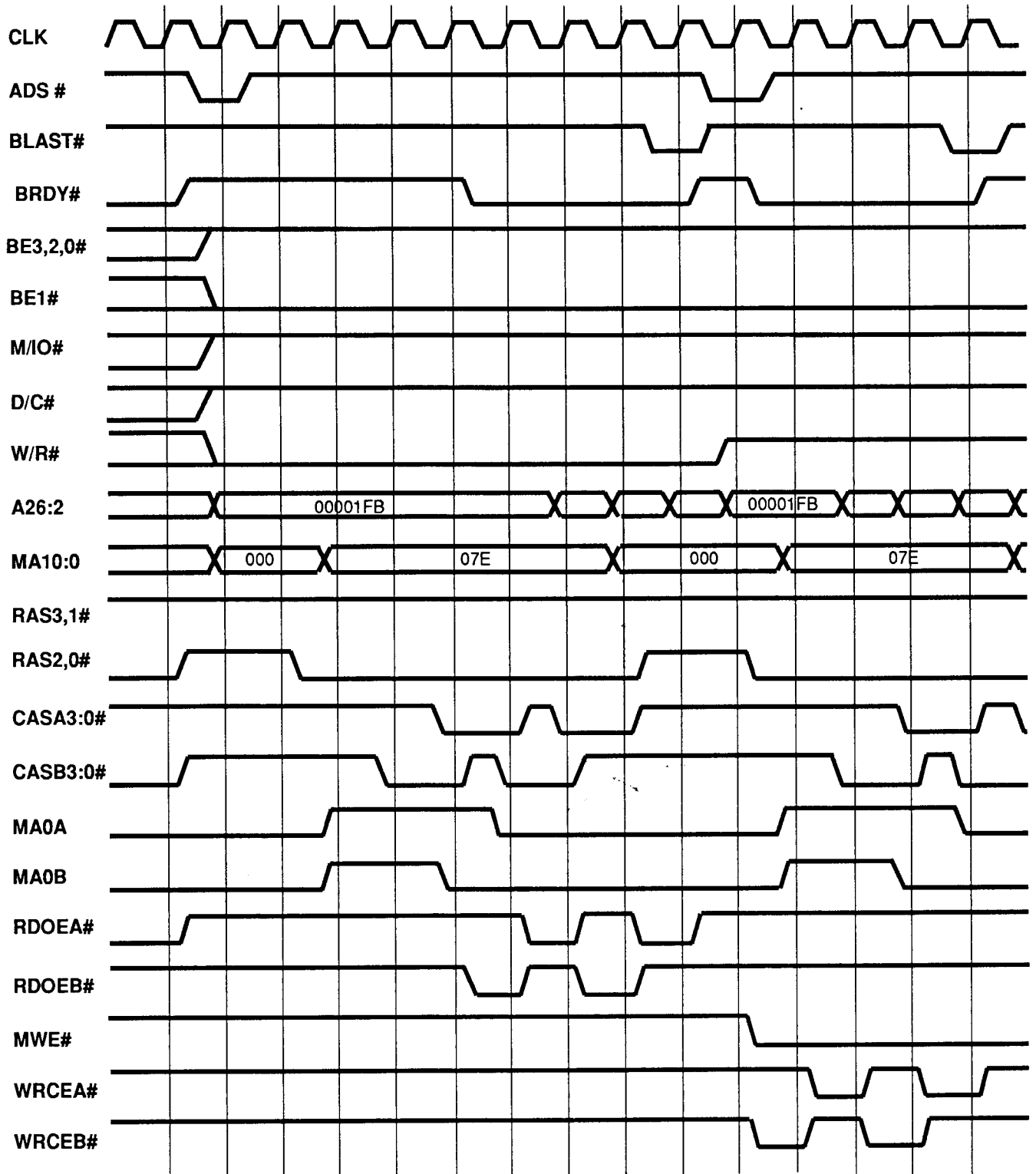
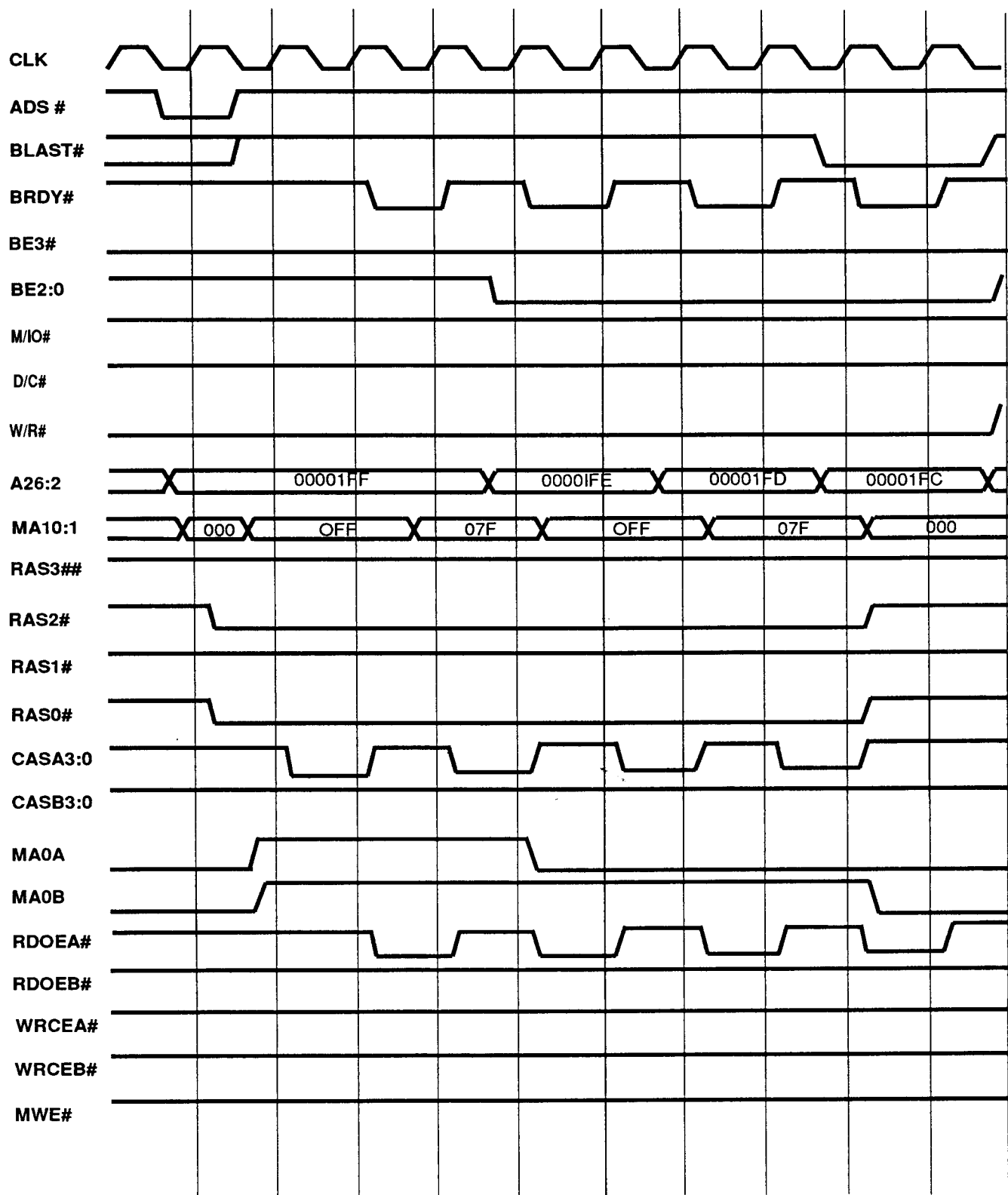


FIGURE - BURST READ FOLLOWED BY BURST WRITE



4 TRANSFER READ CYCLE FROM ONE BANK

MS400

5.4 AT Bus Interface

5.4.1 AT Bus Overview

The MS400 provides the state machines that control all bus accesses and procedures. The control signals interfacing with the AT bus are 100% compatible with an IBM PC/AT.

The AT state machine controls the AT bus cycle and generates bus commands. The AT state machine gains control of the buses when the current cycle is not a local cycle. The AT bus clock is programmable; the state machine controls the duration and wait states of AT bus command. Both 8- and 16-bit I/O devices are supported on the I/O channel.

The MS400 supports 8-, 16-, and 32-bit transfers between the CPU/cache and 8- or 16-bit memory and I/O peripherals which exist on the AT bus. Mis-aligned transfers are supported by the MS400. Byte swapping will be performed if the transfer requests data from a peripheral device which is not physically connected to the requested bytes. Since the AT bus is only 16 bits wide while the CPU and cache SPD data buses are 32-bits in width, Dword (32-bit) requests from the CPU will be generate multiple accesses to the AT bus. These multiple accesses will be held in latches and assembled by the MS400 before BRDY# is returned to the CPU. CPU Dword writes will be dis-assembled, creating multiple write cycles to the AT bus.

An AT bus cycle is initiated by asserting BALE. MEMCS16# and IOCS16# are sampled to determine if the bus cycle is to an 8- or 16-bit device when the I/O command becomes active. The AT bus state machine provides sequencing and timing controls for status and command of various bus cycles. The command cycle is terminated by either of the 0WS# or IOCHRDY signals being activated.

The bus control logic generates all buffer control signals for the local bus, system bus, and the peripheral bus. The bus control logic also controls data conversion and byte swapping when a 16-bit cycle accesses an 8-bit peripheral device.

The default size for AT peripheral devices is 8 bits. However, an accessed device residing in a 16-bit slot may assert IOCS16# or MEMCS16# during a 16-bit transfer to indicate that it can drive or accept both bytes of data. The MS400 responds to IOCS16# and MEMCS16# when running AT bus cycles. IOCS16# and MEMCS16# are sampled by the MS400 on the falling edge of BALE. If neither of these signals is

asserted by the accessed device, two 8-bit AT cycles will occur to transfer the 16 bits of data. In addition, MEMCS16# and IOCS16# are ignored during 8 bit transfers.

The MS400 provides support for several types of cycles on the AT bus. Standard cycles, ready cycles, and zero wait state cycles from the CPU are all supported.

The state of the AT signals SA0 and SBHE# indicate whether the request is for 8- or 16 bits of data. The table below is used to decode the states of these two signals:

<u>SA0</u>	<u>SBHE#</u>	<u>TRANSFER</u>
0	0	16 bits on SD15:0
0	1	8 bits on SD7:0
1	0	8 bits on SD15:8

The combination of SA0 high and SBHE# high is not allowed.

In addition to transferring more data per cycle, 16-bit cycles on the AT bus may occur faster than 8-bit cycles. The AT bus command lines IOW#, IOR#, MEMW#, and MEMR# have one wait state if the transfer is a 16-bit transfer and four wait states if the transfer is to an 8-bit device.

Default Wait States for AT Cycles:

<u>TYPE</u>	<u>Wait States</u>
8-bit I/O & Memory	4
16-bit I/O & Memory	1
8- and 16-bit DMA	1
ROM cycles	3

In addition to the standard cycle lengths listed above, the MS400 provides support on the AT bus for zero wait state and Ready cycles.

5.4.2 Synchronous AT clock

The basic AT clock, S1CLK, is generated by the MS400 and sent to the AT bus. S1CLK is generated synchronously as a divided signal from the CLKIN clock of the MS400. However, the AT bus generally operates as an asynchronous bus.

5.4.3 Basic AT Bus Cycle

Figure 11 shows a basic 8-bit AT memory write cycle. BALE is asserted in order to begin the cycle. The falling edge of MEMR# indicates that a read cycle is occurring. If the cycle address is within the lowest megabyte of memory, SMEMR# will also be asserted. Since only 8 bits of data are requested for the cycle, the state of MEMCS16# returned by the accessed device is ignored by the MS400. 8-bit reads on the AT bus have a standard access time of 4 wait states.

In this example, the accessed device is not fast enough to respond for standard cycles; as a result, the device de-asserts IOCHRDY. This assertion converts the standard cycle into a ready cycle. The cycle is extended until IOCHRDY is re-asserted by the device.

5.4.4 Aligned/mis-Aligned Transfers

For AT compatibility, the MS400 can perform both aligned and mis-aligned transfers. For operations on the AT bus, the MS400 state machines do not support burst operations. BLAST# is not monitored, and BRDY# is used to terminate the cycle. Even when the CPU/cache desires to perform burst accesses, the MS400 will not support such bursts to/from the AT bus.

Aligned Transfers

Aligned transfers on the AT bus are those transfers which complete in one bus transfer. Examples of aligned transfers are 8-bit read/write cycles to either memory or I/O devices, or 16-bit transfers to 16-bit memory and I/O devices which use the MEMCS16# or IOCS16# lines.

Aligned AT Accesses

Aligned AT operations are those that can complete in one transfer on the AT bus. All 8-bit cycles will complete in one transfer, and any 16-bit cycles which access a 16-bit device will complete in one transfer.

Mis-Aligned Transfers

The MS400 classifies mis-aligned transfers are those transfers which take more than one transfer to complete, or involve byte swapping. Extra AT cycles which are due to mis-aligned transfers are transparent to the applications software. All that the CPU/cache sees is one long cycle, with BRDY# being de-asserted until all

necessary AT cycles complete on the AT bus. The MS400 will generate all addresses and any necessary control signals in order to perform additional AT bus cycles due to mis-aligned transfers.

5.4.5 Byte Swapping

The AT system data bus is 16 bits wide, but peripheral devices residing on the bus can be either 8- or 16-bit devices. If an accessed device is 16 bits wide, it may supply data on either or both bytes of the AT system data bus. However, if the device is only 8 bits wide, it is connected to only the lower byte (D7:0) of the bus. For accesses which request data on the upper byte of the bus, the data supplied by the peripheral device must be transferred to the upper byte. Such cycles which request data on only the upper byte are indicated by SA0 being asserted high, and SBHE# being active (low).

The MS400 drives signals in order to correctly perform byte swapping for mis-aligned byte transfers. In such swapping, the data bits of the lower byte (D7:0) are swapped through a 245 transceiver in order to appear on the higher byte (D15:8) for read operations. For mis-aligned writes to 8-bit devices, D15:8 is swapped to D7:0.

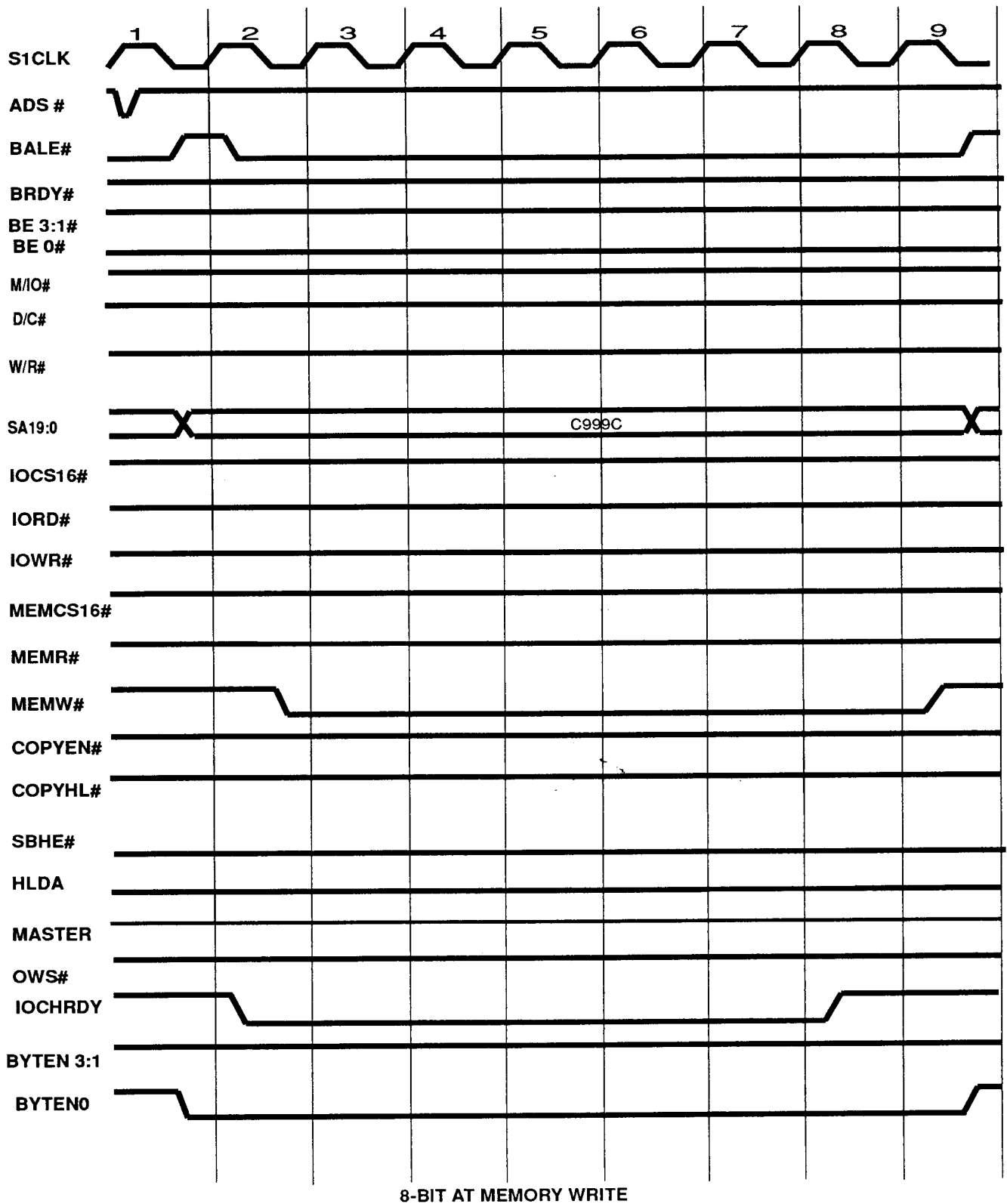
The COPYEN# and COPYHL# signals from the MS400 are used to control the 245 buffer byte swapper. COPYEN# opens the output enables of the device, while COPYHL# chooses the direction that data will be gated. For swapping on write operations, COPYHL# will be asserted (low) to pass the high byte to the low byte, while read swaps will require COPYHL# to be de-asserted (high), to pass the low byte to the high byte.

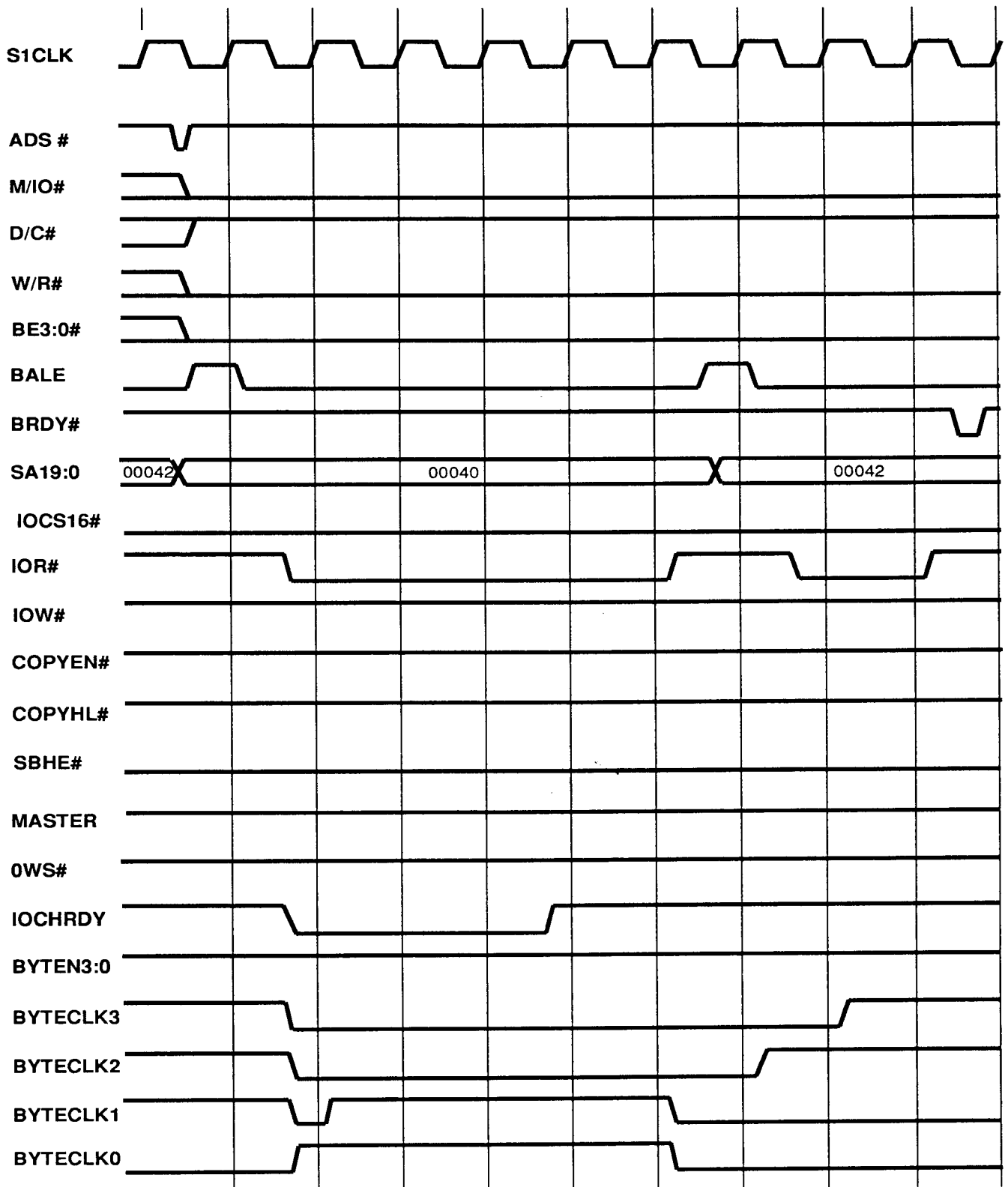
5.4.6 Assembly on AT reads

Mis-aligned reads on the AT bus can be either 16-bit reads to 8-bit devices, or 32-bit (Dword) read operations to any device. These operations can be either in memory or I/O space. The MS400 will withhold the BRDY# signal to the cache/CPU, in order to execute multiple AT read cycles.

Figure 12 shows a 32-bit I/O read operation to the AT bus, which is to a 16-bit device. As the addresses become valid for the bus cycle, the accessed device asserts IOCS16#, indicating its 16-bit size. As a result, the MS400 generates two cycles on the AT bus. The first AT read operation is for bytes 0 and 1 of the request, and occurs on the both bytes of the AT bus, fetching the low word of the needed data. At

MS400





32-BIT AT I/O READ FROM 16-BIT DEVICE

the completion of this first transfer, the read data is stored in the SPD bus byte 0 and byte 1 latch devices. Note the BYTECLK 0 and BYTECLK1 signals during the first cycle. This AT cycle is shown as a ready cycle, with the accessed device de-asserting IOCHRDY to extend the cycle.

The MS400 then re-asserts BALE to perform another AT bus cycle. The second read operation fetches CPU bytes 2 and 3. Since neither OWS# is asserted or IOCHRDY de-asserted, this cycle occurs as a standard 16-bit AT cycle, and incurs 1 wait state, finishing in 3 AT clocks. Like the first transfer, the read data is stored in two of the latch devices, in this case the latches corresponding to bytes 2 and 3.

Since the 16-bit device is capable of supplying data directly to both bytes of the AT bus, the AT byte swap logic is not needed for the second cycle.

At the end of the second transfer, the requested four-byte Dword is available and being driven by the latches; to complete the cycle, BRDY# is returned to the cache/CPU.

A 32-bit request which was to an 8-bit device would occur in a similar fashion, with four 8-bit accesses occurring to fetch all requested bytes from memory. Likewise, the CPU cycle would be extended until all four bytes were assembled in the latches.

5.4.7 Dis-Assembly on AT Writes

Similar to the assembly case described above, some AT write cycles will require more than one transfer to complete. Such cycles are 16-bit writes to 8-bit devices, and all 32-bit writes. As before, the MS400 will not assert BRDY# until all writes are completed.

Since all data is present at the beginning of the first transfer, latches are not required. The MS400 will use external buffers to temporarily store the data, and enable the buffer outputs as needed for each transfer.

Figure 13 shows a 32-bit write to a 16-bit device. Two transfers are needed to complete the cycle. The first AT transfer writes bytes 0 and 1 directly to the peripheral on the AT bus. Note that two of the BYTEN# (byte enable) signals are active (low) for the first transfer. This cycle is shown as a ready cycle, with the accessed device de-asserting IOCHRDY to extend the cycle.

The second write cycle enables two buffers, supplying data for bytes 2 and 3 to the device. At the end of the second transfer, BRDY# is returned to the CPU. This bus cycle occurs as a standard 16-bit AT cycle.

5.4.8 AT DMA Operations

Due to the inclusion of the MOSEL Simulcache in an MS400 system, some considerations are necessary for DMA operations. Both snoop read and snoop writes will operate slightly differently than do those for the i486 CPU. Since the Simulcache has a write-back architecture, there are instances where the cache contains dirty data. This 'dirty' data consists of locations modified by the CPU during write operations, for which the data has not yet been written to main memory. As such, the Simulcache will supply data for some snoop read operations. In addition, snoop writes to the Simulcache which are hits will cause an update of the write data into the cache.

Snoop/invalidate cycles are indicated by the MS400 to the CPU/cache by use of the EADS# (External Address Strobe) signal. In the same clock that EADS# is asserted, the MS400 drives the address for the snoop cycle. In addition to the line address (A27:4) needed by the i486 CPU, the Simulcache requires that A3, A2 and the byte enables be driven valid.

The Simulcache adds three pins to the i486 CPU bus definition in order to support snoop operations. These three pins are SMEMW/R#, SNPBUSY, and SMEMDIS. Essentially, SMEMW/R# is driven to the cache in the same clock that EADS# is asserted. This informs the cache whether the snoop is a read or write operation. In response, SNPBUSY toggles high as the tag lookup operation.

DMA Read Operations

Due to the 'dirty' data potentially contained in the Simulcache, DMA read operations always generate snoop cycles in the MOSEL Simulcache. During such a read cycle, the MS400 presents the valid memory address to the Simulcache, such that the Simulcache can check its internal tag directories to see if any 'dirty' data exists at that address. Along with the address, the MS400 drives the AT bus signal MEMR# low to indicate a read operation. MEMR# should be connected to the Simulcache SMEMW/R# input. If the Simulcache does not contain any dirty data at the specified address, the DMA read proceeds normally, with the MS400 controlling the DRAM's to supply data to satisfy the read request.

MS400

However, if the Simulcache does contain 'dirty' data at the specified address, it will intervene and supply the requested data onto the MOSEL cache data bus.

On the falling edge of SNPBUSY, SMEMDIS may be sampled for snoop read operations in order to determine the necessary action. SMEMDIS being sampled high on a snoop read indicates a hit, and as a result the Simulcache will supply the requested read data. SMEMDIS being sampled low indicates a miss, and that the system should supply the data. In either the hit or miss case, the MS400 will assert BRDY# after SNPBUSY has fallen in order to terminate the cycle. For a further discussion of snoop operations, refer to the MOSEL MS441 Simulcache controller datasheet.

DMA Write Operations

When DMA write operations occur, the MS400 will provide the correct address to the Simulcache or CPU, along with driving MEMR# (on the AT bus) high to indicate a write operation. In addition to this address, the MS400 will supply the data for the write operation.

If the write operation is a hit in the Simulcache, the data will simply be updated into the cache. If a miss occurs, no action will occur in the Simulcache. The update for hits will be transparent to the MS400. It will not be able to distinguish between a snoop write hit or miss. All snoop writes will trigger an invalidate cycle to the i486 CPU, if present in the system. As before in the read case, the MS400 will assert BRDY# to terminate the snoop write cycle after SNPBUSY has been de-asserted by the Simulcache.

Note that there is one consideration for MS400 systems which do not contain the Simulcache. Since the MS400 does not contain the full address bit range that the i486 CPU does, the upper address lines are not driven to the CPU during snoop write operations. The system designer must ensure that the upper addresses not contained in the MS400 are driven low to the CPU, such that the correct address in the CPU cache is invalidated.

5.4.9 ROM Accesses

Shadowing must be disabled to allow ROM reads to occur. If shadowing is enabled (by setting a bit in the appropriate control register), the ROM read will instead occur from the DRAM array.

ROM reads will occur with 3 wait states on the AT bus, and will occur as a regular AT bus access. The

data from the EPROM is output onto the ID bus, where it is gated to the SD bus. ROMEN# controls both the enabling of ROM outputs and the enabling of the 244 buffers.

Once on the SD bus, the ROM data is latched into the latches and finally returned to the CPU.

ROM write accesses are allowed to occur, but will obviously have no effect on the contents.

5.4.10 Memory refresh cycle

One of the channels in the MS400's 8254 compatible timer will be used to control refresh operations. Every 15.6 microseconds, this channel will generate a refresh request.

If the 386 or i486 CPU currently controls the bus, the refresh will immediately occur. However, if the DMA controller currently owns the bus, the refresh must be delayed until the CPU again has control of the bus.

An add-on card that is currently the bus owner can also request refresh cycles. Since the add-on card obtained the bus through arbitration using the DMA controller, the refresh will only occur in this case if specifically requested. The add-on card should assert REF# to request a refresh.

When a refresh occurs on the AT bus, it will also occur to the DRAM memory array. All installed DRAM banks will be concurrently refreshed. The MS400 contains an 11-bit counter which is incremented each time a refresh occurs. This ensures that all addresses are periodically refreshed.

5.4.11 AT Bus Owner Cycles

To support AT compatibility, add-on cards in an MS400 system may be designed to be bus master cards. Before the add-on card can execute any cycles, it must obtain the AT bus through an arbitration cycle.

To arbitrate for the bus, the add-on card uses the DMA controller contained within the MS400. The card asserts one of the DREQ lines and waits for assertion of the corresponding DACK# signal from the MS400. Upon receiving DACK#, the card should assert MASTER# low to indicate ownership of the bus. In response, the MS400 will de-assert AEN. At this point, the add-on card is free to run cycles.

When the add-on bus owner card obtains the bus after the arbitration cycle has occurred, it is free to run cycles. Only standard and ready cycles are sup-

ported for add-on card cycles. In addition, DMA cycles cannot be executed by the add-on owner card. Otherwise, other bus resources and add-on cards should not be able to distinguish whether the CPU or the add-on owner card is driving the cycles.

Upon completion of any cycles that the add-on owner card wishes to execute, it should release the AT bus. The release begins by de-asserting the MASTER# and DREQ signals. In response to these actions, the MS400 will de-assert the corresponding DACK# signal and assert AEN.

5.5 Interrupt Acknowledge Cycles

Some signals from the CPU do not affect the operation of an external cache in the system. These signals should bypass the cache, and be connected directly to the MS400. Such signals are:

INTR

NMI

Assertion of the INTR and NMI pins to the CPU will result in an interrupt acknowledge (INTA) cycle being driven from the CPU, if interrupts are internally enabled in the CPU. The INTA cycle will consist of two bus cycles. During the second bus cycle, the MS400 will respond by placing the interrupt number on the data lines. If the MOSEL Simulcache is present, the cycles will still occur in the same manner. The Simulcache will treat these cycles as non-cacheable. The Simulcache will re-drive the INTA cycles to the MS400, and bypass returned vectors to the CPU.

Interrupt Acknowledge cycles will occur as AT cycles, since the 8259 interrupt controllers reside inside the MS400. The interrupt vector will be generated during the second cycle onto the X (peripheral) data bus, gated through the system SD data bus, and to the CPU/cache.

5.6 Halt/Shutdown Cycles

Either the 386 or i486 CPU can generate halt and shutdowns. A Halt bus cycle is triggered in response to the Halt instruction being executed by the CPU. A shutdown cycle will occur due to abnormal processing or error conditions, when the CPU is no longer capable of performing legal instructions.

When either Halt or Shutdown cycles occur, the MS400 will decode these cycles and complete them by returning BRDY#.

Note that the MS400 will only see and respond to bus

cycle encodings as if an i486 CPU is in the system. Even if a 386 CPU is installed, the Simulcache (which must then be present) will intercept the 386-style Halt and Shutdown encodings, and translate these encodings to i486 CPU-style encodings which the MS400 will respond to.

5.7 Flush Cycles

When a flush of the cache(s) of the Simulcache occurs, it will be necessary for the Simulcache to write back any dirty data which has not yet been written to the system. The Simulcache will perform these write-backs as a series of burst write operations.

For the MS400, these write bursts will be treated exactly as normal burst write operations. The MS400 will not know that a flush sequence is occurring. With the Simulcache, up to 2048 flush sequences can occur. During this sequence, the Simulcache will recognize HOLD from the MS400 and grant HLDA at the end of write cycles, so that refresh and other necessary bus master operations may occur.

When an MS400 system is operating without inclusion of the Simulcache, the i486 CPU may drive Flush bus cycles directly to the MS400. Since the MS400 does not contain a cache, no direct action is necessary. The MS400 will simply return BRDY# in order to terminate the bus cycle.

5.8 Misc Functional features

5.8.1 Memory Remap

In AT system architecture, the 384 kBytes of memory space which is present at the top of the first Mbyte is reserved as AT memory space. As a result, any memory cycles in the 640k-1M address range will generate AT cycles rather than DRAM shadowing. Thus, 384 kBytes of DRAM memory may be 'lost' and unused in a system. This memory may be utilized in one of two ways in an MS400 system, by being either remapped or shadowed. Shadowing will be further discussed in section 5.8.2.

Memory remap is controlled by bit 7 in index register 00. When this bit is set to '1', the 384 kBytes will become available and will appear at the top of physical installed memory. This memory will appear as normal memory and will be cacheable unless bit 7 of index register OEH is cleared. The remap will be transparent to software. All that applications software or operating systems will see is an additional 384

MS400

kBytes of available memory.

Remap is only available if the amount of memory installed in the MS400 system is either 1 or 2 Mbytes. If more than 2 Mbytes is installed, setting the Memory Remap bit will have no effect on system operation. In addition, Memory Remap can not be employed if Memory Shadowing is in use, since both functions entail different use for the same memory.

5.8.2 Shadow RAM support

For faster system operation, the MS400 supports shadow RAM operations. Since programs will execute faster from DRAM memory rather than slower EPROM devices, it is desirable to copy the contents of the EPROM's into DRAM's. Read cycles to shadowed memory can then be directed to the DRAM's rather than EPROM's.

The MS400 allows for the top 256 Kbytes of the first Mbyte of physical memory to be shadowed into the DRAM array. This function does not 'consume' any additional memory, since this 256 Kbytes of the DRAM array would otherwise go unused. This 256 K is normally reserved for devices on the AT bus, and video and system BIOS EPROM's appear here. This 256 Kbytes are broken into seven separate regions by the MS400. Each region can be shadowed independently of the status of the other regions. The size of these regions is either 16 Kbytes or 64 Kbytes per region. The regions are as follows:

	<u>Address Range</u>	<u>Range Size</u>
1)	C0000:C3FFF H	16 Kbytes
2)	C4000:C7FFF H	16 Kbytes
3)	C8000:CBFFF H	16 Kbytes
4)	CC000:CFFFF H	16 Kbytes
5)	D0000:DFFFF H	64 Kbytes
6)	E0000:EFFFF H	64 Kbytes
7)	F0000:FFFFF H	64 Kbytes

Some mechanism is necessary to prevent the memory locations copied into DRAM's from being over-written by subsequent CPU write operations. The MS400 supports such write protection for the shadowed portions of memory. Any write cycles to shadowed addresses for which the WD bit is set will be terminated by the MS400 without altering the contents of the DRAM's. The system BIOS should ensure that the WD bit is set for all shadowed re-

gions, as write operations to shadowed regions whose associated WD bit is cleared will effectively alter the contents of EPROM. This may lead to system failure.

Control of memory shadowing operations is handled through the control registers of the MS400. Each region has two control bits associated with it, which are used to control the DRAM response for that region of memory. The first bit is the RE (DRAM Read Enable) bit, which either enables or disables reads for the portion of DRAM which maps into the region. The second bit, WD (DRAM Write Disable), disables or enables writes to the DRAM associated with the region. The combination of these two bits is interpreted as follows:

<u>WD</u>	<u>RE</u>	<u>DESCRIPTION</u>
0	0	Writes are directed to DRAM, reads are directed to the AT bus.
0	1	Reads and writes both are directed to DRAM. In this mode, the RAM is a scratch pad.
1	0	All reads and writes are directed to AT bus. In this setting, the EPROM's will provide data for read operations, and write operations will also be directed at the EPROM's.
1	1	Reads go to RAM, writes go to AT bus (which is used to disable writing). This setting is used after the ROM has been copied to RAM.

To shadow memory from EPROM's to DRAM's, both WD and RE should be cleared to zero. This allows reads to receive data from the EPROM, and writes to be written to the DRAM array. After these bits are set, CPU string move operations may be used in order to read the entire region from the EPROM's and write to the DRAM's, copying the contents. Finally, both WD and RE should be set to one to write protect the DRAM's and turn on the shadowing feature.

The default setting for all regions is WD=0 and RE=0. Essentially, the regions are prepared for shadowing initially after RESET occurs.

Read cycles to DRAM's which contain shadowed EPROM code will be burst by the MS400 to the CPU/

cache. In addition, use of the Simulcache allows the EPROM contents to be contained in the cache and write-protected. This will allow very high performance for sections of code which make many BIOS/EPROM accesses.

5.8.3 Turbo/Normal Operation

The combination of the i486 CPU, the Simulcache controller, and the MS400 memory controller allow for a very high performance system. However, some this rate of operation can cause problems with some PC/AT software which was intended for older slower performing systems. For such programs to reliably and correctly operate, the MS400 system must be slowed.

However, directly controlling the i486 CPU rate of execution is difficult due to the existence of caches in both the CPU and the external MOSEL cache. Because of these caches, the CPU rate of execution is somewhat independent of the bus speed. Hits in both caches will mean that CPU usage of the system bus may be quite low for periodic stretches during program execution. Due to these factors, the system BIOS should be modified to allow the system user to program the execution rate that he/she prefers. This will allow each user to tailor system performance to their requirements.

The MS400 supports this programmability through use of the counter 1 in the 8254-compatible timer-2. Counter 1 is otherwise unused in PC/AT systems. This counter is clocked by a signal of 1.19318 MHz. By writing various clock counts into the counter, the BIOS may control the output of this counter. Both the frequency and duty cycle may be directly controlled.

When the Turbo/Normal# bit is set to a logical one, the system operates at full speed. When this bit is cleared to zero, the MS400 responds by asserting the HOLD request to the CPU while the output of timer-2 counter 1 is high. HOLD is recognized by the CPU/cache at the termination of bus cycles, and its assertion will prevent the CPU/cache from using the system bus. By controlling the percentage of time that HOLD is applied, the BIOS may effectively throttle the CPU and reduce system performance. The timer should not be programmed such that the HOLD request is asserted for a period of time greater than 15.6 microseconds, or refresh requests may be missed, resulting in unreliable system operation.

To further increase the effectiveness of this throttling, the BIOS may combine Turbo/Normal# bit with the cache flushing option that the MS400 provides. By

toggling the flush bit in the MS400, the caches of both the MS400 and i486 CPU will be cleared. This will cause all cycles to be cache misses, and bus utilization will become much greater. This will enhance the effectiveness of the HOLD request signal.

5.8.4 A20 Gate/Fast A20 Gate

During normal operation of the MS400 system, some software will require the CPU to switch from Protected Virtual Address Mode into Real Address Mode. Due to an inherent limitation in the Intel 80286 CPU which was not capable of performing this switch, external system hardware in the original PC/AT was added. This hardware resets the CPU whenever such a switch was required. Such switching is very time-consuming, and thus limiting the performance of operating systems and applications which perform many of these switches.

For better performance, MS400 systems offer fast A20 Gate switching. Bit 6 of the index register 06H is the Fast Gate A20 bit. When this bit is set to one, the A20 bit from the CPU is passed through unchanged for DRAM and system cycles. When this bit is cleared to zero, A20 is forced to a value of zero.

5.8.5 Reset & Fast reset

Similar to the Fast Gate A20 discussed in the previous section, the operation of some software will require resetting of the CPU dynamically. In the original IBM PC/AT design, this reset also came from the keyboard. The MS400 supports a Fast Reset function.

By setting bit 7 of the index register 07H, a fast reset of the CPU will occur.

5.8.6 Parity

The MS400 supports parity for all memory and I/O operations.

During writes to DRAM, the i486 CPU supplies even parity on the ninth data bit of each byte. This parity bit is simply latched and stored into the DRAM array for later use.

When a DRAM memory read occurs, the DRAM's in the MS400 system supply the stored parity bit previously provided by the CPU. These bits are returned to the CPU, along with the requested read data. If the Simulcache is operating in the system, it will also pass along the parity bits.

The CPU will also drive parity for writes which are to

MS400

the AT bus. However, the MS400 has no mechanism for storing these bits. Since the AT bus has no parity bit(s) associated for either of the SD data bus bytes, these bits will simply be lost.

Because of this, CPU reads from the AT bus must make special provision for reads from the AT bus. To support this function, four external 657 parity generator/checkers are used to support AT bus parity. These generators are placed on the SPD bus. One 657 is used to generate parity for each of the four bytes of the SPD bus.

Each read for which data comes from the AT SD data bus onto the SPD bus will activate the 657 devices. This should be implemented externally by ANDing the LOED# and DIRRD# signals. As read data becomes valid from the AT bus, the 657's will generate the ninth (parity) bit for each valid byte. Even parity will be generated.

Only the parity generation function of the 657's is used. This is because, as previously explained, the i486 CPU provides a parity check feature.

The i486 CPU always generates and expects to receive even parity. If odd parity is found for a read cycle, the i486 CPU will signal the error by activation of the PCHK# output. PCHK# is signalled to the MS400 as the PERROR# input. Activation of this input will result in an NMI (Non-Maskable Interrupt) being generated to the CPU. This is the only action that the MS400 will take as a result of a parity error. The system BIOS which will be called due to the NMI has the responsibility for undertaking corrective action in response to the error.

5.8.7 BIOS Size and Location

The MS400 is capable of operation with either one or two ROM BIOS devices. These devices should each be 64kx8, 150 ns access speed.

In addition, register bit 4 of the register at index 00H controls the occurrence of the BIOS within the 16th Mbyte of memory, which is the top of AT address space. When this bit is set, BIOS ROM appears just below the 16 Mbyte address limit. Otherwise, the BIOS ROM appears only at the top of the first 1M byte, and at the top of the 4 G byte address space.

Either 64 kBytes or 128 kBytes of BIOS space can be selected. Bit 6 in the register at index 0 selects between the two options. If this bit is set to '0', 64 kBytes is chosen, while a '1' selects 128 kBytes. If 64

kBytes is selected, the BIOS space will appear from xF0000-xFFFFFFH, (where x means either the first, 16th, or highest megabyte of memory). If 128 kBytes is chosen, addresses from xE0000-xFFFFFFH will generate BIOS cycles. This is subject to considerations due to having a Combined BIOS. See section 5.8.8 for more details on Combined BIOS.

Operation with one BIOS will produce 8-bit accesses to the ROM's, and only utilize one 244 buffer device. If the second BIOS device is installed, 16-bit accesses should occur. This will be accomplished by asserting MEMCS16# for BIOS accesses. The MS400 will not perform this assertion, and it must be done externally.

If shadowing of the BIOS to RAM memory has been employed, reads and/or writes to the BIOS area may instead cause DRAM accesses rather than ROM accesses. See section 5.8.2 for a full discussion of shadowing.

5.8.8 Combined System/Video BIOS

To reduce part count and system cost, the VGA BIOS and system BIOS codes may be combined into one EPROM device. The MS400 is capable of generating cycles to this single device for both system and video BIOS cycles. With this option, up to 64 kBytes of system BIOS and 64 kBytes of video BIOS may be supported by the MS400.

To enable this option, two steps must be taken. First, bit 6 of index register 00H must be set to '1'. This will enable 128 kBytes of EPROM. Second, bit 4 of index register 0CH must be set to '1'. This will enable the Combined BIOS.

When Combined BIOS is enabled, any cycle address from 0C0000-0CFFFFFFH in the first megabyte will generate an EPROM cycle, as will a cycle with an address from 0F0000-0FFFFFFH. Also, the video BIOS will always appear at the top of the CPU memory space, in the highest megabyte of the 4 Gbyte address space, along with the system BIOS. However, video BIOS will never appear in the 16th megabyte (top of AT memory space), even if system BIOS is enabled to appear in that megabyte.

Appendix A: MS400 Standard Peripherals

DMA Controllers

The MS400 supports seven DMA channels using two Intel 8237-compatible DMA controllers, each having four channels. These controllers are capable of operation up to an 8.33 MHz DMA clock (16.67 MHz SYSCLK) rate. The two 8237 compatible DMA controllers in the MS400 provide a total of seven external DMA channels. Each channel has a 24-bit address output to access data throughout the lower 16 Mbyte system address space.

DMA controller-1 (channels 3:0) supports 8-bit peripherals and 8/16 bit memory peripherals. Each channel can transfer data in 64 Kbyte pages.

DMA controller-2 contains channels 7:4. Channel 4 is used to cascade DMA controller-1, and as such is not externally available. Channels 5 through 7 support 16-bit I/O transfers, as well as 16-bit system memory devices. Each channel can transfer data in 128 Kbyte pages. The DMA function improves the computer system by allowing external devices to transfer information directly to and from the system memory, without the data having first to go through the CPU.

The 74LS612 memory mapper also assists during DMA transfers, since the 8237-compatible controllers cannot directly support 16 Mbytes of address space. During the DMA cycle, one of two internal 8-bit latches hold the middle range address bits while the 74LS612 generates the upper range address bits. Once the hold request has been acknowledged, the DMA controller drives 24 address bits (for a total 16 Mbytes). The middle address bits of the 24-bit address range are held in two sets of 8-bit registers, one register for each DMA controller. The DMA controller drives the value to be loaded onto the data bus, and then issues an address strobe signal to latch the data bus value into these registers.

Memory Mapper

The MS400 has the equivalent of a 74LS612 to generate the upper address bits during a DMA cycle.

Source for the Memory Mapper 8237

for DMA-1 (channels 3:0), Address 23:16 A15:0

for DMA-2 (channels 7:5), Address 23:17 A16:1

Timer/Counters

Two timer/counters are also included in the MS400 architecture. These devices are functionally equivalent to Intel 8254 timers. Each timer provides three frequencies, or counters, for the system. The clock input frequency for these timers is 1.19318 MHz.

The timer-1, counter 0 is connected to the 8259A interrupt controller-1 IRQ0, and provides a system timer interrupt for a time-of-day, diskette time-out, or other system timing functions. Counter 1 generates a refresh-request signal. Counter 2 generates the tone for the speaker.

The timer-2 counter 0 is the fail-safe timer that can generate interrupts on the CPU's NMI line at regular intervals. This interrupt can be used by the operating system in order to prevent system lock-up during device failure or sizing tests. Counter 1 is not used. Counter 2 is placed in the one-shot mode, and is triggered by the refresh-request signal for the CPU speed control.

Refresh Generation Logic

Refresh operations are used to refresh memory on the DRAM memory bus and the 8/16 bit expansion bus. In order to simplify the design of 8/16 bit memory expansion boards and DRAM memory, the refresh address is provided by SA7:0, and SA9:8 are driven to the signal states of SA1:0. SA11 and SA10 have two additional counter bits allocated for RAM refresh requirements. The remaining addresses are in an undefined state during the refresh cycle. The refresh operations are driven by Timer-1 counter.

1. A refresh request is generated every 15.6 microseconds.

Refresh & DMA Arbitration Logic

The MS400 contains circuitry to control a refresh cycle. A 74LS612 equivalent 8-bit counter outputs the refresh addresses onto the memory bus when the refresh signal is pulled low.

There are two possible sources for a hold request to the CPU. Either the DMA controller issues a hold request or the output of counter 1 in the 8254 makes a low to high transition. The HOLD line is active when either source is requesting a hold. The hold request from the DMA clock and the request from the timer is sampled on the falling edge of the DMA clock.

MS400

If the DMA controller hold request wins the arbitration, the HOLD is asserted, and it waits for a signal back from the CPU. When the DMA controller is finished, it negates its hold request signal to the arbiter. The arbitration then switches signal to the REFRESH cycle if a HOLD is pending from the counter of Timer-1. Otherwise, the arbiter inactivates the HOLD line and returns control to the CPU.

If the refresh cycle wins the arbitration, the HOLD is asserted and the MS400 pulls the REFRESH# line low. REFRESH# remains low for four SYSCLK rising edges. On the fourth rising edge of SYSCLK, the HOLD line is deactivated. However, if there is a pending hold request from the DMA controller on the fourth rising edge of SYSCLK, the REFRESH cycle is extended for one more SYSCLK cycle. The HOLD request arbiter then acknowledges the hold request from the DMA controller.

NMI and Port B Logic

The MS400 contains non-maskable interrupt (NMI) signal generation logic. An NMI can be caused by an I/O error or by a parity error, or any expansion boards that pull the IOCHCK# line low. An NMI can also be caused by the timeout of the fail-safe timer (Timer-2, counter 0).

At power-up, the NMI signal is masked off. NMI is enabled by writing to I/O address 070H with bit 7 low. NMI is disabled by writing to I/O address 070H with bit 7 high. I/O address 061H bits 7 and 6 indicate the source of an NMI interrupt. I/O 061H bit 7 is set (parity error) when system memory detects a parity error. This interrupt is enabled by setting I/O 061H bit 2 to low. To reset the parity error, set 061H bit 2 to high (disable parity interrupt), and then clear it to low (enable parity interrupt). I/O port 061H bit 6 is enabled (IOCHCK# NMI) if an expansion board asserts IOCHCK# on the AT bus. This interrupt is set by setting I/O port 061H bit 3 to high (disable IOCHCK#, NMI) and then clearing it to low.

Interrupt Controllers

Two programmable interrupt controllers, compatible with the Intel 8259A devices, act as a system wide interrupt manager for an IBM PC/AT system. The interrupt controller efficiently determines when and which I/O device is serviced by the CPU. The inter-

rupt controller-1 is set to master mode, and the interrupt controller-2 is set to slave mode.

The two cascaded interrupt controllers provide a total of 15 possible interrupt sources. One of these interrupt request lines is used internally, providing a total of 14 possible external interrupt sources.

<u>Line</u>	<u>Source</u>
IRQ0	Timer-2 Counter 0 Output
IRQ1	Keyboard Controller
IRQ2	Interrupt from 8259-2
IRQ3	Serial Port 2 (COM2)
IRQ4	Serial Port 1 (COM1)
IRQ5	Parallel Port (Mouse Port)
IRQ6	Floppy Disk Controller
IRQ7	Parallel Port 1 (Printer)
IRQ8	Real Time Clock
IRQ9	Expansion Bus Pin B04
IRQ10	Expansion Bus Pin D03
IRQ11	Expansion Bus Pin D04
IRQ12	Expansion Bus Pin D05
IRQ13	Co-processor Error
IRQ14	Hard Disk
IRQ15	Expansion Bus Pin D07

SECTION 6. DC SPECIFICATIONS

DC CHARACTERISTICS ($V_{CC} = 5V \pm 5\%$)

Symbol	Parameter	Test Conditions	Min.	Max.	Units
V_{IL}	Input Low Voltage	(Note 1)	-0.3	0.8	V
V_{IH}	Input High Voltage		2.2	$V_{CC}+0.5$	V
V_{OL}	Output Low Voltage		2.2	0.4	V
V_{OH}	Output High Voltage		2.4	—	V
I_{CC}	Power Supply	25 MHz 33 MHz	—		mA mA
I_{LI}	Input Leakage Current	$0V < V_{IN} \leq V_{CC}$	0	10	μA
I_{LO}	Output Leakage Current	$0.4 < V_{OUT} < V_{CC}$	0	50	μA
C_{IN}	Input Capacitance	(Note 2)	—	7	pF
C_{CLK}	CLK2 Input Capacitance	(Note 2)	—	20	pF

NOTES:

1. Minimum value is not 100% tested.
2. Sampled only.

SECTION 7. THERMAL SPECIFICATIONS

This section gives the thermal characteristics of the 208-pin PQFP package, and the resulting maximum ambient temperature allowable to ensure proper operation of the MS400. Calculations were made by the formula:

$$\text{MAX } T_A = \text{MAX } T_J - P\Theta_{JA}$$

Where: $\text{MAX } T_J = xxx^\circ$ and
 $P = 5V * I_{CC \text{ MAX}}$

	Airflow — ft/min			
	0	200	400	600
Θ_{JA}	34.0	25.1	21.5	

Table x. Θ_{JA} ($^\circ C/W$) vs. Airflow — ft/min

	Airflow — ft/min			
	0	200	400	600
20 MHz Max. T_A	71.4	75	76.4	
25 MHz Max. T_A	68	72.5	74.2	
33 MHz Max. T_A	62.5	69.5	70.8	
50 MHz Max. T_A	51	60	63.5	

Table x. Maximum Ambient Temperature Allowable vs. Airflow ($^\circ C$)

MS400

SECTION 8. AC SPECIFICATIONS & TEST MEASUREMENT POINTS

33 MHz MS400 AC CHARACTERISTICS — ADVANCE INFORMATION

($V_{CC} = 5V \pm 5\%$, $C_L = 85 \text{ pF}$)

Symbol	Parameter	Min.	Max.	Units	Notes
t_1	Frequency	8	33	MHz	
t_2	CLKOSC Period	30	125	ns	
t_3	CLK High Time	12		ns	
t_4	CLK Low Time	12		ns	
t_5	ADS* to CLKIN Setup Time	5		ns	
t_6	ADS* to CLKIN Hold Time	5		ns	
t_7	MIO*, D/C*, W/R*, A31, A26:2, BE3:0 to CLKIN Hold Time	10		ns	
t_8	MIO*, D/C*, W/R*, A31, A26:2, BE3:0 to CLKIN Setup Time	2		ns	
t_9	HLDA to CLKIN Setup Time	5		ns	
t_{10}	HLDA to CLKIN Hold Time	5		ns	
t_{11}	SNPBUSY to CLKIN Setup Time	7		ns	
t_{12}	SNPBUSY to CLKIN Hold Time	3		ns	
t_{13}	SMEMDIS to CLKIN Setup Time	10		ns	
t_{14}	SMEMDIS to CLKIN Setup Time	2		ns	
t_{15}	BLAST* to CLKIN Setup Time	8		ns	
t_{16}	BLAST* to CLKIN Hold Time	4		ns	
t_{17}	MWB to CLKIN Setup Time	12		ns	
t_{18}	MWB to CLKIN Hold Time	2		ns	
t_{19}	Valid Address to Valid KEN#		12	ns	
t_{20}	EADS* Period		1CLKIN		
t_{21}	EADS* to CLKIN Valid Delay		14	ns	
t_{22}	SNPBUSY to IOCHRDY Active Delay		15	ns	
t_{23}	SNPBUSY to IOCHRDY Inactive Delay		17	ns	
t_{24}	HOLD to CLKIN Active Delay		10	ns	
t_{25}	NMI to CLKIN Active Delay		20	ns	
t_{26}	KBP21 to A20M* Active Delay		10	ns	
t_{27}	LBA# to CLKIN falling Setup time		4	ns	
t_{28}	PERROR* to CLKIN Setup Time	10		ns	
t_{29}	PERROR* to CLKIN Hold Time	5		ns	
t_{30}	CLK1 to CLK2 skew		1	ns	
t_{31}	MA10:1, MA0A, MA0B to CLKIN Active Delay		17	ns	
t_{32}	MA10:1, MA0A, MA0B to CLKIN Inactive Delay		14	ns	
t_{33}	RAS3:0* to CLKIN Active Delay		16	ns	
t_{34}	RAS3:0* to CLKIN Inactive Delay		16	ns	
t_{35}	CAS3:0A*, CAS3:0B* to CLKIN Active Delay		15	ns	
t_{36}	CAS3:0A*, CAS3:0B* to CLKIN Inactive Delay		13	ns	
t_{37}	RDOEA*, RDOEB* to CLKIN Active Delay		16	ns	
t_{38}	WRCEA*, WRCEB* to CLKIN Inactive Delay		14	ns	
t_{39}	WRCEA*, WRCEB* to CLKIN Active Delay		15	ns	
t_{40}	WRCEA*, WRCEB* to CLKIN Inactive Delay		13	ns	
t_{41}	MWE* to CLKIN Active Delay		17	ns	
t_{42}	MWE* to CLKIN Inactive Delay		14	ns	
t_{43}	BRDY* to CLKIN Active Delay		14	ns	
t_{44}	BRDY* to CLKIN Inactive Delay @15pf		17	ns	

NOTES:

1. Guaranteed by characterization. Not 100% tested.
2. To guarantee recognition on a specific clock edge.

33 MHz MS400 AC CHARACTERISTICS — ADVANCE INFORMATION

(V_{CC} = 5V ±5%, C_L = 85 pf)

Symbol	Parameter	Min.	Max.	Units	Notes
t ₅₁	LA17-23 valid before BALE asserted	56		ns	
t ₅₂	LA17-23 valid before BALE negated	116		ns	
t ₅₃	SA0-19 & SBHE valid before BALE negated	28		ns	
t ₅₄	SA0-19 & SBHE valid before IOW#, IOR# asserted	88		ns	
t ₅₅	BALE asserted before BALE negated	30		ns	
t ₅₆	BALE asserted before IOW#, IOR# asserted	90		ns	
t ₅₇	BALE asserted before LA17-23 invalid	90		ns	
t ₅₈	BALE negated before LA17-23 invalid	22		ns	
t ₅₉	MEMR#, MEMW#, SMEMR#, SMEMW#, IOR#, IOW# negated before SA0-19 invalid	32		ns	
t ₆₀	MEMR#, MEMW#, SMEMR#, SMEMW#, IOR#, IOW# negated before next BALE asserted	36		ns	
t ₆₁	LA17-23 valid to MEMCS16# valid		96		
t ₆₂	OVS# setup to S1CLK falling edge	10		ns	
t ₆₃	OVS# hold from S1CLK falling edge	20		ns	
t ₆₄	IOCHRDY# asserted to MEMR#, MEMW#, SMEMW#, SMEMR#, IOR#, IOW# negated	116		ns ns	
t ₆₅	IOCHRDY# asserted to the next BALE invalid	164		ns	
t ₆₆	IOCHRDY# asserted to SA0-19, SBHE	164			
t ₆₇	MEMR#, IOR#, SMEMR# negated to read data invalid	0		ns	
t ₆₈	MEMR#, IOR#, SMEMR# negated to data bus float		30	ns	
t ₆₉	MEMW#, SMEMW#, IOW# negated to write data invalid	25		ns	
t ₇₀	IOCHRDY# negated hold time	40		ns	
t ₇₁	IOCHRDY# asserted setup time to S1CLK rising edge	34		ns	
t ₇₂	SA0-19, SBHE# valid before IOCS16# valid		160	ns	
t ₇₃	BALE asserted before IOCS16# valid		160	ns	
t ₇₄	AEN valid before BALE asserted	45		ns	
t ₇₅	AEN valid before BALE negated	100		ns	
t ₇₆	AEN valid before IOR#, IOW# asserted	100		ns	
t ₇₇	IOR#, IOW# negated before AEN invalid	30		ns	
t ₇₈	MEMR#, IOR#, SMEMR# asserted to read data enable	30		ns	
t ₇₉	LA17-23 invalid to MEMCS16# float delay	0		ns	
t ₈₀	SA0-19 invalid to IOCS16# float delay	0		ns	
t ₈₁	BYTEN0#-3# active to S1CLK (high to low) delay		54	ns	
t ₈₂	BYTEN0#-3# inactive to S1CLK delay		3	ns	

NOTES:

1. Guaranteed by characterization. Not 100% tested.
2. To guarantee recognition on a specific clock edge.

MS400

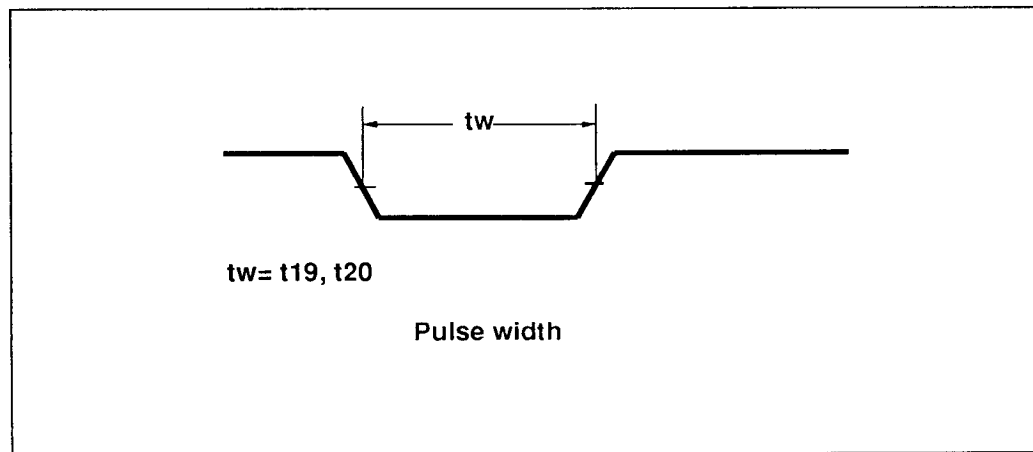
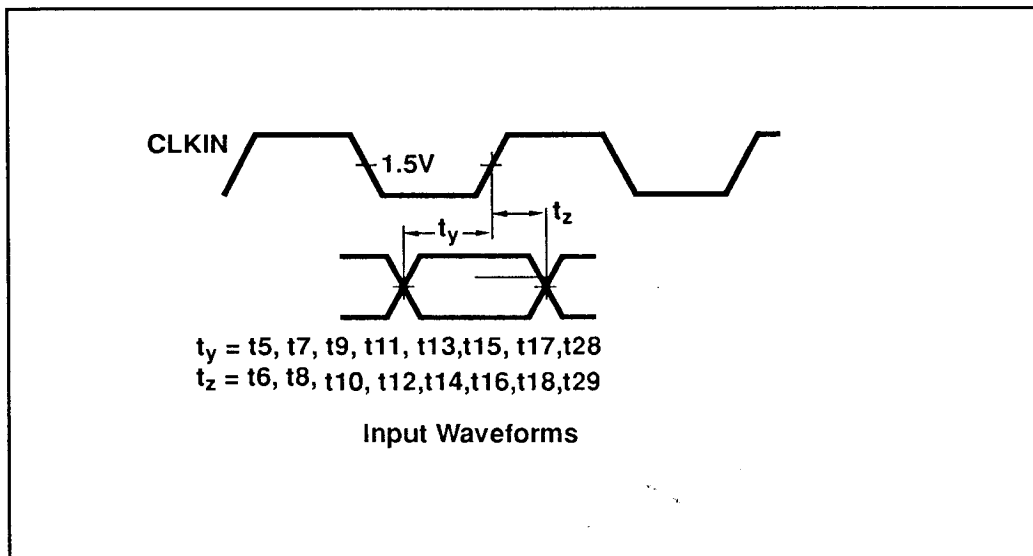
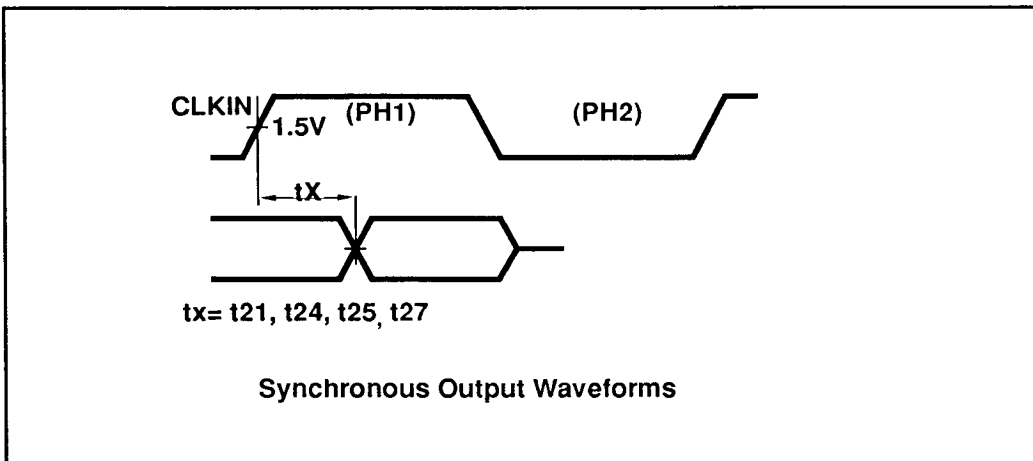
33 MHz MS400 AC CHARACTERISTICS — ADVANCE INFORMATION

($V_{CC} = 5V \pm 5\%$, $C_L = 85$ pf)

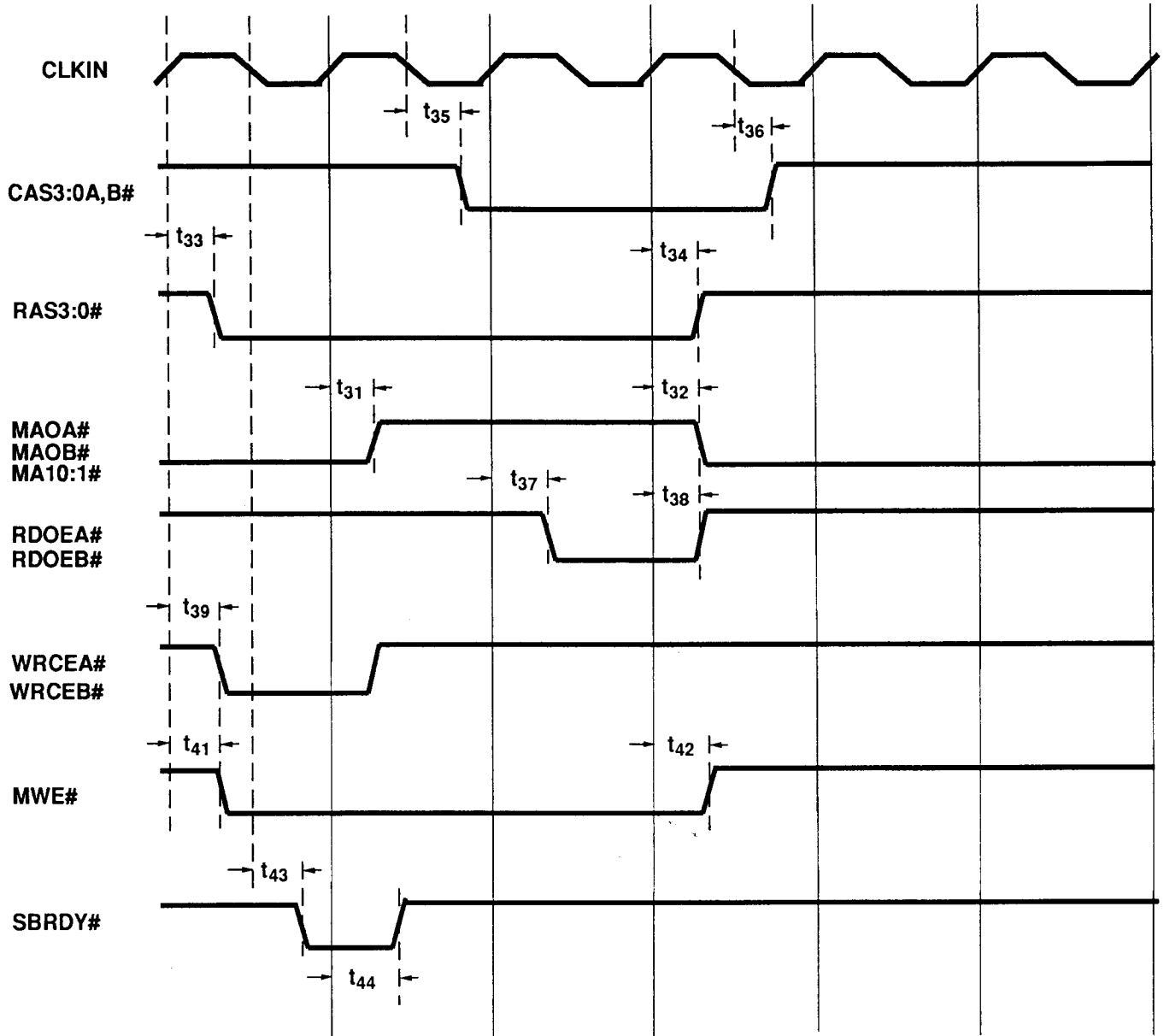
Symbol	Parameter	Min.	Max.	Units	Notes
t_{83}	BYTECK0/2 (low to high) to BALE delay		16	ns	
t_{84}	BYTECK0/2 (high to low) to BALE delay		30	ns	
t_{85}	IOR0# (high to low) to S1CLK delay		64	ns	
t_{86}	IOR0# (low to high) to S1CLK delay		40	ns	
t_{87}	CPYHL (high to low) to S1CLK delay		48	ns	
t_{88}	CPYHL (low to high) to S1CLK delay		46	ns	
t_{89}	CPYEN# (high to low) to S1CLK delay		48	ns	
t_{90}	CPYEN# (low to high) to S1CLK (high to low) delay		2	ns	
t_{91}	RTR/W# (high to low) to S1CLK delay		64		
t_{92}	RTR/W# (low to high) to S1CLK delay		60	ns	
t_{93}	RTDS active to S1CLK delay		63	ns	
t_{94}	RTDS inactive to S1CLK delay		45		
t_{95}	RTAS active to S1CLK delay		44	ns	
t_{96}	RTAS inactive to S1CLK delay		46	ns	
t_{97}	KBCS# (high to low) to BALE delay		44	ns	
t_{98}	KBCS# (low to high) to BALE delay		35	ns	
t_{99}	IRQn (high to low) to INTR delay		100	ns	
t_{100}	S1CLK (low to high) to CLKIN delay		18	ns	
t_{101}	S1CLK (high to low) to CLKIN delay		20	ns	
t_{102}	REFRESH# active to S1CLK delay		25	ns	
t_{103}	REFRESH# inactive to S1CLK delay		30	ns	
t_{104}	BYTECK1/3 (high to low) to S1CLK delay		16	ns	
t_{105}	BYTECK1/3 (low to high) to S1CLK delay		30	ns	
t_{106}	LOED# (low to high) to S1CLK (high to low) delay		54	ns	
t_{107}	LOED# (high to low) to S1CLK (low to high) delay		24	ns	
t_{108}	MEMR* active to S1CLK delay		33	ns	
t_{109}	MEMR* inactive to S1CLK delay		30	ns	
				ns	
				ns	
				ns	
				ns	
				ns	

NOTES:

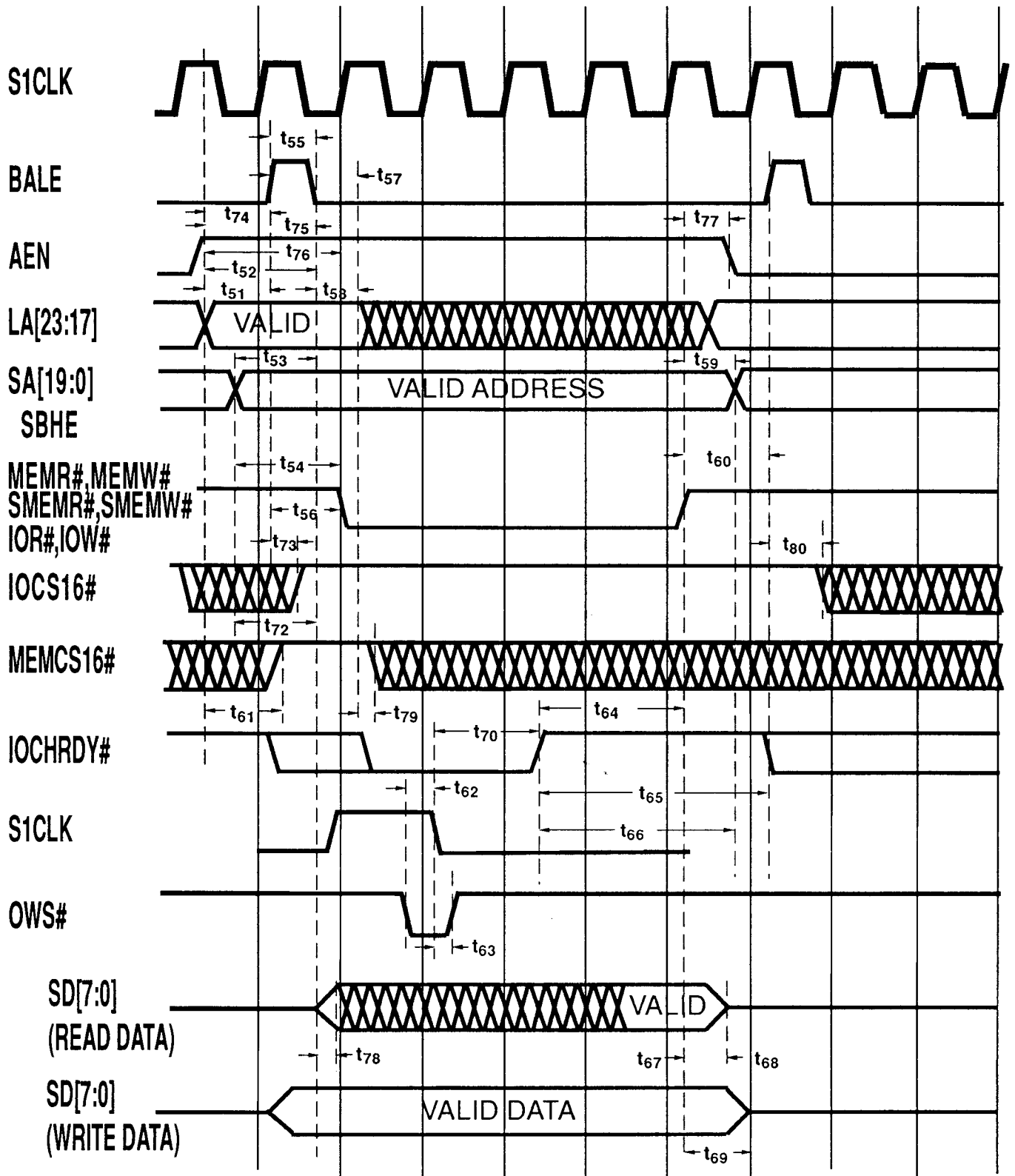
1. Guaranteed by characterization. Not 100% tested.
2. To guarantee recognition on a specific clock edge.



MS400

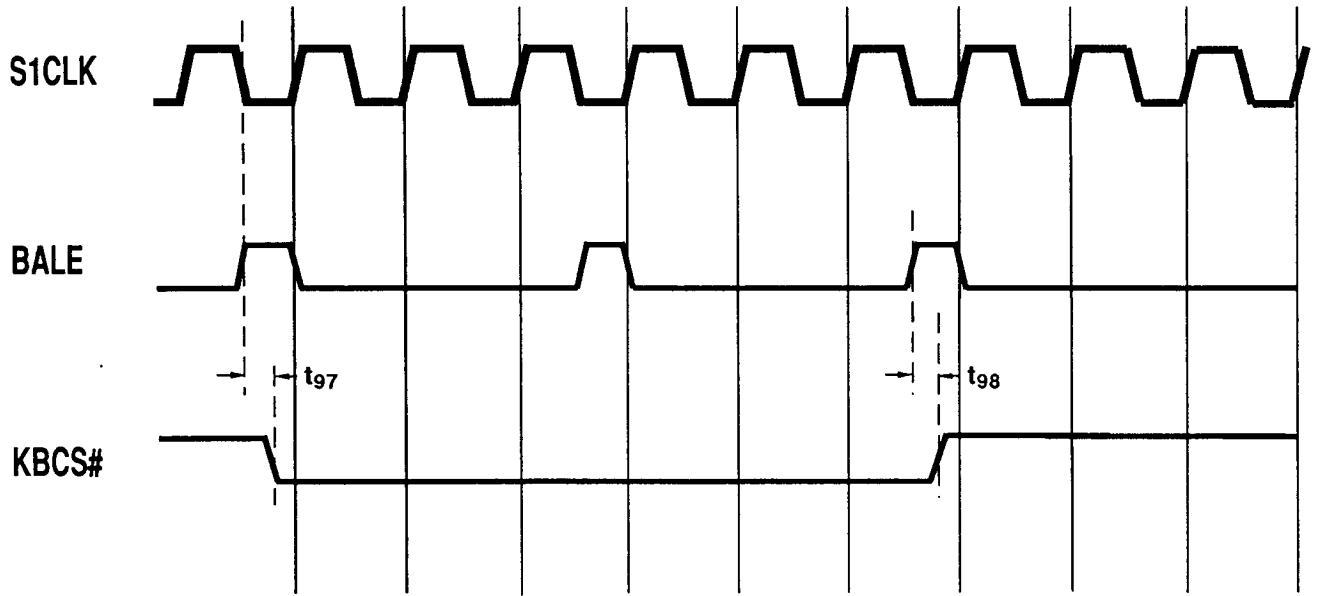


DRAM TIMING

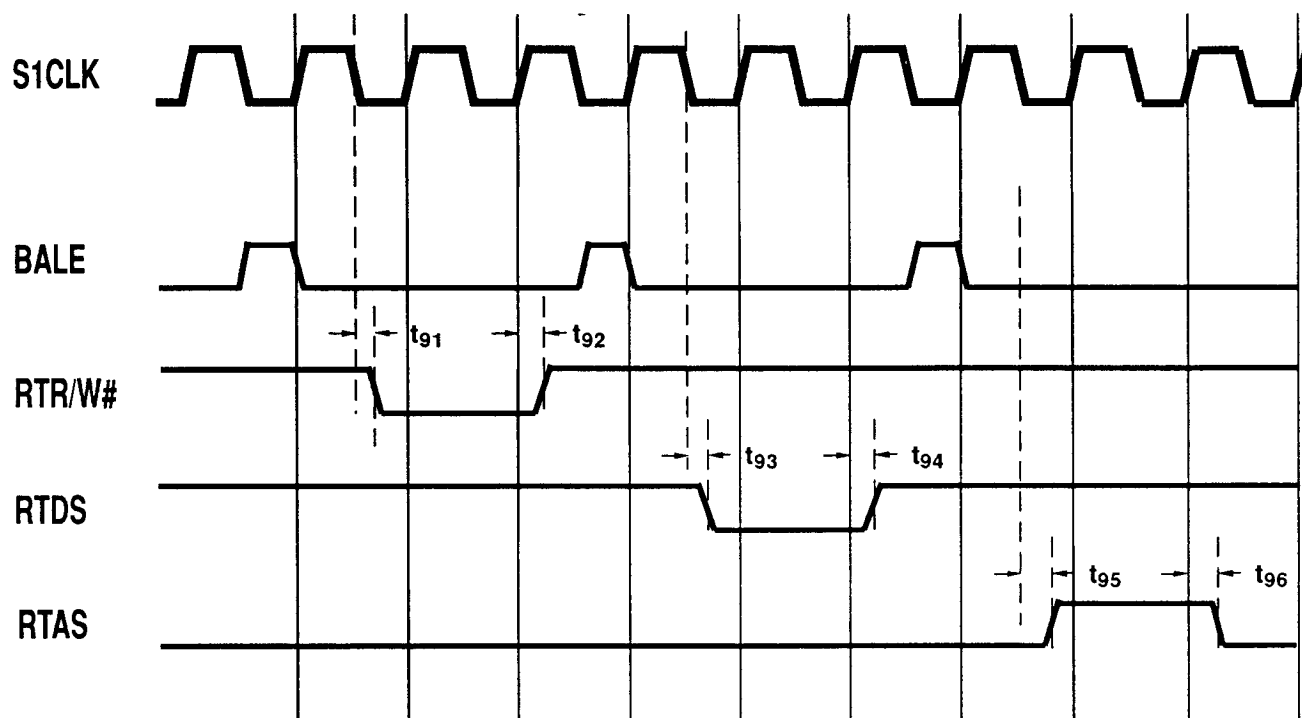


AT BUS TIMING

MS400

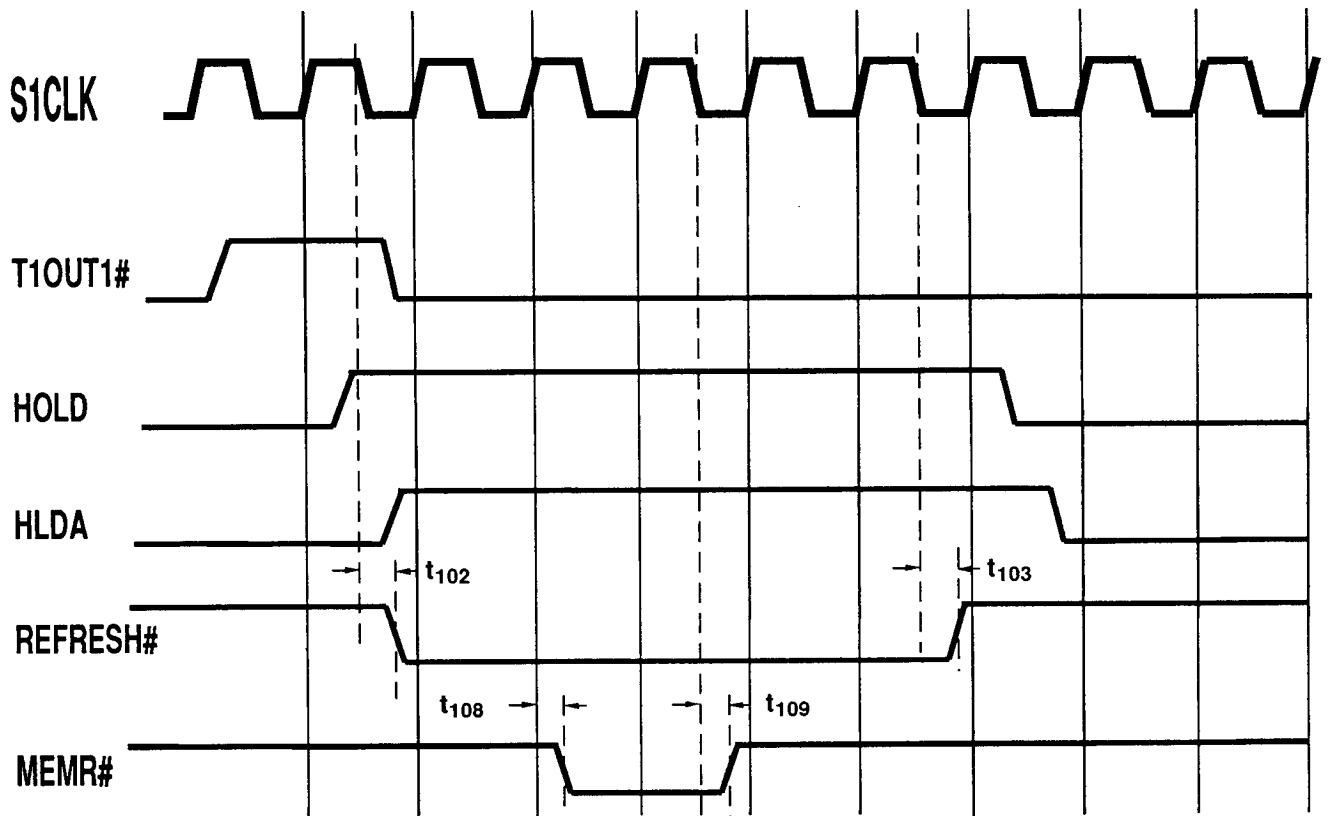


KEY BOARD CONTROLLER TIMING



REAL TIME CLOCK TIMING

MS400



REFRESH CYCLE

MS400

Appendix B: DATASHEET REVISION UPDATE

Following is the list of sections which have been significantly modified for the -002 version of this datasheet, and a brief description of the changes entailed.

Section 1. The pin-out was modified. The functionality of eleven pins was changed.

Section 2. The pin descriptions were modified in order to support the pin changes. Other pin descriptions were modified for clarification or completeness.

Section 4. Several new registers were added. The default register settings were slightly modified. Note that the two versions of the part require separate BIOS'es because of the register modifications.

Section 5.1. The functionality of CLK1 and CLK2 were altered.

Section 5.2.2 CPU clock switching description was added. CLKIN can now also be S1CLK/2 and S1CLK/2.5.

Section 5.2.3. The Local Bus resource section was added.

Section 5.2.4 The CPU/Simulcache Reset section was added.

Section 5.2.5. The Cacheability Support section was added.

Section 5.3.1. DRAM memory mixing description was added.

Section 5.3.3. DRAM read and write cycle functionality was modified, to interface both with 386/486 CPU's and the Simulcache.

Section 5.8.1. The memory remap description was added.

Section 5.8.9. The combined BIOS description was added.

Section 8. Some AC measurements were added to support the new pins now present.