MC88200UM/AD
REV 1

MC88200 USER'S MANUAL SECOND EDITION

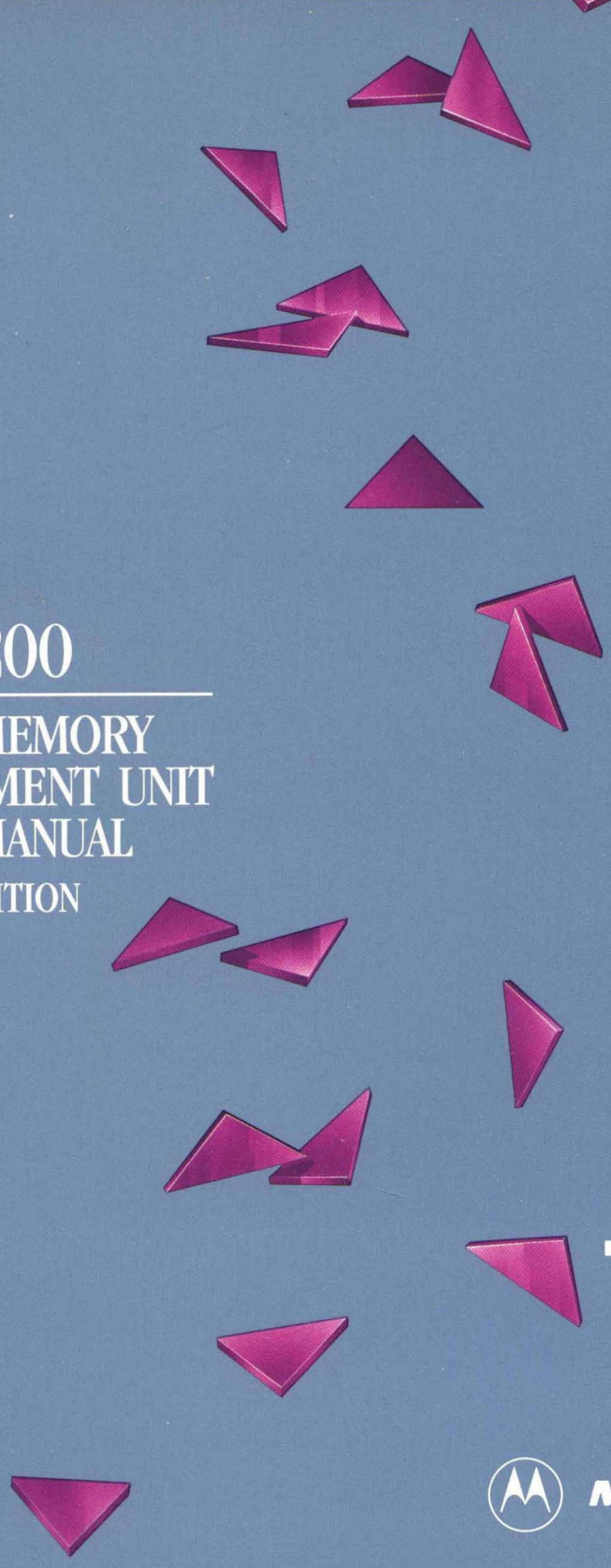
MOTOROLA

PRENTICE HALL

MC88200

CACHE/MEMORY MANAGEMENT UNIT USER'S MANUAL

SECOND EDITION



MOTOROLA

MOTOROLA

MC88200

CACHE/MEMORY MANAGEMENT UNIT USER'S MANUAL

second edition

PRENTICE HALL, Englewood Cliffs, N.J. 07632

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-567033-0

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall Hispanoamericana, S.A., *Mexico*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Simon & Schuster Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Concluded)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Concluded)

LIST OF TABLES

SECTION 1
INTRODUCTION

The MC88200 cache/memory management unit (CMMU) is a high-performance, HCMOS VLSI device, which combines demand-paged virtual memory management with 16K bytes of high-speed cache memory. Memory management performance is increased by two address translation caches (ATCs), which provide zero-wait-state address translation. The M88200 includes a nonmultiplexed, pipelined processor bus (P bus) interface and a multiplexed memory bus (M bus) interface. The MC88200 is specifically designed for the MC88100 reduced instruction set computer (RISC) processor, and both components feature 32-bit addressing and data with the P bus interface.

This section provides an introduction to the features, system configuration, bus interfacing, and registers of the MC88200. Figure 1-1 shows a simplified block diagram of the MC88200 CMMU.

1.1 FEATURES

The MC88200 ATCs consist of the page address translation cache (PATC) and the block address translation cache (BATC). The PATC is a 56-entry, fully-associative cache containing recently used translations for 4K-byte memory pages and is maintained by MC88200 hardware. The BATC is a 10-entry cache, containing translations for 512K-byte memory blocks and is loaded by system software.

The data cache is a four-way set-associative cache for instruction or data storage. The data cache features two software-selectable memory update modes (copyback and write-through). Integral cache coherency features allow the MC88200 to be used in multiprocessing systems incorporating shared memory. Some of the major features of the MC88200 are as follows:

Memory Management Unit (MMU) Features
- Two Logical Address Spaces of 4 Gbytes Each (User/Supervisor)
- Supervisor Access Protection
- Write Protection for User and Supervisor Accesses
- Used and Modified Flags Maintained in Page Translation Tables
- Probe Capability for Testing the Status of a Memory Location

Cache Features
- 16K-Byte, Four-Way, Set-Associative Physical Cache
- Zero-Wait-State Physical Cache Accesses
- Concurrent Address Translation and Cache Access

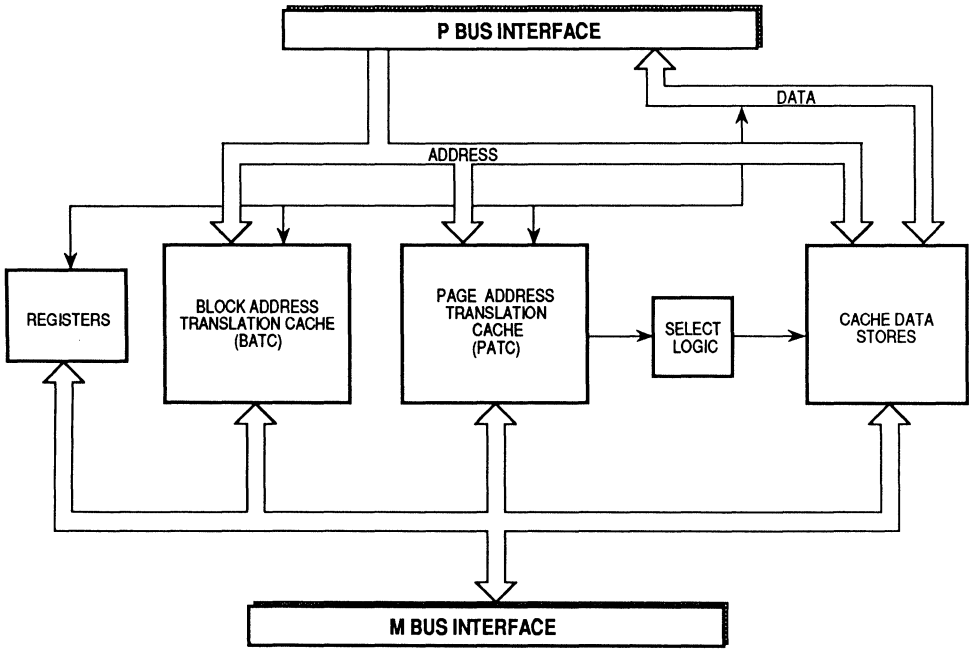


Figure 1-1. MC88200 Block Diagram

- Writethrough/Copyback with Area, Segment, Page Granularity
- Cache Flush and Invalidate Initiated Selectively by Software or Automatically by Hardware
- Cache Inhibit with Area, Segment, Page, and Block Granularity

Multiprocessor Support Features

- Bus Snoop Protocol for Maintaining Cache Coherency
- Lockable Bus Tenure That Provides Atomic Access to Semaphores to Support Multiprocessor Synchronization
- Data Cache and ATCs Flushable by Any Processor or Input/Output Device

1.2 SYSTEM OVERVIEW

In a simple microprocessor-based system, the central processing unit (CPU) is connected directly to memory. With this type of configuration, no memory mapping or protection capabilities are provided, and the addresses generated by the CPU directly identify the physical locations to be accessed. The number of physical devices in the system determines the range of usable logical address space of the CPU. This type system is unsuitable for multiple-task operations since there is no protection to prevent corruption of one task by another.

The MC88200 is designed to provide the mapping and protection needed to construct a multitasking, demand-paged virtual memory system. The MC88200 resides logically between the processor and the physical memory as shown in Figure 1-2. In this type configuration, the MC88200 controls all accesses to physical devices, and tasks can be prevented from accessing the resources owned by other tasks. When under the control of an operating system with virtual memory capabilities, the logical-to-physical mapping functions allow tasks to utilize the entire address space of the CPU without detailed knowledge of the physical attributes of the system. As shown in Figure 1-2, the logical address is generated by the processor and received by the MC88200 on its P bus address inputs. The MC88200 then performs translation and privilege checking for the logical address and, if the mapping is valid, drives the corresponding physical address to the cache. If the data requested by the processor is in the data cache (cache hit), the MC88200 returns this data to the processor one clock cycle after its address was driven on the P bus. If the data does not reside in the data cache (cache miss), an M bus access is performed to retrieve the data from memory (line fill).

A processor can be coupled to multiple MC88200s for increased data cache and ATC sizes. The number of MC88200s is limited only by the electrical characteristics of the P bus and M bus implementations.

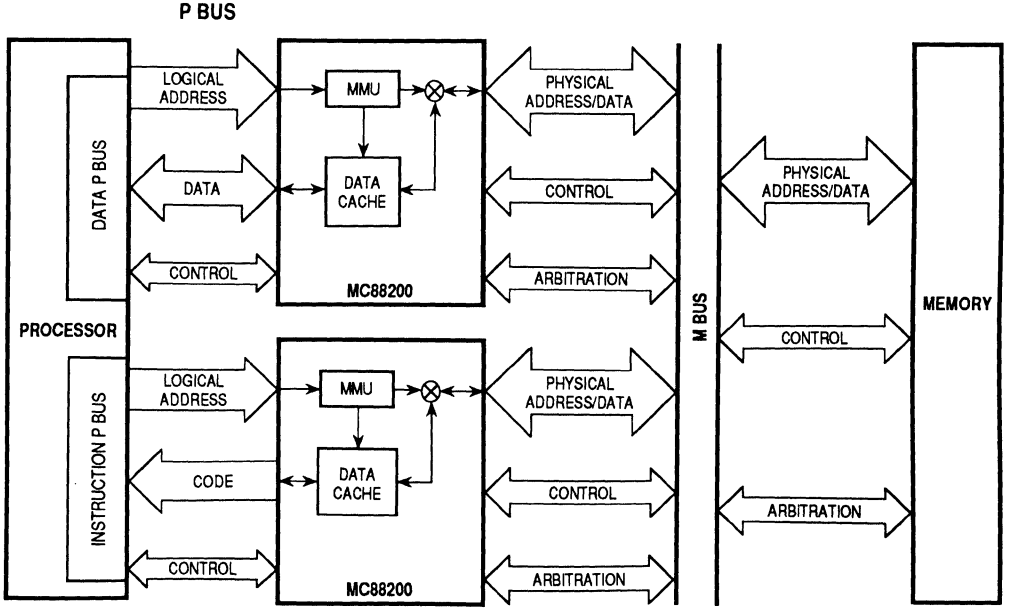


Figure 1-2. MC88200 System Configuration

1.3 ADDRESS TRANSLATION

The MMU translates logical addresses into physical addresses. Most translations are performed using one of the two ATCs, which translate fast enough to support processor reads and writes which complete in one clock cycle.

Both ATCs are fully associative memories, with entries that contain a logical address, the corresponding physical address, and status and protection attributes. Attributes for the translation include write protection, supervisor protection, data cache inhibit, and other status information. A valid bit in each ATC entry identifies the entry as being a valid translation. Two translation enable (TE) bits in the MMU define address translation as being enabled or disabled.

When a transaction is initiated by the processor and address translation is enabled, the logical address is presented to the ATCs where it is simultanously compared to all valid entries. If the logical address matches on the ATC entries (a "hit"), the MC88200 concatenates the physical base address in the entry with low-order bits from the logical address to form the complete physical address. If there is a hit on both ATCs, the BATC translation takes precedence.

If address translation is not enabled, the logical address from the processor is fed through the MMU unchanged to form the physical address. This mode is useful for nonvirtual memory systems that do not need address translation. One of two sets of protection and control attributes may be applied to the transaction, with selection dependent on the supervisory or user mode of the processor.

If there is no ATC entry that matches the processor's logical address, an ATC "miss" occurs, and the MC88200 searches translation tables in memory for a correct translation.

1.3.1 Address Translation Tables

When an ATC miss occurs, the MC88200 issues waits on the P bus and requests ownership of the M bus. When the M bus is available, the MC88200 completes the bus arbitration sequence, assumes M bus ownership, and performs the table search. If no exception conditions occur, the required translation descriptor is loaded into the PATC. The MC88200 then completes the memory access, translating the address through the PATC. When the memory access is complete, the MC88200 signals the processor that the memory transaction was successful.

Once the PATC is loaded with a valid page transaction, memory access to that page does not require a translation table search. This significantly improves the performance of the CMMU and reduces memory bus bandwidth requirements.

The translation tables describe the current logical-to-physical mappings of the system to the MC88200. These tables are organized as a tree structure in the physical address space

(see Figure 1-3). The operating system initializes and updates the tables; the MC88200 accesses the tables to obtain translations and maintain status information. The tree structure supported by the MC88200 contains two distinct levels in the supervisor/user address space:

1. The segment level divides each of the two 4 Gbyte logical address spaces (supervisor/user) into 1024 equivalent 4 Mbyte segments.
2. The page level divides each segment into 1024 equivalent page frames, each 4K bytes in size.

The tree structure allows the MC88200 to maintain only the required working set of translation tables; physical memory does not have to be allocated to tables corresponding to

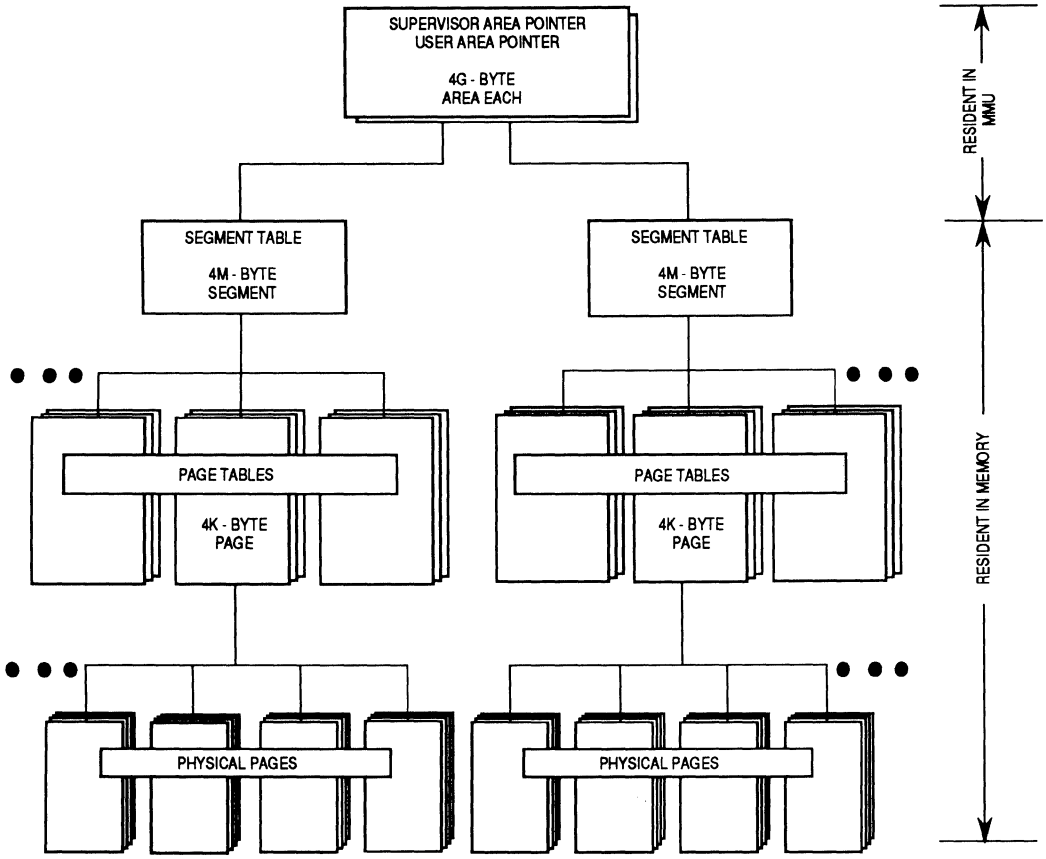


Figure 1-3. Translation Table Organization

unused portions of the logical address space. The tables and entries can be dynamically configured to meet the requirements of the operating system and the availability of physical memory.

1.3.2 Access Protection

The MC88200 protection mechanism provides privilege and write protection for the physical address space. The logical address space is partitioned into supervisor and user spaces; the supervisor address space is privileged, and the user address space is unprivileged. The levels of privilege allow a system to be built with protection from accidental or malicious corruption. The processor accesses either supervisor or user memory according to its current operating state (reflected by the P bus S/$\overline{U}$ signal); the MC88200 automatically enforces the proper access rights. Any segment, page, or block may be write protected or declared supervisor only.

The descriptors in the address translation tables contain information that indicates the protection at each particular level of the table structure. During a table search operation, the protection attributes at each level of the table structure are logically ORed to form the protection for the physical address. For example, if a segment or a page descriptor indicates write protection, then the physical address mapped by those descriptors is write protected. When the MC88200 detects a write transaction to a write-protected address, it aborts the transaction and signals a fault to the processor (write-protection violation). Similarly, when the MC88200 detects a user access to a supervisor-protected address, it signals a privilege-violation fault to the processor. Whenever a fault occurs, address and status registers are updated with information pertaining to the fault.

1.4 DATA CACHE

Because of locality of reference, instructions and data that are used in a program have a high probability of being reused within a short time (temporal locality). Additionally, instructions and data that reside in proximity to the instructions and data currently in use have a high probability of being used within a short time (spatial locality). To exploit these locality characteristics, the MC88200 contains a 16K-byte data cache.

The data cache improves the overall system performance by reducing average memory access time, by reducing the number of M bus cycles required to access memory, and by increasing the bus bandwidth available to other M bus masters. The MC88200 MMU can designate certain parts of memory as cachable and can designate a choice of memory update modes. In the copyback mode, memory is updated only when a cache line is replaced by other data. In the writethrough mode, memory is updated whenever the cache is updated by a write transaction. The MC88200 can be programmed to monitor (snoop) the M bus transactions of other M bus masters to maintain data cache coherency.

Figure 1-4 shows the memory organization of the MC88200 data cache. The cache is organized as four-way set-associative, consisting of 256 sets containing four lines of four

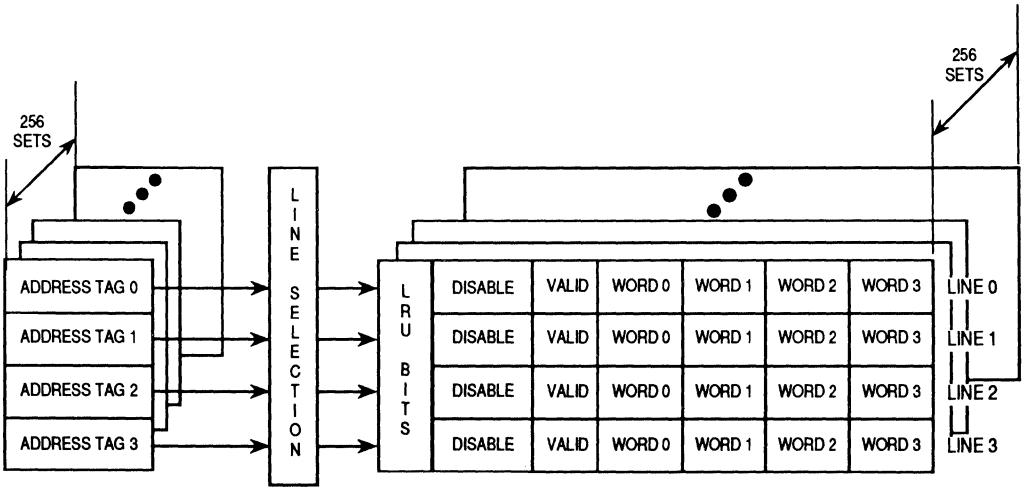


Figure 1-4. Data Cache Organization

words each. Each line has an associated physical address tag and associated disable and status bits. Each set has associated least recently used (LRU) information. When the processor initiates a memory access, the MC88200 selects a cache set, using the low-order 12 bits of the logical address. Simultaneously, the MC88200 translates the logical address into a physical address. The physical address is associatively compared to the four address tags in the cache set; if there is a match, cached data is returned to the processor. If there is no match, the MC88200 access memory via the M bus and loads the data into the cache. The selected line is loaded with the data read from physical memory; if the memory transaction was a write transaction, then the cached data is updated.

Using a four-word burst mode transaction, the MC88200 fills an entire cache line. The burst mode of operation not only fills the cache efficiently, but also captures adjacent instructions or data that are likely to be used in the near future due to locality of reference. The MC88200 implements the LRU line replacement algorithm.

1.5 BUS INTERFACES

The MC88200 CMMU interfaces with the P bus and the M bus. The following paragraphs provide a brief description of the bus interfaces.

1.5.1 P Bus Interface

The P bus is the dedicated communication link between a single MC88100 processor and one or more MC88200 CMMUs. The P bus is a high-speed, nonmultiplexed, synchronous

32-bit bus. The bus protocol is pipelined and optimized to provide the processor with either instructions or data at a peak rate of one word per cycle (80 Mbytes/sec at 20 MHz). Full 32-bit addressing and 32-bit data transfers are supported.

The MC88200 is always a P bus slave. All transactions are initiated by the processor; the MC88200 indicates completion or fault conditions through reply signals and the P bus fault registers. A chip select input allows for multiple CMMUs on the same P bus. Refer to **SECTION 7 APPLICATIONS INFORMATION** for detailed information on multiple CMMU interfacing.

1.5.2 M Bus Interface

The 32-bit synchronous M bus has multiplexed address and data. The MC88200 M bus interface maintains sufficient control signals to specify read and write operations, to establish exclusive resource usage (locks), to inhibit external caching, to signal bus errors, and to provide initialization. In addition, the MC88200 can function as a master or a slave device. The MC88200 acts as a slave when its control registers are being read or written via M bus transactions. The MC88200 also acts as a slave when monitoring (snooping) M bus transactions to maintain cache coherency. The MC88200 acts as a bus master when accessing memory or other M bus devices.

1.6 REGISTERS

The MC88200 registers occupy a page in the top 1M byte of the supervisor address space (control memory). The registers are accessible from either the P bus or the M bus. This accessibility allows both the local processor and other M bus masters to configure the MC88200 dynamically to perform diagnostics and to read the device status. The registers for the MC88200 are as follows:

Identification (ID) Register — contains a programmable 8-bit number used to map the CMMU register page in the top 1M byte of the supervisor address space. Also contains a device type code.

P Bus Fault Registers — used exclusively by the local processor, providing fault information related to transactions initiated by the P bus (local) processor.

System Interface Registers — control and status registers used when performing various CMMU operations. These registers also include the user and supervisor memory area pointers.

BATC Write Ports Registers — address loading ports for the BATC.

Cache Diagnostic Ports Registers — data exchange points and status information on the data cache sets.

Table 1-1 summarizes the functions of the MC88200 registers. For detailed information on the registers, refer to **SECTION 2 MEMORY MANAGEMENT, SECTION 3 DATA CACHE,** and **SECTION 6 OPERATIONS TIMING AND REGISTER DESCRIPTIONS.**

Table 1-1. Register Summary

Register	Title	Function	Access
IDR	ID Register	Contains the Device Type Code and Defines the MC88200 Device Address in the Control Memory Space	Read/Write
PFSR	P Bus Fault Status Register	Indicates P Bus Exception or Results of a Probe Transaction	Read/Write
PFAR	P Bus Fault Address Register	Indicates the Physical Address of a P Bus Exception	Read/Write
SCR	System Command Register	Programmed with MC88200 Commands for Probe Transactions, ATC Flushes, and Cache Flushes	Read/Write
SSR	System Status Register	Indicates Probe Results, M Bus Error, Data Copyback Error, Data Cache Flush Error, and Results of a Probe Transaction	Read/Write
SAR	System Address Register	Written to Pass Addresses for MC88200 Commands; Read to Obtain Address Where an M Bus Error Occurred	Read/Write
SCTR	System Control Register	Programmed with Parity Enable, Bus Snoop Enable, and Bus Arbitration Scheme	Read/Write
SAPR	Supervisor Area Pointer Register	Written with Area Pointer Segment Table Address and Control/Protection Information for Supervisor Address Space	Read/Write
UAPR	User Area Pointer Register	Written with Area Pointer Segment Table Address and Control/Protection Information for User Address Space	Read/Write
BWP7–BWP0	BATC Write Port Registers	Written to Load Block Address Translations into the BATC	Write Only
CTP3–CTP0	Cache Tag Port Registers	Provide Access to the Cache Tags of a Set for Diagnostic Purposes	Read/Write
CDP3–CDP0	Cache Data Port Registers	Provide Access to the Words in a Cache Set for Diagnostic Purposes	Read/Write
CSSP	Cache Set Registers	Provide Access to the LRU Information, Disable Bits, and Line Status of a Cache Set for Diagnostic Purposes	Read/Write

1

SECTION 2
MEMORY MANAGEMENT

This section provides the complete details of the MC88200 memory management functions. The MC88200 cache/memory management unit (CMMU) includes a memory management unit (MMU) that can support a demand-paged virtual memory environment. The memory management is 'demand' in that programs executed by the MC88100 do not specify required memory areas in advance but request them by accessing logical addresses. The physical memory is paged, meaning that it is divided into blocks of equal size, called page frames. The logical address space is divided into pages of the same size. The operating system assigns pages to page frames as they are required to meet the needs of programs.

One principal function of the MMU is the translation of logical addresses to physical addresses using translation tables stored in memory. These translation tables contain access and protection information for the translation. Most translations are performed using one of the two fully associative address translation caches (ATCs), which translate a logical address into a physical address in less than one clock cycle. The size of these caches is proportional to the number of CMMUs connected to the processor bus (P bus). The ATCs eliminate many of the memory accesses required for translation table searches, which significantly improves the performance of the CMMU and reduces memory bus bandwidth requirements.

An ATC entry includes the protection attributes for that mapping (e.g., write and/or supervisor protected), a data cache inhibit indicator, and other protection and control information. Before an address is translated through the ATC, the MC88200 checks the protection information. If the transaction conflicts with the protection (e.g., write to write-protected memory), the memory transaction is aborted with a fault, and the translation is not performed. Control registers are updated with information concerning the fault.

2.1 ADDRESS TRANSLATION

The function of the MMU is to translate logical addresses to physical addresses according to control information stored by the system in the MMU registers and in translation table trees resident in memory. The following paragraphs explain the address translation used by the MC88200.

2.1.1 General Flow for Address Translation

The MMU translates P bus logical addresses to memory bus (M bus) physical addresses using one of the two ATCs. The block address translation cache (BATC) contains entries

that translate 512K-byte blocks of memory. The page address translation cache (PATC) contains entries that translate 4K-byte pages of memory. The BATC and PATC entry lookups are performed concurrently. During address translation, cache set selection is performed. If the logical address misses both ATCs, then a translation table search creates a new PATC entry. Figure 2-1 illustrates the address translation process.

The MC88200 receives the processor logical address on the P bus. The MMU performs four functions concurrently:

1. The MMU performs an identify translation (physical address = logical adddress) if the translation enable (TE) bit is clear in either the
   a. Supervisor area pointer register (in supervisory mode) or
   b. User area pointer register (in user mode).
2. The MMU compares bits 31–19 of the logical address and the supervisor/user bit to each logical block address (LBA) entry in the BATC. If there is a hit, then the MMU creates a physical address by concatenating logical address bits 18–2 to the physical block address (PBA) from the BATC entry.
3. The MMU compares bits 31–12 of the logical address and the supervisor/user bit to each logical page address (LPA) in the PATC. If there is a hit, then the MMU creates a physical address by concatenating logical address bits 11–2 to the page frame address (PFA) from the PATC entry. If there is a BATC and a PATC hit, then the BATC entry is used for address translation.
4. Data cache set selection is performed using bits 11–4 of the logical address. The cache line is selected at the end of address translation if there is a hit in the data cache.

If the logical address misses both ATCs, then the MMU creates a new PATC entry by performing a translation table search, which involves traversing a two-level table of descriptors in memory to find a new PFA. The PFA is placed in the PATC along with its corresponding LPA and control bits. If the PATC is full, an entry is replaced using a first-in, first-out (FIFO) scheme. Protection and control bits accumulated during the table search are placed in the PATC with the new created entry.

The ATCs are searched again for an entry matching the upper bits of the logical address. The address translation process completes when one of the following two events occur:

1. An ATC match (hit) or
2. Faulted termination of the table search done either
   a. immediately, due to
      1) an invalid descriptor, or
      2) a supervisor privilege violation, or
      3) an M bus error, or
      4) a parity error, or
   b. at the end of a table search sequence due to a write-protection violation.

If the address translation is terminated due to a fault, the CMMU informs the processor that the transaction has faulted and returns the faulting physical address in the P bus fault

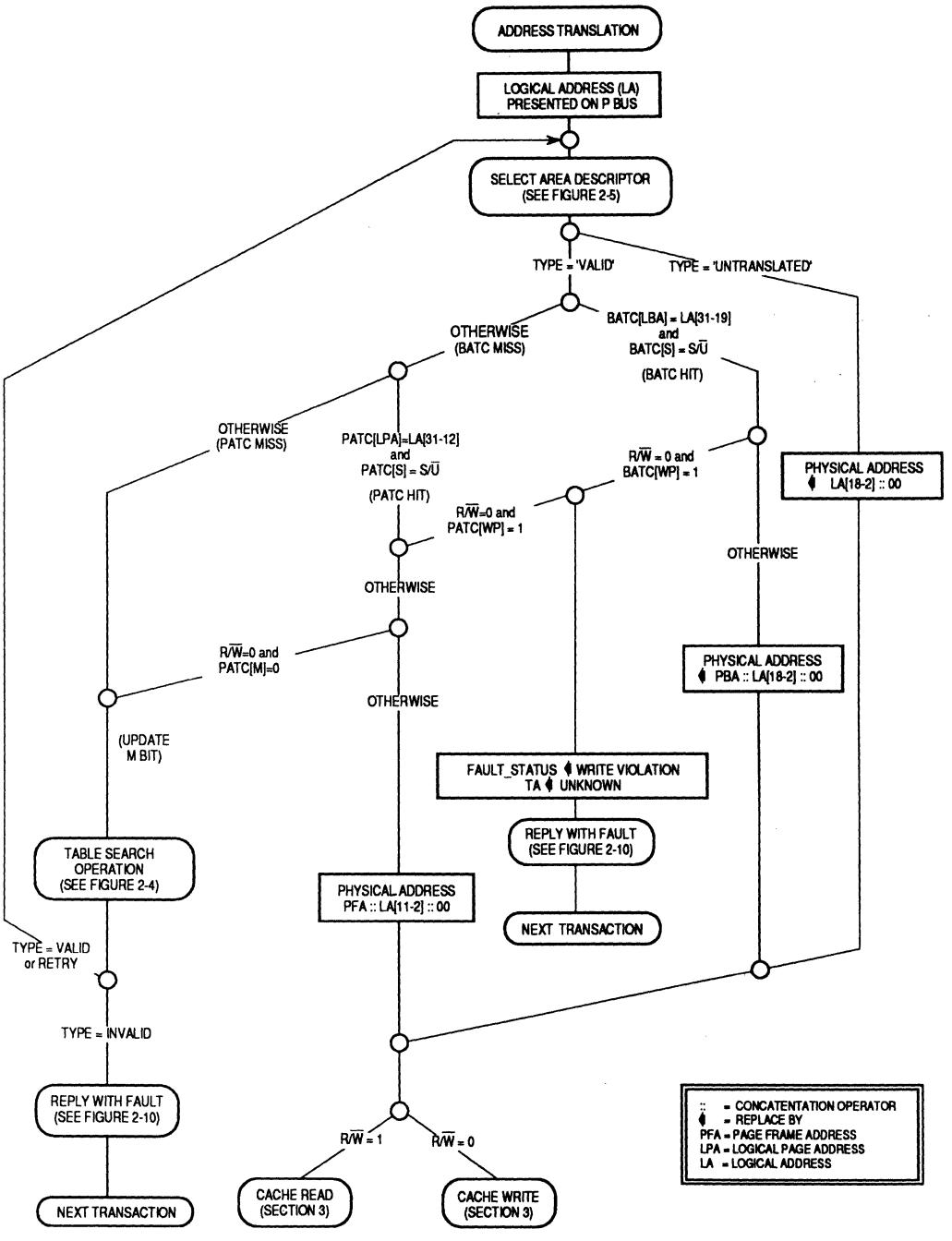


Figure 2-1. Address Translation Flowchart

address register (PFAR) and the fault type in the P bus fault status register (PFSR). Refer to **SECTION 5 BUS OPERATIONS** and **2.4 FAULTS** for more information on fault processing.

If there is an ATC hit and no fault, the physical page address generated in the transaction is presented to the data cache. If there is a 'hit' in the cache set selected at the begining of the translation and caching is not inhibited, data is read from/written to the cache. If there is a miss in the cache, a memory access is performed to fill a line in the cache. A memory access is also performed when caching is disabled (the address is cache inhibited) or the transaction is locked from the P bus.

2.1.2 Unmapped Translation

The MC88200 MMU provides the capability to present untranslated logical addresses from the P bus to the M bus, which is useful in nonvirtual memory applications where the use of translation tables is unnecessary. It is also used after system reset and before memory mapping is initialized.

Unmapped translation in supervisor mode is accomplished by clearing the translation enable (TE) bit in the supervisor area pointer register (SAPR). Unmapped translation in user mode is accomplished by clearing the TE bit in the user area pointer register (UAPR). In both cases, the physical address generated by address translation is identical to the logical (P bus) address. Protection and control bits in the area pointer registers are applied toward their respective identity translations.

2.2 ADDRESS TRANSLATION CACHES

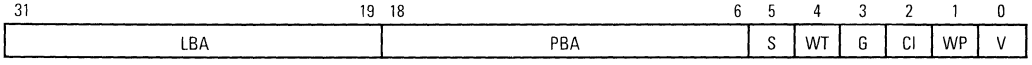
The ATCs contain entries for translating logical addresses into physical addresses. Each entry consists of three parts: the logical address, the physical address, and the protection and control information. The logical address corresponds to the memory address received from the processor over the P bus; the physical address is the address used for cache access, or placed on the M bus if needed to access physical memory. When the MC88200 receives a logical address over the P bus, it simultaneously searches all ATC entries for a matching logical address. When a match occurs (an ATC hit), the MC88200 replaces bits in the logical address with the corresponding physical address bits from the ATC entry. This translated physical address specifies a block or page of physical memory. The remaining bits in the P bus logical address are untranslated; these bits are an offset that specifies the exact data word in the physical block or page. The control and protection information is applied to the memory transaction.

The BATC provides physical address translation for 512K-byte blocks of memory. Normally, these blocks contain the operating system kernel or other high-use software. Since these are high-use blocks, caching address translation at the block level provides faster data cache access while avoiding ATC misses and their associated table searches. Eight of the ten BATC entries are available for use to the system software. The two remaining BATC entries are hardwired to map addresses in the control address space (peripheral devices).

The PATC is a 56-entry cache containing page (4K byte) logical addresses and their corresponding physical addresses. The PATC is maintained by the MMU. If a memory access misses both ATCs, then a PATC entry is created by a translation table search (see **2.3.3 Table Search Algorithm**). From then on, all translations for that page are performed by the PATC until the entry is replaced or invalidated.

2.2.1 BATC Entries/Write Ports

The BATC contains ten 32-bit entries that provide the address translation, control, and protection data. The format of these BATC entries is shown in the following list; they are loaded by writes to the BATC write ports.



LBA — Logical Block Address
This field contains the upper 13 bits of the logical address that maps to the associated physical address. If the V bit is set, logical address that matches the corresponding LBA field and S bit is mapped by the BATC.

PBA — Physical Block Address
This field contains the upper 13 bits of the physical address of the memory block.

S — Supervisor Mode Bit
This bit represents the value of the supervisor/user bit in the logical address (i.e., this bit should be considered an extension of the logical address). If the LBA matches the processor logical address but the value of the S bit is different, a BATC miss occurs, not a privilege violation.

WT — Writethrough
Memory update policies are described in detail in **3.2 MEMORY UPDATE POLICIES**. If the CI bit is set, this bit has no effect.
1 = Cache memory updates are performed using writethrough policy.
0 = Cache memory updates are performed using copyback policy.

G — Global
The value of the G bit is driven on M bus signal C5 if the BATC mapped access requires an M bus transaction (e.g., cache miss or write-once). Other M bus devices snoop the address driven on the M bus if this signal is asserted (see **SECTION 3 DATA CACHE**).
1 = Memory mapped by this entry is global memory.
0 = Memory mapped by this entry is local to this MC88200.

CI — Cache Inhibit

The CI bit affects caching only in this MC88200, although the value of the CI bit is driven on M bus signal C4. Other M bus devices may or may not use this signal.

1 = Data and/or instructions mapped by this entry are not cached in the MC88200.

0 = Data and/or instructions mapped by this entry can be cached in the MC88200.

WP — Write Protect

A write to write-protected memory fails and causes a write-protection violation.

1 = Memory mapped by this entry is write protected by the MC88200.

0 = Memory mapped by this entry can be written by the MC88200.

V — Valid

This bit must be set for address translation to occur. (Other BATC entries and the PATC can still be used for address translation.)

1 = Entry is valid.

0 = Address translation will not be performed using this entry in the BATC.

The following two hardwired BATC entries provide an identity (one-to-one) mapping of logical-to-physical addresses in the upper 1M byte of the supervisor address space (control memory). These entries are used even if TE = 0 in the SAPR.

LBA												PBA													S	WT	G	CI	WP	V
1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0	1

LBA												PBA													S	WT	G	CI	WP	V
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1

The control bits are programmed to indicate 1) supervisor access, 2) local address, 3) cache inhibit, 4) no write protect, 5) valid descriptor, and 6) writethrough policy.

The BATC write ports are used by system software to write entries into the BATC. Each port contains the logical block address, the corresponding physical address, and various protection and access bits. Eight ports are accessible by software, corresponding to BATC entries 0–7. Entries 8 and 9 are hardwired to map addresses in the control memory address space. Each port is one word (four bytes).

When the MC88200 is reset, the software-accessible BATC entries are marked invalid and must be loaded through these write ports. There must not be more than one BATC entry with the same value for the LBA field. This is considered a programming error and could result in unreliable operation of the MC88200.

31	19	18	6	5	4	3	2	1	0
LBA		PBA		S	WT	G	CI	WP	V

LBA — Logical Block Address

Contains the 13-bit logical address for the mapped memory block.

PBA — Physical Block Address

Contains the 13-bit physical address for the mapped memory block.

S — Address Space

This bit is considered an extension of the logical address: for memory blocks in the user address space, logical address bits 31–19 must match bits 31–19 of the BATC entry, and the S bit must be clear for translation to take place. For the supervisor address space, the S bit must be set. If the logical address matches the logical block address but the supervisor/user bit does not match the S bit, the BATC entry is not used for address translation. The S bit is not a protection bit; a mismatch does not cause a protection violation but instead causes a BATC miss.

1 = Supervisor address space.

0 = User address space.

WT — Writethrough

This bit is loaded to establish the memory update policy for the block. See **3.2 MEMORY UPDATE POLICIES** for more information on the memory update policy. If the CI bit is set, this bit has no effect.

1 = Writethrough memory update policy.

0 = Copyback memory update policy.

G — Global

Software sets this bit to indicate that the memory area may be shared among multiple MC88200s. The value of the G bit is driven on M bus signal C5 if the BATC mapped access sequences an M bus transaction (e.g., cache miss or write-once). Other M bus devices snoop the address driven on the M bus if this signal is asserted (see **SECTION 3 DATA CACHE**).

1 = Memory mapped by this entry is global memory.

0 = Memory mapped by this entry is local to this MC88200.

CI — Cache Inhibit

The CI bit affects caching only in this MC88200, although the value of the CI bit is driven on M bus signal C4 if the BATC mapped access requires an M bus transaction. Other M bus devices may or may not use this signal. A P bus locked transaction forces a cache inhibited M bus access.

1 = Data and/or instructions mapped by this entry are not cached in the MC88200.

0 = Data and/or instructions mapped by this entry can be cached in the MC88200.

WP — Write Protect

Software sets this bit to establish write protection for the memory block. A write transaction to the memory block causes a write protection violation exception.

1 = Memory mapped by this entry is write protected by the MC88200.

0 = Memory mapped by this entry can be written by the MC88200.

V — Valid

If this bit is clear, address translation will not be performed using this entry in the BATC. (Other BATC entries and the PATC can still be used for address translation.)

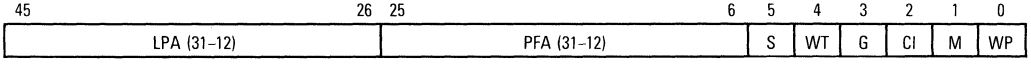
1 = Descriptor is valid.

0 = Descriptor is not valid.

2

2.2.2 PATC Entries

The PATC contains 56 entries that provide address translation as well as control and protection information for logical-to-physical page translation. The format of the PATC entries is as follows:



LPA — Logical Page Address

This field contains the upper 20 bits of the logical address that maps to the associated physical address.

PFA — Page Frame Address

This field contains the the upper 20 bits of the physical address corresponding to the LPA. This value and the status bits described below are loaded by the CMMU at the completion of a translation table search.

S — Supervisor Mode Bit

This bit represents the value of the supervisor/user bit in the logical address (i.e., this bit is an extension of the logical address). It does not function as a protection bit in the strict sense; if the LPA matches the processor logical address but the value of the S bit is different, a PATC miss occurs, not a privilege violation.

WT — Writethrough

Memory update policies are described in detail in **SECTION 3 DATA CACHE**. If the CI bit is set, this bit is ignored.

1 = Cache memory updates are performed using writethrough policy.

0 = Cache memory updates are performed using copyback policy.

G — Global

The value of the G bit is driven on M bus signal C5 if the BATC mapped access requires an M bus transaction (e.g., cache miss or write-once). Other M bus devices snoop the address driven on the M bus if this signal is asserted (see **SECTION 3 DATA CACHE**).

1 = Memory mapped by this entry is global memory.

0 = Memory mapped by this entry is local to this MC88200.

CI — Cache Inhibit

The CI bit affects caching only in this MC88200, although the value of the CI bit is driven on M bus signal C4. Other M bus devices may or may not use this signal.

1 = Data and/or instructions mapped by this entry are not cached in the MC88200.

0 = Data and/or instructions mapped by this entry can be cached in the MC88200.

M — Modified

Once a write access is performed to any location on the page, the M bit remains set until the entry is replaced or invalidated. The M bit may be set by the CMMU, but is never cleared.

1 = Page mapped by this entry has been modified (write access).

0 = Page mapped by this entry has not been written.

WP — Write Protect

A write to write-protected memory causes a write-protection violation (P bus error response).

1 = Memory mapped by this entry is write protected.

0 = Memory can be written by the MC88200.

2.2.3 PATC Load Algorithm

When ATC misses cause a table search (see **2.3.3 Table Search Algorithm**), a new entry is created for the PATC. The 20 most significant bits of the resulting physical address are placed in the PFA field of the entry; the 20 most significant bits of the logical address are placed into the LPA. The S bit reflects the value of the P bus S/$\overline{U}$ signal. Control and protection bits (WT, G, and CI) are filled in with the logical OR of the corresponding area, segment, and page descriptor control and protection bits. The WP bit is filled with the logical OR of the corresponding segment and page descriptor fields, and the V bit reflects the logical AND of the area TE bit and the segment and page V bits. No PATC entry is created if the appropriate TE bit is clear or a V bit is found clear during a table search. The M bit is set if the memory access was a write. The page descriptor U bit is set the first time the page descriptor is accessed, due to a table search or probe. Refer to **2.5 PROBE TRANSACTIONS** for information on probes.

The PATC load algorithm uses a first-in, first-out (FIFO) policy to select an entry for replacement. When a new entry is loaded into the PATC, all other entries are shifted by one, and the oldest entry is shifted out (replaced). In this manner, the PATC caches the 56 most recently created page address translations.

2.2.4 Invalidation of PATC Entries

The PATC reduces the number of memory accesses, thereby increasing system performance. Certain application programs may require specific mapping and protection changes during operation. When a map change is required (due to a task swap), the affected PATC

entries must be invalidated by system software. When a PATC entry is invalidated, addresses will no longer be translated through that entry. The PATC entries are invalidated through software by writing the invalidation command to the system command register (SCR).

To invalidate the PATC entries, the system software must perform the following steps:

1. For systems implementing semaphore protection, the software must gain ownership of the required semaphore(s).
2. For segment and page invalidations, the system address register (SAR) must be loaded with the logical address to be invalidated.
3. The invalidate command is written to bits 5–0 of the SCR (bits 31–6 should be zero):

   11×0gg Invalidate user PATC entries
   11×1gg Invalidate supervisor PATC entries

   where gg is the desired granularity (the 'x' bit is not used):

   01 Page (one single PATC entry)
   10 Segment (all PATC entries that map into one segment)
   11 All (user or supervisor PATC entries depending on command)

   Once the command is written, the MMU automatically begins the invalidation process. The process is complete in two clock cycles.

When multiple MC88200s are connected to the P bus, the invalidation command must be issued to all MC88200s. When multiple MC88200s are attached to a processor, the logical addresses are partitioned across the MC88200s by external hardware (e.g., the address bits are decoded to generate the chip select signal).

2.3 TRANSLATION TABLE DETAILS

When there is no translation for an address in either ATC, the MMU creates a new PATC entry by performing a translation table search. The logical address space is partitioned into three levels: area, segment, and page. The user and supervisor memory areas correspond to the state of the supervisor/user (S/$\overline{U}$) input. The memory in each area is defined by the descriptors in the two area pointer registers (SAPR, UAPR). Each area descriptor has a segment table base address (STBA) field pointing to a segment table. The segment tables divide memory into 4M-byte segments. Each descriptor in the segment tables (segment descriptors) points to a page table. The descriptors in the page tables contain the page frame addresses. The descriptors at the three levels form a tree structure in memory. When the MMU performs a translation table search, it follows the tree structure to arrive at the required physical page address.

Figure 2-2 shows the translation table structure. Not all the segment or page descriptors need to be initialized; when the MMU finds an invalid descriptor, it signals a page or segment fault. (The exception handler may then load a valid descriptor as described in **2.4 FAULTS**.) For a table search to occur, either the supervisor access needs the SAPR to be valid (enabled) or the user access needs the UAPR to be valid (enabled).

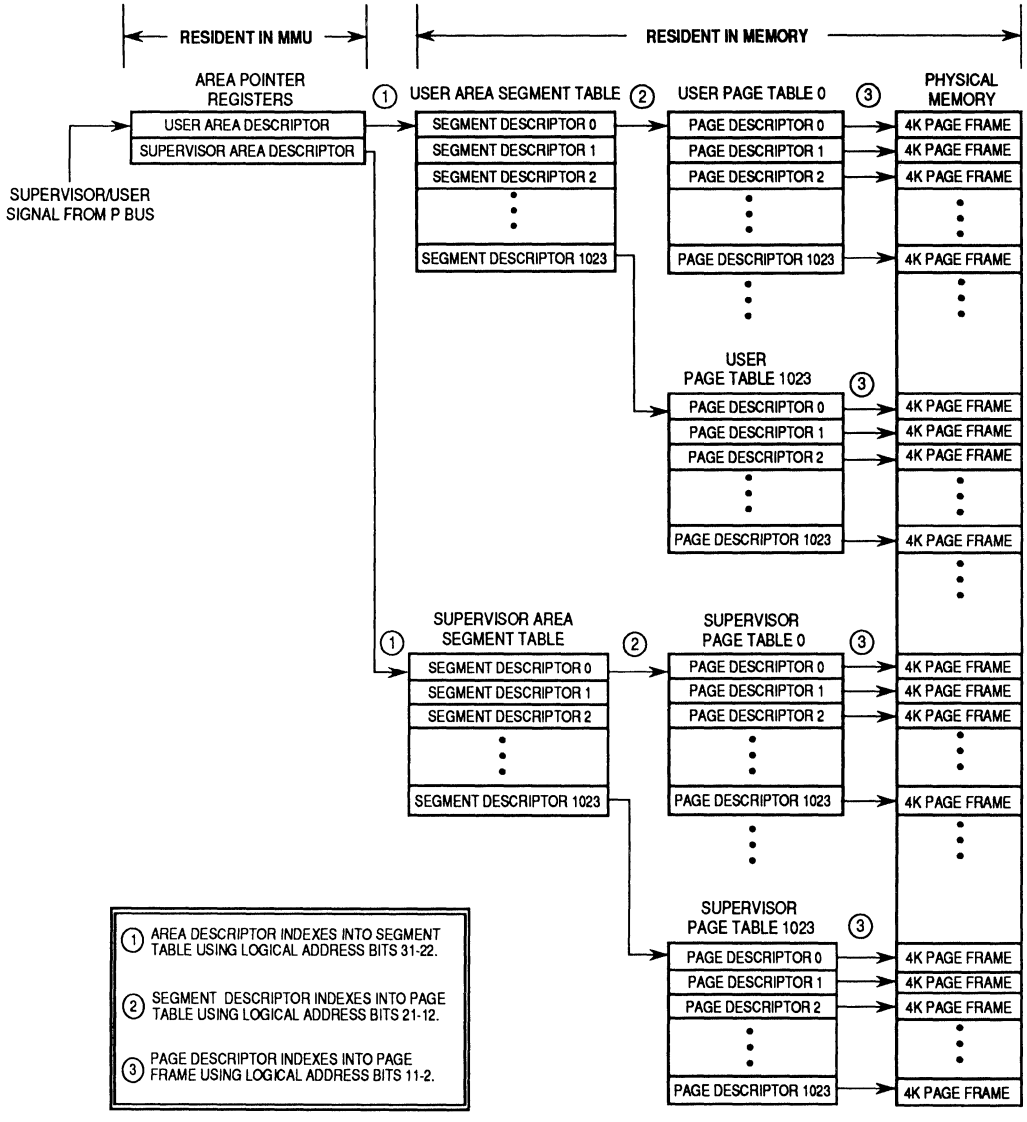


Figure 2-2. Translation Table Structure

2.3.1 Translation Descriptors

The translation descriptors contain a pointer to the next level in the translation hierarchy, either another translation table or a physical page. These descriptors also contain control and protection information for the representative portions of memory. Software initializes the descriptors as required (i.e., at system initialization or on demand) with appropriate addresses and information. Software can modify the descriptors as needed during operation.

Protection and control information for a particular memory location is accumulated from the various descriptors; each protection or control bit for a physical page is the logical OR of the corresponding bit in the area, segment, and page descriptors. The valid bit is the logical AND of segment and page descriptors. The resulting binary value is placed in the PATC entry. The MMU protection mechanism allows the operating system to provide sharing of information between multiple tasks while maintaining distinct access privileges for each task.

For example, two instances of an application program can be running on an MC88200-based memory system (see Figure 2-3). These task instances (A and B) share a memory page frame for the common portions of the application. Task A operates at a higher privilege level because it has read-write privileges; whereas, task B is restricted to read-only access. The operating system can assign the different privileges by giving each task a separate page descriptor that maps to the same physical page. Task A's descriptor has the WP bit clear (no write protect); task B's descriptor has the WP bit set (write protected).

Before table searches can occur, a minimum number of descriptors must be initialized. At least one area pointer must point to an initialized segment table. The descriptors in these

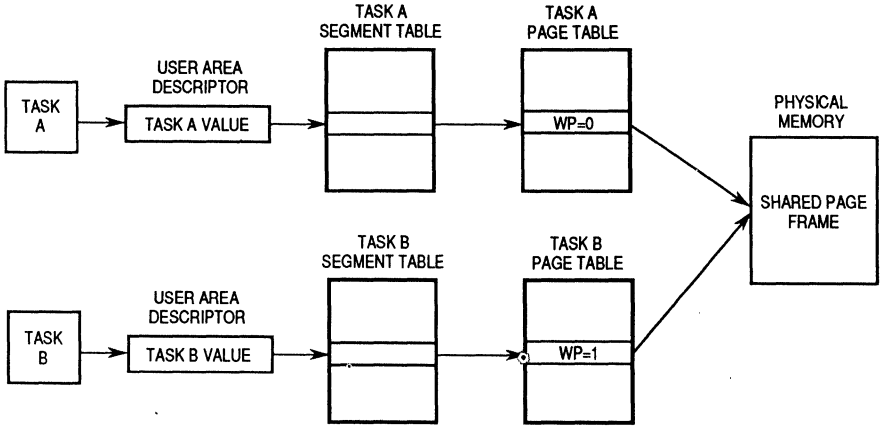


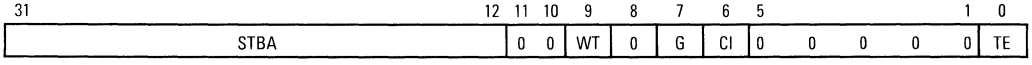
Figure 2-3. Memory Protection for Shared Memory Page

tables do not have to be valid, but they must be marked valid or invalid as appropriate. This notation guarantees that either correct PATC entries will be generated (through valid segment, then page descriptors) or that appropriate faults will occur during the table search.

2.3.1.1 AREA DESCRIPTORS. The area descriptors in the SAPR and UAPR contain the base address of the segment tables and the area control bits. The control bits include the translation enable bits, which are set by software to enable address translation, and the cache inhibit bits, which are set by software to enable the data cache.

For translation table searches to be performed, at least one area descriptor must be initialized. Initialization is performed when the local processor or an M bus master writes a valid descriptor to one or both area pointer registers. The segment table base address field contains the address where the software places the segment table (this location is not constrained by the MMU except that it is aligned on a 4K-byte page boundary). The WT, G, and CI bits are set as appropriate. The TE bit (bit 0) must be set to enable translation through the descriptors. Only one area descriptor needs to be initialized for address translation to occur.

The TE bit should not be set unless the descriptor contains a valid segment table base address. If an area descriptor is invalid (TE clear), addresses are presented to memory without translation. The WT, G, and CI bits in the area descriptor are applied toward the transaction regardless of the status of the TE bit.



STBA — Segment Table Base Address
These bits are the physical base address (upper 20 bits) of the segment.

Bits 11–10, 8, 5–1 — Reserved
Reserved for future extension. These bits should be cleared for future compatibility.

WT — Writethrough
Memory update policies are described in detail in **SECTION 3 DATA CACHE**. If the CI bit is set, this bit is ignored.
1 = Cache memory updates are performed using writethrough policy.
0 = Cache memory updates are performed using copyback policy.

G — Global
The value of the G bit is driven on M bus signal C5 during the address phase of the M bus transaction. Other M bus devices use this signal to perform bus snooping (see **SECTION 3 DATA CACHE**). When set, snoopers monitor the access to determine if their copies of the data are needed to be copied back to memory and/or invalidated.
1 = Memory mapped by this entry is global memory.
0 = Memory space is local to this MC88200.

CI — Cache Inhibit

The CI bit pertains only to the local MC88200, although the value of the CI bit is driven on M bus signal C4 during the address phase. Other M bus devices may or may not use this signal.

1 = Data and/or instructions mapped by this descriptor are not cached in the MC88200.

0 = Data and/or instructions mapped by this descriptor may be cached in the MC88200.

TE — Translation Enable

The TE bit must be set in one of the area descriptors for address translation to occur. If clear, addresses are presented one-for-one (logical-to-physical) to the memory system. The WT, G, and CI bits are still valid and are applied to the memory address. Software must set this bit when it loads the area descriptor.

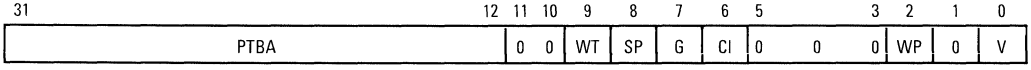
1 = Enables address translation through the ATCs or by a translation table search.

0 = Segment table base address is invalid, so a table search cannot be performed.

2.3.1.2 SEGMENT DESCRIPTOR. Each segment descriptor contains the base address of a page table. In addition, each segment descriptor contains segment-level protection and control information, which is logically ORed with the information in the other descriptors.

A segment table consists of 1024 32-bit translation descriptors. None of the translation descriptors need to be valid; when an invalid descriptor is encountered, the MC88200 signals a segment fault. The processor exception handler can load a valid segment descriptor into the appropriate entry of the table. Valid segment descriptors can also be loaded by software at system initialization, task initialization, or at other times as appropriate. The page table base address (PTBA) field contains the address where the software places the page table. This location is not constrained by the MMU except that it is aligned on a 4K-byte page boundary. The protection and control bits may be set as appropriate; the valid bit (bit 0) must be set.

To prevent a table search from accidentally using an invalid descriptor, all invalid descriptors should be explicitly marked as invalid by clearing bit 0 of each descriptor in the segment table



PTBA — Page Table Base Address

These bits are the physical base address (upper 20 bits) of the page.

Bits 11–10, 5–3, 1 — Reserved

Reserved for future extension. These bits should be cleared for future compatibility.

WT — Writethrough

Memory update policies are described in detail in **3.2 MEMORY UPDATE POLICIES**. If the CI bit is set, this bit is ignored.

1 = Cache memory updates are performed using the writethrough policy.

0 = Cache memory updates are performed using the copyback policy.

SP — Supervisor Protection

1 = Translations through the address page table can only be done in supervisor mode.

0 = Translations through the address page table can be done in supervisor or user mode.

G — Global

The accrued value of the G bit is driven on M bus signal C5 during the address phase of the M bus transaction. Other M bus devices use this signal to perform bus snooping (see **SECTION 3 DATA CACHE**).

1 = Memory mapped using this descriptor is global memory.

0 = Memory mapped by this descriptor is local to this MC88200.

CI — Cache Inhibit

The CI bit accrued after a table search affects caching only in this MC88200. The CI bit of the ATC entry used during address translation is driven on M bus signal C4. Other M bus devices may or may not use this signal.

1 = Data and/or instructions mapped by this entry are not cached in the MC88200.

0 = Data and/or instructions mapped by this entry can be cached in the MC88200.

WP — Write Protect

A write to the page write-protected by this descriptor fails, causing a write-protection violation. The MC88200 replies to the processor with a fault after the entire table search operation is complete.

1 = Memory mapped by this entry is write protected by the MC88200.

0 = Memory mapped by this entry can be written by the MC88200.

V — Valid

This bit must be set when software initializes the descriptor so that table searches use this descriptor.

1 = Descriptor is valid.

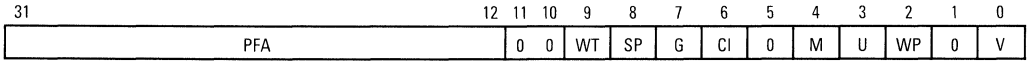
0 = Table search ends with a segment fault.

2.3.1.3 PAGE DESCRIPTOR. Each page descriptor contains the address of a physical page frame into which the 4K bytes of logical page address is mapped. In addition, each descriptor contains page-level protection and control information, which is logically ORed with the area and segment information to form the protection/control for a particular page.

The page descriptors contain two reference bits that indicate the status of the memory page. These bits are set by the MC88200 and cleared by software. The used bit (bit 3) is

set whenever the page descriptor is used in a memory access, including probe transactions. The modified bit (bit 4) is set whenever a memory write is mapped through the page descriptor. In addition, if a PATC hit satisfies the translation of a write transaction, a table search is performed to set the modified bit in the page descriptor and to update the PATC entry with the modified bit set.

The page table initialization is similar to the segment table initialization: descriptors can be initialized when a page fault occurs or can be initialized by software at appropriate times, such as at system or task initialization. The page frame address (PFA) field contains the physical address of the page (4K boundary), and the protection and control bits may be set as appropriate. The modified and used bits (bits 4 and 3, respectively) should be cleared to prevent a table search from accidentally using an invalid descriptor, and all invalid descriptors should be explicitly marked as invalid by clearing bit 0 of each descriptor in the page table.



PFA — Page Frame Address

Bits 11, 10, 5, 1 — Reserved
Reserved for future extension. These bits should be cleared for future compatibility.

WT — Writethrough
Memory update policies are described in detail in **3.2 MEMORY UPDATE POLICIES**. If the CI bit is set, this bit is ignored.
1 = Cache memory updates are performed using the writethrough policy.
0 = Cache memory updates are performed using the copyback policy.

SP — Supervisor Protection
1 = Addressed page frame can only be accessed in supervisor mode.
0 = Page frame can be accessed in supervisor or user mode.

G — Global
The accrued value of the G bit is driven on M bus signal C5 during the address phase of the M bus transaction. Other M bus devices use this signal to perform bus snooping (see **SECTION 3 DATA CACHE**).
1 = Memory mapped using this descriptor is global memory.
0 = Memory mapped by this descriptor space is local to this MC88200.

CI — Cache Inhibit
The CI bit accrued after a table search affects caching only in this MC88200. The CI bit of the ATC entry used during address translation is driven on M bus signal C4. Other M bus devices may or may not use this signal.
1 = Data and/or instructions mapped by this entry are not cached in the MC88200.
0 = Data and/or instructions mapped by this entry can be cached in the MC88200.

M — Modified

1 = The page frame mapped by this descriptor has been modified by a memory write.

0 = The page frame has not been modified.

U — Used

1 = Data and/or instructions residing in the page frame mapped by this descriptor have been accessed.

0 = Page frame has not been accessed.

WP — Write Protect

A write to the page frame write protected by this descriptor fails, causing a write-protection violation. The MC88200 replies to the processor with a fault after the entire table search operation is complete.

1 = Memory mapped using this descriptor is write protected by the MC88200.

0 = Memory mapped using this descriptor can be written by the MC88200.

V — Valid

This bit must be set when software initializes the descriptor so that table searches use this descriptor.

1 = Descriptor is valid.

0 = Table search ends with a page fault.

2.3.2 Protection

The MC88200 provides two types of memory access protection: write protection and supervisor protection. These protections are invoked by bits in the various descriptors. If the write protect bit is set and a write or locked read access is performed, a write-protection violation occurs. The processor is notified with a fault reply at the end of the entire table search sequence. If the supervisor bit is set and the current access is for user memory, a privilege violation occurs, the table search sequence aborts, and the processor is notified immediately with a fault reply.

The protection bits in the table descriptors are set and cleared by software. They can be different for a physical address at the area, segment, and page descriptor levels. The privilege level of a physical memory location is the logical OR of the area, segment, and page privilege levels.

2.3.3 Table Search Algorithm

The table search algorithm sequentially searches each level of the memory hierarchy, following the appropriate descriptors until the PFA is reached. Figures 2-4 through 2-10 provide flowcharts for the table search algorithm. Figure 2-11 summarizes how the physical

address is assembled from a segment descriptor, page descriptor, and low-order bits of the logical address.

1. The supervisor/user bit of the logical address specifies whether the supervisor area pointer ($S/\overline{U}=1$) or user area pointer ($S/\overline{U}=0$) will be used.
   a. If the selected area pointer is invalid, then the address is presented to memory without mapping. However, the WT, G, and CI bits are applied to the address.
   b. If the selected area pointer is valid, then the WT, G, and CI bits from the area pointer are accumulated for the final physical address. The MC88200 performs the next step in the table search.
2. The segment table base address from the area pointer points to the segment table; bits 31–22 of the logical address specify the individual segment descriptor within the table.
   a. If the segment descriptor is valid, the protection and control bits of the segment descriptor are logically ORed to their accrued values. The MC88200 performs the next step in the table search. The WP bit is not tested at this point.
   b. If the selected descriptor is invalid, a segment fault occurs. The table search stops, the P bus fault status register (PFSR) is updated, and the processor is informed of the fault. The exception handler executed by the processor must load a valid segment descriptor into the segment table and must retry the memory transaction.
   c. If the supervisor privilege (SP) bit is set and the access is a user access, a privilege violation occurs. The table search stops, the PFSR is updated, and the processor is informed of the fault. The exception handler executed by the processor must perform an appropriate action, such as retrying or aborting the transaction.
   d. If a bus error occurs while accessing the descriptor, a bus error fault occurs. The table search stops, the PFSR is updated, and the processor is informed of the fault. The exception handler executed by the processor must perform an appropriate action, such as retrying or aborting the transaction.
3. The page table base address from the selected segment descriptor specifies the page table; bits 21–12 of the logical address specify the individual page descriptor within the table:
   a. If the selected descriptor is invalid, a page fault occurs. The table search stops, the PFSR is updated, and the processor is informed of the fault. The exception handler executed by the processor must load a valid page descriptor into the page table and must retry the memory transaction.
   b. If a bus error occurs while accessing the descriptor, a bus error fault occurs. The table search stops, the PFSR is updated, and the processor is informed of the fault. The exception handler executed by the processor must perform an appropriate action, such as retrying or aborting the transaction.
   c. If the SP bit is set and the access is a user access, a privilege violation occurs. The table search stops, the PFSR is updated, and the processor is informed of the fault. The exception handler executed by the processor must perform an appropriate action, such as retrying or aborting the transaction.

d. If no fault condition is encountered, the page descriptor is valid. The protection and control bits from the page descriptor are logically ORed with accumulated values in the area and segment descriptors. The MC88200 performs the next step in the table search. The WP bit is not tested at this point.

4. The MC88200 creates a PATC entry:
   a. The S/$\overline{U}$ bit and bits 31–12 of the logical address form the LPA field of the PATC entry.
   b. The PFA from the selected page descriptor is used as the physical address of the page (PFA field in the PATC entry).
   c. The protection and control bits are loaded with the values accumulated during the table search.
5. The MC88200 translates the logical address through the PATC. At this point, the WP bit (bit 0) is tested. If the memory page is write protected and the memory transaction is a write transaction, then a write-protection violation occurs. Otherwise, the memory access is performed.

2.3.3.1 TABLE SHARING BETWEEN TASKS. A page frame, page table, or segment table can be shared between different tasks by placing a pointer to the shared table in the address translation tables of more than one task. The nonshared tables can contain a different setting of protection bits, allowing different tasks to use the area with different permissions. In Figure 2-12, two tasks share the memory translated by the page table. Task "A" cannot write to the shaded area; however, since task B has the WP bit clear in its segment descriptor, it can read and write to the shaded area. The shared area could appear at different logical addresses for each task since each task uses a different segment descriptor.

2.3.3.2 PAGING OF TABLES. The entire address translation tree for an active task need not be resident in main memory at the same time. In the same way that only the working set of pages must reside in main memory, only the tables that describe the resident set of pages need be available in main memory. This paging of tables is implemented by clearing the valid bit of the table descriptor that points to the absent table(s). When a task attempts to use an address that would be translated by an absent table, the MC88200 is unable to locate a translation and takes a segment or page fault. It is the responsibility of system software to determine that the 'invalid' descriptor corresponds to nonresident tables.

2.3.3.3 DYNAMIC ALLOCATION OF TABLES. Similar to the case of paged tables, a complete translation tree need not exist for an active task. The translation tree can be dynamically allocated by the operating system based on request for access to particular areas.

As in the case of demand paging, it is difficult, if not impossible, to predict the areas of memory that are used by a task over any extended period of time. Instead of attempting to predict the requirements of the task, the operating system performs no action for a task

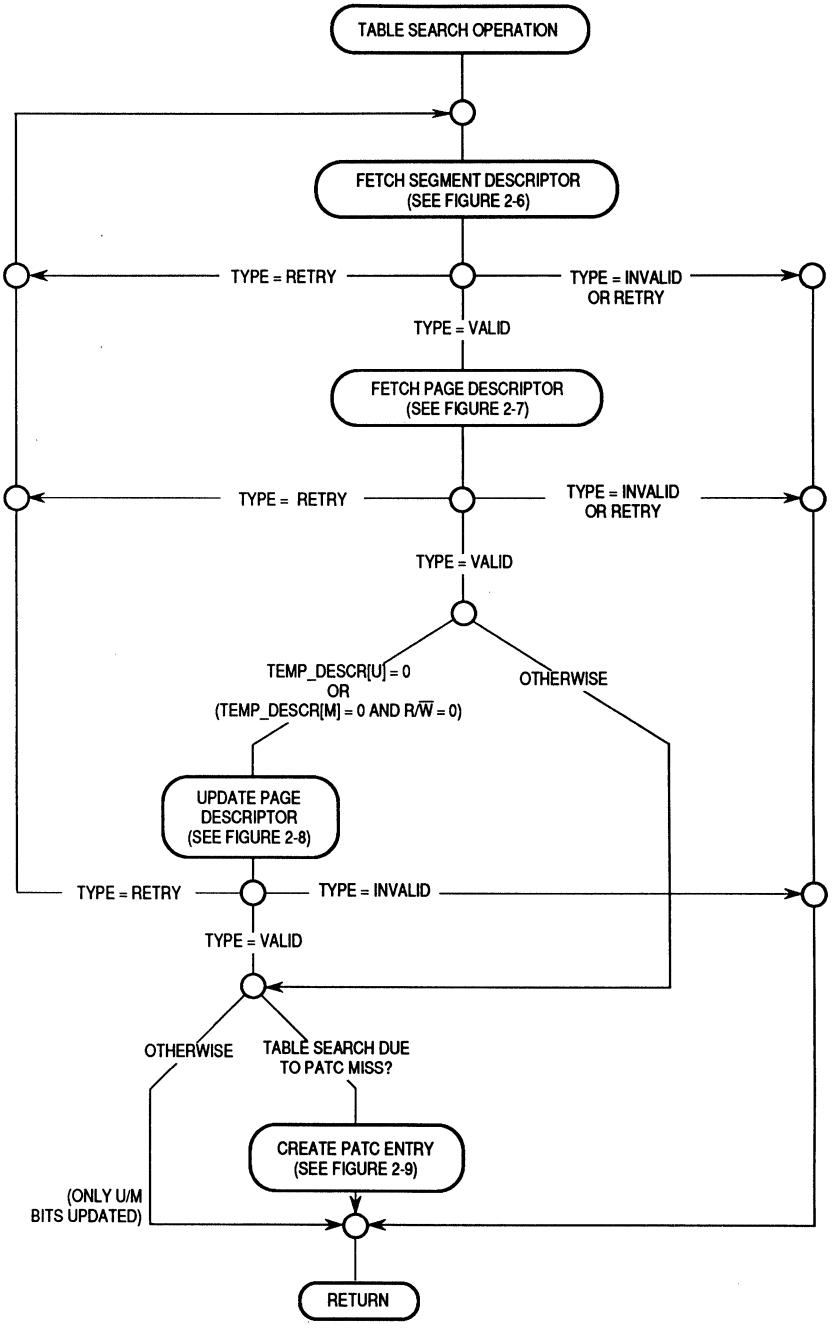


Figure 2-4. Table Search Operation Flowchart

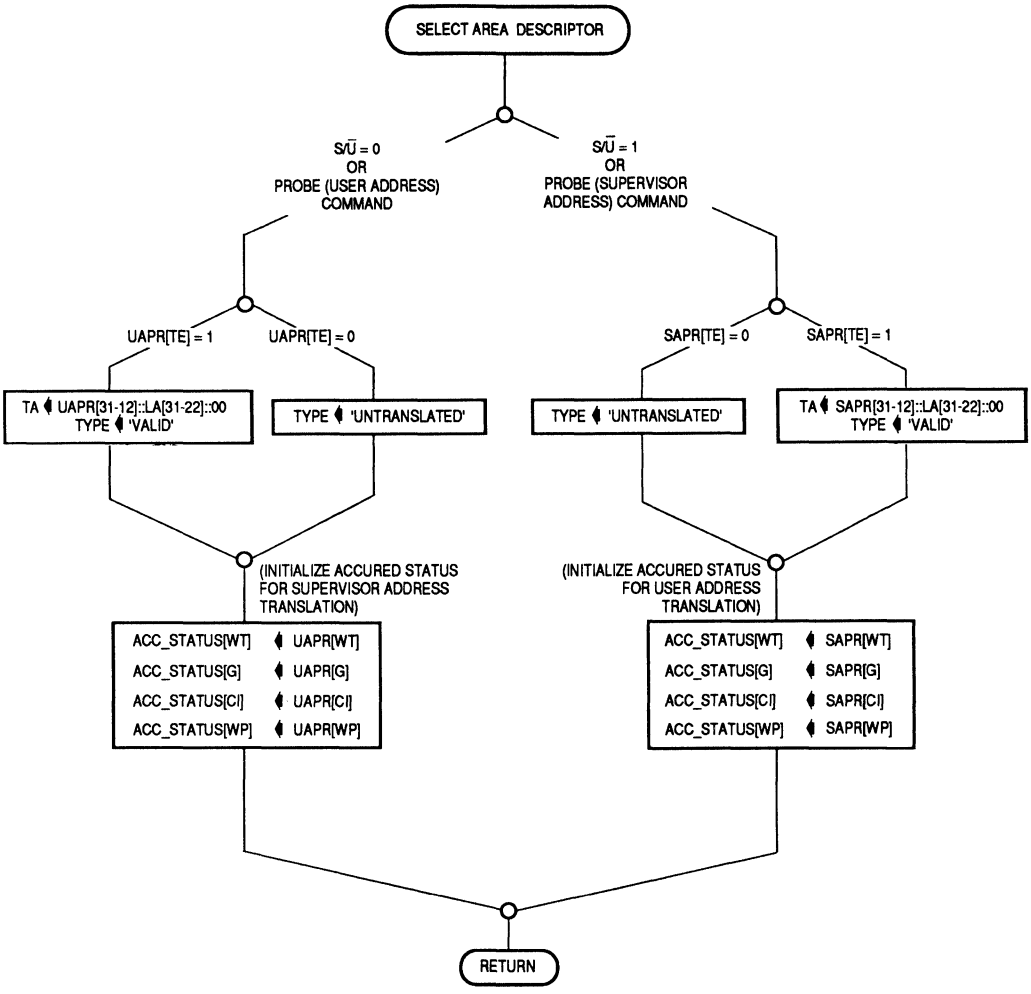


Figure 2-5. Area Descriptor Selection Flowchart

until a 'demand' is made requesting access to a previously unused area or an area that is no longer resident in memory. This same technique can be used to efficiently create a translation tree for a task.

For example, consider an operating system that is preparing the system to execute a previously unexecuted task that has no translation tree. Rather than guessing what the memory-usage requirements of the task are, the operating system creates a translation tree for the task that maps one page corresponding to the initial value of the program counter for that task and, possibly, one page corresponding to the initial stack pointer of

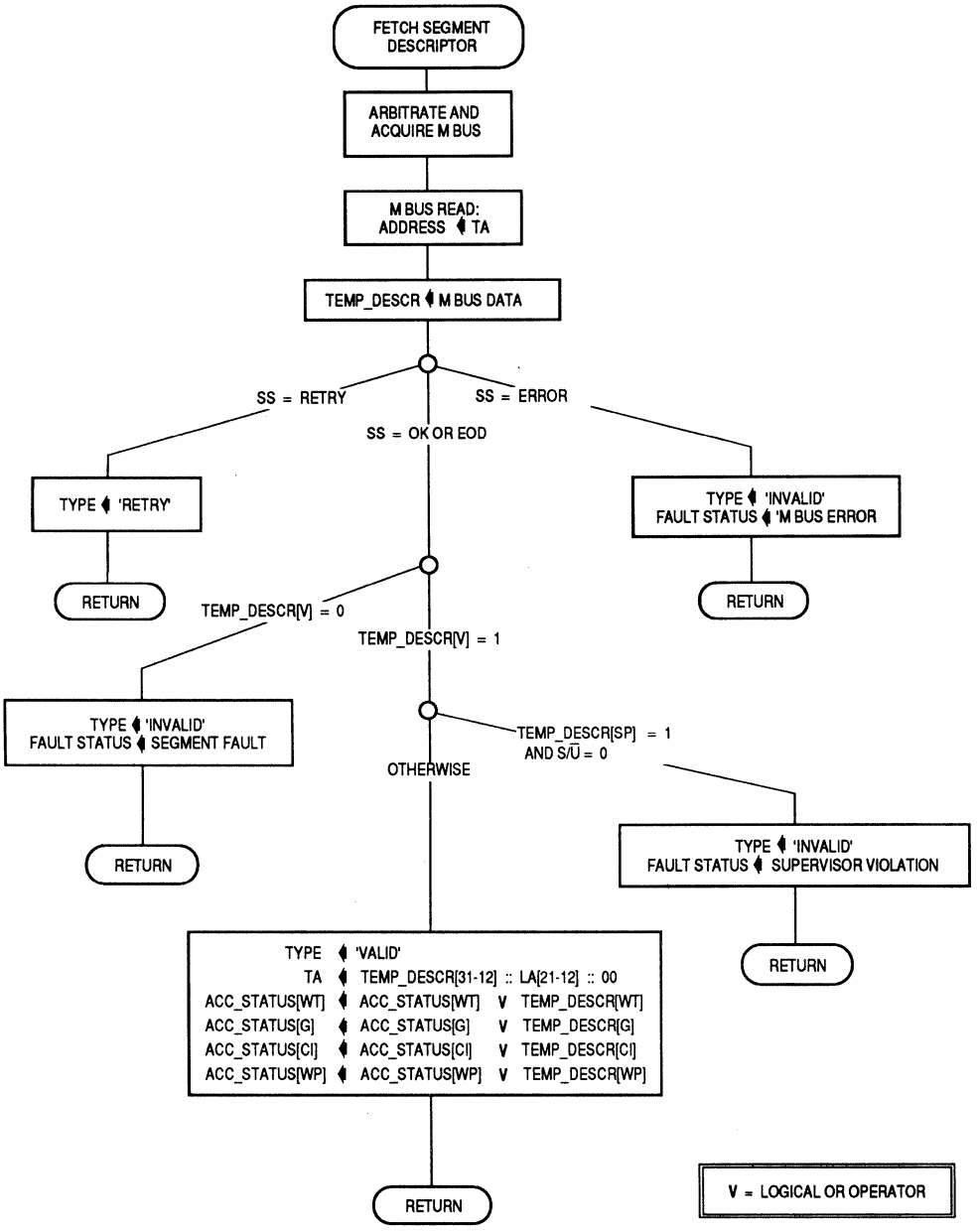


Figure 2-6. Segment Descriptor Fetch Flowchart

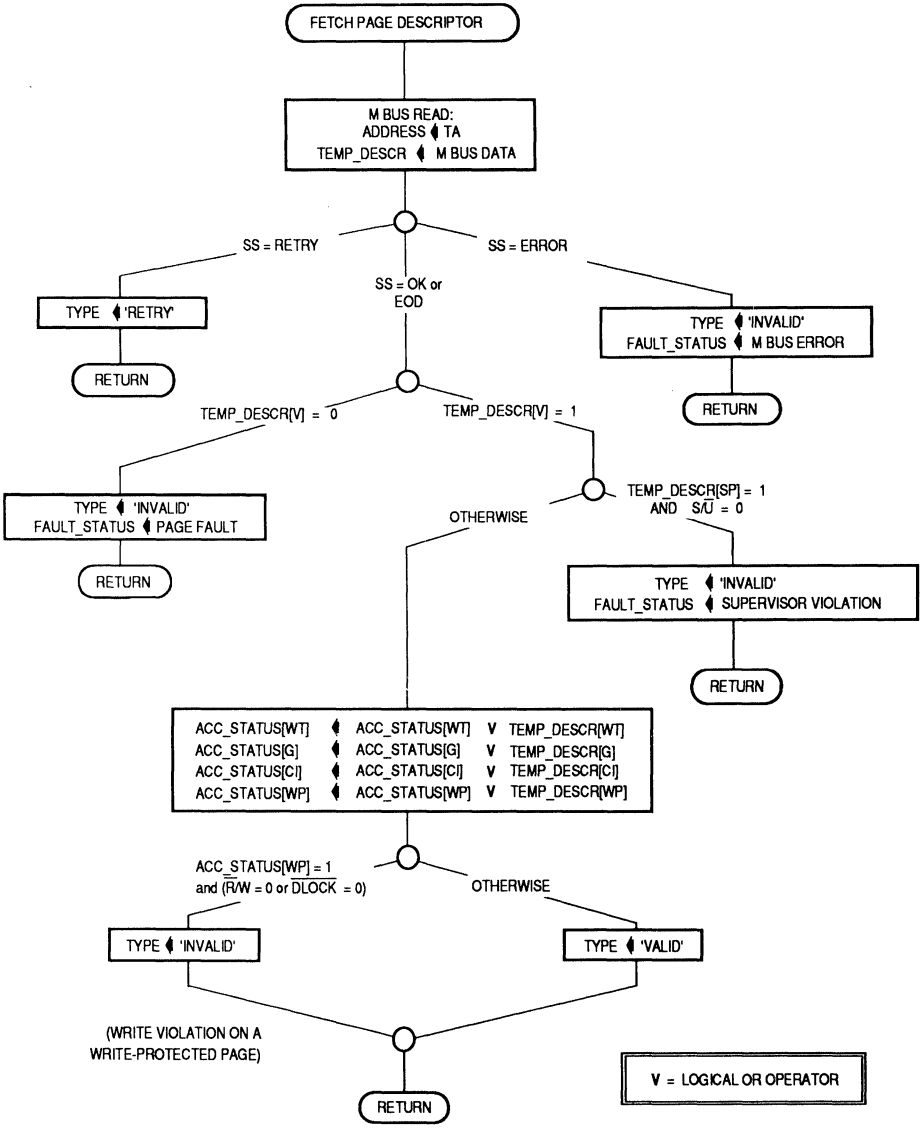


Figure 2-7. Page Descriptor Fetch Flowchart

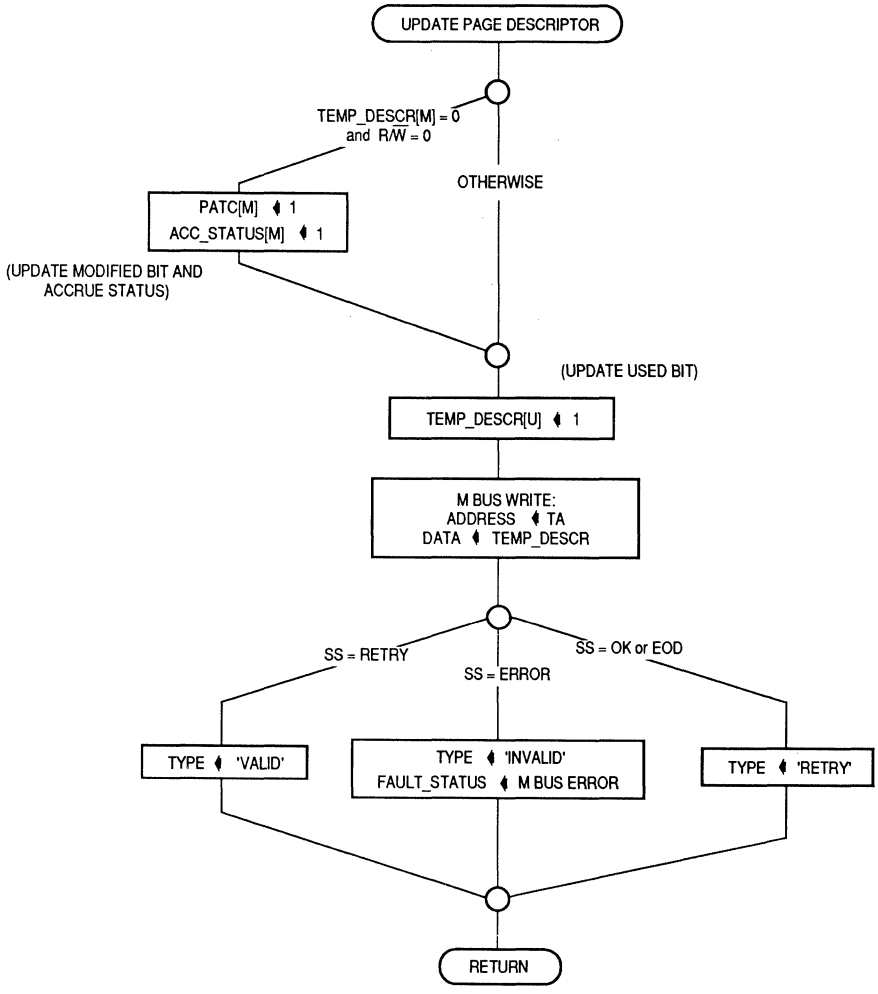


Figure 2-8. Page Descriptor Update Flowchart

the task. All other branches of the translation tree for this task remain unallocated until the task requests access to the areas mapped by these branches. This technique allows the operating system to construct a minimal translation tree for each task conserving physical memory utilization and minimizing operating system overhead.

2.4 FAULTS

Faults occur in response to a P bus transaction (initiated by the P bus processor). A fault indicates that the MC88200 could not complete the current transaction due to an unusual or error condition. Faults can occur because of the following conditions.

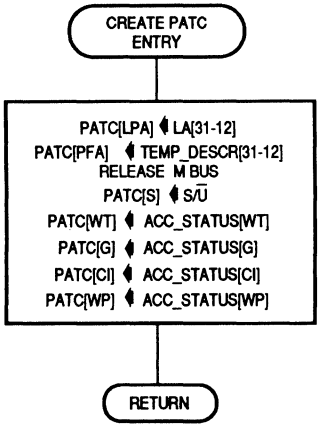


Figure 2-9. PATC Entry Creation Flowchart

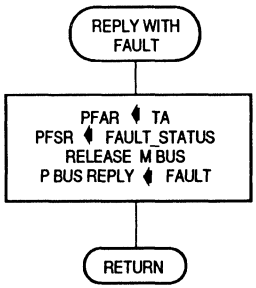


Figure 2-10. Fault Register Update Flowchart

1. An M bus error or parity error was encountered during the memory access. Refer to **SECTION 5 BUS OPERATIONS** for more information on M bus error.
2. A translation table search encountered an invalid descriptor.
3. Memory protection was violated.

In addition to faults, the MC88200 may encounter M bus errors during cache flushes, data copyback errors during bus snopping, and errors during control register accesses. These errors are discussed in **SECTION 3 DATA CACHE** and **SECTION 5 BUS OPERATIONS**.

When a fault is encountered, the MC88200 aborts the current transaction. The P bus fault status register (PFSR) and P bus fault address registers (PFAR) are updated, and a 'fault' reply is driven on the P bus reply lines. The processor monitors the reply lines (done automatically by the MC88100 processor), and when the fault reply is received, the processor takes some action. Normally, a P bus fault causes an exception in the processor,

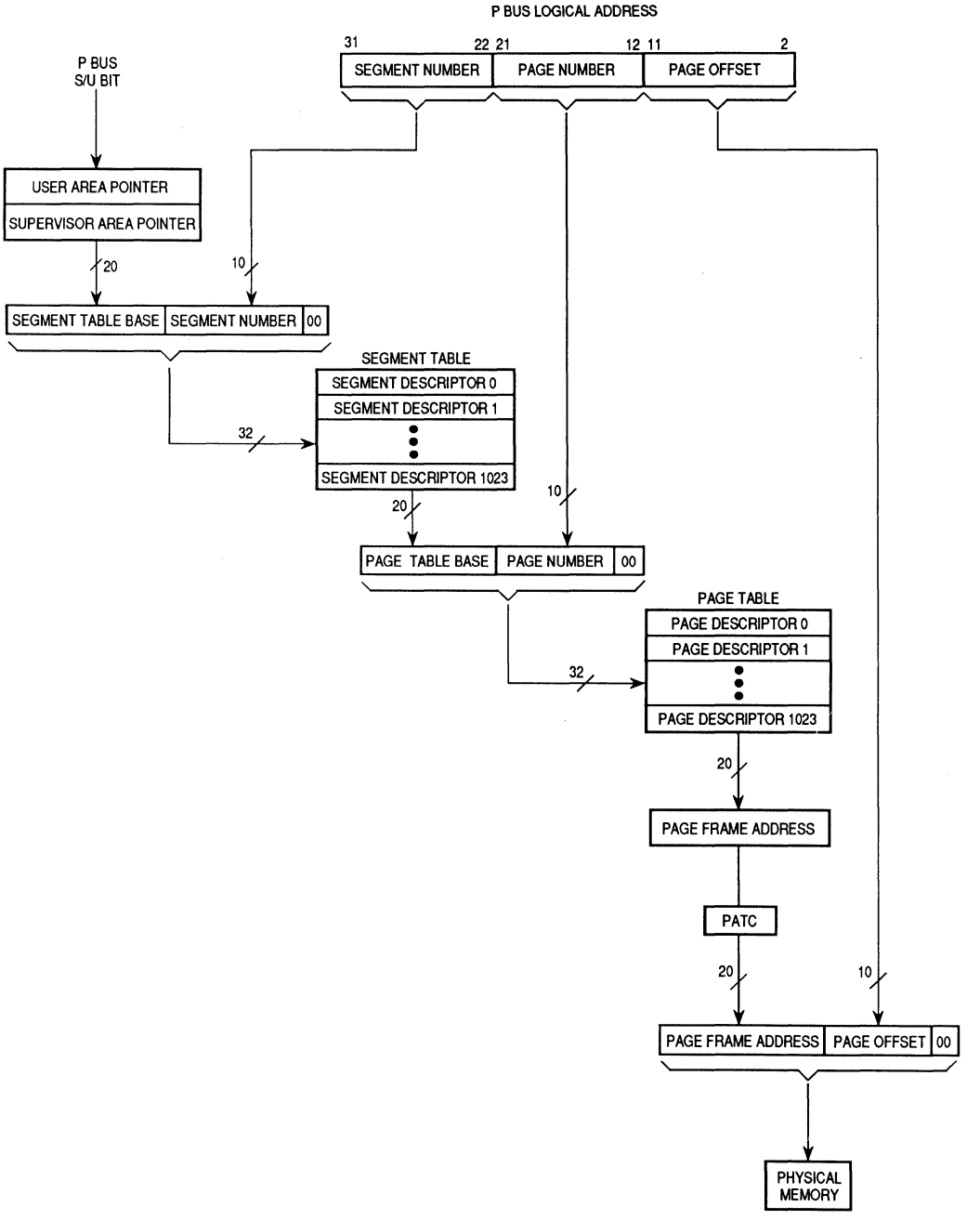


Figure 2-11. Physical Address Generation

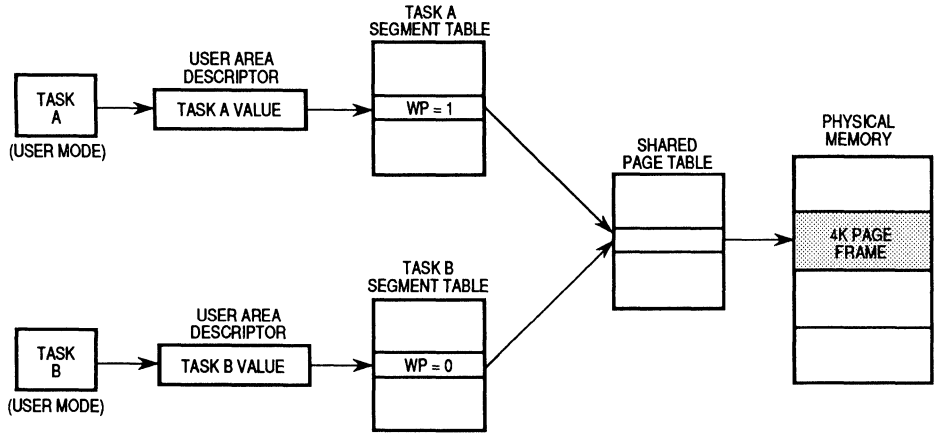


Figure 2-12. Shared Page Tables

and the processor executes an exception handler routine. The exception handler corrects or processes the condition that caused the fault, and may retry the P bus transaction.

A fault does not necessarily indicate an error. For example, the MC88200 supports a demand-paged virtual memory system. The MC88200 indicates that a segment or page is required by signaling a segment or page fault, respectively.

2.4.1 Fault Processing

Whenever the processor initiates a P bus transaction, it must monitor the reply signals (R1–R0) returned from the MC88200. These signals indicate the status of the transaction. When the MC88200 encounters a fault condition, it performs the following actions to signal the fault to the processor.

1. The MC88200 aborts the transaction that caused the exception.
2. The fault code field (bits 18–16) of the PFSR is updated to indicate the cause of the fault. The fault code field is coded as follows:
   - 000 — No Fault
   - 011 — Bus Error
   - 100 — Segment Fault
   - 101 — Page Fault
   - 110 — Supervisor Violation
   - 111 — Write-Protection Violation
3. The PFAR is updated with the physical (M bus) address of the memory transaction. For write-protection violtation, the PFAR is updated with invalid data.

4. The P bus reply lines are driven with a 'fault' reply to inform the processor that the fault occurred.

5. The next P bus transaction is ignored. This prevents a second fault from occurring because of pipelined bus transactions. For example, the MC88100 processor places the transaction address on the P bus before it receives the reply from the previous transaction. If the previous transaction ends in a page fault and the current transaction accesses the same memory page, then that transaction would also cause a fault. Ignoring the transaction allows the processor time to stop issuing pipelined transactions.

6. After ignoring the transaction following the fault, the MC88200 returns to the normal state of accepting the P bus transaction.

When a fault is signaled, the processor normally takes an exception and begins execution of an exception handler routine. This routine determines the fault type from the PFSR and may read the PFAR to determine the faulting address. If only the logical address is needed and the processor is an MC88100, then the logical address can be read from the MC88100. Once the cause of the fault is determined, the exception handler may correct the fault condition or may ignore the memory transaction. If the fault condition is corrected, the transaction can be retried.

A fault may cause an inconsistency in the modified bit of a segment or page descriptor. When a write transaction is performed that results in a PATC hit, the modified bit is set in the PATC entry. The MC88200 then performs a table search to set the modified bits in the corresponding page descriptors. If that table search results in a bus error, then the modified bits in the descriptors may not be set. As a rule, bus error exception processing should invalidate the PATC entry that caused the error to avoid M bit inconsistency with memory. The logical address that faulted is recorded by the MC88100 processor.

2.4.2 Fault Conditions

The following paragraphs describe the different faults that may be encountered.

2.4.2.1 SEGMENT FAULT. This fault occurs when the MC88200 encounters an invalid segment descriptor during a translation table search. When this fault occurs, the MC88200 loads the physical address of the faulting segment descriptor into the PFAR and updates the PFSR.

2.4.2.2 PAGE FAULT. A page fault occurs when the MC88200 encounters an invalid page descriptor during a translation table search. When this fault occurs, the MC88200 loads the physical address of the faulting page descriptor into the PFAR and updates the PFSR.

2.4.2.3 SUPERVISOR VIOLATION. This fault occurs when a translation table search for a user logical address encounters a segment or page descriptor with supervisor protection. The MC88200 loads the physical address of the faulting descriptor into the PFAR and updates the PFSR.

2.4.2.4 WRITE-PROTECTION VIOLATION. This fault occurs when a memory transaction encounters a write-protected BATC or PATC entry and the memory access is a write access. Only the PFSR is updated; the PFAR contains invalid data. If the transaction requires a translation table search, the write violation is not detected until after the table search.

2.4.2.5 M BUS ERROR. The M bus error can occur during any M bus transaction resulting from a P bus transaction. The M bus error is caused by the MC88200 detecting a parity error or occurs when the $\overline{SS3}$ system status signal is asserted by another M bus device. Modified (M) bit updates occur in the PATC before the M bit is written to the page descriptor in memory (Figure 2-8). Because a page descriptor write may be aborted with an M bus error, the PATC should be flushed during M bus error exception processing to avoid M bit incoherency between a PATC entry and it's corresponding page descriptor in memory. Refer to **SECTION 5 BUS OPERATIONS** for more information on M bus errors.

2.5 PROBE TRANSACTIONS

Probe transactions are performed to load page descriptors into the PATC and to accumulate status information for a logical-to-physical address translation. Probes may be implemented by the operating system to test the protection and control status of a logical-to-physical address translation. Probes are useful to see if a local (exclusive) page has been modified so that any cached data may be written back to memory when a task terminates. Probes are also useful to determine the physical address that is mapped from a certain logical address.

2.5.1 Use of the System Address, Command, and Status Registers

Before performing a probe transaction, the operating system initializes the system address register (SAR) with the logical address it wishes to probe.

It then writes a probe command to the SCR. There are two types of probe commands: probe user address and probe supervisor address. Both commands search the BATC and PATC for the logical address in the SAR. If none is present, a table search is performed through the SAPR or UAPR, depending on which probe command is used. During the table search, access and protection information is accumulated from the area, segment, and page tables. Table 2-1 lists the command code for the probe types.

Table 2-1. Probe Commands

Command Code SCR (Bits 5–0)	Probe Type
10 × 0xx	Probe User Address
10 × 1xx	Probe Supervisor Address

If an invalid descriptor is not encountered during the table search, a new PATC entry is created, and status of the translation is written to the system status register (SSR). The physical address of the translation is written to the SAR, overwriting the logical address from which it was mapped. If an invalid descriptor is read during the table search, the search is aborted, and the PFAR is updated with the physical address of the fault. The PFSR is updated with a fault code corresponding to the descriptor that was invalid. (see **2.3.3 Table Search Algorithm**).

If an M bus error occurs during the table search, the physical address at which the error occurred is written to the SAR, and the bus error (BE) bit of the SSR is set. The complete probe transaction operation is shown in Figure 2-13.

System Status Register

31 … 16	15	14	13 … 10	9	8	7	6	5	4	3	2	1	0
RESERVED	CE	BE	RES	WT	SP	G	CI	R	M	U	WP	BH	V

Address: Base + $008

Bits 31–16, 13–10, 5 — Reserved
These bits are returned as zeros on a register read.

CE — Copyback Error
This bit indicates that an error occurred during a data copyback initiated by a snoop transaction; not involved in memory management.

BE — Bus Error
This bit indicates that an M bus error occurred during a probe transaction or a cache flush operation. The SAR is updated with the physical address of the error.

WT — Writethrough
1 = Data at the probed address is cached with writethrough memory.
0 = Data at the probed address is cached with the copyback memory update policy.

SP — Supervisor Privilege
1 = Probed address can only be accessed in the supervisor mode.
0 = Probed address can be accessed in the user or supervisor mode.

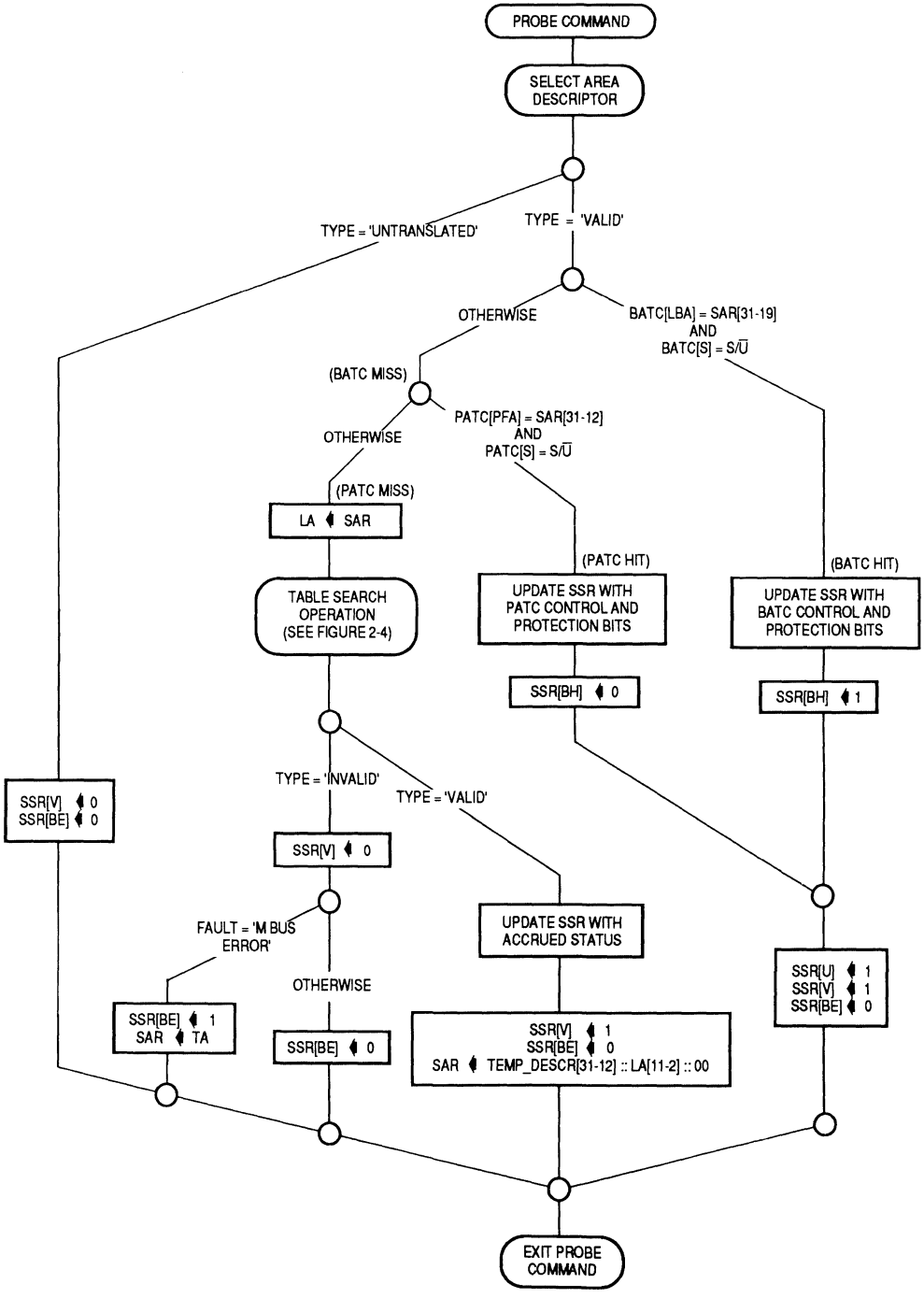


Figure 2-13. Probe Operation

G — Global

1 = One or more of the descriptors for the probed address are marked global.

0 = All of the descriptors for the probed address have clear global bits.

CI — Cache Inhibit

1 = Data at the probed address cannot be cached.

0 = Data can be cached.

M — Modified

1 = Data at the probed address has been modified in memory or with respect to memory (cached data).

0 = Data has not been modified.

U — Used

This bit is always set since the probe transaction is interpreted as a use of the probed memory area.

WP — Write Protection

1 = Probed address is write protected.

0 = Probed address can be read or written.

BH — BATC Hit

1 = Probed address resulted in a BATC hit instead of a PATC hit or table search.

0 = Probed address was found in the PATC or was generated by a table search.

V — Valid

1 = All descriptors or ATC entries encountered for the probed address were marked valid. If this bit is set and no fault occurred, the probe results are valid

0 = One or more descriptors encountered during the table search were marked invalid, or transaction was not enabled in the area pointer register used for the probe.

2.5.2 Probe Error

A probe transaction may result in a translation table search as the MC88200 determines the translation information for a particular P bus address. If the MC88200 encounters an M bus error ($\overline{SS3}$ signal asserted by memory or another M bus device), the BE bit (bit 14) of the SSR will be set and the V bit cleared. The device that initiated the probe must sample this bit periodically to determine if the probe is successful. If the BE bit is set, the probe is unsuccessful, the probe result information is invalid, and the SAR contains the physical address where the fault occurred. A probe error is reported through the system registers and is not signaled as a fault. A bus error (which is treated the same as a parity error) or an invalid descriptor are the only types of errors that will generate a probe error condition and halt the associated table search operation.

2.6 CMMU REGISTER UPDATE SUMMARY (MEMORY MANAGEMENT)

Table 2-2 summarizes all of the MC88200 control register update operations as a result of normal, fault, and M bus error transactions. The table indicates how the CMMU registers are updated and illustrates the states of the signals that indicate exceptions.

Table 2-2. CMMU Register Update Summary (Memory Management)

Exception	Transaction	P Bus State	Update PFSR	Update PFAR	Update SSR	Update SAR	Reply P Bus Fault
Page Fault	**P Bus R/$\overline{W}$**	Slave	Bits 18–16 = 101	Page Descriptor Address	No	No	Yes
Segment Fault		Slave	Bits 18–16 = 100	Segment Descriptor Address	No	No	Yes
Privilege Violation		Slave	Bits 18–16 = 110	Physical Address of Error	No	No	Yes
M Bus Error		Slave	Bits 18–16 = 011	Physical Address of Error	No	No	Yes
Parity Error		Slave	Bits 18–16 = 011	Physical Address of Error	No	No	Yes
Write Violation		Slave	Bits 18–16 = 111	Contents Destroyed	No	No	Yes
Success		Slave	No	No	No	No	No
M Bus Error	**Probe**	N/A	No	No	Bit 14 Set Bit 9–0 Destroyed	Physical Address of Error Initialization Is Required for Meaningful Results	No
Success		N/A	No	No	Bit 14 Cleared Bits 9–0 = Results	Physical Address of Probe	No
Success	**PATC Invalidate**	N/A	No	No	Bit 14 Cleared Bits 9–0 Destroyed	Contents Destroyed Initialization Is Required for Meaningful Results	No

2.7 EFFECTS OF RESET ON MEMORY MANAGEMENT

The MC88200 MMU is reset when the $\overline{RST}$ signal is asserted. No pending or partially completed transactions are retained after $\overline{RST}$ is asserted. Control registers are modified,

external signals enter specially defined states, and the MC88200 becomes receptive to new P bus accesses. Memory management resources are initialized as shown in Table 2-3. It is the responsibility of the operating system to initialize the supervisor and user area pointers, segment table, page table, and the related access and protection information.

Table 2-3. Effects of RESET on Registers

Register	Memory Management Function	State During Reset
System Command Register	Invalidation of PATC descriptors initiate probes.	Cleared
System Status Register	Return status of probe.	Cleared
System Address Register	Contains logical address of memory location to be probed. Written with physical address of faulted transaction during M bus error of a probe.	Undefined
P Bus Fault Status Register	Indicates P bus fault code (bus error segment, page fault supervisor, and write violations).	Cleared
P Bus Fault Address Register	Written with physical address that caused P bus fault.	Undefined
Supervisor Area Pointer Register	Contains pointer to supervisor segment table and protection and control bits.	CI bit set, all other bits cleared
User Area Pointer Register	Contains pointer to user segment table and protection and control bits.	CI bit set, all other bits cleared
BATC Write Ports	Used to initialize BATC descriptors.	Undefined

SECTION 3
DATA CACHE

This section describes the architecture and operation of the MC88200 data cache. The data cache provides 16K bytes of high-speed instruction or data storage. This four-way set-associative cache has entries organized by physical address. Cache management facilities provide least recently used (LRU) line replacement, protection and diagnostic facilities, and two software-selectable memory update policies.

In this manual, "data cache" is used to specify the 16K-byte cache used for storage, as opposed to the page address translation cache (PATC) or the block address translation cache (BATC). The data cache may contain either instruction or data operands as defined by the application.

3.1 CACHE ORGANIZATION

The data cache portion of the MC88200 is organized into three functional components:

- 16K bytes of static random-access memory (SRAM) for data storage
- Cache address tags, used to determine the physical memory address corresponding to the cached data
- Control logic

The cache is configured as 256 sets of four lines each. Each line contains four 32-bit words, has an associated address tag, and associated disable and status bits. Each set has associated LRU information for line replacement. Figure 3-1 shows the cache memory organization.

A cache line contains four contiguous words from memory, loaded from a quad-word aligned address (A3–A0 = 0). All bus operations that load (fill) a cache line from memory are performed on a line basis. Operations that store (copyback) a cache line to memory are also done on a line basis. This load/store scheme simplifies the cache control circuitry. (When a write transaction misses the cache, only the affected word is written to memory. See **3.4.2.1 CACHE WRITE: CACHE MISS**.) Processor accesses to the cached data are designed to match the data types supported by the MC88100 processor: byte (8 bits), half-word (16 bits), and word (32 bits).

Each line in the cache has an associated address tag. The tag contains a 20-bit physical address that corresponds to the base address of a 4K-byte page in physical memory. When a memory location is accessed, the tag is used to determine if there is cached data corresponding to the memory location.

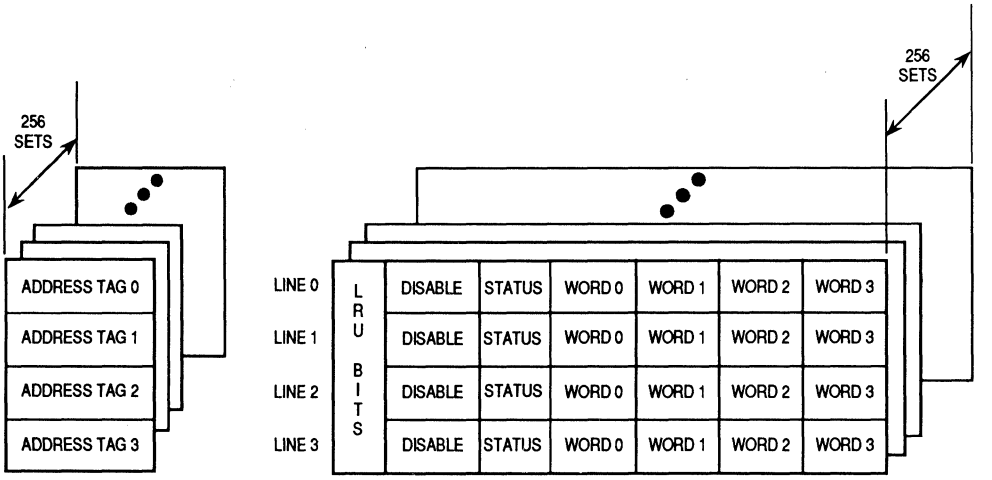


Figure 3-1. MC88200 Cache Memory Organization

The status (VV) bits indicate the state of the cached data with respect to other caches and memory: invalid, shared unmodified, exclusive modified, and exclusive unmodified.

The disable bit (D) is used for cache line fault tolerance. The MC88200 provides facilities for testing each line of the cache and determining its integrity. If a cache line causes errors, then it is disabled by setting the disable bit. More information on the disable bit and diagnostic facilities is provided in **3.9 CACHE DIAGNOSTIC PORTS**.

Whenever the MC88200 misses the data cache, it loads the cache line (unless the address used is cache inhibited). At the same time, it loads the address tag with the upper 20 bits of the translated physical address. If the physical address was translated through the BATC, the address tag contains the 13-bit physical block address with the upper seven bits of the block offset appended. If the physical address was translated through the PATC, the address tag contains the 20-bit page frame address. Figure 3-2 shows the address-tag generation.

Each cache set contains six bits that indicate the line usage. These bits are used by the LRU algorithm to select a cache line for replacement when the cache set is full. More information on the LRU algorithm can be found in **3.3 LRU LINE REPLACEMENT**.

3.2 MEMORY UPDATE POLICIES

The MC88200 provides two memory update policies: copyback and writethrough. In the copyback mode, processor writes are written to memory the first time that the line address

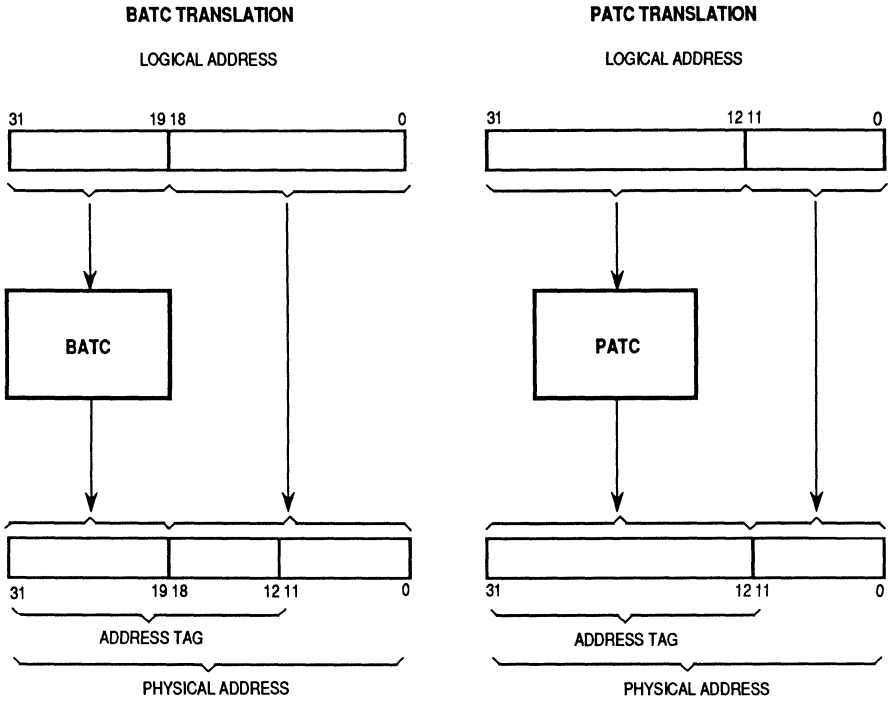


Figure 3-2. Physical Address Generation Using ATCs

is written (write-once). Subsequent updates are not written to memory until an explicit cache flush command is issued, a snoop hit causes a copyback operation, or, if the set is full and there is a miss. This mode reduces M bus traffic and decreases the latency of write transactions from the processor. In the copyback mode, memory data is not always updated, but in writethrough mode, data is written to memory every time the cache line is modified. With this mode, cache data is always up to date with memory.

When address translation is enabled, the memory update policy is controlled by the write-through (WT) bit in the BATC or PATC entry that was used to translate the corresponding logical address. When address translation is disabled, the WT bit in the SAPR and UAPR control the memory update policy. When this bit is set, the corresponding memory write transactions are performed in the writethrough mode. When the bit is clear, the copyback mode is used.

3.2.1 Copyback Mode

Copyback mode is optimal for high-use data that is closely coupled to a processor such as stacks and local variables. Both reads and writes that hit in the cache will complete in zero wait states. In general, addresses at which data is used by only one processor and no other M bus master should be mapped as local (G = 0), copyback (WT = 0) for maximum performance.

If more than one processor uses data stored in a page or block marked as copyback, snooping must be enabled to allow automatic copyback operation and cache invalidation of modified data. In this case, the page is mapped as global (G = 1), copyback (WT = 0). The MC88200 incorporates bus snooping to prevent other M bus devices from accessing stale data. When bus snooping is enabled, the MC88200 monitors the M bus transactions of other devices. When another device accesses memory data that is cached by the MC88200 and the M bus global (G) bit is set, the MC88200 pre-empts the other transaction and updates memory with the cached data (if the data is modified). See **3.5 CACHE COHERENCY** for complete information on bus snooping.

3.2.2 Writethrough Mode

Writethrough mode is used when cache and memory images should always agree (e.g., video memory) or with high-use shared (global) data. In this manner, memory is always updated on writes, and global transactions cause snoop logic to simply invalidate their cached images. Automatic copyback of cached data is not performed since valid cache data always agrees with memory.

3.2.3 Cache Inhibit

The memory update policy has no meaning when an access is cache inhibited. There are two cases where transactions are cache inhibited. First, a transaction is cache inhibited when the ATC entry used for the address translation has the CI bit set. Second, a transaction is cache inhibited when it is part of a locked operation (P bus $\overline{\text{DLOCK}}$ signal is asserted). See **3.6 CACHE AVOIDANCE** for more information.

3.3 LRU LINE REPLACEMENT

When a cache miss occurs and the cache set is full, the cache access algorithm replaces the LRU line in the set. When a line is selected for replacement, the MC88200 checks the VV bits. If the selected line is modified (exclusive modified state), then it is written back to memory. The MC88200 then reads the missed line from memory. This line overwrites the selected line in the set. The new line is marked as the most recently used, and the usage order of all other lines in the set is recomputed. If there are invalid (unused) lines in the cache, then data is loaded into those lines until the cache is full, and the LRU algorithm takes effect.

To maintain the LRU policy, each cache set includes LRU bits. This 6-bit field contains the current LRU line ordering and is updated after each cache access. The values of these bits are as follows:

L5 When set, line 3 more recently used than line 2.

L4 When set, line 3 more recently used than line 1.

L3 When set, line 3 more recently used than line 0.

L2 When set, line 2 more recently used than line 1.

L1 When set, line 2 more recently used than line 0.

L0 When set, line 1 more recently used than line 0.

As an example of the replacement line selection, when the line register has the value 111001 (L5–L0), then line 2 is the LRU line:

- Line 3 is the most recently used, since L5, L4, and L3 are set.
- Line 1 is next in the sequence, since L0 is set (more recent than line 0), L2 is clear (more recent than line 2), and L4 is set (less recent than line 3).
- Line 0 is next in sequence, since L3 and L0 are set (less recent than lines 3 and 1) and L1 is clear (more recent than line 2).

3.4 CACHE ACCESS

Each time the processor performs a P bus transaction with the MC88200, the MC88200 initiates a cache lookup operation. The MC88200 simultaneously performs the cache set selection and address translation. To achieve this concurrency, the MC88200 exploits the fact that the low-order 12 bits of an address are the same for both the logical and physical address. Since these bits do not have to be translated, cache set selection occurs in parallel with address translation performed by the memory management unit (MMU). The following steps provide a functional description of the cache set selection and data access. Refer to **2.1 ADDRESS TRANSLATION** for the flowchart of the following steps.

1. The MC88200 receives the processor logical address on the P bus. Three functions are performed concurrently:
   a. If data caching is enabled, set selection begins. Bits 11–4 of the logical address are used to select one of the 256 cache sets.
   b. The MMU compares bits 31–19 of the logical address and the supervisor/user bit to each entry in the BATC. If there is a hit, then the physical address is the 13-bit physical block address from the BATC, and the remaining bits of the logical address provide the block offset.
   c. The MMU compares bits 31–12 of the logical address and the supervisor/user bit to each entry in the PATC. If there is a hit, then the physical address is the 20-bit page frame address, and the remaining bits of the logical address provide the page offset. If both a BATC and PATC hit occur, then the BATC entry is used.

2. If there is no match in the BATC or PATC, the MC88200 performs a translation table search to create a PATC entry. The MC88200 performs the BATC/PATC lookups again, which result in a PATC hit.

3. When the physical address is generated, the upper 20 bits of the address are compared to the address tags in the selected set:

   a. If there is a match, the data access is performed with the selected line of the cache. Bits 3 and 2 of the logical address select a word from the specified line, and the data byte enable signals specify a byte, half-word, or word transaction. The data is read from/written to the cache, and the status bits and LRU information are updated accordingly.

   b. If there is not a match, memory is accessed to load a cache line. The cache line is selected according to the LRU algorithm; data is loaded into the line from the translated physical address that is quad-word aligned (A3–A0 = 0). Once the line is loaded, the data access is performed with the selected line in the cache.

If the logical address is cache inhibited, a memory transaction is always performed to access the required data. If a cache hit occurs on a cache-inhibited address, then the cache line is invalidated.

Detailed descriptions of the data cache operations are provided in **3.4.1 Cache Access: Read Transaction** and **3.4.2 Cache Access: Write Transaction**. These operations are presented as a series of actions with illustrative flowcharts. These descriptions represent a logical model of the MC88200 during cache access and do not necessarily indicate the exact operations of the MC88200. The MC88200 implements a high degree of concurrency; although the cache operations follow a given logical sequence, many of the individual actions are performed concurrently.

During the cache access, the line that contains the data read or written by the processor may change state. The two VV bits in each cache line reflect the state of the line with respect to memory, and whether or not the CMMU has exclusive ownership of the cached data. The encoded states are invalid (11), shared modified (10), exclusive modified (01), and exclusive unmodified (00).

The invalid state means the cache line contains no meaningful data. This status is used in two cases. First, software should invalidate the cache when the MC88200 is reset because these bits are undetermined at reset. Second, a line is marked invalid under certain conditions by the bus snooping logic. The shared unmodified state indicates that other caches may have a copy of this line and that this line is unmodified with respect to memory. The exclusive modified state signifies that this cache is the only cache with copy of this line and the line has been modified with respect to memory (dirty). If any word in the line is dirty, then the entire line is dirty. The exclusive unmodified state indicates that this cache is the only cache with a copy of this line and that the line is unmodified with respect to memory.

3.4.1 Cache Access: Read Transaction

Figure 3-3 shows the MC88200 action due to a read by the MC88100. It is assumed that a physical address has already been generated by the address translation. Figure 3-4 shows the functional timing for the read transaction.

If the read operation results in a data cache hit, the data is presented to the MC88100 one clock period after the address is latched. The MC88200 drives a "success" reply to the processor, signifying that the P bus transaction has been completed.

If there is a cache miss, the MC88200 drives the wait status on the reply lines. The wait is held until after an M bus transaction is performed to fill the line in the cache. The MC88200 then acquires the M bus through arbitration and selects the LRU cache line for replacement. If the line is modified, the MC88200 copies the line back to memory (writeback), marks the replacement cache line invalid, and performs an M bus read to read the required data from memory into the cache. The MC88200 then releases the M bus and marks the cache line as shared unmodified data. The requested data is provided to the processor on the P bus, and reply lines are driven with the "success" status.

If the MC88200 encounters an error condition, it releases the M bus. The P bus fault status register (PFSR) indicates the cause of the error, and the P bus fault address register (PFAR) contains the physical address where the error occurred. The P bus reply lines are driven with the "fault" status. The processor then takes the appropriate actions to handle the error condition (load a valid descriptor, abort the transaction, etc.).

If the MC88200 receives a retry M bus status, it releases the M bus and waits one clock cycle. Retry is normally issued by another MC88200 that is performing bus snooping. The one clock cycle allows the snooping device to acquire the M bus so that it can update memory with cached data.

3.4.2 Cache Access: Write Transaction

If the data is not in the cache (cache miss), the line containing the data is read from memory, then the modified data is written back to memory and to the cache. If the data is in the cache (cache hit), the operation depends on the memory update policy in effect. The MC88200 operation is significantly different for a write transaction with a cache hit and a write transaction with a cache miss. Figure 3-5 illustrates the transaction with a cache miss; Figure 3-6 illustrates the transaction with a cache hit.

3.4.2.1 CACHE WRITE: CACHE MISS. For a processor write with a cache miss, the MC88200 selects a new cache line using the LRU replacement algorithm. If the selected line is modified, the CMMU performs a copyback operation to write the line to memory. It then reads the required line with the intent-to-modify (IM) bit set, indicating that the next

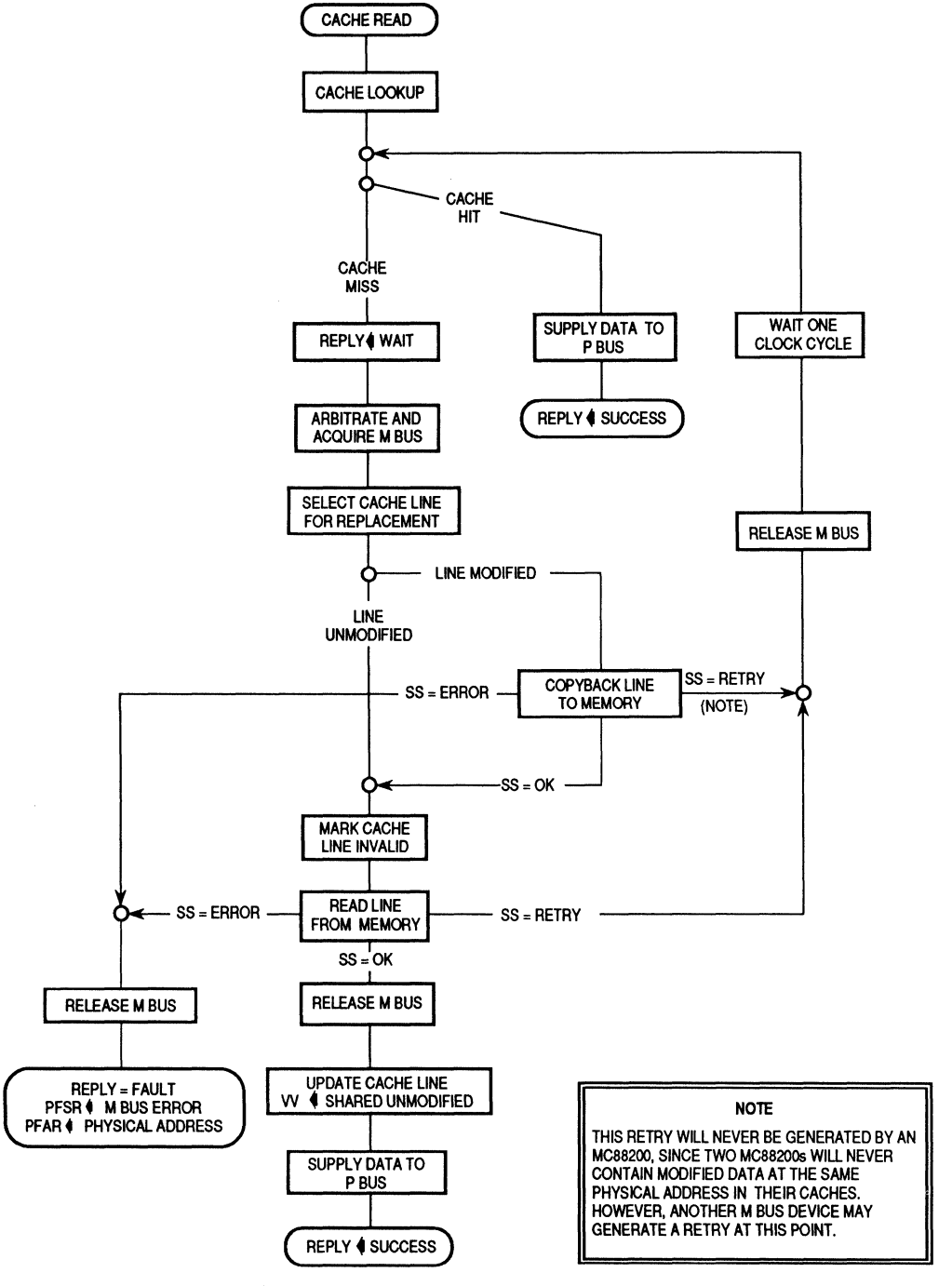


Figure 3-3. Cache Read Transaction

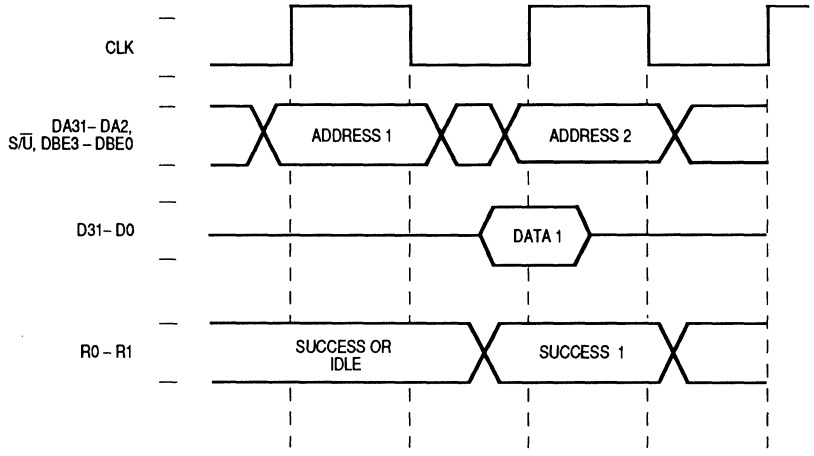


Figure 3-4. Functional Timing (Read Hit)

transaction will be a write to part of the one line's image in memory. Once the MC88200 writes the processor data to memory, it updates the cache entry and marks it as exclusive unmodified. The MC88200 releases the M bus and services the next transaction.

If the MC88200 encounters an error condition, it releases the M bus. The PFSR indicates the cause of the error, and the PFAR contains the physical address where the error occurred. The P bus reply lines are driven with the "fault" status. The processor then takes the appropriate actions to handle the error condition (load a valid descriptor, abort the transaction, etc.).

If the MC88200 receives a retry M bus status, it releases the M bus and waits one clock cycle. Retry is normally issued by another MC88200 that is performing bus snooping. The one clock cycle allows the snooping device to acquire the M bus so that it can update memory with cached data.

3.4.2.2 CACHE WRITE: CACHE HIT. For a processor write with a cache hit, the MC88200 determines if the line is a) exclusive, b) mapped as writethrough, c) mapped as global copyback, or d) mapped as local copyback.

If the line is marked as exclusive, the MC88200 simply updates the cache line with the processor data and marks the line as exclusive modified. No memory update is performed since the CMMU has exclusive ownership of this data.

If the processor access is mapped as writethrough, then the MC88200 writes the processor data to the cache line and the corresponding memory location. It marks the line as shared unmodified and services the next transaction.

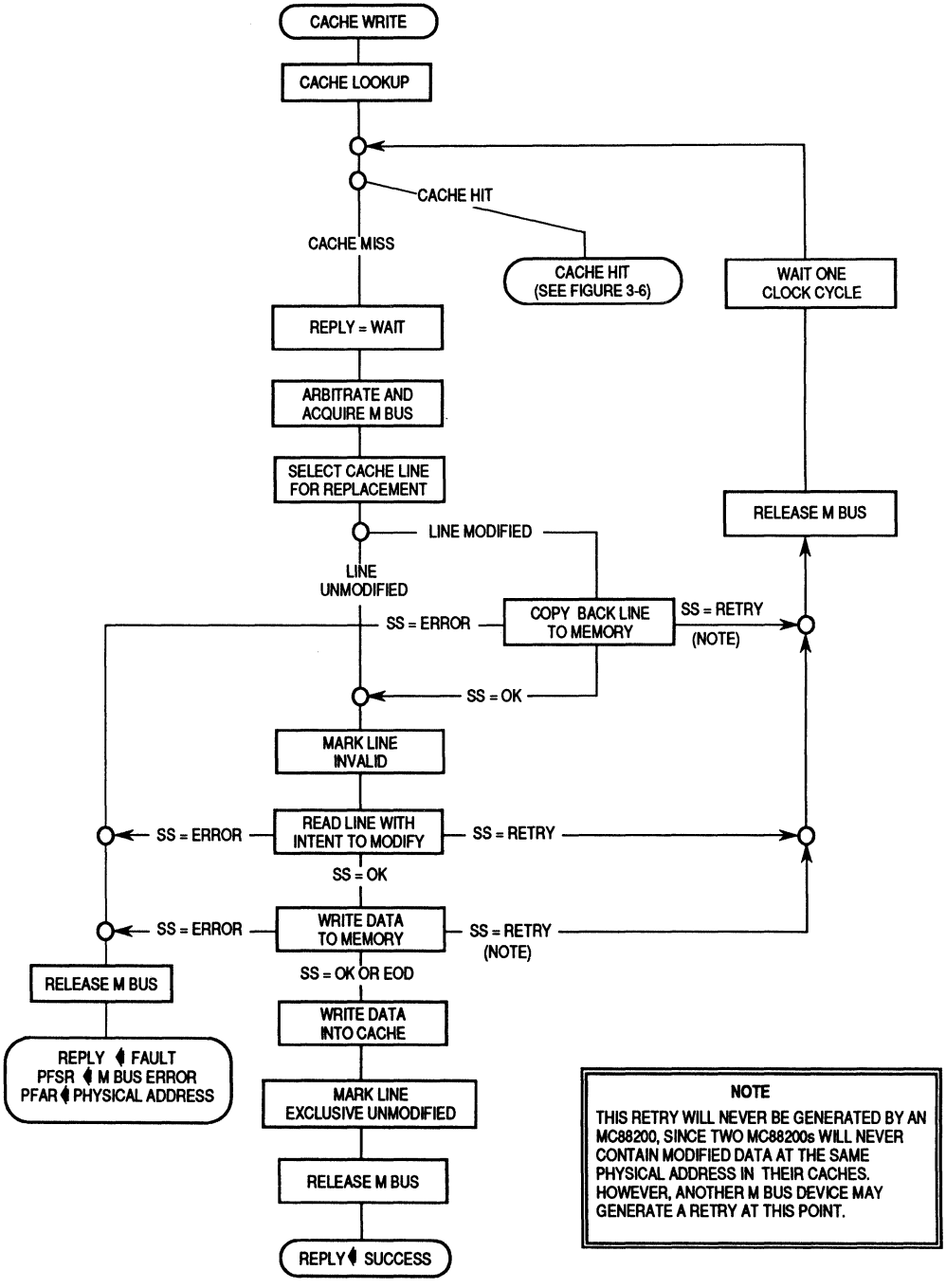


Figure 3-5. Cache Write with Cache Miss

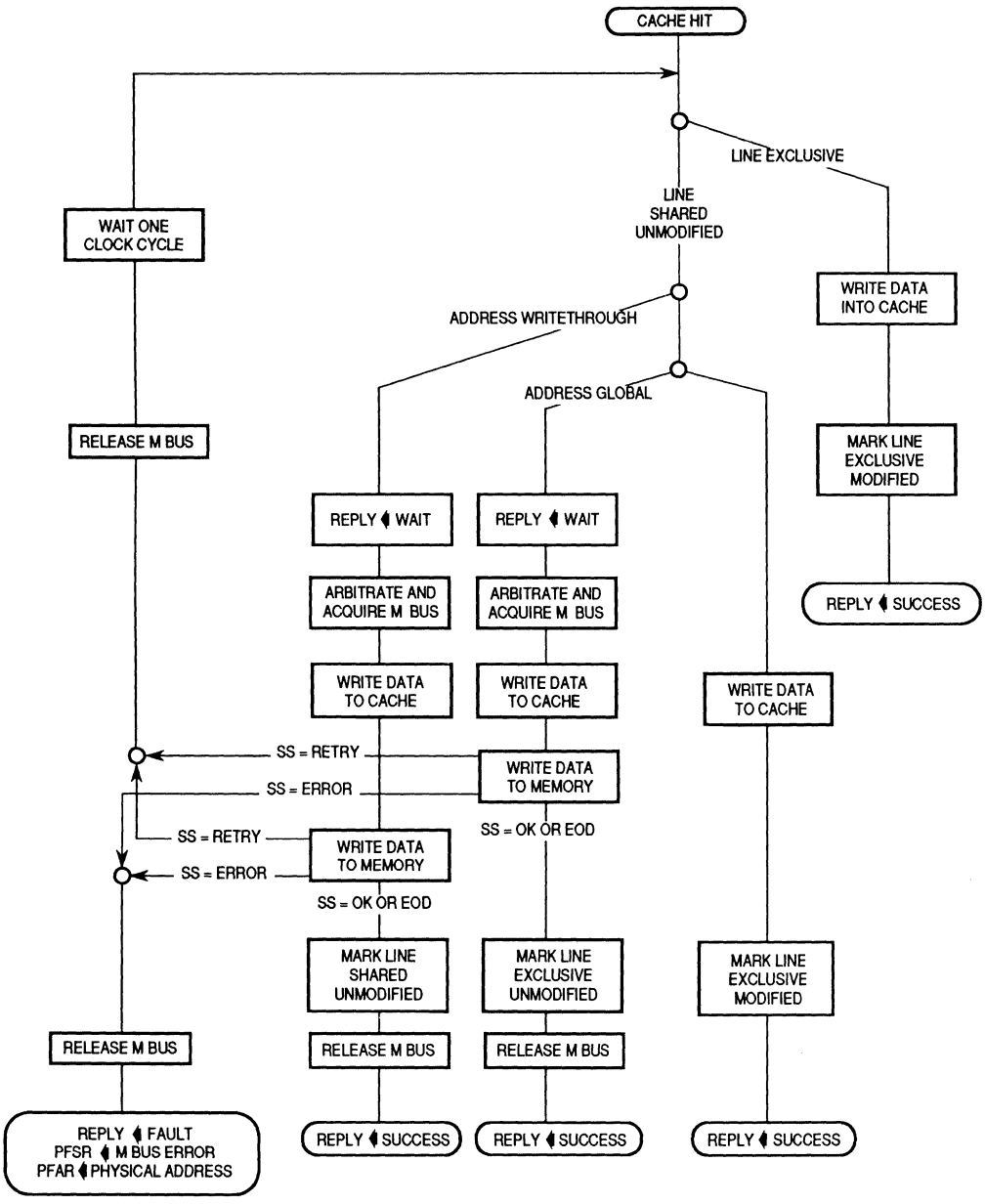


Figure 3-6. Cache Write with Cache Hit

If the process access is mapped as global copyback, the MC88200 writes data to the cache and to the memory to enforce the write-once copyback policy. It marks the line as exclusive unmodified and services the next transaction. Subsequent writes that hit the cache entry will not write through to memory until the cache is flushed, the line is replaced, or snoop logic forces a line copyback of modified data.

Finally, if the processor access is mapped as local copyback, the MC88200 simply updates the cache line with the processor data. No memory update is necessary because the data is local to this processor. The MC88200 marks the line as exclusive modified and services the next transaction. Figure 3-6 illustrates the actions done by the MC88200 during a processor write that hits in the data cache.

If the MC88200 encounters an error condition, it releases the M bus. The PFSR indicates the cause of the error, and the PFAR contains the address where the error occurred. The P bus reply lines are driven with the "fault" status. The processor then takes the appropriate actions to handle the error condition (load a valid descriptor, abort the transaction, etc.).

If the MC88200 receives a retry M bus status, it releases the M bus, waits one clock cycle, and requests ownership of the bus again. Retry is normally issued by another MC88200 that is performing bus snooping. The one clock cycle allows the snooping device to acquire the M bus so that it can update memory with cached data.

3.4.3 Cache Access: xmem Instruction

The MC88100 supports an exchange memory (**xmem**) instruction that is effectively a combination of a load and store instruction. The MC88100 implements the **xmem** instruction as a read access followed by a write access; the bus lock ($\overline{\text{DLOCK}}$) signal is maintained by both the processor and the MC88200. The MC88200 automatically cache inhibits all P bus locked transactions. As in all cache transactions, a data cache lookup is performed, and, if a hit occurs, the line is invalidated (refer to **3.6 CACHE AVOIDANCE** for more detailed information). However, unlike normal loads and stores to addresses mapped as cache inhibited, the locked transactions will copy back a modified line to memory before invalidating it. This guarantees that a lock (**xmem**) transaction that hits a modified cache line (in an address space marked as cachable) does not cause a loss of exclusive modified data. Figure 3-7 shows a flowchart of the **xmem** operations. **SECTION 5 BUS OPERATION** provides functional timing for the **xmem** operation.

If the MC88200 encounters an error condition during the **xmem** sequence, it releases the M bus. The PFSR indicates the cause of the error, and the PFAR contains the physical address where the error occurred. The P bus reply lines are driven with the "fault" status. The processor then takes the appropriate actions to handle the error condition (load a valid descriptor, abort the transaction, etc.).

If the MC88200 receives a retry M bus status during the copyback or data read operation, it releases the M bus and waits one clock cycle. Retry is normally issued by another MC88200 that is performing bus snooping. Snooping CMMUs will only hit the data read operation.

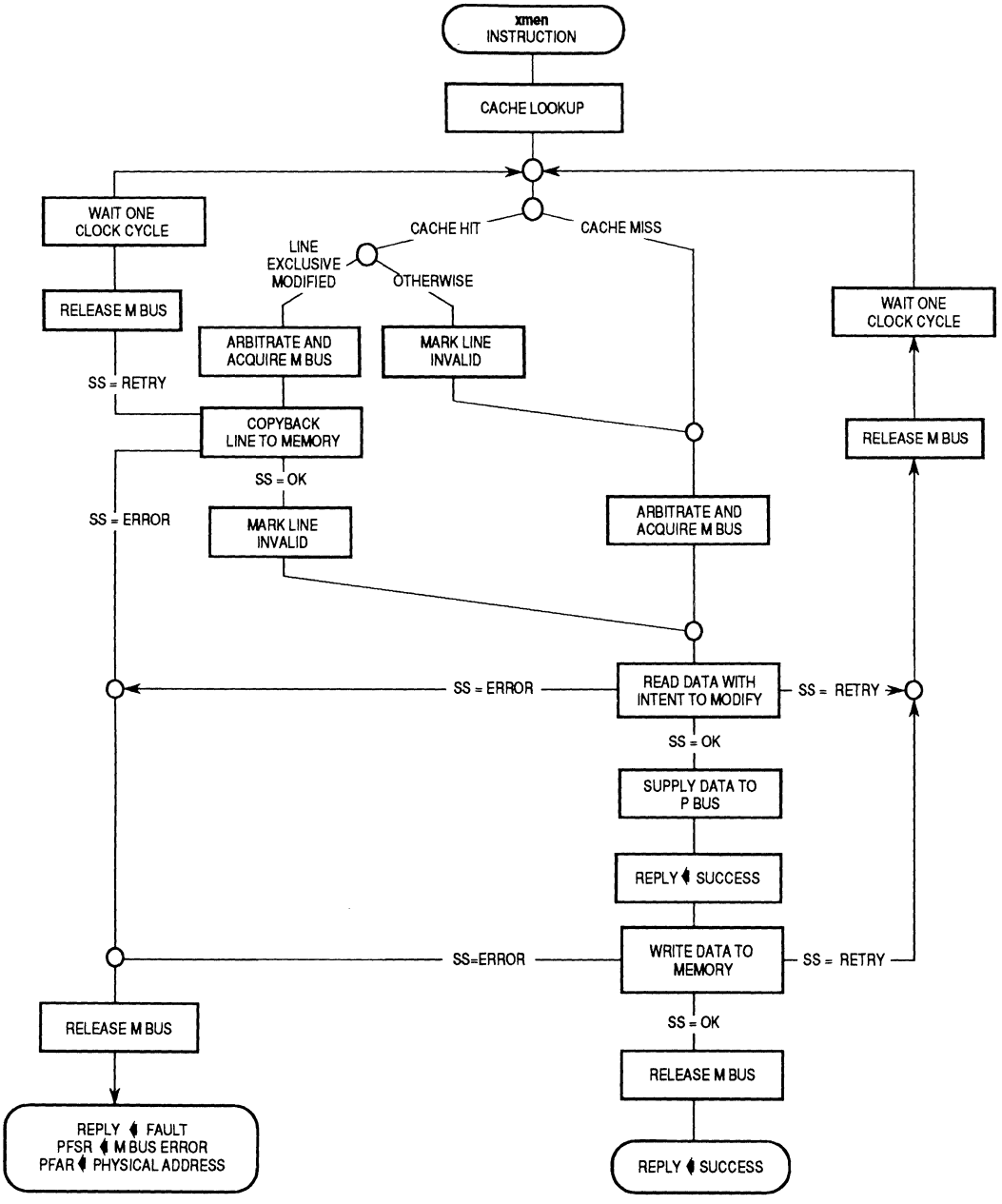


Figure 3-7. xmem Instruction Operations

By the time the data write operation is performed, all snooping CMMUs have invalidated and/or copied back their corresponding cache lines. Therefore, retries of the data write by a CMMU will never occur. In general, the MC88200 will not support a retry of a locked (LK = 1) M bus write transaction. The one clock cycle wait after retry allows the snooping device to acquire the M bus so that it can update memory with cached data.

3.5 CACHE COHERENCY

The MC88200 can automatically maintain coherency between cached and in-memory copies of data. To maintain this coherency, the MC88200 incorporates bus snooping. With bus snooping, the MC88200 monitors the M bus transactions of other M bus devices. If another device accesses data that is modified in the MC88200 cache, the MC88200 takes control of the bus and updates the data in memory. Then, the other M bus device can access the correct data.

The MC88200 snoops M bus transactions by monitoring the M bus control and information lines. When the global (G) control bit is asserted, the MC88200 compares the M bus address to the address tags in the cache. If there is a hit and the data is modified, the MC88200 pre-empts the other transaction and writes the modified data to memory. In addition, if the global transaction has the intent-to-modify (IM) contol bit set, the MC88200 marks the cache line invalid.

The IM bit is asserted during M bus writes, and reads that are followed by a write to the same address. It signifies that the MC88200 is intending to modify data at the address driven when the IM bit is asserted. In most cases, the IM bit is clear on M bus reads and set on M bus writes. Table 3-1 lists the setting of the IM bit for M bus transactions.

Table 3-1. IM Bit Settings

M Bus Transaction	IM Bit Setting
Normal Read (LK = 0)	0
xmem Read (LK = 1)	1
Cache Lines Read Due to a Write Miss (LK = X)	1
Write (LK = X)	1

Exceptions that affect cache coherency are during locked read/write **xmem** transactions and cache line fills (reads) due to write misses. In this case, the IM bit is set on both the read and write cycles. A snooping CMMU can hit on a global read with intent to modify, copy back its modified data, and invalidate the line in the data cache. The snooping CMMU will monitor the following **xmem** write but will never hit since the line was copied back already. Therefore, a CMMU will never retry a locked (**xmem**) write.

In the second case, line fills during a write miss will always be followed by a write (once) to the same location (refer to **3.4.2.1 CACHE WRITE: CACHE MISS** for more information).

Therefore, the IM bit will be set on the line read, and snooping MC88200s may retry this read to copy back their modified data to memory.

3.5.1 M Bus Snooping: Transaction without Intent to Modify

The MC88200 performs the following actions (see Figure 3-8) when snooping an M bus read with IM = 0. These actions represent the logical flow of operations; since the MC88200 incorporates a high degree of concurrency, some of the operations are performed in parallel. The MC88200 is an M bus slave until it acquires the M bus. The device initiating the M bus transaction is the M bus master until it receives the retry signal or completes the transaction.

When a CMMU snoops a global read transaction that hits the data cache, it determines if the cache data is modified or not. If it is unmodified, the MC88200 marks the line as shared unmodified and services the next transaction. In this manner, the MC88200 recognizes that other CMMUs have read access to the global data. If the line is modified, it aborts the read operation with retry, copies back the modified line, marks it shared unmodified, then allows read operations to restart.

If the MC88200 encounters an error condition, it releases the M bus and the CE bit in the SSR is set. If the MC88200 receives a retry M bus status, it releases the M bus and waits one clock cycle. Retry is normally issued by another MC88200 that is performing bus snooping. The one clock cycle allows the snooping device to acquire the M bus so that it can update memory with cached data.

3.5.2 M Bus Snooping: Transaction with Intent to Modify

The MC88200 performs the following actions when snooping an M bus write or an M bus read with intent to modify (see Figure 3-9). They represent the logical flow of operations; since the MC88200 incorporates a high degree of concurrency, some of the operations are performed in parallel. The MC88200 is the M bus slave until it acquires the M bus; the device initiating the memory transaction is the M bus master until it receives the retry signal.

A snooping CMMU that hits on a global write or global read with intent to modify must determine if the cache line is unmodified. If the line that hit is modified, it must be copied back to memory before the CMMU performing the global intent to modify can complete its transaction. Once the line is copied back, the snooping CMMU invalidates its copy in the data cache and services the next transaction. If the line that hit was not modified, the snooping CMMU (slave) simply invalidates its cache line and services the next transaction.

If the MC88200 encounters an error condition, it releases the M bus, and the CE bit (bit 15) in the SSR is set. If the MC88200 receives a retry M bus status, it releases the M bus and waits one clock cycle. Retry is normally issued by another MC88200 that is performing bus snooping. The one clock cycle allows the snooping device to acquire the M bus so that it can update memory with cached data.

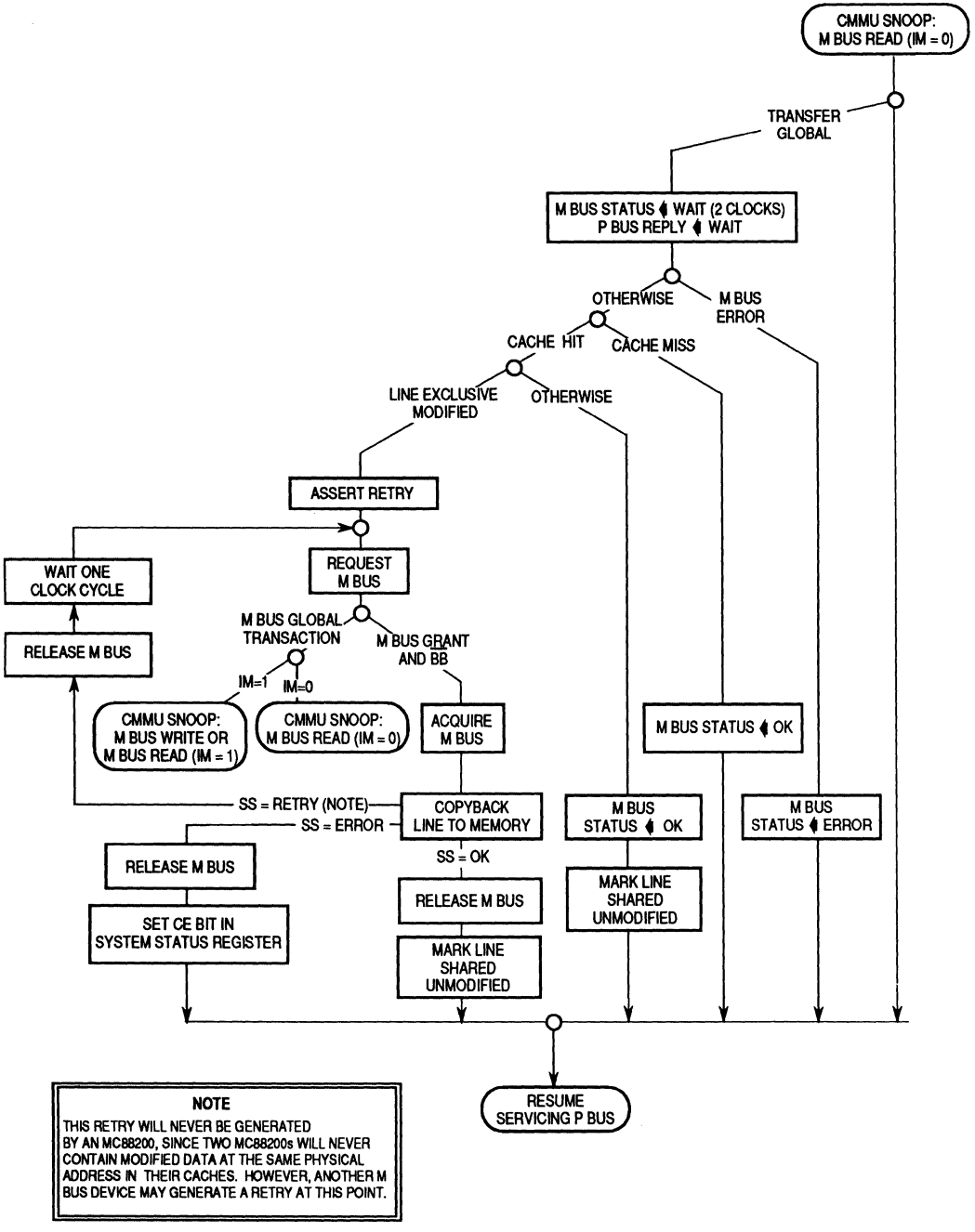


Figure 3-8. M Bus Snoop: Read Transaction (IM = 0)

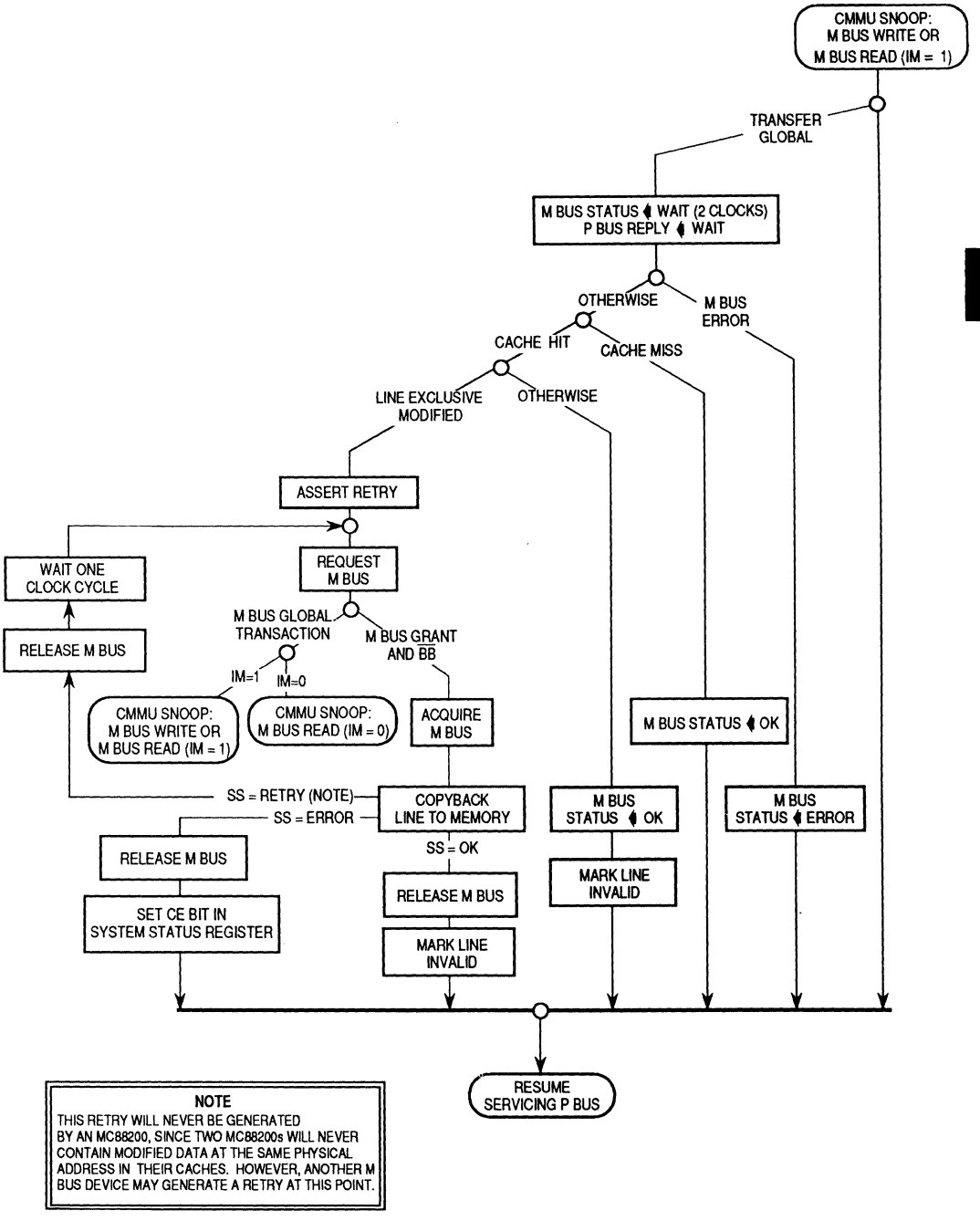


Figure 3-9. M Bus Snooping: Transaction with Intent to Modify

3.6 CACHE AVOIDANCE

Memory locations can be declared cache inhibited; data at these locations will never be stored in the MC88200 data cache. A location is cache inhibited when the CI bit is set in the location's address translation. The BATC and PATC entries both contain a CI bit. The CI bit in a BATC entry is set by software. (The two hardwired BATC entries permanently have CI set). The CI bit in a PATC entry is generated from the logical OR of the CI bits in the area, segment, and page secondary descriptors encountered when the PATC entry is created. The value of the CI bit is echoed on the M bus (C4) so that external (secondary) data caches can use the signal if appropriate.

Certain transactions are automatically cache inhibited. The data cache is bypassed whenever the MC88200 executes a table search, meaning that segment and page descriptors are never cached in the data cache. However, the cache-inhibit signal (C4) is not asserted on the M bus during a table search, which allows descriptors to be cached in secondary external caches. Control space accesses are always cache inhibited since they reside in the top 1M byte of supervisor memory mapped by the two hardwired BATC entries. P bus locked transactions are always cache inhibited; these transactions are signaled when the $\overline{\text{DLOCK}}$ P bus signal is asserted. This causes the cache-inhibited (C4) signal to be asserted on the M bus during P bus locked transactions regardless of the CI bit determined through address translation.

A P bus transaction that is translated as a cache-inhibited access and hits in the data cache causes the corresponding data cache line to be invalidated (VV = 11). No line copyback is performed if the line is marked as exclusive modified.

3.7 CACHE FLUSHES

The MC88200 supports software-initiated cache flushes. When the software initiates a cache flush, the MC88200 performs one of three actions:

1. Invalidates data in the cache
2. Writes modified data back to memory
3. Writes modified data back to memory, then invalidates data in the cache

Software initiates a cache flush by writing a command into the command code field (bits 5–0) of the system command register (SCR):

(Bits 5–0 of SCR)	(Command Action)
0101gg:	Data Cache Invalidate
0110gg:	Data Cache Copyback to Memory
0111gg:	Data Cache Copyback and Invalidate

where gg specifies the granularity:

00 — Line
01 — Page
10 — Segment
11 — All

For the line, page, and segment granularities, the software writes the physical address to the system address to specify the desired data to flush. For the "all" granularity, the entire cache is flushed.

When a cache flush is in progress, the control registers cannot be read. If another M bus device tries to access the MC88200 control registers, the MC88200 issues a retry status on the system status lines ($\overline{SS3}$–$\overline{SS0}$). If the access is made from the P bus (local processor), then the P bus reply lines (R1–R0) are driven with the wait status. These statuses are driven until the cache flush finishes, meaning that the control registers cannot be accessed until the flush finishes.

To determine the success or failure of a cache flush that copies data back to memory, the initiating device (P bus processor or other M bus device) must test the BE bit (bit 14) of the SSR. Since this register is not available until the flush completes, the initiating device should test the register only after a given time interval. Delaying the register test reduces unnecessary M bus traffic. For an invalidate, the delay is measured in clock cycles. For a copyback or copyback plus invalidate, the delay entails one burst transaction for each line copyback plus the clock cycles needed for the invalidations. Table 3-2 lists the clock cycles for cache invalidate.

Table 3-2. Clock Cycles for Cache Invalidate

Granularity	Minimum Cycles
Line	6
Page or All	260
Segment	1024

The following steps describe the actions of a cache flush:

1. If the cache flush will be for a line, page, or segment, then the software writes the appropriate physical address to the system address register (SAR).
2. The software writes the appropriate command to the command code field of the system control register (SCTR).
3. For a flush of the entire cache (granularity "all"), the MC88200 performs the following actions for each cache line.
   a. If the status (VV) bits are 01 (exclusive modified), the MC88200 writes the line to memory for data write and data write with invalidate flush commands.
   b. For invalidate and data write with invalidate flush commands, the MC88200 marks each line invalid by setting the VV bits (11).

4. For a segment flush, bits 31–22 of the SAR specify the segment address. The MC88200 performs the following actions:
   a. The MC88200 tests the upper 10 bits of each address tag in the cache. If the address matches, the MC88200 performs the following actions:
      (1) If the status (VV) bits are 01 (exclusive modified), the MC88200 writes the line to memory for data write and data write with invalidate flushes.
      (2) For invalidate and data write with invalidate flushes, the MC88200 marks each line invalid by setting the VV bits (11).
5. For a page flush, bits 31–22 of the system address register specify the page address. The MC88200 performs the following actions:
   a. The MC88200 tests all 20 bits of each address tag in the cache. If the address matches, the MC88200 performs the following actions:
      (1) If the status (VV) bits are 01 (exclusive modified), the MC88200 writes the line to memory for data write and data write with invalidate flushes.
      (2) For invalidate and data write with invalidate flushes, the MC88200 marks each line invalid by setting the VV bits (11).
      (3) After the MC88200 writes and/or invalidates the line, it tests the next set in the cache. These actions are repeated until all cache sets are checked.
6. For a line flush, bits 31–12 of the SAR specify the page address, and bits 11–4 specify a cache set. The MC88200 performs the following actions:
   a. The MC88200 compares the upper 20 bits of each address tag in the set with bits 31–12 of the system address register. If the address matches, the MC88200 performs the following actions:
      (1) If the status (VV) bits are 01 (exclusive modified), the MC88200 writes the line to memory for data write and data write with invalidate flushes.
      (2) For invalidate and data write with invalidate flushes, the MC88200 marks each line invalid by setting the VV bits (11).

The cache is not accessible to the processor while the flush is in progress. However, the MC88200 will perform bus snooping (if enabled) while the flush is occuring. On multiple copybacks, a bus tenure is requested on a per-dirty-line basis.

3.8 CACHE DATA STATE

The following three diagrams summarize the cache data states and the processor/M bus operations that place data in one of the four states (invalid (I), shared unmodified (SU), exclusive unmodified (EU), exclusive modified (EM)). The invalid state means that data in the cache line does not agree with the copy of the line in memory. This situation arises when an M bus master updates data in memory cached by a snooping CMMU. Lines are also invalidated by software upon reset or through a cache flush operation. The shared

unmodified state indicates that multiple CMMUs may be caching the same data, but none of the lines containing the data are modified with respect to memory. The exclusive unmodified state means that the CMMU has written data in the line to memory in accordance with the write-once policy. Both cache line and memory agree, and the CMMU has "exclusive ownership" of the data. No other CMMU can cache data at this address at the same time. The exclusive modified state means that data in the line has been modified (written by the MC88100) with respect to memory.

Global data is data that is transferred on the M bus with the global (G) bit (C5 pin) set during the address phase. The M bus states of global and local are important to snooping. Data that is marked global on the M bus is snooped by each CMMU. Data that is marked local is not snooped. The status of the global bit is driven during the address phase of the M bus master and is tested by each CMMU that is snoop enabled via the system control register SE bit.

If the global bit is set due to a P bus to M bus translation or set during a CMMU snoop operation, Figure 3-10 shows the state diagram to modify the state of the data in the cache and enforce cache coherency. If the global bit is not set, indicating data local to a particular CMMU, Figure 3-11 shows the state diagram to enforce cache coherency. If the writethrough (WT, bit 4) field of the BATC or PATC entry used in the translation is set, Figure 3-12 shows the state diagram to maintain cache coherency. Once the data is loaded into the data cache, no information regarding the status of local, global, or writethrough is retained. The global bit, M bus intent-to-modify bit, and transaction type (CPU read/write, copyback, and/or invalidate command) is used to determine the next state of the line of data in the cache. Table 3-3 shows which state diagram to use, depending on values for the G and WT bits.

Table 3-3. State Diagram Selection

G Bit (M Bus C5 — Address Phase)	WT Bit (BATC/PATC Bit 4)	Cache Coherency State Diagram
1	0	Figure 3-10
0	0	Figure 3-11
X	1	Figure 3-12

Due to the cache coherency scheme, no two CMMUs may cache exclusive data in cache lines having the same physical address tags. The transition of global data from invalid or shared to exclusive will invalidate all copies of the data in other CMMUs if they are snoop enabled. If one of those copies is exclusive modified, it is written back to memory before the line is invalidated. It is not recommended that data at the same physical address be mapped as global by certain CMMUs and as local by others, or mapped locally by more than one CMMU. Cache coherency is not maintained, and this practice could result in stale data resident in CMMUs.

Figure 3-10 shows the states for global data, Figure 3-11 shows the states for local data, and Figure 3-12 shows the states of data (global or local) when the writethrough memory update policy is in effect.

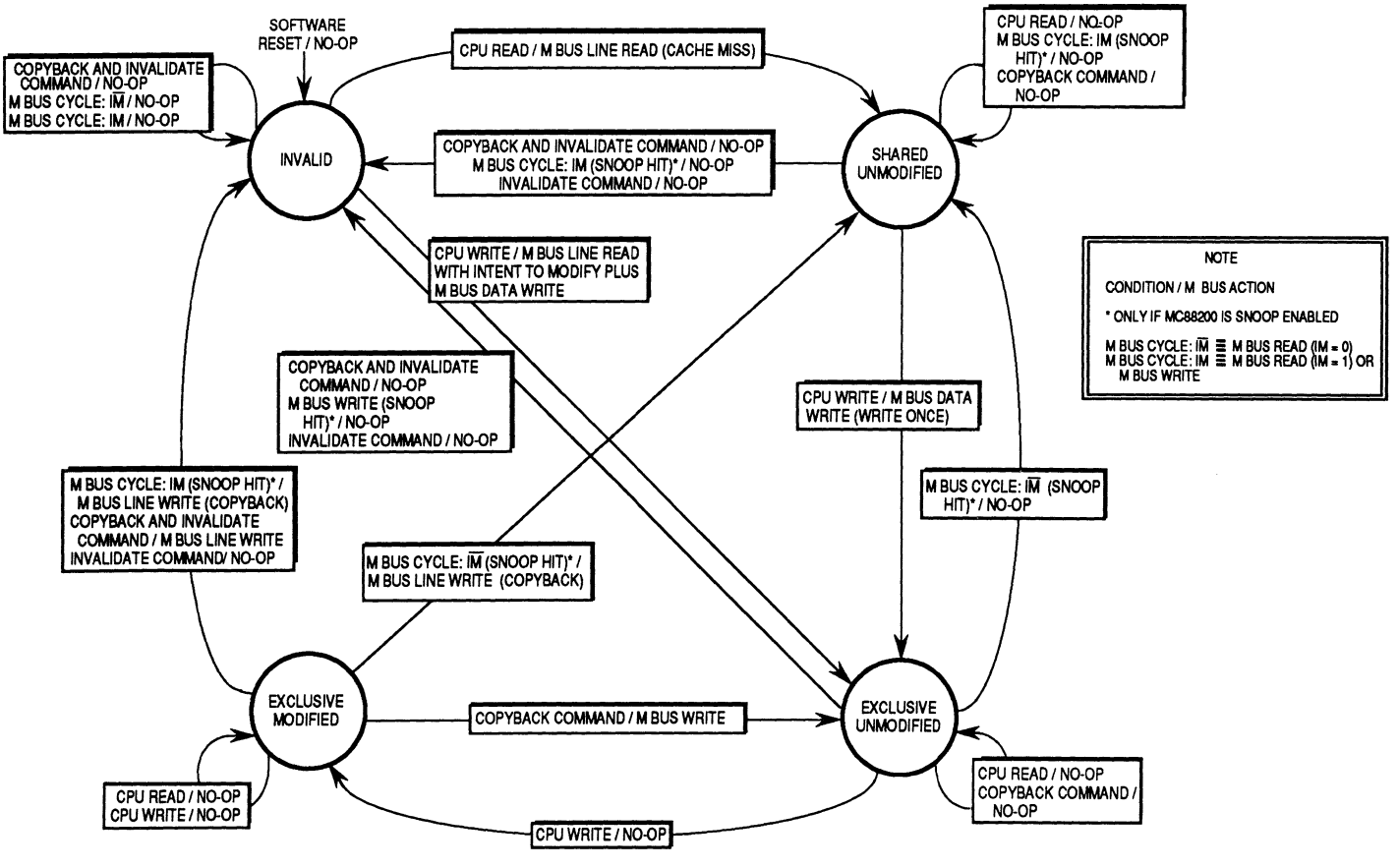


Figure 3-10. Cache Data States — Global Data

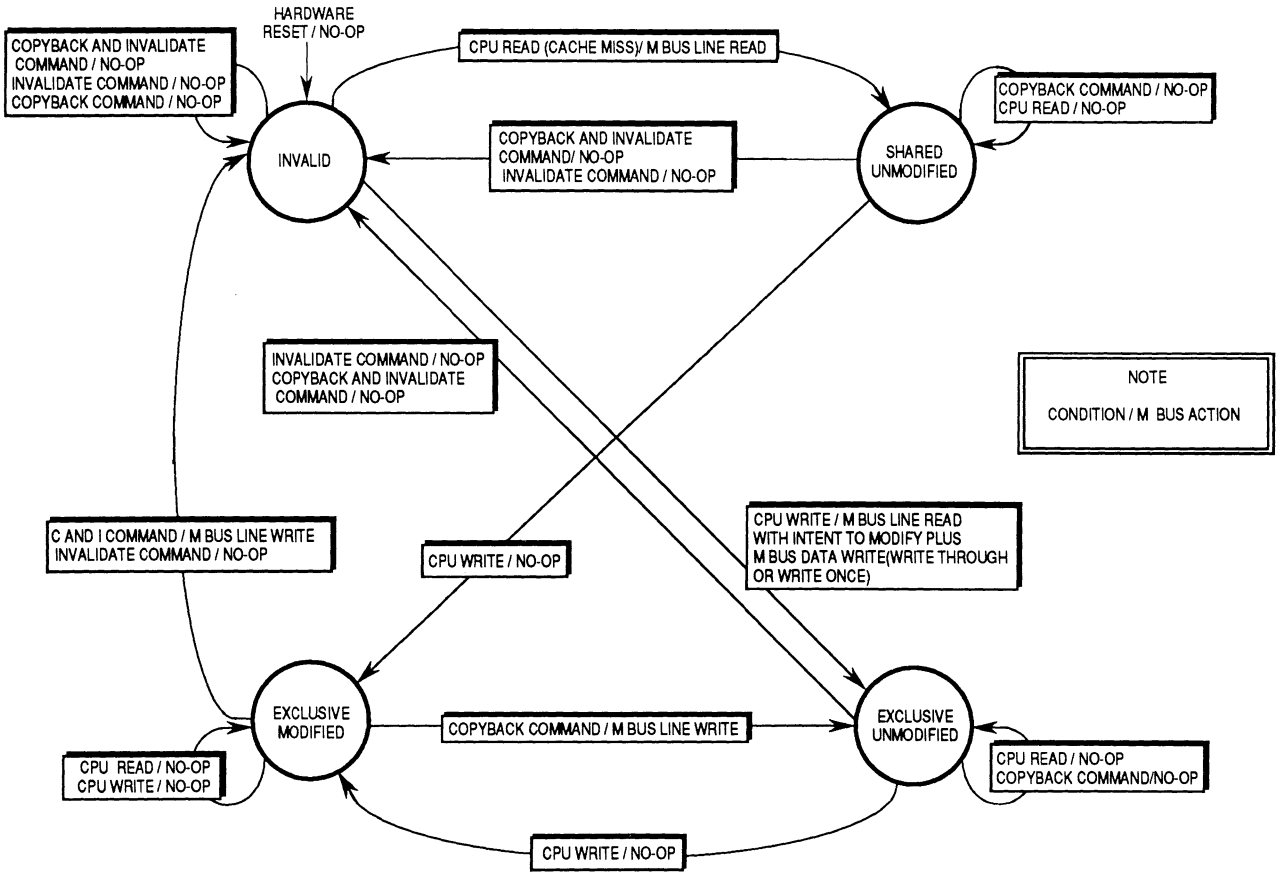


Figure 3-11. Cache Data States — Local Data

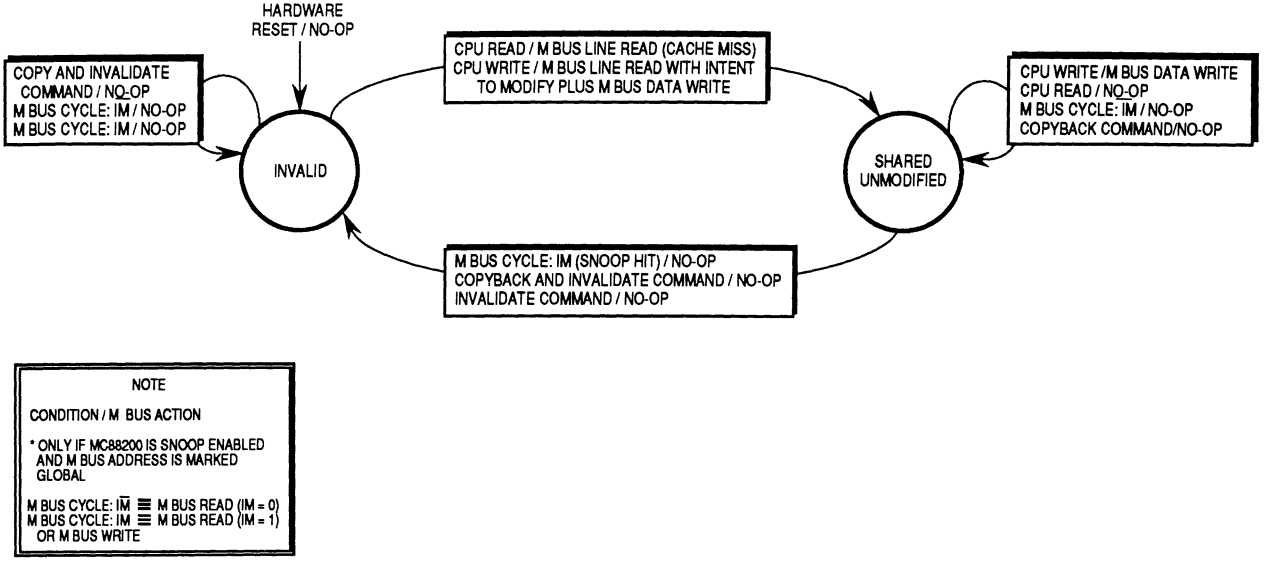


Figure 3-12. Cache Data State — Writethrough Memory Update Policy

Figures 3-13 through 3-20 illustrate how snooping maintains cache coherency in a multiprocessing configuration. In the figures, there are two MC88100 CPUs, each with one MC88200 CMMU on the data P bus (for the sake of clarity, instruction CMMUs are not shown). Shown inside the CMMU blocks is the present state of a line (invalid (INV), shared unmodified (SU), exclusive unmodified (EU), or exclusive modified (EM)) and the next state as a result of P bus transactions or snooping. Also shown inside the CMMUs are a line address tag and the associated line data. This example uses the same line in the data caches for simplicity. P bus transactions use a logical address (LA); whereas, M bus transactions use the corresponding physical address (PA) and tags only the upper 20 bits of the physical address (PA[31–12]). Logical address (LA) selects word 0 of the cache line, LA + 4 selects word 1, LA + 8 selects word 2, and LA + 12 selects word 3. Physical address (PA), PA + 4, PA + 8, and PA + 12 select the corresponding copies of the data in memory. Line reads perform four consecutive word reads from memory address PA, PA + 4, PA + 8, and PA + 12 to the cache line, using an efficient burst mode transfer. Line copybacks write (burst) the four words from the cache line back to memory. All addresses are mapped as global, copyback, cachable, with no write-protect (WT, CI, WP = 0; G = 1). During the example, refer to Figure 3-10 and follow the corresponding state transitions on the state diagram.

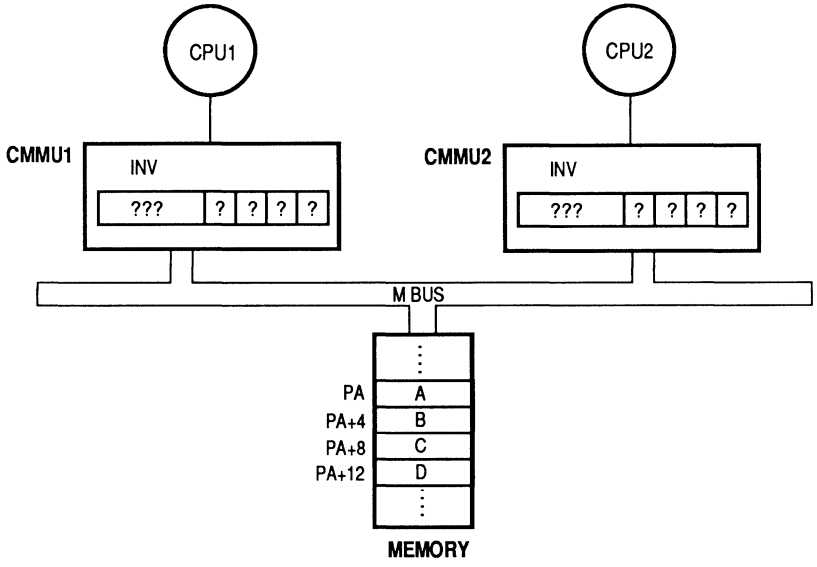


Figure 3-13. Initial State of System

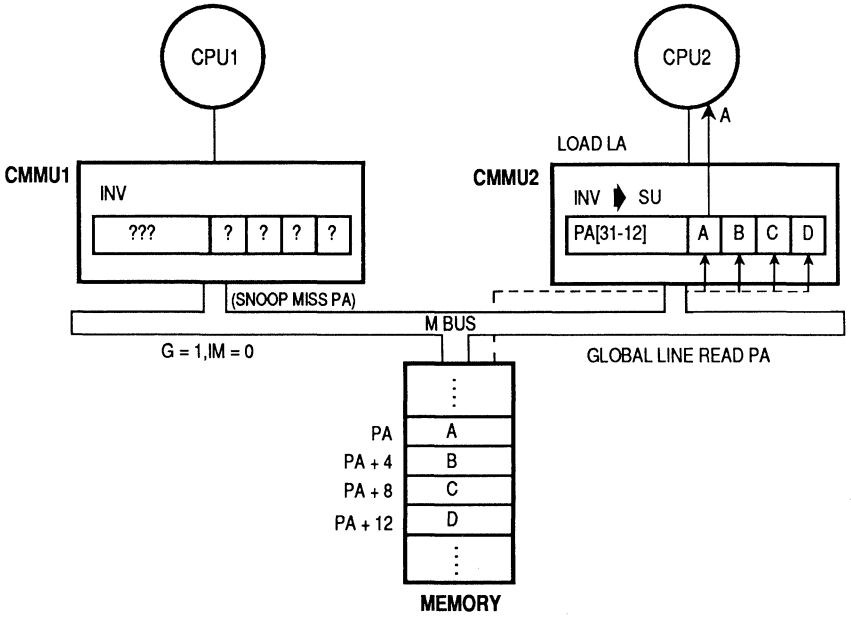


Figure 3-14. CPU2 Load, Data Cache Miss

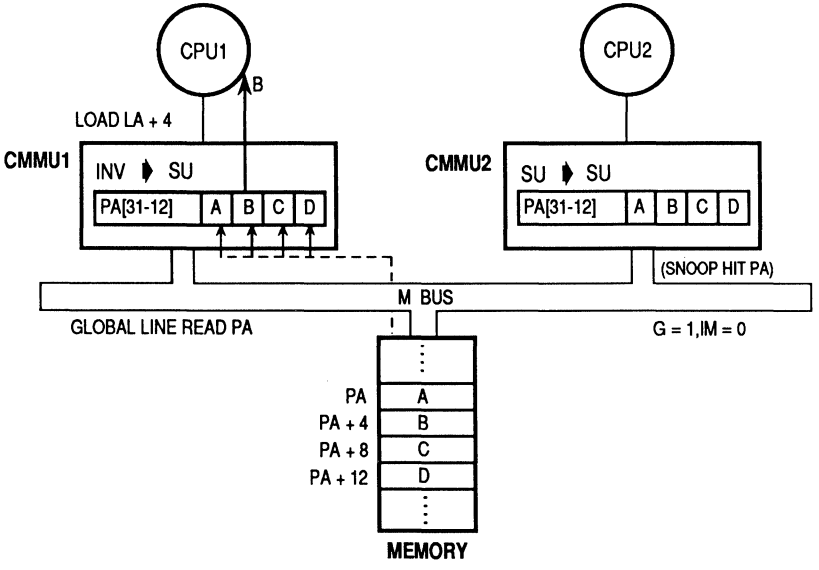


Figure 3-15. CPU1 Load, Data Cache Miss

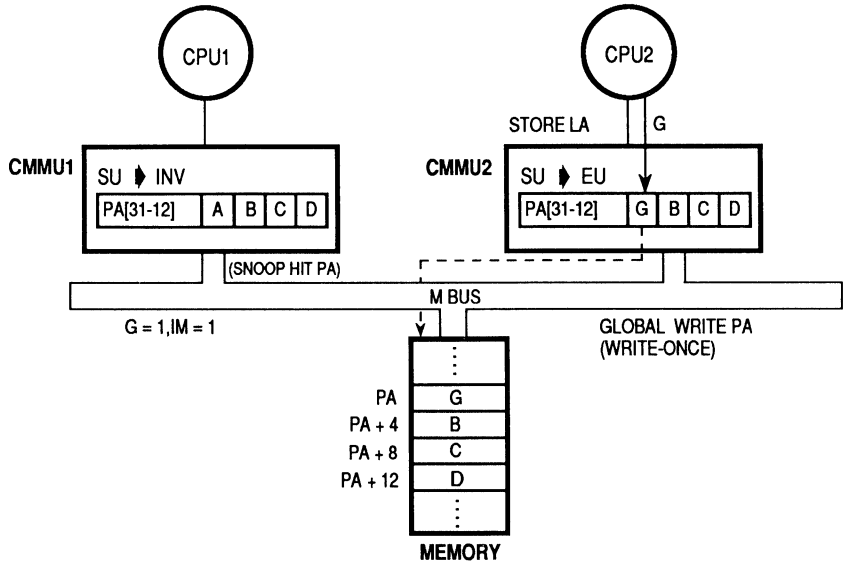


Figure 3-16. CPU2 Store, Data Cache Hit (Write-Once)

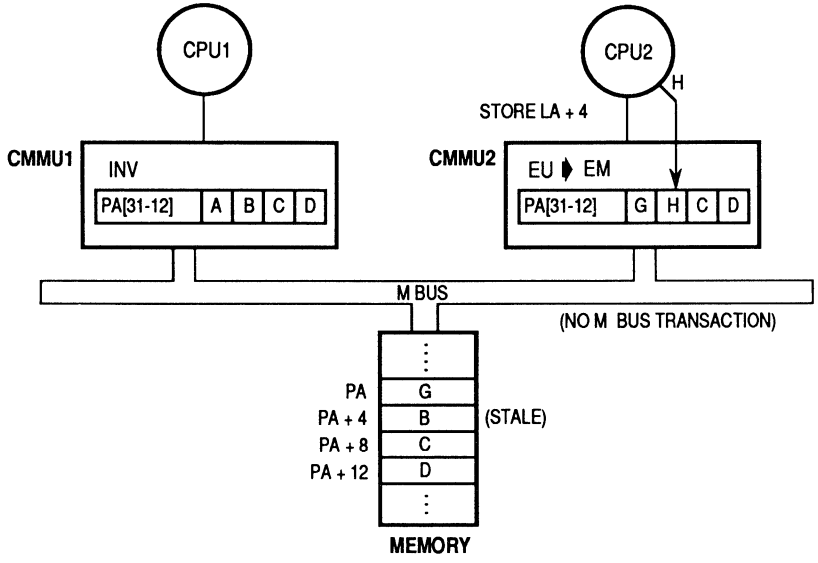


Figure 3-17. CPU2 Store, Data Cache Hit

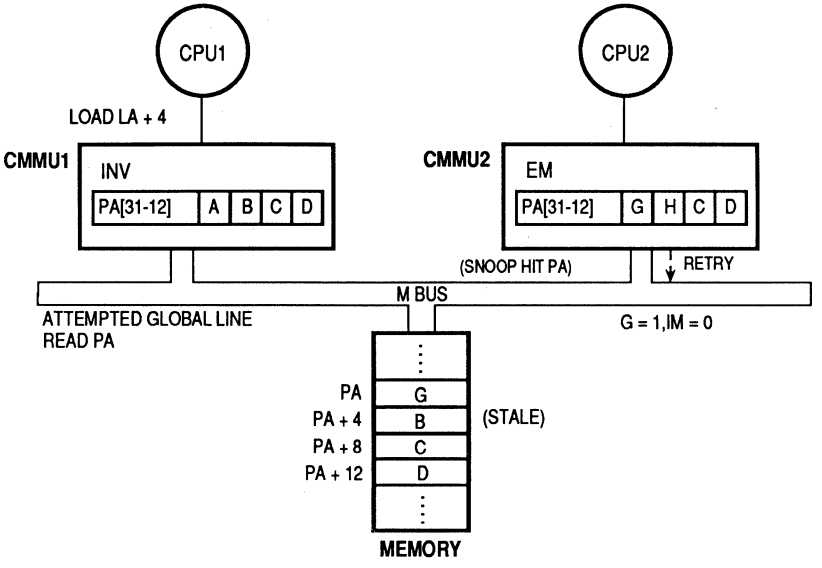


Figure 3-18. CPU1 Load, Cache Miss, Line Read Retried

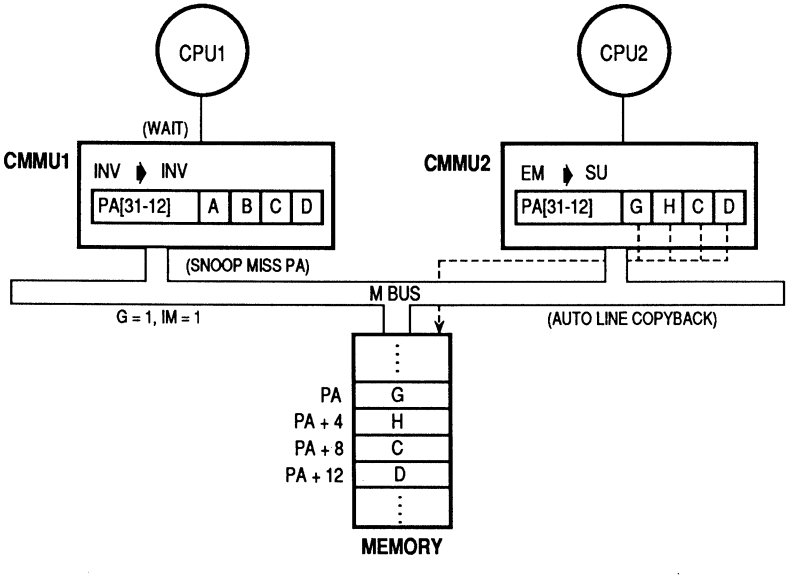


Figure 3-19. CMMU2 Line Copyback

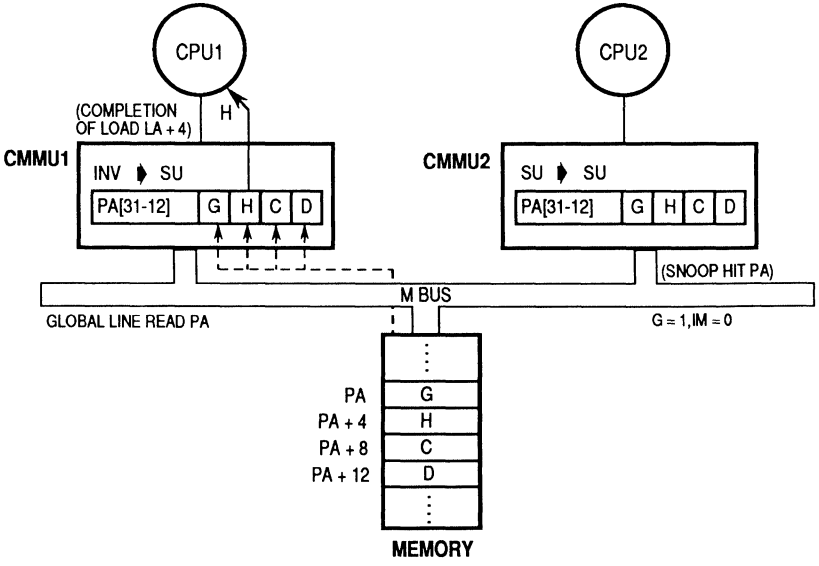


Figure 3-20. Completion CPU1 Load, Cache Miss

Figure 3-13 shows the CMMUs in their initial state, with both lines invalidated and their contents unknown. This would be the state of the CMMU data cache after reset, with software invalidation of all cache lines.

Figure 3-14 shows CPU2 performing a load word from location LA. There is a data cache miss and the CMMU reads a line from memory to fill the cache line. CMMU1 monitors (snoops) the M bus transaction, but does not find a tag match (a "miss") since the line the snoop operation selects is invalidated. CMMU2 supplies CPU2 with the required word and the state of the cache line is updated to shared unmodified.

Figure 3-15 shows CPU1 loading a word from LA + 4, which misses the selected cache line. A line-fill operation is performed as before, which loads four words from memory starting at location PA. CMMU2 snoops the global transaction and finds a tag match ("hit") in the line the snoop operation selects. The state of the line remains as shared unmodified since CMMU1's copy is the same (IM = 0).

Figure 3-16 shows CPU2 storing a word to address LA. A cache hit occurs, and both the cache line and memory are updated with the new data in accordance with the write-once policy. The effect of the single-word write going out on the M bus causes CMMU1 to invalidate the line containing its copy of the word, since it no longer agrees with memory. CMMU2 updates its line to the exclusive unmodified state. CMMU2 now has exclusive ownership of the entire line of data that is unmodified with respect to memory. The exclusive status means CMMU2 is guaranteed that no other CMMU on the M bus is caching a valid copy of the line.

Figure 3-17 shows CPU2 storing a word to logical address LA+4, which maps into the second word of the line. The line has already been claimed by the previous word write, which invalidates copies of the line in all other snooping CMMUs. CMMU2 has no need to write the word to memory since it has exclusive ownership of the line containing the word. The data cache is updated with the new data, and the line is changed to the exclusive modified state since it is modified with respect to memory. All subsequent loads and stores by CPU2 that map to this line will complete without accessing memory.

Figure 3-18 shows CPU1 attempting a load from location LA+4. The transaction misses in the cache, which forces CMMU1 to access memory. CMMU2 snoops the access, realizes it is caching modified (dirty) data requested by CMMU1, and suspends CMMU1's line read by forcing the RETRY status on the M bus.

Figure 3-19 shows CMMU2 copying back the exclusive modified line to memory and updating the state of the line to shared unmodified. Since CMMU2 had exclusive ownership of the line, no CMMU will snoop hit on any copyback of exclusive modified data. Exclusive ownership implies that it is the only copy of the line cached in the system.

Figure 3-20 shows CMMU1 regaining control of the M bus to complete the line read that was previously retried. The cache line is updated from memory, the required word is supplied to CPU1, and the line is marked shared unmodified.

3.9 CACHE DIAGNOSTIC PORTS

The cache diagnostic ports provide a way for the local processor or for M bus devices to access the components of a data cache set: the data, the physical address tags, and the cache status bits. These registers are used for performing cache diagnostics as described in **7.5 CACHE DIAGNOSTICS**.

3.9.1 Cache Data Ports (CDP3–CDP0)

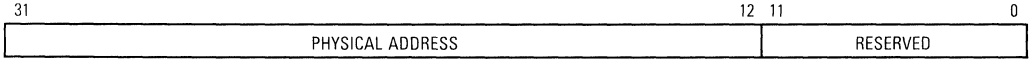
Address: Base+$800 (CDP0)
Base+$804 (CDP1)
Base+$808 (CDP2)
Base+$80C (CDP3)

These ports provide read/write access to each column in a cache set (e.g., word 2 of each line in the selected set). There are four cache data ports corresponding to each line in the set. The set and word numbers are specified by the value in the SAR: bits 11–4 specify the set number, and bits 3–2 specify the word number.

3.9.2 Cache Tag Ports (CTP3–CTP0)

Address: Base + $840 (CTP0)
Base + $844 (CTP1)
Base + $848 (CTP2)
Base + $84C (CTP3)

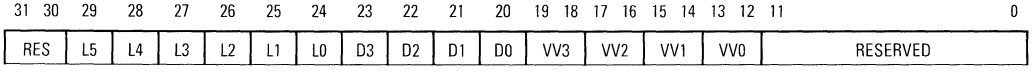
These ports provide read/write access to the four address tags of a cache set. The cache tag contains the 20-bit physical page address of the data stored in each line of the cache set (the other 12 bits of the register are reserved). There are four cache tag ports corresponding to each line in the set. The set is selected by the value written to the SAR, bits 11–4.



3.9.3 Cache Set Status Port (CSSP)

Address: Base + $880

This port provides information on the state of a cache set: the LRU algorithm information, the bits for disabling faulty lines, and the information on the state of each line. The set is specified by the value in the SAR, bits 11–4.



Bits 31–30, 11–0 — Reserved
These bits are returned as zeros on a register read.

L5 — Line 5
When set, line 3 more recently used than line 2.

L4 — Line 4
When set, line 3 more recently used than line 1.

L3 — Line 3
When set, line 3 more recently used than line 0.

L2 — Line 2
When set, line 2 more recently used than line 1.

L1 — Line 1
When set, line 2 more recently used than line 0.

L0 — Line 0
When set, line 1 more recently used than line 0.

D3–D0 — Line Disable
These bits are set by software to disable the line when diagnostics finds an error with the line. Once the line is disabled, subsequent reads of these bits return a one (set).
1 = Line 3–0 disabled due to fault.
0 = Line 3–0 operational.

VV3–VV0 — Line Status Bits
The MC88200 sets these bits according to the operations performed on the data cache.
00 — Exclusive Unmodified
01 — Exclusive Modified
10 — Shared Unmodified
11 — Invalid

3.10 CMMU REGISTER UPDATE SUMMARY (DATA CACHE)

Table 3-4 summarizes the MC88200 control register update operations as a result of M bus snooping, snoop copybacks, flush copybacks, and flush invalidations. The table indicates how the MC88200 is serving the P bus and M bus as master or slave, shows how the CMMU registers are updated, and illustrates the states of the signals that indicate exceptions.

Table 3-4. CMMU Register Update Summary (Data Cache)

Transaction	P Bus State	M Bus State	Exception	Update PFSR	Update PFAR	Update SSR	Update SAR	Assert M Bus ERR	Reply P Bus Fault
M Bus Snooping	N/A	Slave	M Bus Error	No	No	No	No	No	No
M Bus Snooping	N/A	Slave	Parity Error (Address Only)	No	No	No	No	Yes	No
	N/A	Slave	Success	No	No	No	No	No	No
Snoop Copyback	N/A	Master	M Bus Error	No	No	Bit 15 Set	No	No	No
	N/A	Master	Success	No	No	No	No	No	No
Flush Copyback	N/A	Master	M Bus Error	No	No	Bit 14 Set Bit 8–0 Destroyed	Physical Addess of Error	No	No
	N/A	Master	Success	No	No	Bit 14 Cleared Bits 8–0 Destroyed	Contents Destroyed	No	No
Flush Invalidate	N/A	Master	Success	No	No	Bit 14 Cleared Bits 8–0 Destroyed	Contents Destroyed	No	No

3.11 EFFECTS OF RESET ON DATA CACHE

The MC88200 data cache is not initialized by hardware upon reset. To properly configure the data cache, the software must clear the status and control information associated with each tag in the data cache. This is done by writing the set number (0–255) in bits 11–4 of the SAR, then writing 3F0FF000 to the cache set status port (CSSP). All CMMUs in the system must be initialized in this manner before caching is enabled. Refer to **3.9.3 Cache Set Status Port** for more information on configuring the status of cache sets. Table 3-5 list the data cache state upon reset.

Table 3-5. Data Cache State on Reset

Register	Data Cache Function	State After Reset
Cache Tag Ports	Allows Read/Write of Cache Tag in Set SAR[11–4]	Undefined
Cache Data Ports	Allows Read/Write Cache Data in Set SAR[11-4], Word SAR[3–2]	Undefined
Cache Set Status Port	Allows Read/Write of Cache Set Status in Set SAR[11–4]	Undefined

3

SECTION 4
SIGNAL DESCRIPTION

MC88200 signals are grouped into the five functional categories: processor bus (P bus), memory bus (M bus), interrupt and control, power and clock, and miscellaneous signals. The following paragraphs are organized according to these categories. Figure 4-1 shows the MC88200 signals.

NOTE

The terms assert and negate are used exclusively in this manual to avoid confusion between active high and active low signals. **Assert** or **assertion** indicates that a signal is active or true, regardless of whether the signal is active high or active low. **Negate** or **negation** indicates that the signal is inactive or false.

4.1 P BUS SIGNALS

The P bus signals provide the interface between the MC88200 and the processor. The MC88200 always acts as a P bus slave device, receiving commands from the processor and driving signals only in response to processor transactions. The P bus interface is supported by the MC88100 microprocessor. The following paragraphs describe the signals that interface with the P bus.

4.1.1 P Bus Data (D31–D0)

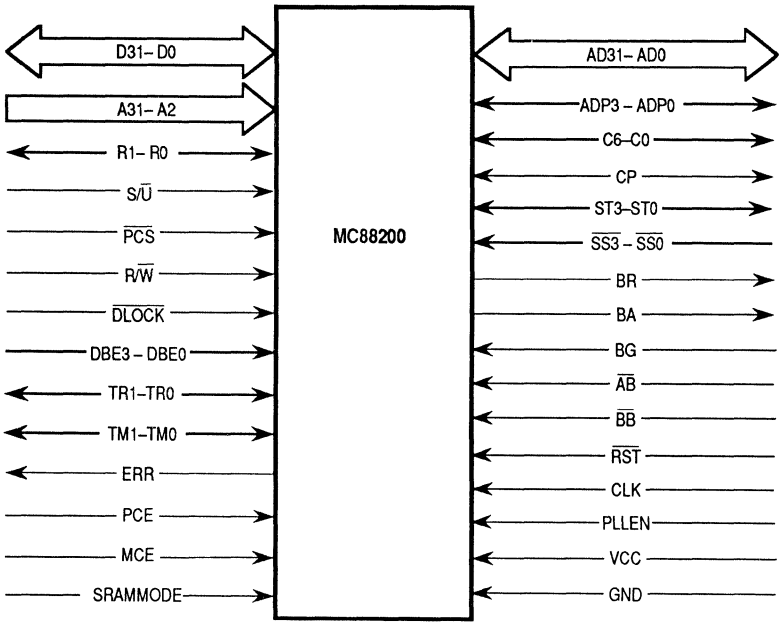
Signals D31–D0 provide the data or instruction transfer paths between the processor and the MC88200.

4.1.2 P Bus Address (A31–A2)

Signals A31–A2 provide the 30-bit word address for memory accesses. Since MC88100 instructions are word aligned, the two least significant bits of the address are not needed for instruction fetches. For data accesses, the data byte enable signals (see **4.1.8 P Bus Data Byte Enable (DBE3–DBE0)**) specify the required byte or half-word.

4.1.3 P Bus Reply (R1–R0)

The P bus reply signals provide the status of the P bus transaction to the processor; Table 4-1 lists the setting of the P bus reply signals. R0 and R1 should be pulled up externally through 10KΩ resistors, defaulting the reply to the fault condition.



Function	Mnemonic	Type	Active	Count	Reset State
P Bus					
Address	A31–A2	Input	High	30	Ignored
Supervisor/User	S/U	Input	High	1	Ignored
Chip Select	PCS	Input	Low	1	Ignored
Data Byte Enable	DBE3–DBE0	Input	High	4	Ignored
Data	D31–D0	I/O	High	32	High Impedance
Read/Write	R/W	Input	High	1	Ignored
Lock	DLOCK	Input	Low	1	Ignored
Reply	R1–R0	I/O	High	2	High Impedance
M Bus					
Bus Request	BR	Output	High	1	Negated
Bus Grant	BG	Input	High	1	Ignored
Bus Acknowledge	BA	Output	High	1	Negated
Arbitration Busy	AB	Input	Low	1	Ignored
Bus Busy	BB	Input	Low	1	Ignored
Address/Data	AD31–AD0	I/O	High	32	High Impedance
Address/Data Parity	ADP3–ADP0	I/O	High	4	High Impedance
Control	C6–C0	I/O	High	7	High Impedance
Control Parity	CP	I/O	High	1	High Impedance
Local Status	ST3–ST0	I/O	High	4	Active Input
System Status	SS3–SS0	Input	Low	4	Ignored
Reset	RST	Input	Low	1	Asserted
Clock	CLK	Input	High	1	Active
Phase Lock Enable	PLLEN	Input	Low	1	Active
Tag Monitor	TM1–TM0	I/O	High	2	Active Input
Trace	TR1–TR0	I/O	High	2	Active Input
Cache Static RAM Mode	SRAMMODE	Input	High	1	Active Input
P Bus Checker Enable	PCE	Input	High	1	Active Input
M Bus Checker Enable	MCE	Input	High	1	Active Input
Error	ERR	Output	High	1	Low
Power	V_{CC}			18	
Ground	GND			18	

Figure 4-1. MC88200 Signal Functional Diagram

Table 4-1. P Bus Reply Signals

R0	R1	Transaction
0	0	Reserved
0	1	Success (Transaction Completed)
1	0	Wait (Transaction Delayed)
1	1	Fault (Transaction Aborted)

4.1.4 P Bus Supervisor/User Select (S/$\overline{U}$)

The S/$\overline{U}$ signal selects the user or supervisor memory space. An asserted signal specifies the supervisor memory space; a negated signal specifies the user memory space. The MC88200 uses this signal to determine which memory area the P bus (logical) address pertains to or to select the supervisor or user area pointer at the beginning of a translation table search.

4.1.5 P Bus Chip Select ($\overline{PCS}$)

This signal selects the MC88200 as a P bus slave. When multiple MC88200s reside on the P bus, external circuitry is needed to decode the chip select signal from the P bus address lines. Only one MC88200 should be selected at a time, except for MC88200s operating in checker mode. Checker MC88200s should be selected when their associated master is selected. Refer to *MC88100 User's Manual* for a description of the checker mode.

4.1.6 P Bus Read/Write (R/$\overline{W}$)

The read/write signal indicates the type of memory transaction. When asserted, R/$\overline{W}$ indicates a memory read; when negated, R/$\overline{W}$ indicates a memory write. If the MC88200 is used as an instruction memory controller, this signal can be tied high since all memory accesses are reads (instruction fetches).

4.1.7 Bus Lock ($\overline{DLOCK}$)

The $\overline{DLOCK}$ signal is the transaction lock signal. When $\overline{DLOCK}$ is asserted, the MC88200 retains M bus ownership and bypasses the data cache. In particular, this signal is used by the MC88100 microprocessor during an exchange memory (**xmem**) instruction. When $\overline{DLOCK}$ is asserted, the MC88200 maintains its M bus tenure for both the write and read accesses of the **xmem** instruction, which prevents other M bus masters from changing the memory location between accesses. When $\overline{DLOCK}$ is asserted, the data cache is bypassed and data is written to/read from memory. A cache lookup is performed; and if the data is found in the cache, it is invalidated. Whenever the $\overline{DLOCK}$ signal is asserted, the M bus lock signal (C3) during the address phase is asserted.

4.1.8 P Bus Data Byte Enable (DBE3–DBE0)

The data byte enable signals indicate which bytes of an addressed location are used during the memory access. If one signal is active, a single byte is used; if two signals are active, a half-word is used. When all four signals are active, a word transaction occurs. Unaccessed bytes are not affected by the memory transaction. Any asserted byte enable signal indicates a valid transaction. When the byte enable signals are all negated (low), the P bus transaction is a no-operation.

The byte enable signals are not individually recognized during control register accesses; all accesses to control registers must be word accesses.

If the MC88200 is used as an MC88100 instruction memory controller, the byte enable inputs can be tied to the code fetch (CFETCH) output of the MC88100, since all instruction accesses are word accesses. The R/$\overline{W}$ input to the MC88200 can be tied high.

4.1.9 P Bus Checker Enable (PCE)

This signal is used in systems redundantly incorporating two or more MC88200s. The two MC88200s are wire-ORed together on the P bus. The 'master' MC88200 (PCE negated) operates normally. The checker MC88200 (PCE asserted) places all of its P bus signals in the high-impedance state; all outputs are monitored as inputs. The checker compares its internal results with the results read from the pins. If a mismatch occurs between the master and checker, the checker asserts ERR. External logic must then be used to determine the appropriate action for the system.

The master MC88200 also checks that the internal value of the signals it is driving are the same as the external values. The MC88200 makes this comparison by testing the signals before and after the internal line drivers. If there is an error, the master asserts ERR. This feature is useful for determining P bus shorts, because the internal and external values of a signal will be different when a bus signal is shorted.

PCE is sampled during reset but must remain in the required state throughout its operation.

4.1.10 M Bus Checker Enable (MCE)

The MCE signal is used for master/checker applications on the M bus: two (or more) MC88200s are wire-ORed together on the M bus, one acting as the master and the other acting as the checker (identical to the PCE signal described previously). When the checker finds a mismatch, it asserts the ERR signal.

4.2 M BUS SIGNALS

The M bus signals provide the interface between the MC88200 and the memory system. The M bus is a shared, multiplexed bus that accesses memory and memory-mapped

peripheral devices. The M bus includes a bus arbitration scheme that grants bus ownership to a single M bus device. Bus arbitration is discussed in **7.2.2 M Bus Arbitration**. The following paragraphs describe the signals that interface with the M bus.

4.2.1 M Bus Address/Data Bus (AD31–AD0)

These signals are the multiplexed address/data lines. The function of these signals depends on the M bus transaction phase defined by the M bus control signals (see **4.2.3 M Bus Control (C6–C0)**). During the address phase (AP), the MC88200 drives the 30 most significant bits of the physical word address onto AD31–AD2 (AD1 and AD0 are zero). During the data phase, AD31–AD0 are the data input/output lines. When the MC88200 is not the M bus master, these lines are placed in the high-impedance state.

4.2.2 M Bus Address/Data Parity (ADP3–ADP0)

These signals indicate the parity of the M bus information lines. The MC88200 always uses even parity, checking parity on reads and generating parity for addresses and memory writes. Each parity signal is associated with eight information signals (see Table 4-2).

Table 4-2. M Bus Address/Data Parity Signals

Signal	Lines
ADP3	AD31–AD24
ADP2	AD23–AD16
ADP1	AD15–AD8
ADP0	AD7–AD0

The MC88200 always generates parity for addresses and for memory writes. For the MC88200 to check parity on a memory read, the parity enable (PE) bit (bit 15) must be set in the system control register (SCTR). When the MC88200 is not the M bus master, these line are placed in the high-impedance state.

4.2.3 M Bus Control (C6–C0)

When the MC88200 is the M bus master, these signals define the transaction phase, type of transaction, and provide control information. The C0 signal defines whether the phase is address (address on lines AD31–AD2) or data (data on lines AD31–AD0). Table 4-3 lists other signals that depend on the address and data phase.

These signals are inputs during bus snooping and when the MC88200 is accessed as a slave device (register access). That is, the MC88200 determines the phase by reading C0, determines if the address is global or local by reading C5, etc.

4.2.4 M Bus Control Parity (CP)

When the MC88200 is the M bus master, this signal indicates the even parity of the M bus control signals. This signal is placed in the high-impedance state when the MC88200 is not the M bus master.

Table 4-3. M Bus Control Signals

Address Phase		
C0	AP	Address Phase (Asserted).
C1	IM	Intent to Modify. Asserted if the transaction is a memory write or a read with intent to modify. This signal is used for data cache coherency control.
C2	RD	Asserted for memory read; negated for memory write.
C3	LK	Lock. Asserted to lock out all other M bus devices (e.g., during the MC88100 **xmem** instruction).
C4	CI	Cache Inhibit. Asserted if the MC88200 cannot cache the data transferred during the current memory access. This signal reflects the value of the translation descriptor CI bit for the accessed memory address.
C5	G	Global. Asserted if the addressed memory location may be shared by more than one MC88200. This signal reflects the value of the translation descriptor G bit for the accessed memory address.
C6	—	Reserved. Driven as zero.
Data Phase		
C0	DP	Data Phase (Negated).
C1	LDT	Last Data Transfer. Asserted if the current data phase completes the transfer; negated if additional data phases are required.
C2	RD	Asserted for memory read; negated for memory write.
C3	MBE3	M Bus Byte Enable 3. Asserted if AD31–AD24 contain significant data.
C4	MBE2	M Bus Byte Enable 2. Asserted if AD23–AD16 contain significant data.
C5	MBE1	M Bus Byte Enable 1. Asserted if AD15–AD8 contain significant data.
C6	MBE0	M Bus Byte Enable 0. Asserted if AD7–AD0 contain significant data.

4.2.5 M Bus Local Status (ST3–ST0)

These signals indicate the local M bus status when the MC88200 is being accessed as a slave device (register access) or when the MC88200 is snooping a global M bus transaction. In a typical system configuration, these signals are buffered and inverted to generate the system status lines. Table 4-4 lists the values of these signals. There are no illegal combinations.

These signals, which are inputs during reset, are used to initialize the CMMU ID register (described in **5.7.3 ID INITIALIZATION**).

Table 4-4. M Bus Local Status Signals

Status	ST3	ST2	ST1	ST0
Error	1	x	x	x
Retry	0	1	x	x
Wait	0	0	1	x
End of Data (Data Phase) OK (Phase Completed)	0 0	0 0	0 0	1 0

4.2.6 M Bus System Status ($\overline{SS3}$–$\overline{SS0}$)

These signals are the reply generated by M bus slaves in response to the MC88200 request and data phases. These signals are typically the wired-NOR of the local status signals of all M bus devices. Non-MC88200 M bus slaves may drive these signals directly (i.e., they do not require local and system status lines as used in the MC88200). There are no illegal combinations. Table 4-5 lists the values of these signals.

Table 4-5. M Bus System Status Signals

Status	$\overline{SS3}$	$\overline{SS2}$	$\overline{SS1}$	$\overline{SS0}$
Error	0	x	x	x
Retry	1	0	x	x
Wait	1	1	0	x
End of Data (Data Phase) OK (Phase Completed)	1 1	1 1	1 1	0 1

4.2.7 M Bus Request (BR)

The bus request signal is asserted by the MC88200 to request M bus ownership. For BR to be asserted, the MC88200 cannot be the bus owner, and it cannot be blocked from asserting BR (fairness arbitration scheme). A complete description of M bus arbitration is given in **SECTION 5 BUS OPERATIONS**.

4.2.8 M Bus Grant (BG)

The M bus arbitration logic generates this signal in response to a bus request. The MC88200 recognizes this signal only if the M bus is not busy (see **4.2.10 M Bus Busy ($\overline{BB}$)**). Only one device at a time can receive a bus grant (i.e., only one device can be granted bus ownership).

4.2.9 M Bus Acknowledge (BA)

Bus acknowledge is asserted by the MC88200 when it has received a bus grant in response to a bus request. This signal allows the MC88200 to accept and maintain bus ownership while it accesses memory. The MC88200 keeps the BA signal asserted during the full memory transaction (bus tenure).

4.2.10 M Bus Busy ($\overline{BB}$)

This signal indicates that some M bus device is currently the bus master; it is an input to the MC88200 that qualifies the bus grant signal. For an MC88200 to become the bus master, it must receive a bus grant signal from the arbitration logic, and $\overline{BB}$ must be negated (no

other device is the M bus master). This signal is typically generated from the wired-NOR of the bus acknowledge of all M bus master devices.

4.2.11 M Bus Arbitration Busy ($\overline{AB}$)

This signal indicates that one or more M bus devices are performing a bus request. It is an input to the MC88200 that indicates bus contention is occurring; it may also prevent the MC88200 from issuing a bus request (fairness arbitration scheme). Arbitration busy is typically generated from the wired-NOR of the bus request of all M bus master devices.

4.3 POWER AND CLOCK SIGNALS

The following paragraphs describe the power and clock signals used on the MC88200.

4.3.1 Clock (CLK)

The clock input signal generates the internal timing signals for the MC88200. The MC88200 internal clock is derived from the CLK signal, normally phase locked to minimize the skew between the external and internal signals. Since the CLK signal will be applied to the processor (MC88100) and to other peripherals, exact timing of internal signals is required to properly synchronize the devices to the buses. Phase locking also eliminates the need for asynchronous data strobes found in conventional asynchronous bus protocols. Refer to **SECTION 5 BUS OPERATIONS** for specific details.

4.3.2 Power Signals (V_{CC})

These signals provide +5 V to the processor components. Eighteen power signals are provided to the MC88200 components; these signals are implemented as two separate power buses. The external line drivers are supplied by one bus; all other circuits are supplied by the second bus. This scheme provides more stable power to the internal circuits of the MC88200.

4.3.3 Ground (GND)

These are the ground pins for the power signals. Eighteen ground pins are separated into two separate power buses matching the V_{CC} signals.

4.4 MISCELLANEOUS SIGNALS

The following paragraphs describe the miscellaneous signals used on the MC88200.

4.4.1 Trace (TR1–TR0)

The trace signals perform several functions during MC88200 operation. During normal operation, these signals are generated by the MC88200 to indicate which line of a cache set is being accessed. These signals operate in conjunction with the tag monitor signals (TM1–TM0) to indicate when changes are made to the data cache address tags. TR1–TR0 indicate the line number in the affected set.

4.4.2 Tag Monitor (TM1–TM0)

These signals provide information about the address tags associated with each line in the cache. TM1–TM0 allow external logic to maintain duplicates of the data cache address tags by indicating when changes are made to the tags. These signals are used in conjunction with other M bus signals discussed in the preceding paragraph. Refer to Table 4-6 for TM signals functions.

During cache set test operations, TR1–TR0 signals are MC88200 inputs that select the data line to be tested in a cache set. Cache diagnostics are described in **7.5 CACHE DIAGNOSTICS**. During reset, the TR0, TR1, and TM0 signals are MC88200 inputs used to initialize the CMMU ID register, described in **5.7.3 ID Initialization**.

Table 4-6. Tag Monitor Signal Functions

BA	AP	RD	TR1 TR0	TM1	TM0	Operation
1	1	1	<line number>*	0	1	M bus read, tag is loaded.
1	1	1	<line number>	1	1	M bus read, no effect on the tag (table search) or tag invalidation (cache inhibit).
1	1	0	<line number>	0	1	M bus write, tag is loaded or remains loaded.
1	1	0	<line number>	0	0	M bus write, invalidate tag (CI = 1) or control register access.
1	1	0	<line number>	1	1	M bus write, no effect on the tag (cache inhibit, table search, or control register access).
0	X	X	AH DH	1	0	P bus transaction result: AH = 1 for ATC hit, DH = 1 for data cache hit.

*<line number> encoded as:

TR1	TR0	
0	0	Line 0
0	1	Line 1
1	0	Line 2
1	1	Line 3

4.4.3 Cache Static RAM Mode (SRAMMODE)

The SRAMMODE signal places the MC88200 into diagnostic mode for checking the data cache or simply using the data cache as a 16K-byte scratchpad memory. The data cache

is accessed with no wait states through P bus signals, A11–A2, TR1–TR0, A3–A2, and DBE3–DBE0. A particular location in the cache is addressed as follows:

A11–A4	set number
TR1–TR0	line number
A3–A2	word number
DBE3–DBE0	byte enable

While in the test mode, all M bus transactions are ignored, including global transactions and accesses to the MC88200 internal registers. No memory management is performed while in cache static RAM mode and the CMMU may not be placed in checker mode. Entry into and exit from cache static RAM mode must be performed during RESET sequence. Refer to **SECTION 5 BUS OPERATIONS** for functional timing.

4.4.4 Checker Mismatch Error (ERR)

This signal is generated when a bus comparator error occurs. This signal is asserted by internal bus checking circuitry or by the checker MC88200 in master/checker configurations.

4.4.5 Reset ($\overline{\text{RST}}$)

The reset signal is used to perform an orderly restart of the MC88200, bringing it to a known state. When $\overline{\text{RST}}$ is asserted, the MC88200 halts the current transaction and functionally removes itself from the M bus and P bus. The CMMU ID register is initialized, and the M bus and P bus checker enable signals are sampled (MCE and PCE, respectively). Registers are placed in a predefined state, and all address translation stops. The reset operation is described fully in **SECTION 5 BUS OPERATIONS**.

4.4.6 Phase Lock Enable (PLLEN)

The PLLEN signal controls the internal phase lock circuit that synchronizes the internal clocks to the CLK signal. Refer to **SECTION 5 BUS OPERATIONS** for specific details.

SECTION 5
BUS OPERATION

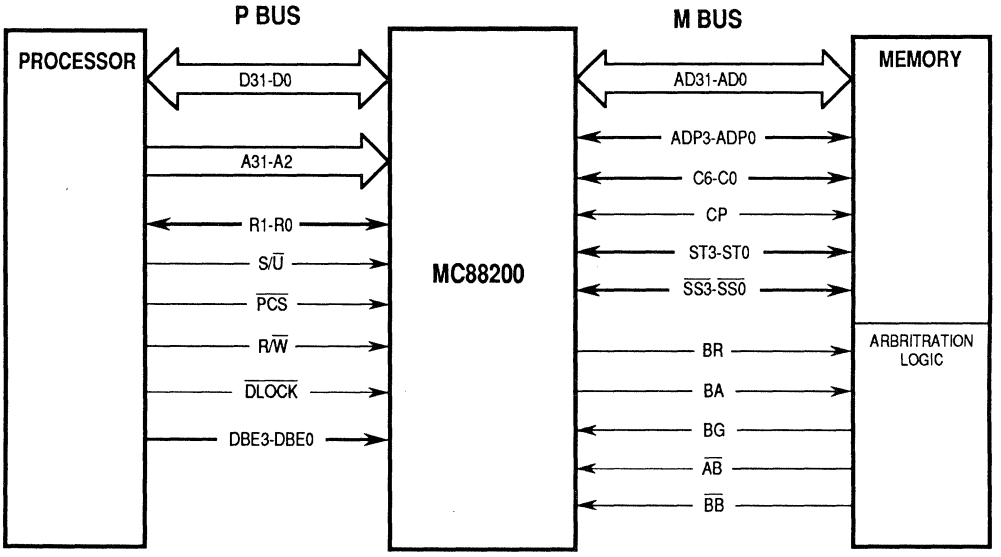
This section provides a functional description of the bus interfaces associated with the MC88200 cache/memory management unit (CMMU). Data transfer operations to and from the MC88100 processor on the processor bus (P bus) and data transfers to and from memory on the memory bus (M bus) are described. Bus error conditions, bus arbitration, and reset operation are also described. Output signals are only driven when the CMMU is operating in master mode. In checker mode, all signals are inputs except ERR, BR, and BA. The bus signals for the MC88200 are shown in Figure 5-1.

The MC88200 services both P bus and M bus transactions. The CMMU always acts as a P bus slave, receiving logical addresses from the processor, and either returning data on the P bus during reads or latching data from the P bus during writes. The CMMU acts as a M bus master when performing a line fill to update the data cache, a line copyback to update memory, or a table search operation. Alternatively, the CMMU acts as a M bus slave when being accessed by another CMMU or when snooping M bus transactions. Cache lines can be transferred to and from memory on the M bus using an efficient burst mode providing four words/five clock cycles plus an additional clock cycle for M bus arbitration.

5.1 BUS CHARACTERISTICS

The P bus and M bus are fully synchronous buses. The bus transfers between the MC88200 and attached MC88100 (P bus) or memory and peripherals (M bus) are clocked by the CLK signal. Byte, half-word, and full 32-bit transfers are supported on both buses. All operands are aligned on word (modulo 4) address boundaries, precluding the need for the two lowest order address lines and misaligned accesses. On a data read of a byte or half-word operand, the memory system supplies either the required bytes or a full 32-bit word to the MC88200 on the M bus; the processor automatically selects the required byte(s) from the MC88200 P bus. All memory transactions access either the user or supervisor address space.

The MC88200 input and output signals are synchronous in that all setup and hold times are specified in reference to the clock signal. MC88200 outputs are driven from a clock edge, and a maximum delay is specified. In addition, minimum hold times are specified in relation to the clock (see Figure 5-2). The requirements for MC88200 P bus inputs are shown in Figure 5-3. The minimum setup and hold times must be met to guarantee proper device operation.



5

Function	Mnemonic	Type	Active	Count	Reset State
P BUS					
Address	A31–A2	Input	High	30	Ignored
Supervisor/User	S/$\overline{U}$	Input	High	1	Ignored
Chip Select	$\overline{PCS}$	Input	Low	1	Ignored
Data Byte Enable	DBE3–DBE0	Input	High	4	Ignored
Data	D31–D0	I/O	High	32	High Impedance
Read/Write	R/$\overline{W}$	Input	High	1	Ignored
Lock	$\overline{DLOCK}$	Input	Low	1	Ignored
Reply	R1–R0	I/O	High	2	High Impedance
00 — Reserved					
10 — Successful Transaction					
01 — Wait					
11 — Transaction Fault					
M BUS					
Bus Request	BR	Output	High	1	Negated
Bus Grant	BG	Input	High	1	Ignored
Bus Acknowledge	BA	Output	High	1	Negated
Arbitration Busy	$\overline{AB}$	Input	Low	1	Ignored
Bus Busy	$\overline{BB}$	Input	Low	1	Ignored
Address/Data	AD31–AD0	I/O	High	32	High Impedance
Address/Data Parity	ADP3–ADP0	I/O	High	4	High Impedance
Control	C6–C0	I/O	High	7	High Impedance
Control Parity	CP	I/O	High	1	High Impedance
Local Status	$\overline{ST3–ST0}$	I/O	High	4	Active Input
System Status	$\overline{SS3–SS0}$	Input	Low	4	Ignored

Figure 5-1. MC88200 Bus Signals

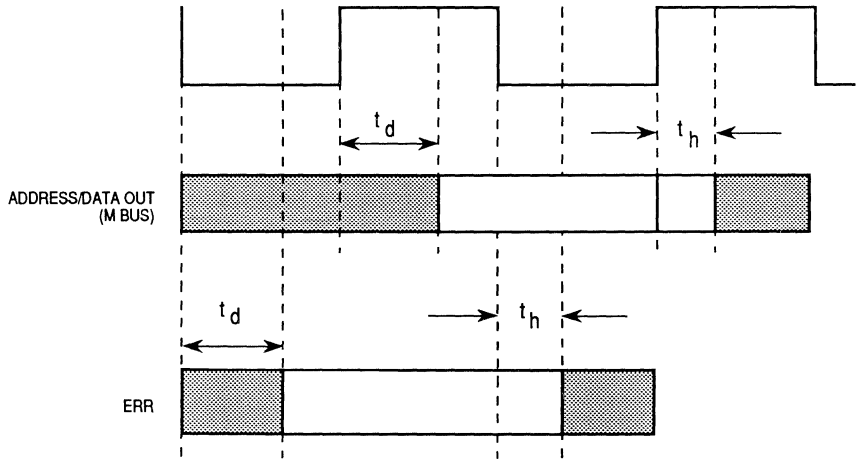


Figure 5-2. Output Signal Relationship to Clock (Example)

5

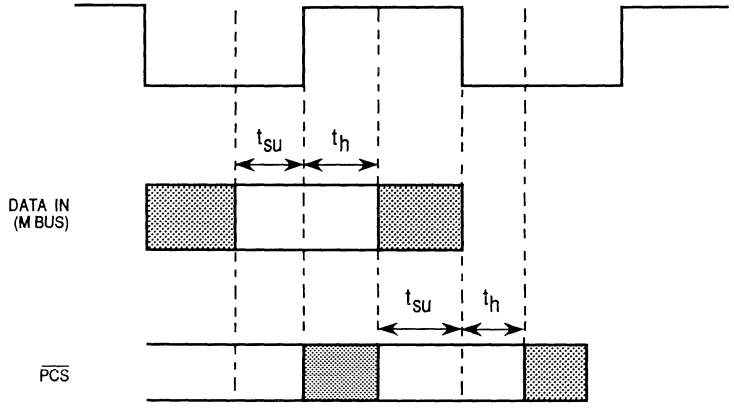


Figure 5-3. Input Signal Requirements (Example)

5.2 P BUS PROTOCOL

The MC88100 is the bus master of the P bus and always drives the address bus, the data byte enable (DBE3–DBE0) signals, and the data bus (except for P bus reads). The MC88200 latches the addresses at the beginning of P bus transactions and drives the P bus data signals (D31–D0) during read accesses, as well as the P bus reply signals. MC88100 P bus transactions are pipelined, which allows the processor to begin a transaction before receiving the result or status of the previous transaction. Since the processor can initiate a new transaction on each clock cycle, the MC88200 monitors the P bus signals on each clock cycle and responds to valid transactions accordingly.

Data transfers can be 8-, 16-, or 32-bit read or write accesses, and transactions can be locked for the exchange memory (**xmem**) instruction. The DBE3–DBE0 signals indicate which byte(s) on the data bus (D31–D0) are valid during a P bus transaction. For a P bus read operation, the upper 30 data address lines select a word in memory, and the asserted data byte enable signals indicate which byte(s) within that word are required by the processor. The MC88200 responds to the read by placing the requested byte(s) on the P bus, corresponding to the asserted data byte enable signals; the processor automatically extracts the desired byte(s) from the bus. Figure 5-4 shows an example of a byte read operation and the relationship of the data byte enable signals to the four bytes on the P bus.

5

During a P bus write operation, the data byte enable signals indicate which byte(s) in memory should be updated. On a byte write, the MC88100 places the data byte in all four byte positions on the data bus. The data byte enable signals indicate the byte in memory where the data is to be stored. Figure 5-5 shows the operation of a byte write transaction.

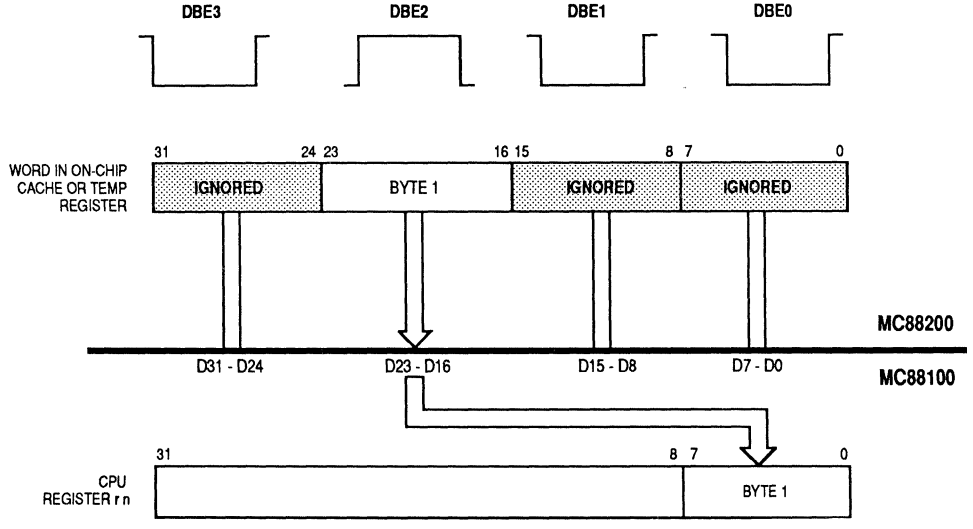


Figure 5-4. Byte Enable Signal Control of Memory Read

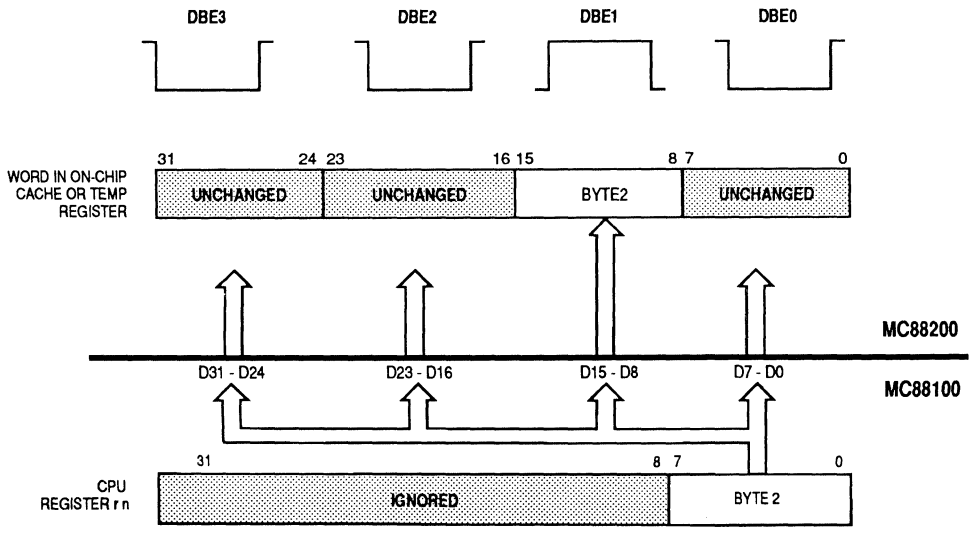


Figure 5-5. Byte Enable Signal Control of Memory Write

Instructions and data bus transactions are not necessarily required by the processor on every clock cycle. The processor inserts null transactions during those clock cycles in which a transfer is not required. The data (data CMMU only) and address lines are driven during a null transaction, but the DBE3–DBE0 signals are negated. The processor ignores the reply signals for all null transactions.

5.2.1 P Bus Read Transaction

During a P bus read transaction, the MC88200 supplies a byte, half-word, or word to the MC88100 processor. The DBE3–DBE0 signals indicate the size and byte offset of the access. Figure 5-6 shows a flowchart of an instruction read transaction. Figure 5-7 shows the relative timing of the signals that perform the read. A chip-select signal ($\overline{PCS}$) must be generated by external logic to select the appropriate MC88200.

P bus transactions have an address phase (AP) and a reply phase associated with each access. The address phase of an access is defined by two states: address low (AL), corresponding to the low time of the CLK, and address high (AH) corresponding to the high time of the CLK. Similarly, the reply phase of the transaction is defined by two states: reply low (RL) and reply high (RH), corresponding to the low time of the CLK and high time of the CLK.

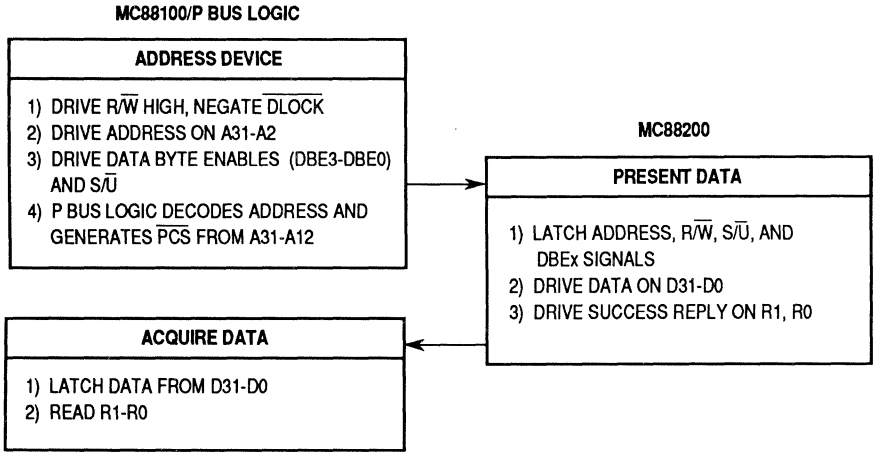


Figure 5-6. Data Read Flowchart

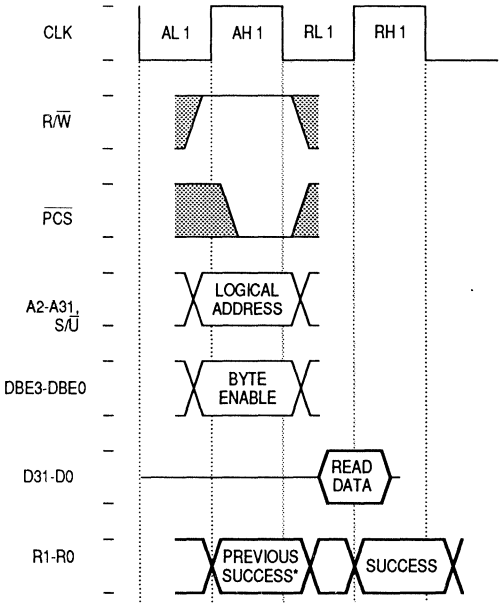


Figure 5-7. Data Read Timing

The address (A31–A2), DBE3–DBE0, and supervisor/user select (S/$\overline{U}$) signals are driven by the falling edge of the processor clock at the beginning of state AL and are set up to the rising edge at the beginning of AH. The appropriate MC88200 responds to the access by placing the data on D31–D0 and by driving the reply signals (R1–R0) with the appropriate setup and hold times during the reply phase. Figure 5-1 shows the various encodings of the reply signals.

The reply signals indicate whether or not the bus transaction is successful. If the data on D31–D0 is guaranteed to have met the appropriate setup and hold times with respect to the rising edge of the clock of state RH, then the reply during RH has the 'success' encoding. If the CMMU is unable to supply the data on the P bus with the required setup time to the rising edge of state RH, then the reply indicates a 'wait' response. This response causes the processor to ignore D31–D0 and to continue driving the next address until a successful or fault encoding is indicated with the reply signals. The CMMU inserts as many wait cycles as necessary until the data is supplied. Finally, if the MC88200 can not supply the required data (e.g., due to a page fault M bus error) then the reply signals indicate 'fault.' This response causes either the MC88100 instruction pointer corresponding to the faulted pre-fetch to be marked as invalid (in the case of instruction CMMU) or the MC88100 to initiate exception processing for a data exception (in the case of a data CMMU). If the MC88100 attempts to execute that instruction (i.e., it is not discarded due to a change of program flow), an instruction access exception is generated. The MC88200 conforms to the P bus operations initiated by the processor by not latching a new address until a successful or fault reply is driven by any CMMU for the previous access. The transaction immediately following a fault reply is ignored by the CMMU. An encoding of '00' on the reply signals is reserved and may cause unpredictable behavior in the current implementation.

5.2.2 P Bus Write Transaction

During a P bus write transaction, the MC88100 transfers a byte, half-word, or word to the MC88200. The DBE3–DBE0 signals indicate the size and byte offset of the access.

Figure 5-8 shows a flowchart of a P bus write transaction. The $\overline{PCS}$ signal is generated by external logic to select the appropriate MC88200. Figure 5-9 shows the relative timing of the signals that perform the data write.

During a memory write transaction, the MC88100 drives A31–A2, DBE3–DBE0, and S/$\overline{U}$ appropriately, and the R/$\overline{W}$ signal is driven low. All these signals are set up to the rising edge of AH and remain valid through the specified hold time from the falling edge of AH. If the reply for the previous data transaction was 'success,' the processor then drives the data on D31–D0 so that the data is set up to the falling edge of RL and remains valid through the rising edge of state RL. However, if the reply for the previous transaction was 'wait', the processor does not drive the data signals of the data P bus. Instead, the address phase of the write transaction is repeated until a 'success' is driven for the previous reply.

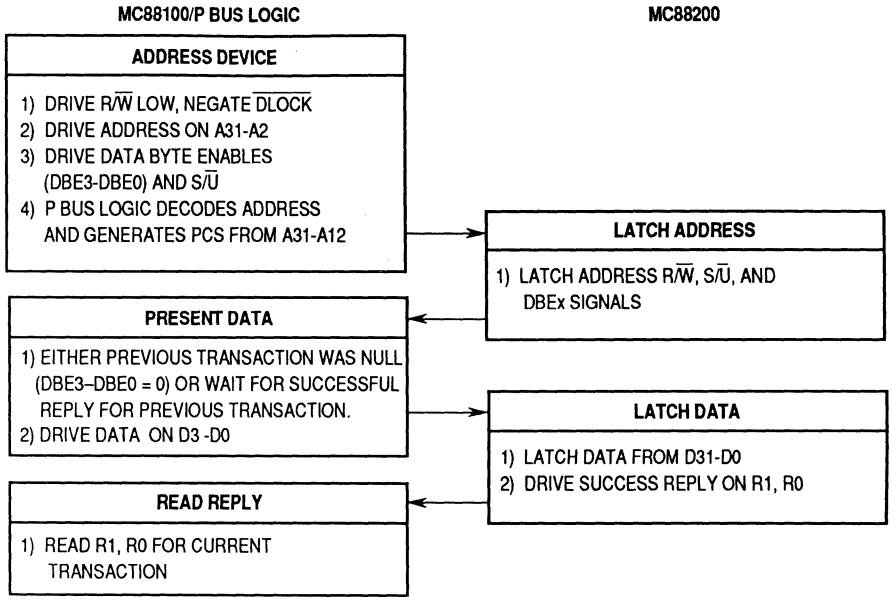


Figure 5-8. P Bus Write Flowchart

If possible, the MC88200 indicates with R1–R0 that the current access was successful and latches the data appropriately. Otherwise, the CMMU signals a 'wait' response on R1–R0, causing the P bus to repeat the reply phase and drive D31–D0 again with the appropriate setup and hold times with respect to the clock. The MC88200 inserts as many wait cycles as necessary until the data can be latched.

If the MC88200 cannot complete the access, the R1–R0 signals drive a 'fault' encoding, causing the MC88100 to initiate exception processing for a data access exception.

5.2.3 Pipelined P Bus Access

The P bus is pipelined so that the address phase of a memory access can overlap with the reply phase for the previous transaction; the processor may begin a new transaction (and drive a new value on A31–A2), even if the initial transaction is extended due to wait replies. Therefore, the MC88200 latches the value on A31–A2 when it is first driven, in case more than one clock of access time is required.

Figure 5-10 shows the relative timing of a P bus read transaction having one wait cycle followed by a P bus write transaction. Although the address phase of the write immediately

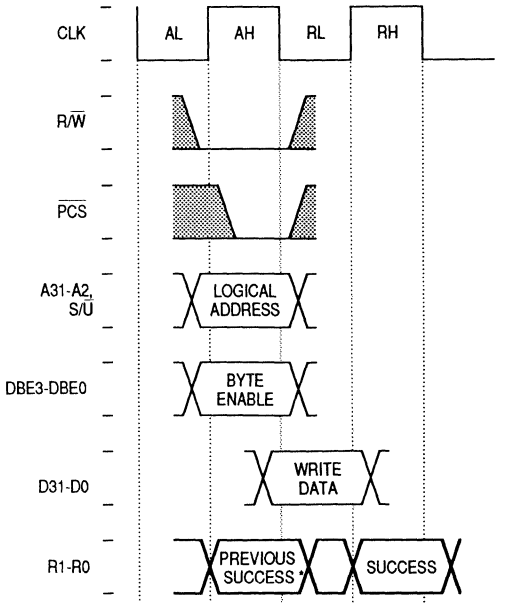


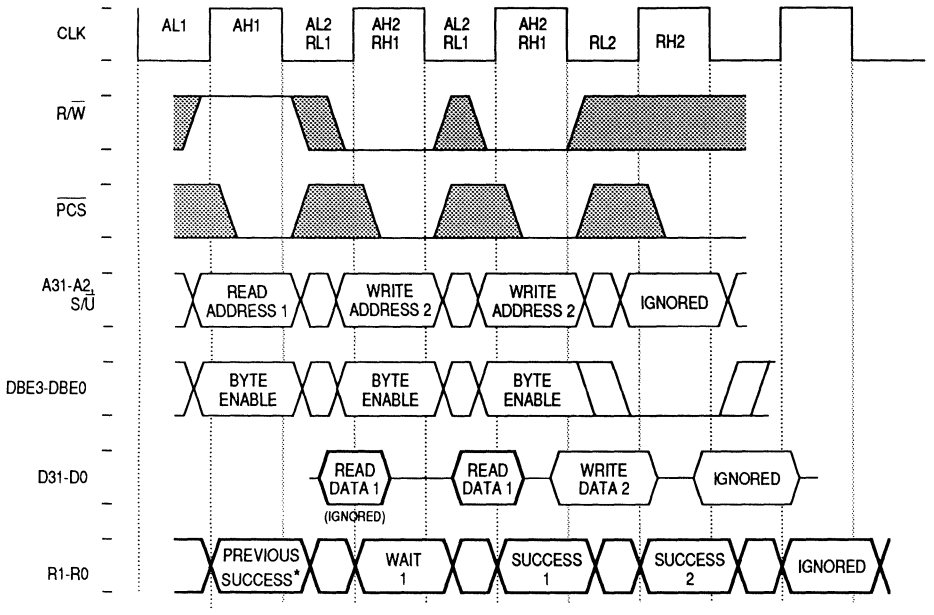
Figure 5-9. P Bus Write Timing

follows the address phase of the read, the data signals are not driven for the write transaction until a successful reply is signaled for the previous access, thus avoiding data bus contention. Figure 5-10 also illustrates a null transaction following the write access.

Figures 5-11 and 5-12 illustrate two other examples of pipelined operations on the data P bus. Figure 5-11 illustrates two successive write transactions (due to two successive store instructions or a store double instruction) with one wait cycle inserted for the first access. Figure 5-12 illustrates the relative timing for a write transaction with one wait cycle followed by one read transaction. Since the timing for the R/$\overline{W}$ signal is only valid during the address phase of a transaction, it is also latched by the MC88200.

5.2.4 Locked P Bus Transactions

Execution of the **xmem** instruction by the MC88100 causes the contents of a processor register to be exchanged with a memory location. To perform this operation, the processor performs a read transaction from the specified address immediately followed by a write



* OR NULL P BUS TRANSACTION (DBE3–0 = 0) ON PREVIOUS CYCLE

Figure 5-10. Pipelined P Bus Accesses (Read, Wait, Write, Null)

transaction to that address. The data bus lock ($\overline{\text{DLOCK}}$) signal is asserted from the beginning of the address phase of the read transaction (with the same timing as A31–A2) and remains asserted through the address phase(s) of the write transaction, regardless of the number of wait cycles in the read. However, $\overline{\text{DLOCK}}$ is not asserted during the reply phase(s) of the write transaction.

The only way to interrupt a locked operation on the P bus is to signal a fault for the read transaction via R1–R0. Either the read or the write transaction may be terminated with a fault.

When the read transaction of an **xmem** instruction is terminated with a fault, the entire **xmem** instruction should be emulated in the processor exception handler. However, if the write transaction of an **xmem** instruction is terminated with a fault, the software handler may not be able to guarantee that the rest of the program is synchronized with the **xmem**. Refer to *MC88100 User's Manual* for more information on the recovery of faults.

The $\overline{\text{DLOCK}}$ signal is used by the M bus as an indication that the read and write operations should not be interrupted by an alternate bus master M bus. The **xmem** instruction can be used by the processor for semaphore manipulation since the MC88200 guarantees indivisible operation between the read and write transactions on the M bus.

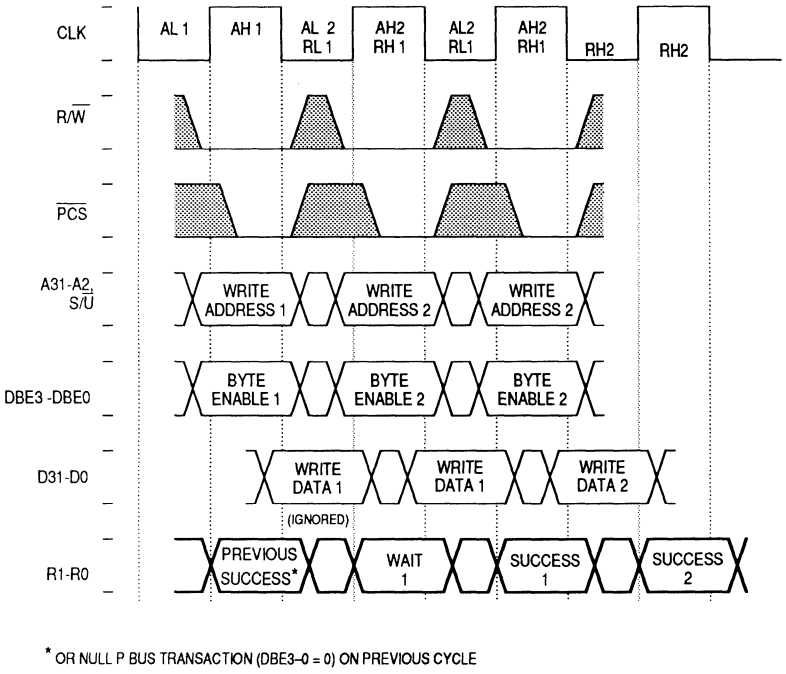


Figure 5-11. Pipelined P Bus Accesses (Write, Wait, Write)

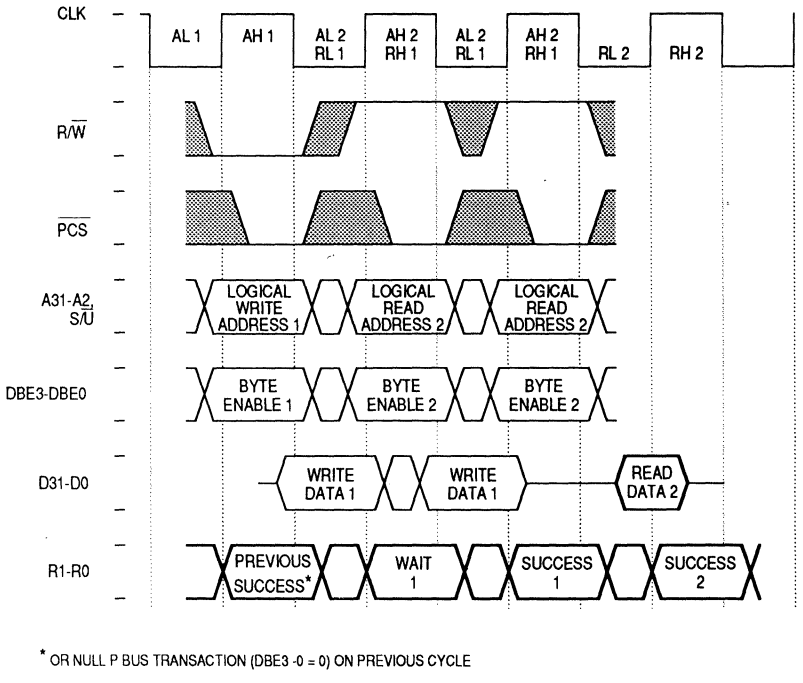
5.2.5 P Bus Faults

When a memory fault occurs on the P bus, the processor ignores the transaction that follows the faulted transaction; therefore, the MC88200 also ignores the access following the fault. Refer to *MC88100 User's Manual* for more information on fault processing.

5.3 M BUS PROTOCOL

The MC88200 M bus is the interface used by M88000 processor/CMMU nodes to access memory (and peripheral devices) as well as other MC88200 devices. The M bus is a synchronous bus with muliplexed address and data buses and control signals that have multiple functions. MC88200 devices can act as both masters and/or slaves of the M bus. These devices are M bus masters when accessing memory in response to a P bus transaction and M bus slaves when the internal registers are accessed or when snooping other transactions on the M bus.

The MC88200 M bus defaults to the slave state. Snooping is controlled by the snoop enable (SE) bit of the system control register (SCTR) and the global signal (C5) during M bus transactions by other masters. Another M bus master accesses an internal register of the



* OR NULL P BUS TRANSACTION (DBE3 -0 = 0) ON PREVIOUS CYCLE

Figure 5-12. Pipelined P Bus Accesses (Write, Wait, Read)

CMMU by appropriately addressing the 4K-byte range in control space defined for that particular MC88200 by the CMMU ID register (initialized during reset).

To perform a memory transaction required by the CPU (initiated on the P bus), the MC88200 arbitrates for mastership of the M bus. The MC88200 also requests mastership of the M bus when a snoop hit occurs and the CMMU must update memory with the modified data. When the MC88200 is master of the M bus, it performs all transactions required for the event that requested the M bus mastership; the MC88200 then relinquishes control of the bus. The MC88200 performs the M bus arbitration sequence independently for the next event that requires the M bus. PATC loads and data cache-to-memory operations are considered as two separate events, even if they result from the same P bus transaction. Therefore, a CMMU requires separate masterships for table searches and data cache operations that need to access memory.

5.3.1 Bus Arbitration

Each MC88200 device in a system arbitrates for mastership (tenure) of the M bus when performing transactions required by the CPU or when a snoop hit is detected and the CMMU must update memory with modified data. The individual bus request (BR), bus

grant (BG), and bus acknowledge (BA) signals of each MC88200 interface with external arbitration logic that determines the next bus master. The arbitration busy ($\overline{AB}$) and bus busy ($\overline{BB}$) inputs are the wired-NOR of the individual BR and BA signals, respectively; they are used by the MC88200 to determine when it is acceptable to request the M bus and when it is acceptable to acknowledge bus mastership.

The MC88200 is configured to operate in either a fairness arbitration mode or priority arbitration mode, depending on the state of the priority arbitration (PR) bit of the SCTR. In the fairness mode, the MC88200 only performs a new request for mastership of the M bus when no other M bus master has a pending request for mastership. In the priority mode, the MC88200 asserts BR independently of the requests of other masters. Both modes depend on external arbitration circuitry to prioritize bus requests that occur simultaneously.

5.3.1.1 BUS ARBITRATION STATE MACHINES. Figure 5-13 shows the state diagram of the MC88200 with all signals shown as active high, regardless of the actual polarity of the signal on the bus. The state diagram shows functional relationships between signals. Figure 5-15 shows bus arbitration timing with signal polarities as seen on the M bus. As shown in Figure 5-13, the inactive state corresponds to the slave (default) state, in which the MC88200 does not require mastership of the M bus. Following reset, the CMMU enters the inactive state. When the CMMU requires use of the M bus, it transitions from the inactive state to the contend state and issues a bus request (BR). In the contend state, the CMMU waits for a bus grant (BG) and bus not busy ($\overline{BB}$) before becoming the bus master.

While in the content state, the MC88200 is still operating in the slave mode and may be snooping other transactions on the M bus, or servicing a remote internal register access (IRA). If the MC88200 detects a snoop hit on exclusive modified data, it asserts RETRY on the local status signals (echoed by external logic onto the system status signals) to force the current bus master to terminate its tenure.

Depending on the value of the priority arbitration (PR) bit of the SCTR, after receiving a BG and $\overline{BB}$, the CMMU enters the master or blocked master state. The term "blocked" means that the CMMU will not issue another bus request until arbitration is not busy ($\overline{AB}$).

In fairness mode, the CMMU enters the blocked master state and performs transactions on the M bus. While in the blocked master state, if AB negates, the CMMU becomes "unblocked" and enters the master state. Otherwise, if AB remains asserted throughout the entire M bus tenure, when the tenure ends the CMMU negates bus acknowledge (BA) and enters the blocked inactive state. The CMMU remains in the blocked inactive state until AB negates, at which time the CMMU may request use of the bus if needed. If no internal request is pending when AB negates, the MC88200 transitions from the blocked inactive state to the inactive state.

In priority mode, the CMMU never enters a blocked state. Following a BG and $\overline{BB}$, the CMMU leaves the contend state and enters the master state in which it performs necessary M bus transaction(s). When the M bus tenure is complete, the CMMU negates BA and

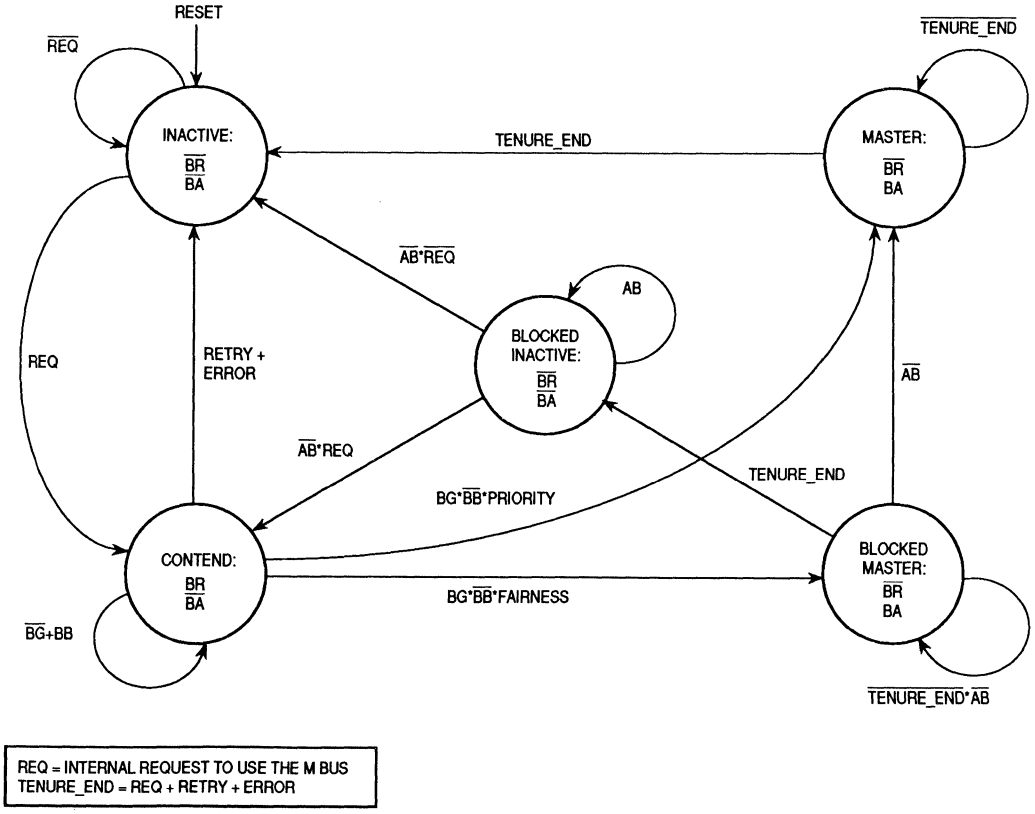


Figure 5-13. M Bus Arbitration State Diagram

transitions to the inactive state. The CMMU issues a bus request again whenever it requires use of the M bus, irrespective of the status of AB.

Finally, if the MC88200 detects the assertion of RETRY or ERROR while it is the bus master (in the master state), the CMMU terminates the tenure and leaves the master state immediately. The device that asserted RETRY can request the bus and receive mastership in this case as well.

All MC88200 devices terminate the bus tenure after the transactions required by one event (either one P bus operation or a snoop hit) are completed, regardless of the arbitration priority assigned by the external logic. This termination allows an alternate master to request (and be granted) the M bus in between any two tenures of even the highest priority device. However, because they can repeatedly request the bus (with one clock of no request in between), two relatively high-priority masters can alternate mastership between themselves and prevent lower priority devices from receiving mastership in the priority mode.

5.3.1.2 BUS ARBITRATION SIGNAL INTERFACE. Figure 5-14 is a block diagram showing the interface between the external arbitration logic and multiple M bus masters. The external arbitration logic prioritizes M bus mastership when multiple masters simultaneously request the bus and asserts the BG to only one device in response to multiple requests. Figure 5-15 illustrates the timing for the M bus arbitration.

5.3.2 M Bus Phases

After an M bus device arbitrates for bus tenure, it performs transactions on the multiplexed address/data bus in phases. The M bus is in one of the following four phases at any given time:

1. Inactive: no master has tenure on the M bus.
2. Idle: master with tenure is not performing a transaction or is between transactions.
3. Address: master with tenure is driving address for the current transaction.
4. Data: master with tenure is driving or receiving data for the current transaction.

Figure 5-16 shows the hierarchy associated with M bus mastership and examples of individual transactions on the M bus.

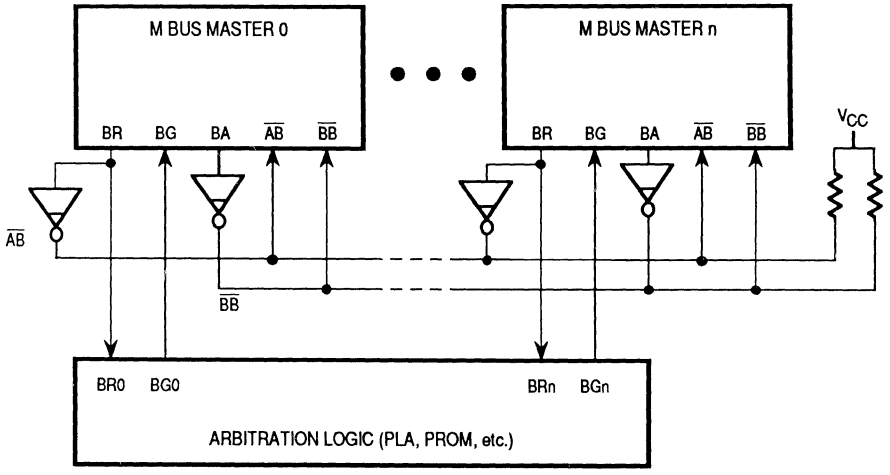


Figure 5-14. External Arbitration Logic

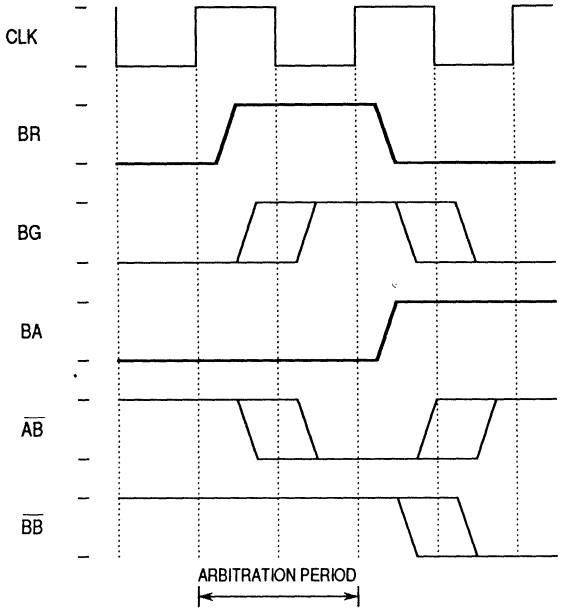


Figure 5-15 . M Bus Arbitration Timing

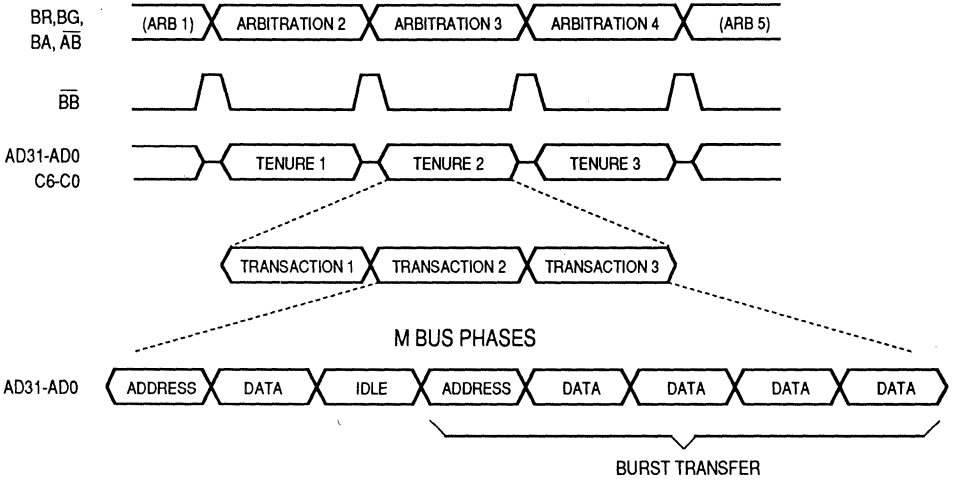


Figure 5-16. M Bus Operation Hierarchy

Valid M bus transactions consist of an address phase followed by one or more data phases. The MC88200 supports up to four consecutive data phases. Idle phases may occur at either the beginning of the transaction (before the address phase) or at the end of the transaction (after the last data phase). Some legal M bus sequences are as follows:

1. <ARB> <IDLE> <ADDRESS> <DATA> <DATA> <DATA> <DATA>
2. <ARB> <ADDRESS> <DATA> <ADDRESS> <DATA> <ADDRESS> <DATA> <ADDRESS> <DATA>
3. <ARB> <IDLE> <IDLE> (a tenure with no transactions)

Some illegal sequences are as follows:

1. <ARB> <ADDRESS> . . . <DATA> <IDLE> <DATA>
2. <ARB> <ADDRESS> . . . <IDLE> . . . <DATA>
3. <ARB> <ADDRESS> . . . <DATA (read)> . . . <DATA (write)>

The second illegal sequence, in which an idle phase follows an address phase, is performed by the MC88200 when its internal registers are accessed by the local MC88100 processor. However, this transaction is self-contained and does not interact with other slaves on the M bus.

Wait states may be inserted by M bus slaves during either the address or data phases of a transaction. Wait states indicate that the master should repeat the phase, driving the same address or data until an OK or end of data (EOD) status is driven on the system status signals ($\overline{SS3}$–$\overline{SS0}$).

The OK state indicates that the master may continue onto the next phase (address or data) if any. EOD indicates that the memory cannot support burst transfers, and requires a new address phase before the next data phase in the transaction. Wait states are never generated by M bus masters; masters are guaranteed to drive valid address or data on every corresponding address or data phase. MC88200 slave devices that detect an address phase always latch the address for snooping or in case the transaction is an access to an internal control register. When the MC88200 master detects an OK (or EOD) status during the address phase, the CMMU proceeds to the data phase of the transaction.

Table 5-1 illustrates the signals that may be used to decode the M bus phases:

Table 5-1. M Bus Control Signals

		C0	C1	C2	C3	C4	C5	C6	
M Bus Phase	$\overline{BB}$	**AP**	**IM**	**RD**	**MBE3**	**MBE2**	**MBE1**	**MBE0**	**Comments**
Inactive	H	X	X	X	X	X	X	X	No Master
Idle	L	L	X	X	L	L	L	L	No Transaction
Address	L	H	X	X	X	X	X	X	Valid Address
Data	L	L	X	X	At Least One = H				Valid Data

5.3.3 M Bus Read Transactions

A single read transaction occurs when the MC88200 reads data from memory mapped as cache inhibited and/or devices in the control memory space (including other MC88200s). The MC88200 arbitrates for bus tenure and becomes the bus master before it initiates the read access. The slave device(s) decode the address, and control is placed on the M bus during the address phase. They then reply with an OK or EOD status when they are ready to continue to the data phases(s). The bus slaves place data and the transaction status (OK or EOD) on the M bus during the data phase, or they signal an error, retry, or wait.

Figure 5-17 shows a flowchart of the basic read operation (no errors, waits, or retries). The shaded areas in this flowchart pertain to errors, waits, and retries, which are discussed in **5.4 M BUS TRANSACTION ERROR**. Figure 5-18 shows functional timing for the read. In the following functional timing diagrams, signal transitions shown with bold lines are driven by the MC88200 performing the M bus transaction or replying to the local P bus transaction. Signal lines in normal typeface are driven by other M bus slaves, or the MC88100 processor driving the P bus. Addresses on the P bus are shown as logical (virtual) address, and addresses on the M bus are shown as physical, translated addresses. M bus control (C6–C0) is shown by the corresponding function names (refer to **4.2.3 M Bus Control**): address phase (AP), data phase (DP), read (RD), etc. The absence of the RD signal implies a write transaction is being performed. MBE3–MBE0 (M bus byte enable 3 through M bus byte enable 0) indicates that all four byte strobes are asserted (word transfer). BYTE ENABLES indicates that the transfer may be either a byte, half-word, or word transfer.

The MC88200 performs burst reads of the M bus whenever possible. The MC88200 begins a burst read by performing the standard address phase for the memory transaction (after arbitration). However, it does not assert C1 last data transfer (LDT) until all of the required data is transferred from memory. The MC88200 internally increments the target address where the data is stored; the slave must also increment the memory address from which data is supplied (if it supports burst accesses). Figure 5-19 shows a burst read transaction when the slave device cannot increment the address, it signals the MC88200 with an EOD status, which terminates the transaction but retains the bus tenure. The MC88200 increments the address by four and starts another transaction by driving a new address phase on the next clock cycle. The MC88200 keeps supplying a new address for every EOD status received on the data phase until it has read in four words of data. During the last data phase, the CMMU signals LDT, during which the memory may either respond with OK or EOD. Figure 5-20 shows a cache line read with an EOD termination on each data phase.

If the slave device cannot decode the address or supply the data in one clock cycle, it signals the MC88200 to wait by driving a wait status on the system status lines ($\overline{SS3}$–$\overline{SS0}$). The MC88200 repeats the current phase until an OK status is received from the slave. Figure 5-21 shows the signal timing for a read access where a one-clock wait is generated during the address and data phases.

Table 5-2 shows the M bus byte enables for various operand sizes read while the MC88100 is in either big Endian or little Endian mode. MBE3–MBE0 on the M bus are driven identically

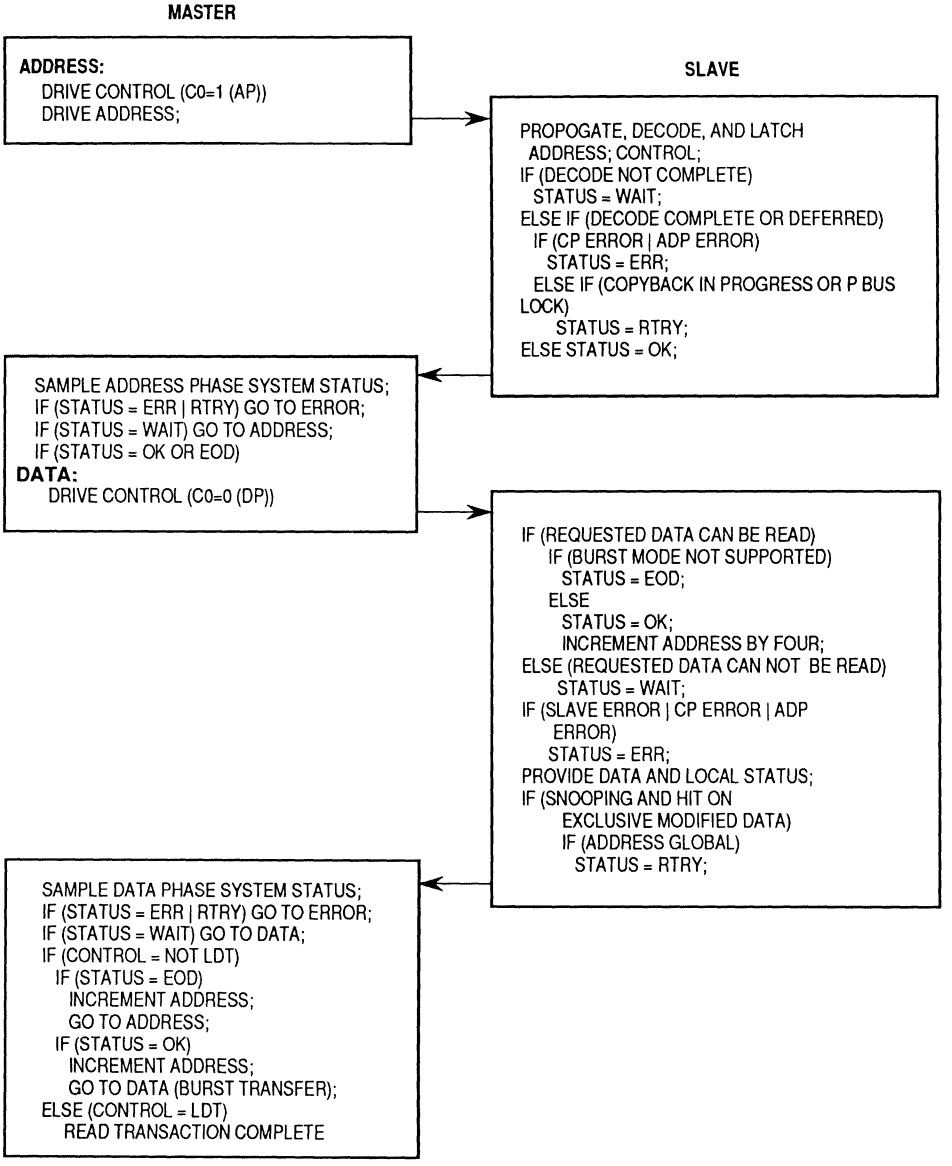
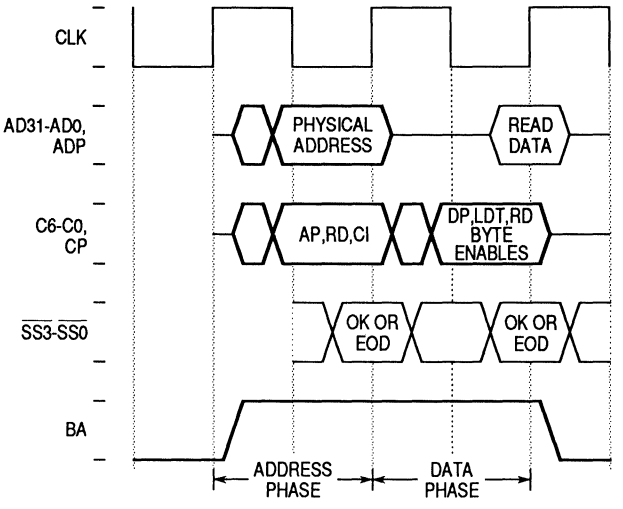


Figure 5-17. M Bus Read Flowchart



5

Figure 5-18. M Bus Read Timing

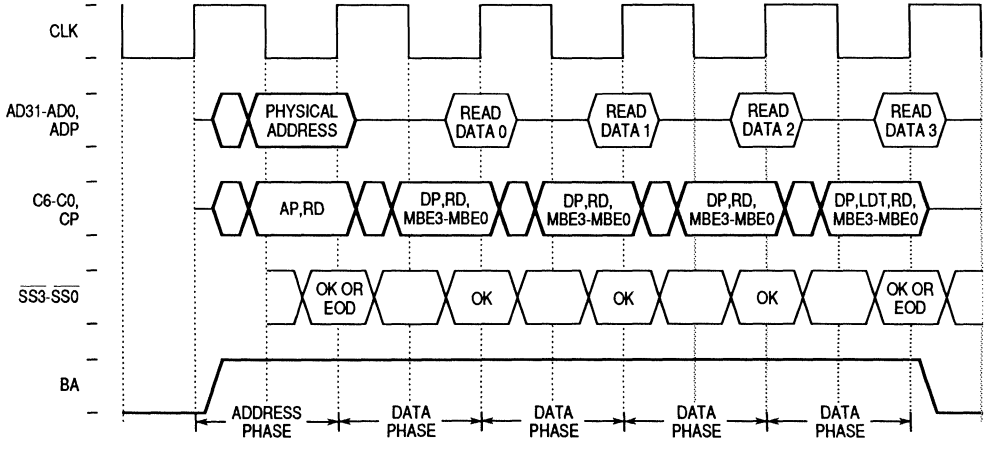


Figure 5-19. M Bus Burst Read Timing

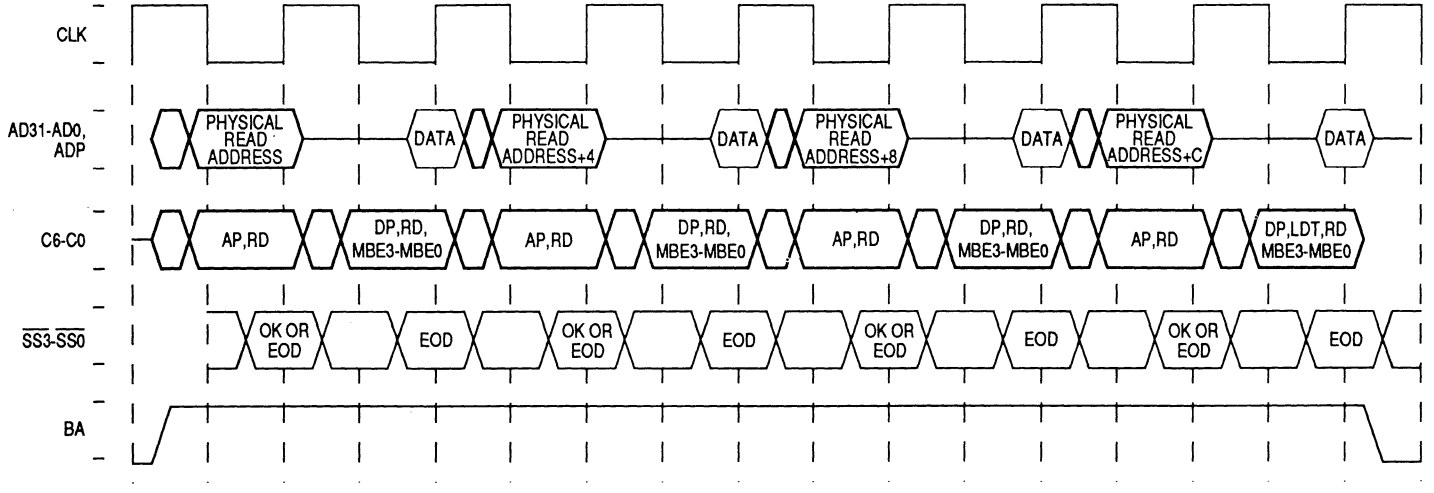


Figure 5-20. Cache Line Read with EOD Termination

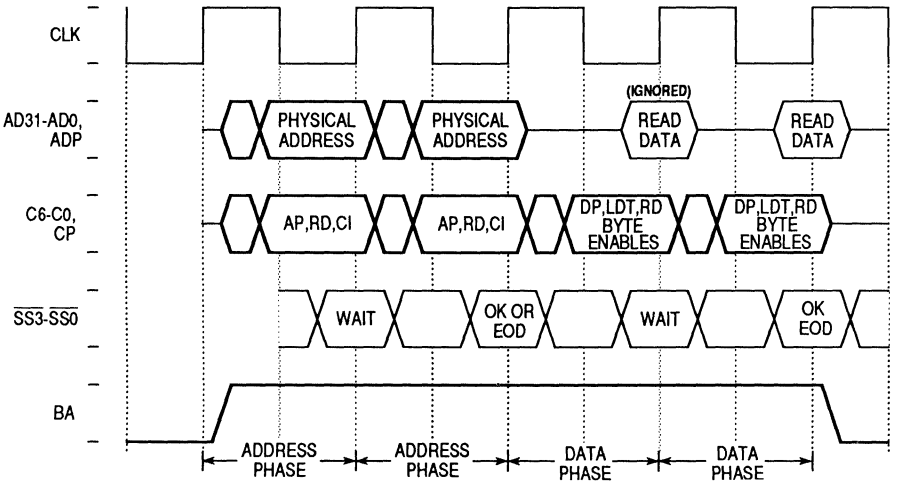


Figure 5-21. Read Access with Wait States

to DBE3–DBE0 for the corresponding P bus transaction. In the tables, bytes 0–3 correspond to the data bytes at memory address locations 0–3. Half-word 0 and half-word 1 correspond to the first two half-word operands in memory, namely those at address locations 0 and 2. Word reads are identical (all byte enables asserted) for both big Endian and little Endian modes. Double-word reads occur as two separate word read transactions. In big Endian mode, the most significant word is stored in lower memory. In little Endian mode, the least significant word is stored in lower memory.

5.3.4 M Bus Write Transactions

A write transaction occurs between the MC88200 and one or more slave devices. The slave device(s) decodes the address and control that the MC88200 places on the M bus during the address phase. During the data phase, the M bus slaves receive data from the MC88200.

Figure 5-22 shows a flowchart of the basic write operation (no errors, waits, or retries). The shaded areas in this figure are steps not applicable to a simple write transaction. Figure 5-23 shows the signal timings for the write.

Cache line copybacks due to snoop hits and data cache flushes that copy back modified data to memory use burst write transfers. The MC88200 begins a burst write by issuing an address phase with intent to modify (IM) asserted and RD negated. It does not assert C1 (LDT) until all the data is transferred to memory. The MC88200 internally increments the write target address; the slaves must also increment the address to store data where it is intended. Figure 5-24 shows the timing of a burst write. When the slave device cannot increment the address, it signals the MC88200 with an EOD status, which terminates the

Table 5-2. Data Bus Requirements for Read Cycles

Big Endian									
Transfer Size	Operand Address	MBE3	MBE2	MBE1	MBE0	DATA			
						AD31:AD21	AD23:AD16	AD15:AD8	AD7:AD0
Byte	0	1	0	0	0	Byte 0	X	X	X
	1	0	1	0	0	X	Byte 1	X	X
	2	0	0	1	0	X	X	Byte 2	X
	3	0	0	0	1	X	X	X	Byte 3
Half Word	0	1	1	0	0	[Half Word 0] MSB	LSB	X	X
	2	0	0	1	1	X	X	[Half Word 1] MSB	LSB
Word	0	1	1	1	1	[Word 0] MSB	HMB	LMB	LSB
Double Word	0	1	1	1	1	[Most Significant Word] MSB	HMB	LMB	LSB
	4	1	1	1	1	[Least Significant Word] MSB	HMB	LMB	LSB
Little Endian									
Transfer Size	Operand Address	MBE3	MBE2	MBE1	MBE0	DATA			
						AD31:AD21	AD23:AD16	AD15:AD8	AD7:AD0
Byte	0	0	0	0	1	X	X	X	Byte 0
	1	0	0	1	0	X	X	Byte 1	X
	2	0	1	0	0	X	Byte 2	X	X
	3	1	0	0	0	Byte 3	X	X	X
Half Word	0	0	0	1	1	X	X	[Half Word 0] MSB	LSB
	2	1	1	0	0	[Half Word] MSB	LSB	X	X
Word	0	1	1	1	1	[Word 0] MSB	HMB	LMB	LSB
Double Word	0	1	1	1	1	[Least Significant Word] MSB	HMB	LMB	LSB
	4	1	1	1	1	[Most Significant Word] MSB	HMB	LMB	LSB

NOTES:
1. With address on, AD1 and AD0 are always driven as zero by the MC88200.
2. X = data and parity ignored.

transaction. The MC88200 increments the address by four and starts another transaction by driving a new address phase on the next clock cycle. The MC88200 keeps supplying a new address for every EOD status received on the data phase until it has written four words of data. During the last data phase, the CMMU signals LDT, during which the memory may either respond with OK or EOD. Figure 5-25 shows a cache line write with an EOD termination on each data phase.

If the slave device cannot decode the address or accept the data in one clock cycle, it signals the MC88200 to wait by driving a wait status on the $\overline{SS3}$–$\overline{SS0}$ lines. The MC88200

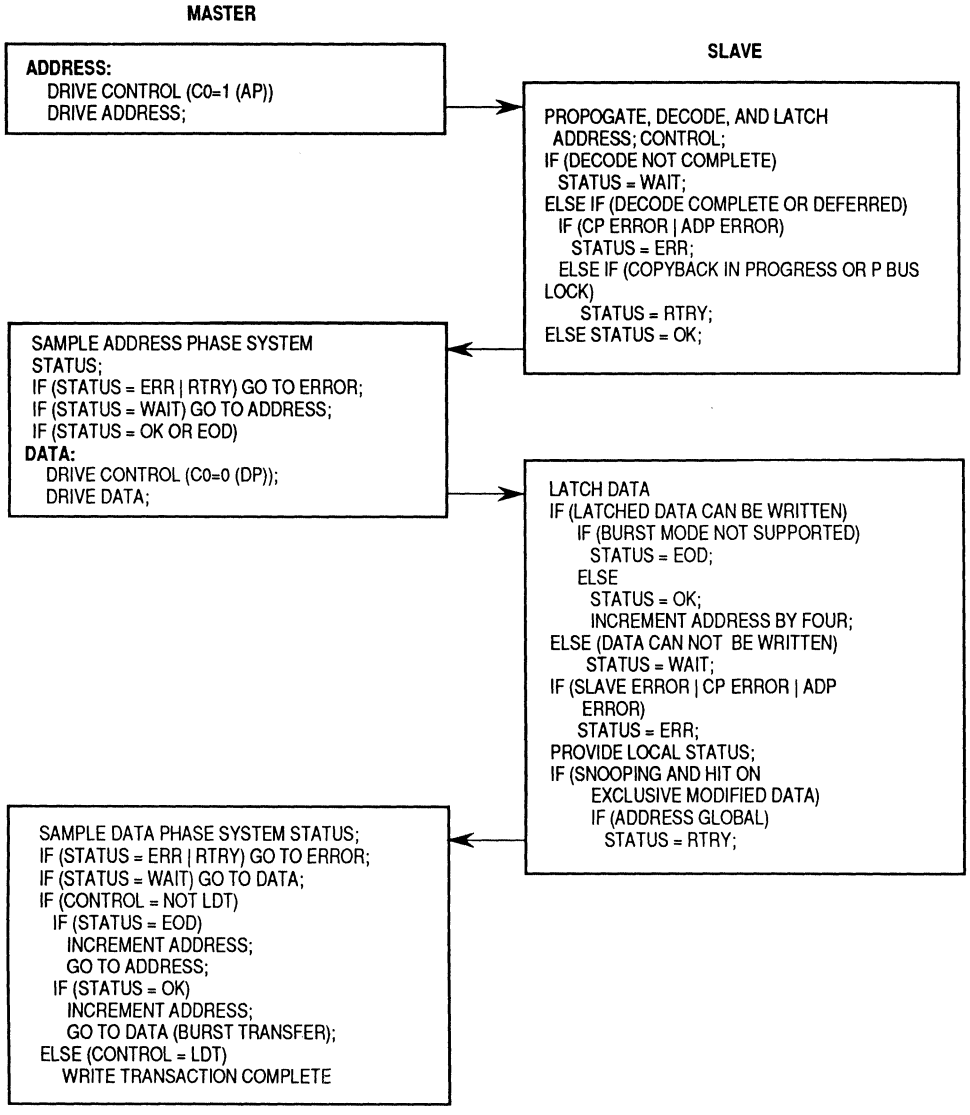


Figure 5-22. M Bus Write Flowchart

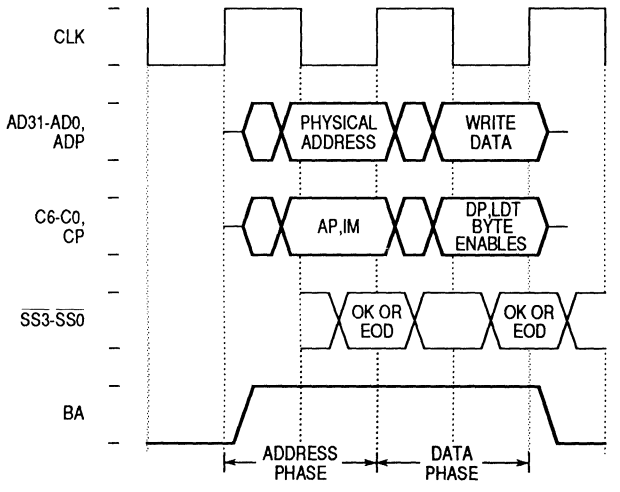


Figure 5-23. M Bus Write Timing

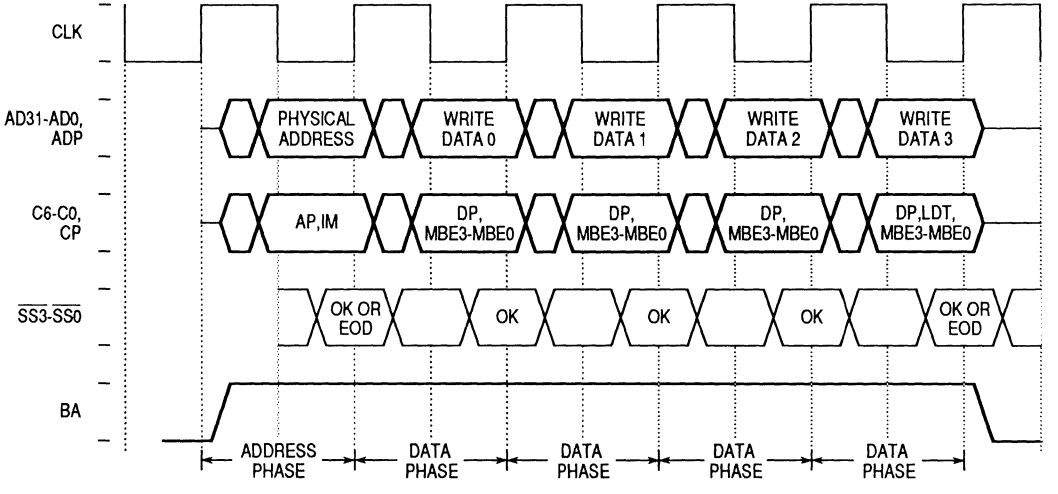


Figure 5-24. M Bus Burst Write Timing

repeats the address phase until an OK status is received, the same as with read wait states. Wait states (see Figure 5-26) during the data phase of a write transaction cause the data phase to be repeated. Data and control are guaranteed around the falling and rising edges of the clock during the data phase in accordance with specifications in **SECTION 8 ELECTRICAL CHARACTERISTICS**.

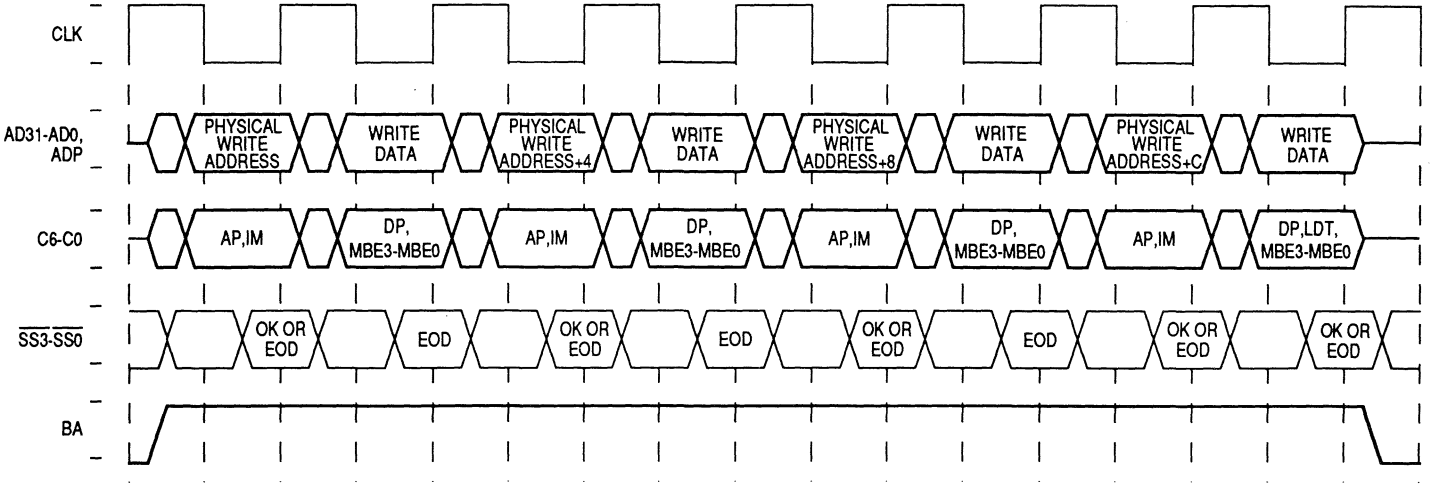


Figure 5-25. Cache Line Write with EOD Termination

Table 5-3 shows the M bus byte enables for various operand sizes written while the MC88100 is in either big Endian or little Endian mode. MBE3–MBE0 on the M bus are driven identically to DBE3–DBE0 for the corresponding P bus transaction. The entire data bus is driven for byte and half-word writes. The byte strobes indicate which part of the bus is valid depending on the byte-ordering mode of the processor. In the tables, bytes 0–3 correspond to the data bytes at memory address locations 0–3. Half-word 0 and half-word 1 correspond to the first two half-word operands in memory, namely those at address locations 0 and 2. Word writes are identical (all byte enables asserted) for both big Endian and little Endian modes. Double word reads occur as two separate word write transactions. In big Endian

Table 5-3. Data Bus Behavior for Write Cycles

Big Endian									
Transfer Size	Operand Address	MBE3	MBE2	MBE1	MBE0	DATA			
						AD31:AD21	AD23:AD16	AD15:AD8	AD7:AD0
Byte	0	1	0	0	0	Byte 0	Byte 0	Byte 0	Byte 0
	1	0	1	0	0	Byte 1	Byte 1	Byte 1	Byte 1
	2	0	0	1	0	Byte 2	Byte 2	Byte 2	Byte 2
	3	0	0	0	1	Byte 3	Byte 3	Byte 3	Byte 3
Half Word	0	1	1	0	0	[Half Word 0] MSB	LSB	[Half Word 0] MSB	LSB
	2	0	0	1	1	[Half Word 2] MSB	LSB	[Half Word 2] MSB	LSB
Word	0	1	1	1	1	[Word 0] MSB	HMB	LMB	LSB
Double Word	0	1	1	1	1	[Most Significant Word] MSB	HMB	LMB	LSB
	4	1	1	1	1	[Least Significant Word] MSB	HMB	LMB	LSB
Little Endian									
Transfer Size	Operand Address	MBE3	MBE2	MBE1	MBE0	DATA			
						AD31:AD21	AD23:AD16	AD15:AD8	AD7:AD0
Byte	0	0	0	0	1	Byte 0	Byte 0	Byte 0	Byte 0
	1	0	0	1	0	Byte 1	Byte 1	Byte 1	Byte 1
	2	0	1	0	0	Byte 2	Byte 2	Byte 2	Byte 2
	3	1	0	0	0	Byte 3	Byte 3	Byte 3	Byte 3
Half Word	0	0	0	1	1	[Half Word 0] MSB	LSB	[Half Word 0] MSB	LSB
	2	1	1	0	0	[Half Word 2] MSB	LSB	[Half Word 2] MSB	LSB
Word	0	1	1	1	1	[Word 0] MSB	HMB	LMB	LSB
Double Word	0	1	1	1	1	[Least Significant Word] MSB	HMB	LMB	LSB
	4	1	1	1	1	[Most Significant Word] MSB	HMB	LMB	LSB

NOTES:
1. With address on, AD1 and AD0 are always driven as zero by the MC88200.
2. X = data and parity ignored.

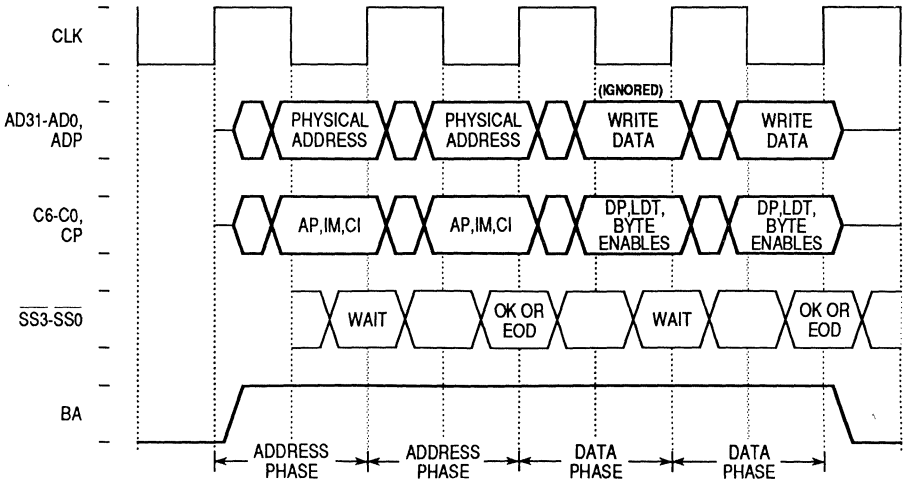


Figure 5-26. M Bus Write Access with Wait States

5

mode, the most significant word is stored in lower memory. In little Endian mode, the least significant word is stored in lower memory.

5.3.5 Error/Retry Termination

Error/retry terminations occur during two circumstances. First, these terminations occur when any participating M bus module generates an error or retry status ($\overline{SS3}$–$\overline{SS0}$), allowing for an orderly termination of the current transaction and current bus tenure. Second, these terminations occur when the MC88200 detects a parity error while reading data, while snooping, or while being accessed as a slave. Figure 5-27 is a flowchart of the error/retry transaction protocol. Refer to Figure 5-52 for more information on error and retry timing.

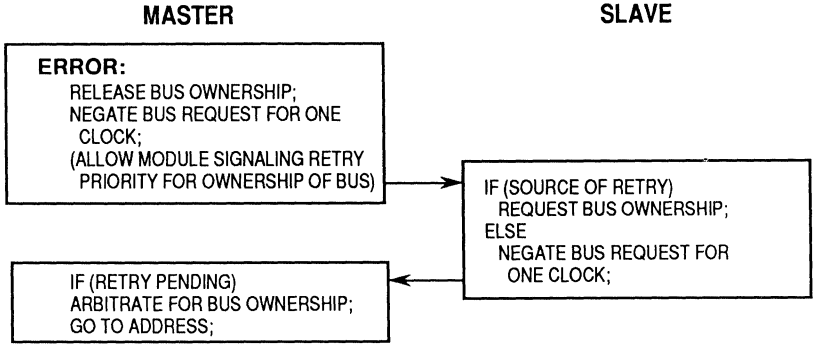


Figure 5-27. Error/Retry Flowchart

5.4 M BUS TRANSACTION ERROR

An M bus transaction error occurs during a memory transaction that pertains only to M bus devices. The transaction is not initiated because of a processor request for instructions or data. Since these transactions do not effect the normal operations of the processor, errors are not reported as faults. However, the MC88200 maintains error information when these errors occur, and this information can be used by other devices (including the P bus processor) to detect that an M bus transaction error occurred.

An M bus transaction error can occur during a snoop copyback, a flush transaction, and a probe transaction. The SSR and/or the SAR is updated when the error occurs, as described in the following paragraphs.

5.4.1 Probe Error

A probe transaction may result in a translation table search as the MC88200 determines the translation information for the logical address in the SAR. If the MC88200 encounters an M bus error ($\overline{SS3}$ signal asserted by memory or another M bus device), the bus error (BE) bit of the SSR will be set. The device that initiated the probe must periodically sample this bit to determine if the probe is successful. If the BE bit is set, the probe is unsuccessful; the probe result information is invalid, and the SAR contains the physical address where the fault occurred. A bus error is the only type error possible for a probe transaction.

If the probe was initiated by the local (P bus) processor, a probe error is still reported through the system registers and is not signaled as a fault.

5.4.2 Flush Error

When a flush is performed using the copyback option, the MC88200 copies cache lines back to memory when those lines are marked exclusive modified. During these line copybacks, the MC88200 can encounter an M bus error ($\overline{SS3}$ signal asserted by memory or another M bus device). As with the probe error, the flush error causes the BE bit to be set in the SSR. The SAR is updated with the physical address where the error occurred. A bus error is the only type error possible during a flush copyback.

5.4.3 Snoop Copyback Error

When M bus snooping results in a cache hit on exclusive modified data, the MC88200 copies the cache line back to memory. During this line copyback, the MC88200 can encounter an M bus error ($\overline{SS3}$ signal asserted by memory or another M bus device). When a snoop copyback error occurs, the copyback error (CE) bit of the SSR is set. The remainder of the SSR is **not** updated.

5.4.4 Parity Error

When operating as an M bus master, the MC88200 always generates even parity on the M bus address, control signals, and write data. Parity checking is enabled via the parity error (PE) bit of the SCTR. The MC88200 checks parity on read data when operating as M bus master; M bus masters do not signal parity errors on the M bus. Parity errors are reported to the processor via a fault reply. When operating as an M bus slave, the MC88200 checks parity on addresses and write data during accesses to its internal control registers. Also, slave MC88200 devices check parity on addresses that are snooped (parity for data is **not** checked). The MC88200 only checks parity on reads for data bytes that have their corresponding byte enables asserted. Parity is generated on all data bytes for a write, regardless of the state of the byte enables.

MC88200s operating as slaves take two clock cycles to compute parity and then signal an M bus error if parity is incorrect. MC88200 masters detecting a parity error during a burst read will complete the entire burst read transaction before signaling a fault to the processor. The cache is not updated with valid data if a parity error occurs.

5

5.5 OPERATING STATES SUMMARY

Table 5-4 summarizes all of the MC88200 normal and fault state operations. This table includes the MC88200 state in relation to the P bus and M bus, the states of various registers,

Table 5-4. Operating States

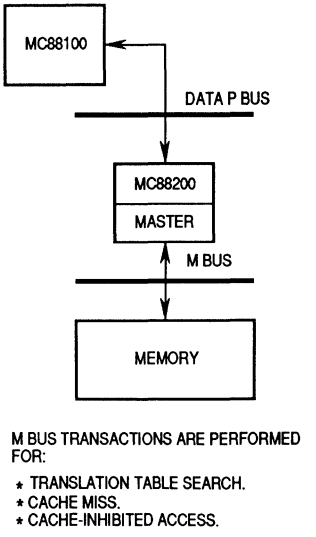
Transaction	P Bus State	M Bus State	Exception	Update PFSR	Update PFAR	Update SSR	Update SAR	Assert M Bus ERR	Reply P Bus Fault
P Bus R/$\overline{W}$ includes Internal Register Access (Figures 5-28, 5-29, 5-30)	Slave	Master	Page Fault	Bits 18–16 = 101	Page Descriptor Address	No	No	No	Yes
	Slave	Master	Segment Fault	Bits 18–16 = 100	Segment Descriptor Address	No	No	No	Yes
	Slave	Master	Privilege Violation	Bits 18–16 = 110	Physical Address of Error	No	No	No	Yes
	Slave	Master	M Bus Error	Bits 18–16 = 011	Physical Address of Error	No	No	No	Yes
	Slave	Master	Parity Error	Bits 18–16 = 011	Physical Address of Error	No	No	No	Yes
	Slave	Master	Write Violation	Bits 18–16 = 111	Contents Destroyed	No	No	No	Yes
	Slave	Master	Success	No	No	No	No	No	No

— Continued —

Table 5-4. Operating States (Continued)

Transaction	P Bus State	M Bus State	Exception	Update PFSR	Update PFAR	Update SSR	Update SAR	Assert M Bus ERR	Reply P Bus Fault
Slave Internal Register Access (Figure 5-31)	N/A	Slave	Parity Error	No	No	No	No	Yes	No
	N/A	Slave	M Bus Error	No	No	No	No	No	No
	N/A	Slave	Success	No	No	No	No	No	No
M Bus Snooping (Figure 5-32)	N/A	Slave	M Bus Error	No	No	No	No	No	No
	N/A	Slave	Parity Error	No	No	No	No	Yes	No
	N/A	Slave	Success	No	No	No	No	No	No
Snoop Copyback (Figure 5-33)	N/A	Master	M Bus Error	No	No	Bit 15 Set	No	No	No
	N/A	Master	Success	No	No	No	No	No	No
Probe (Figure 5-33)	N/A	Master	M Bus Error	No	No	Bit 14 Set Bits 8–0 Destroyed	Physical Address of Error	No	No
	N/A	Master	Success	No	No	Bit 14 Cleared Bits 8–0 = Results	Physical Address of Probe	No	No
Flush Copyback (Figure 5-33)	N/A	Master	M Bus Error	No	No	Bits 14 Set Bits 8–0 Destroyed	Physical Address of Error	No	No
	N/A	Master	Success	No	No	Bit 14 Cleared Bits 8–0 Destroyed	Contents Destroyed	No	No
Flush Invalidate (Figure 5-33)	N/A	Master	Success	No	No	Bit 14 Cleared Bits 8–0 Destroyed	Contents Destroyed	No	No
PATC Invalidate (Figure 5-33)	N/A	Master	Success	No	No	Bit 14 Cleared Bits 8–0 Destroyed	Contents Destroyed	No	No

and the states of the signals that indicate exceptions. Figures following this table illustrate the MC88200 bus orientation for each entry in the table; each table entry includes the appropriate figure reference.



5

Figure 5-28. P Bus Read/Write (M Bus Memory Access)

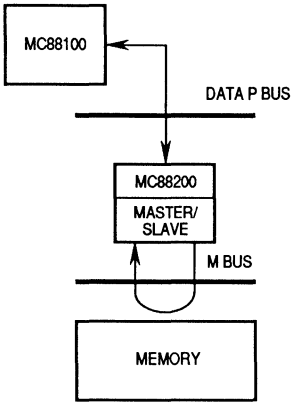
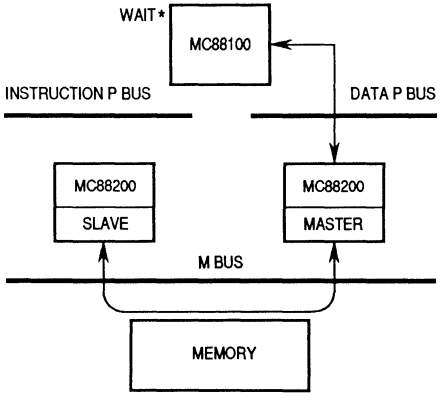
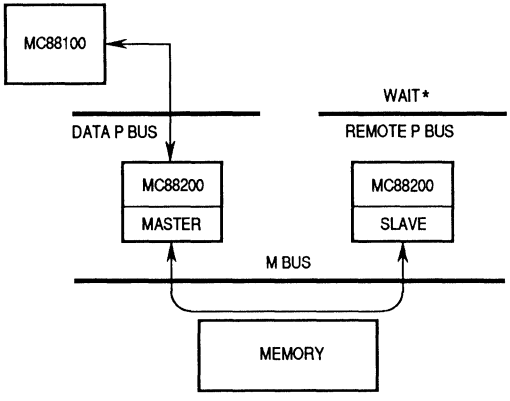


Figure 5-29. Data MC88200 Register Access



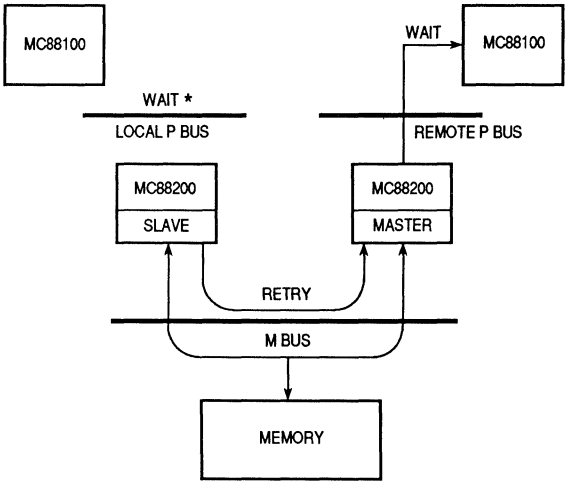
* P BUS WAIT ISSUED IF THE INSTRUCTION CMMU IS SELECTED BY THE PROCESSOR.

Figure 5-30. Instruction MC88200 Register Access



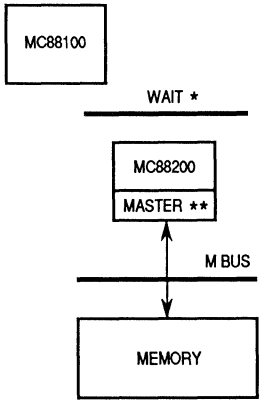
* P BUS WAIT ISSUED IF THE REMOTE CMMU IS SELECTED BY THE PROCESSOR.

Figure 5-31. Remote MC88200 Register Access



* P BUS WAIT ISSUED IF THE REMOTE CMMU IS SELECTED BY THE PROCESSOR.

Figure 5-32. MC88200 Snooping a Global Transaction



* P BUS WAIT ISSUED IF THE REMOTE CMMU IS SELECTED BY THE PROCESSOR.

** M BUS TRANSACTIONS ARE PERFORMED FOR:
- COPYBACK COMMANDS THAT ENCOUNTER MODIFIED CACHE ENTRIES.
- SNOOP OPERATIONS THAT ENCOUNTER MODIFIED CACHE ENTRIES.
- PROBE COMMANDS THAT MISS BOTH ATCS.

Figure 5-33. MC88200 Probe, Flush, and Copyback

5.6 P BUS AND M BUS TRANSACTION SEQUENCES

The functional timing diagrams in Figures 5-34 through 5-56 illustrate various M bus activity as a result of P bus operations. Most M bus sequences start with an optional page address translation cache (PATC) load, which occurs when the logical address presented by the processor misses in both the block address translation cache (BATC) and the PATC. The CMMU automatically accesses translation tables in memory to load the PATC if translation is enabled. The access is then restarted from the address latched at the P bus address inputs. The second time around, the address hits in the PATC, assuming that no fault occurred during the table search sequence or that no write protection violation occurred after the table search operation. When interpreting the M bus operations, refer to the appropriate paragraphs in **SECTION 2 MEMORY MANAGEMENT** and **SECTION 3 DATA CACHE**.

NOTE

In Figures 5-34 through 5-56, traces that appear in bold type are signals driven by the CMMU. Signals in normal type are driven by the processor or another bus master. All four byte strobes asserted during a data phase are shown as MBE3–MBE0, indicating a word transfer. Byte strobes supporting byte or half-word transfers have their signals shown as BYTE ENABLES. The block labelled "M BUS ARBITRATION" in the figures corresponds to Figure 5-15.

5

5.6.1 Table Search Operation

The MC88200 performs a table search when a logical address provided by the processor misses in the BATC and PATC with address translation enabled (MMU turned on). The table search loads the PATC with a new entry so a logical-to-physical address translation may be performed.

Table 5-5 shows a table search sequence defining the address and control asserted during the operation.

Table 5-5. Table Search Functional Chart

Control Signals	M Bus Transactions		
	Read Segment Descriptor	Read Page Descriptor	Write* Page Descriptor
Lock (LK)	0	1	1
Intent to Modify (IM)	0	0	1
Cache Inhibit (CI)	0	0	0
Global (G)	0	0	0
Read (RD)	1	1	0
Address (AP)	[Segment	Page	[Page]]

*Optional Page Descriptor Write

5.6.1.1 TABLE SEARCH WITHOUT PAGE DESCRIPTOR UPDATE. Figure 5-34 shows the effects of an address translation cache (ATC) miss in the MC88200 as seen from the M bus. If logical read address 1 does not hit in either the BATC or PATC, a wait is generated on the P bus reply during the RL1 and RH1 reply phase. On the next rising edge of the clock, M bus arbitration begins with the assertion of bus request (BR). If a bus grant (BG) is received before the next rising edge of the clock, bus acknowledge (BA) is driven and the CMMU becomes the M bus master (see **5.3.1 Bus Arbitration** for more information).

The MC88200 begins the M bus tenure with an address phase (C0 = 0) and the address of the segment descriptor in memory to which the active area descriptor points. When the memory system has data available, it provides the data and OK reply on the system status. For brevity, all M bus cycles shown in this subsection complete with no wait states from the memory. After the segment descriptor is read, the page descriptor is read in the same manner. The PATC is loaded with a new entry, and the ATC lookup is restarted from the logical read address latched internally in the MC88200. On the second ATC lookup, the logical address is translated into a physical address by the new PATC entry. The physical address is presented to the data cache to determine if there is a hit in the set selected with bits 11–2 of the logical address. If there is a hit, data is presented back to the processor as indicated in the Figure 5-34. If there is a data cache miss, refer to Figure 5-37.

Figure 5-35 shows a PATC load operation that occurs as a result of a write transaction which does not have an associated ATC entry. It is similar to Figure 5-34, except the P bus continues to drive the pending write data. Figure 5-34 should be used as a reference in cases where a PATC load is required to replace a section of a functional timing diagram of a write cycle. Aside from P bus activity shown in Figure 5-35, the timing for the M bus is the same as that shown in Figure 5-34.

The operations involving the M bus arbitration, table search, and PATC update are collectively called a "PATC load" operation. Some figures in this section refer to this operation if an ATC miss occurs. In those cases, the section of the functional timing diagram referring to the PATC load should be replaced with the area labeled "PATC load." This also applies to Figure 5-36 which is similar to Figure 5-34, but shows a page descriptor write to update the U or M bits in the descriptor.

5.6.1.2 TABLE SEARCH WITH PAGE DESCRIPTOR UPDATE. Figure 5-36 shows a table search operation like that shown in Figure 5-34, but the page descriptor in memory is written to update its used (U) or modified (M) bits. MC88200 accesses to page descriptors during table searches are marked as locked (LK = 1) transactions on the M bus. This procedure is done to insure that a page descriptor read, followed by a page descriptor write (to update U or M bits), occurs as an indivisible bus tenure. External M bus devices must not interrupt locked M bus transaction sequences, otherwise, the atomicity of the locked sequence is broken.

Some applications mandate the capability to interrupt or suspend any M bus transaction, regardless of whether the transaction is locked or unlocked. In these cases, one may further

specify which locked transactions within the locked sequence are critical for 1) proper software operation and 2) proper MC88200 operation.

For proper software operation, the page descriptor write and the write portion of an exchange memory (**xmem**) sequence must not be terminated with a retry because an external master could acquire the M bus and access inconsistent page descriptor or semaphore data in memory. In cases where bus mastership must be obtained immediately by an external master, termination of these transactions with a bus error is acceptable. However, the entire locked sequence must be rerun in the exception handler. For proper MC88200 operation, the page descriptor write transaction must not be terminated by a retry. If this transaction is retried, MC88200 operation cannot be guaranteed. Specifically, while the MC88200 attempts to reacquire the M bus after the retry, the MC88200 may be the target of a remote internal register access or may perform a snoop operation. If operated as a slave under these circumstances, proper operation cannot be guaranteed.

Locked writes identified by LK = 1 and RD = 1, are detected by latching control signals C3 and C2 during the address phase of an M bus transaction. Retries from an external device that need to break locked transactions must be qualified with the above conditions to insure both proper software (**xmem**) operation and MC88200 operation.

5.6.2 Data Cache Miss

The MC88200 experiences a data cache miss when catching is enabled (CI = 0) and there is a miss of the translated physical address in the selected set of the cache.

5.6.2.1 READ MISS. Figure 5-37 shows M bus operations that occur due to a data cache read miss. If an ATC miss occurs, a PATC entry is loaded as shown in Figures 5-34 or 5-36 where "PATC load" is indicated. After the PATC entry is loaded and two clock cycles later, the CMMU rearbitrates for the M bus to fill the cache line that is selected for replacement via the LRU algorithm. If the selected cache line is marked exclusive modified (EM), a cache line copyback using a burst transfer is performed to write the modified (dirty) line to memory. If the line is shared unmodified (SU), exclusive unmodified (EU), or invalid (I), no copyback is performed. The CMMU then performs a cache line read using a burst mode transfer to read four words into the cache line. If the memory system does not support burst mode transfers, it responds with end of data (EOD) on the system status lines (see Figure 5-20). After the cache line is read, the cache is updated. Then the cache access is restarted from the P bus logical read address 1. The second time through, the access hits in the cache and data is driven back to the processor during the P bus reply phase.

5.6.2.2 WRITE MISS. Figure 5-38 shows a data cache miss as a result of a write transaction from the P bus. The M bus tenure starts similarly to a cache miss on a read transaction. The MC88200 performs a line copyback if the cache line selected for replacement contains EM data. It then does a burst read transaction to fill the line of the cache with four new

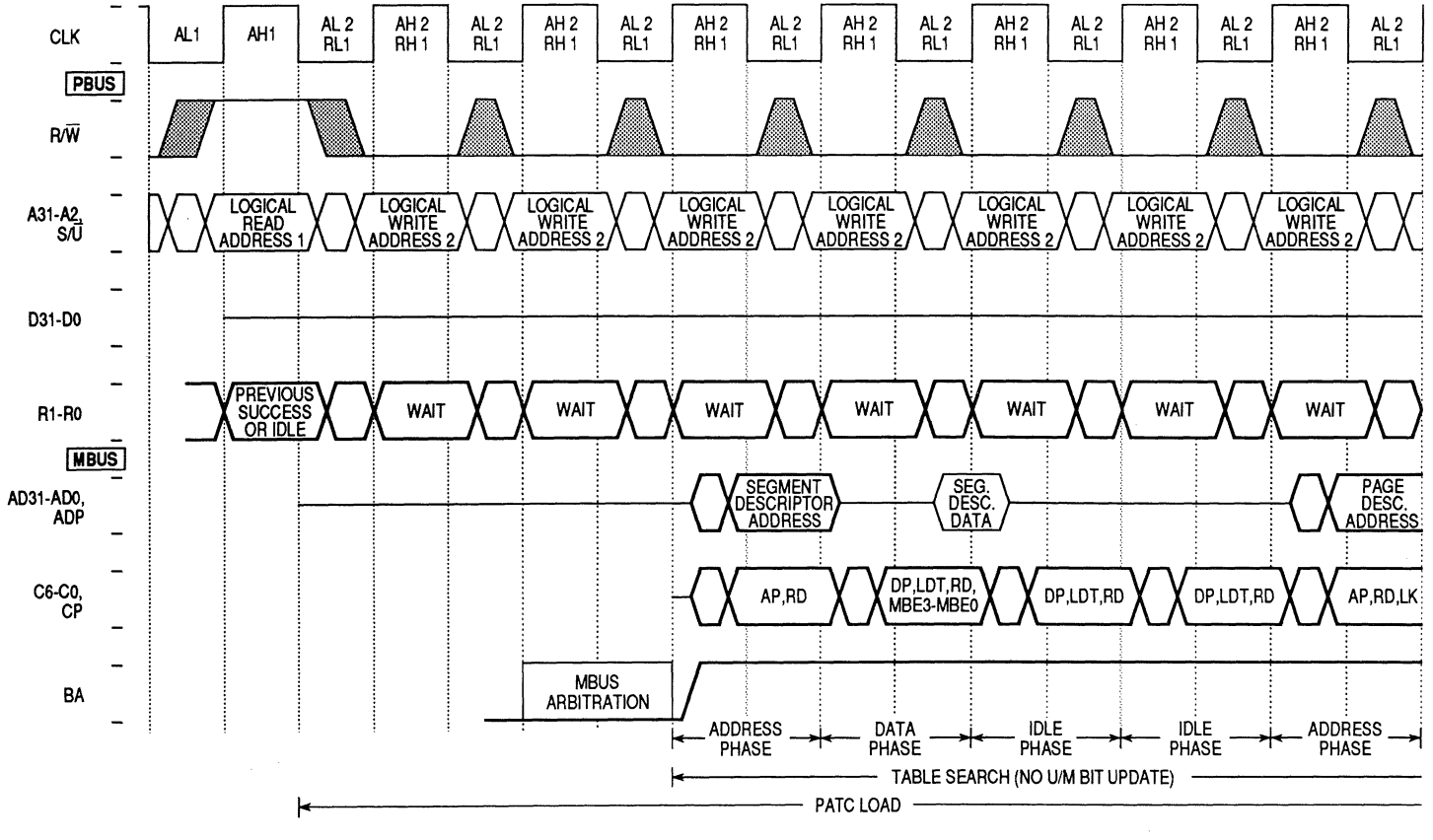


Figure 5-34. Data Cache Read Hit — ATC Miss (No U/M Update) (Sheet 1 of 2)

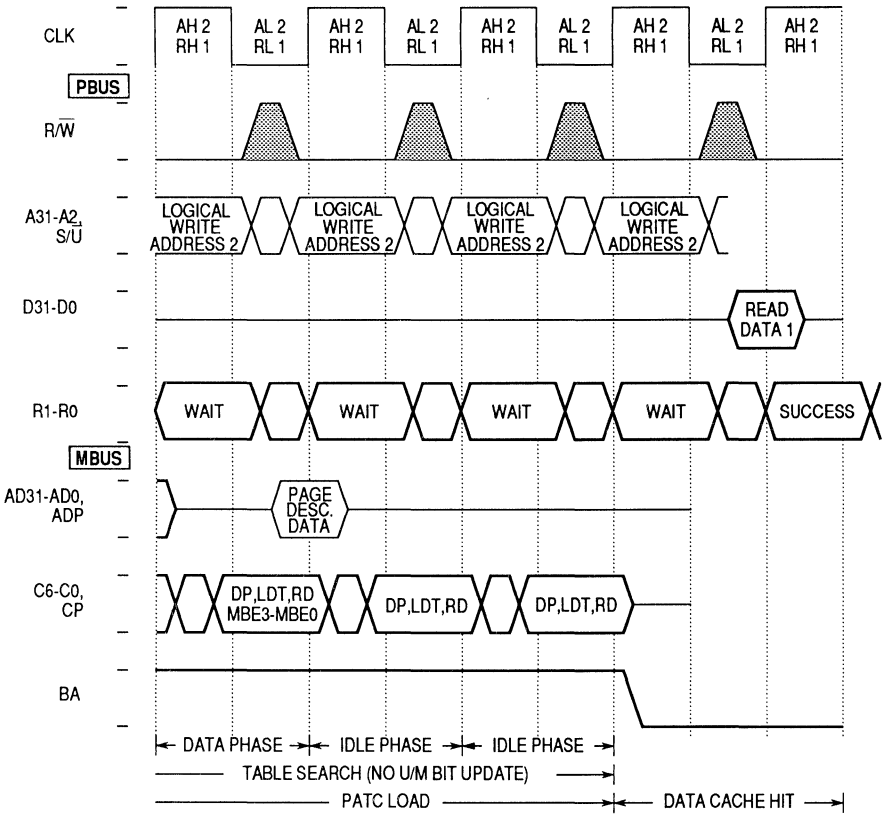


Figure 5-34. Data Cache Read Hit — ATC Miss (No U/M Update) (Sheet 2 of 2)

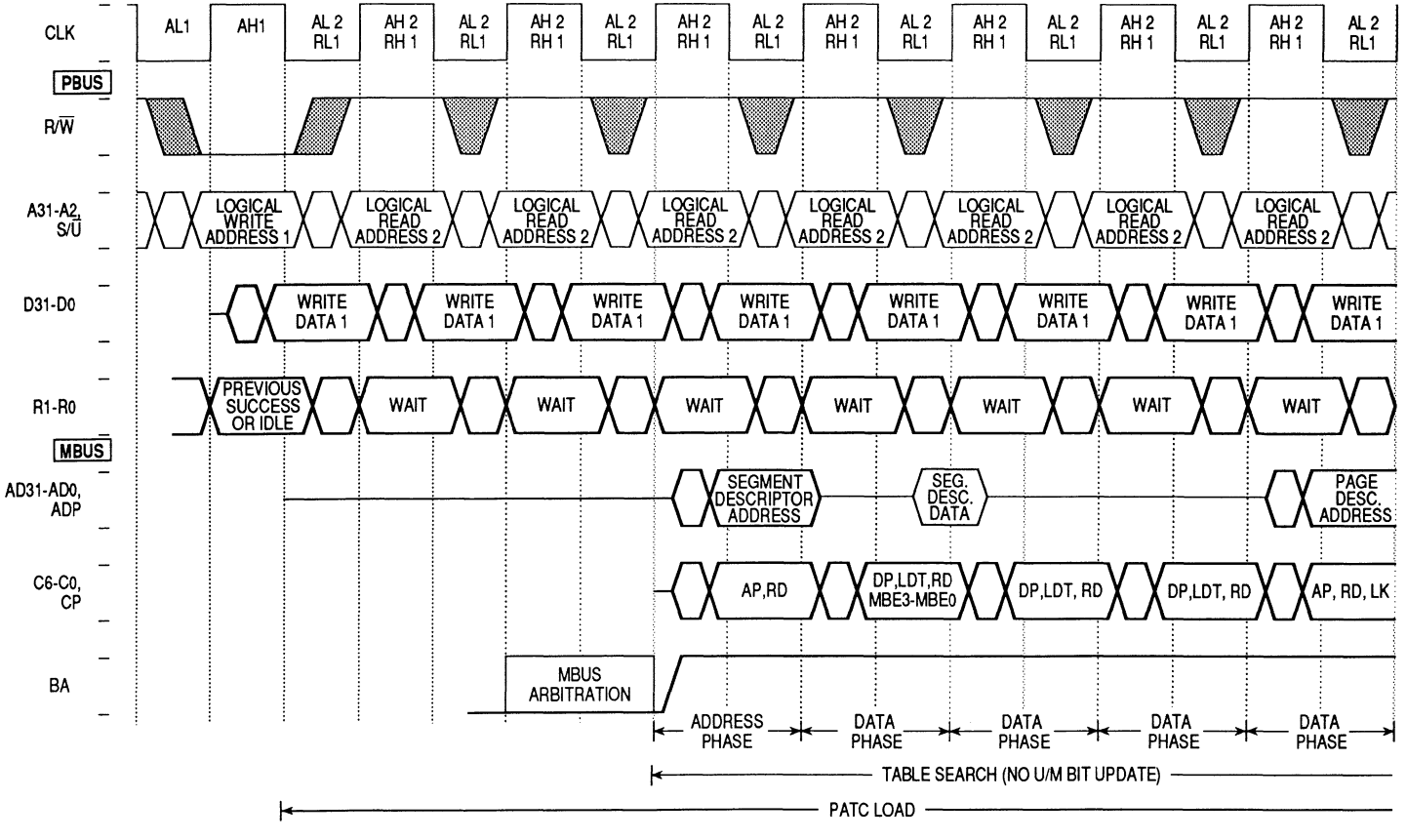


Figure 5-35. Data Cache Write Hit — ATC Miss (No U/M Update) (Sheet 1 of 2)

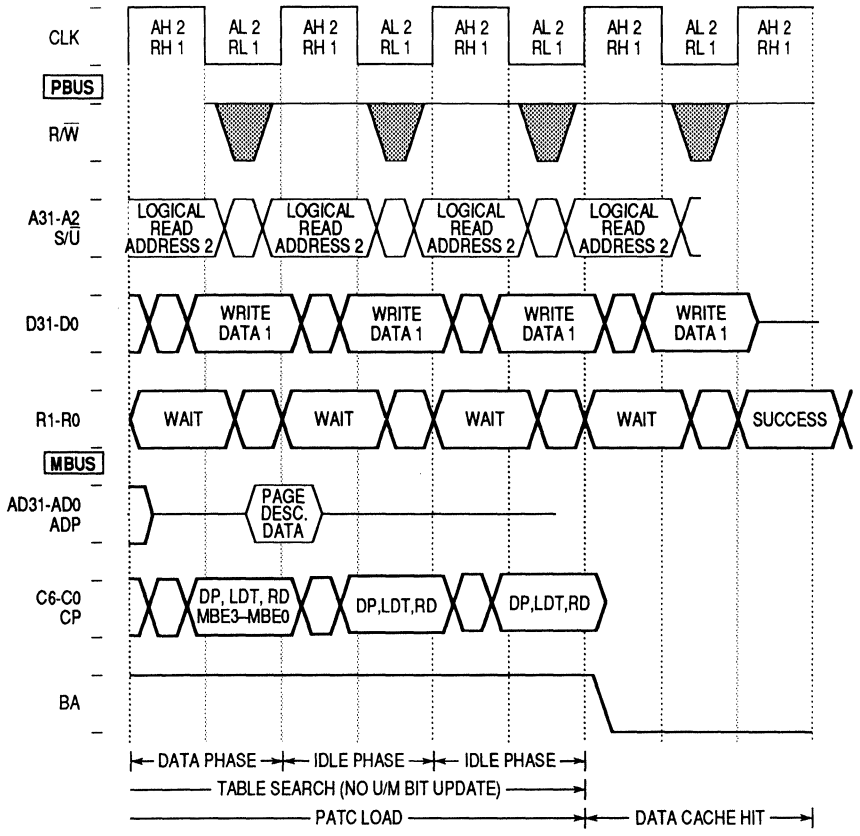


Figure 5-35. Data Cache Write Hit — ATC Miss (No U/M Update) (Sheet 2 of 2)

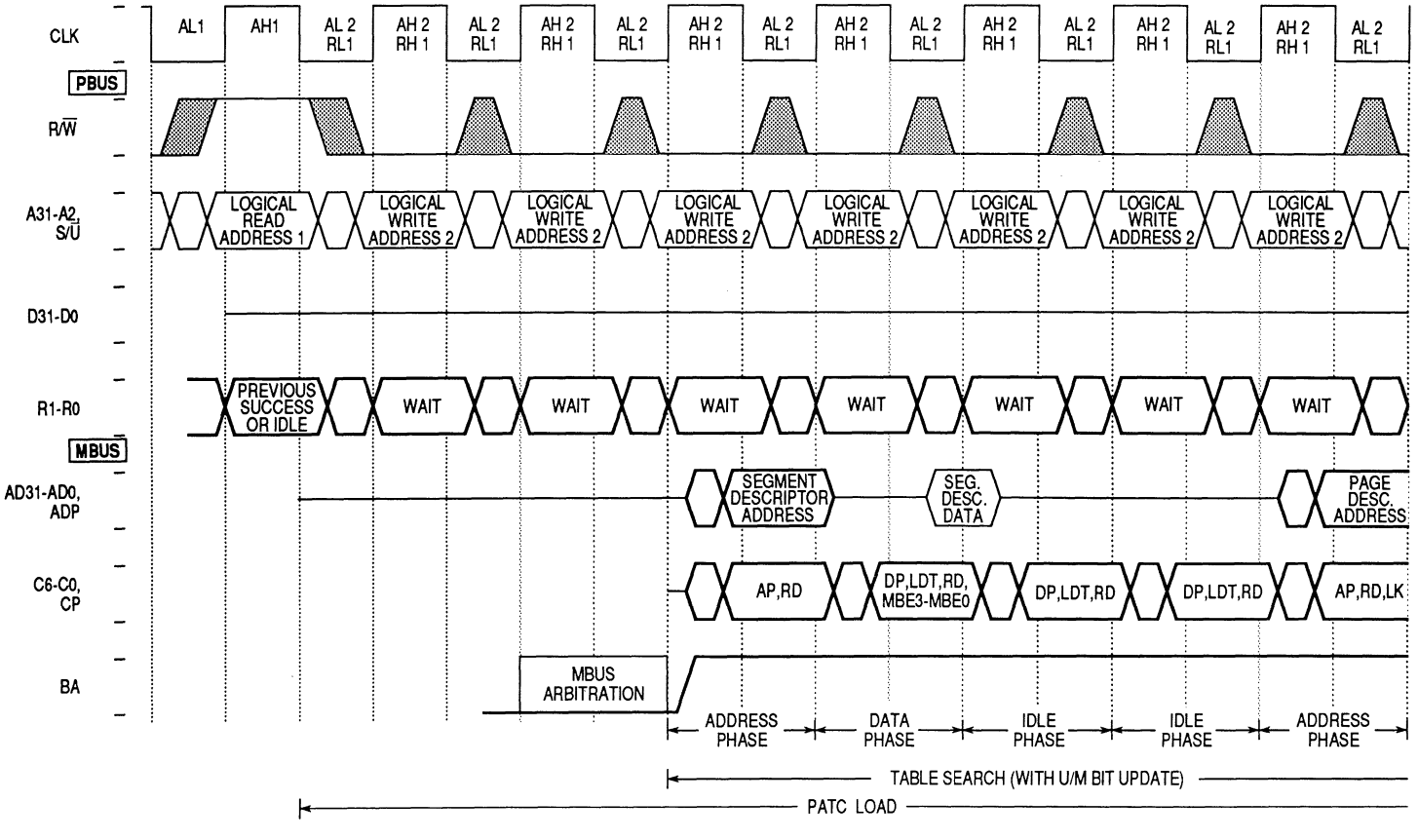


Figure 5-36. Data Cache Read Hit — ATC Miss (U/M Update) (Sheet 1 of 2)

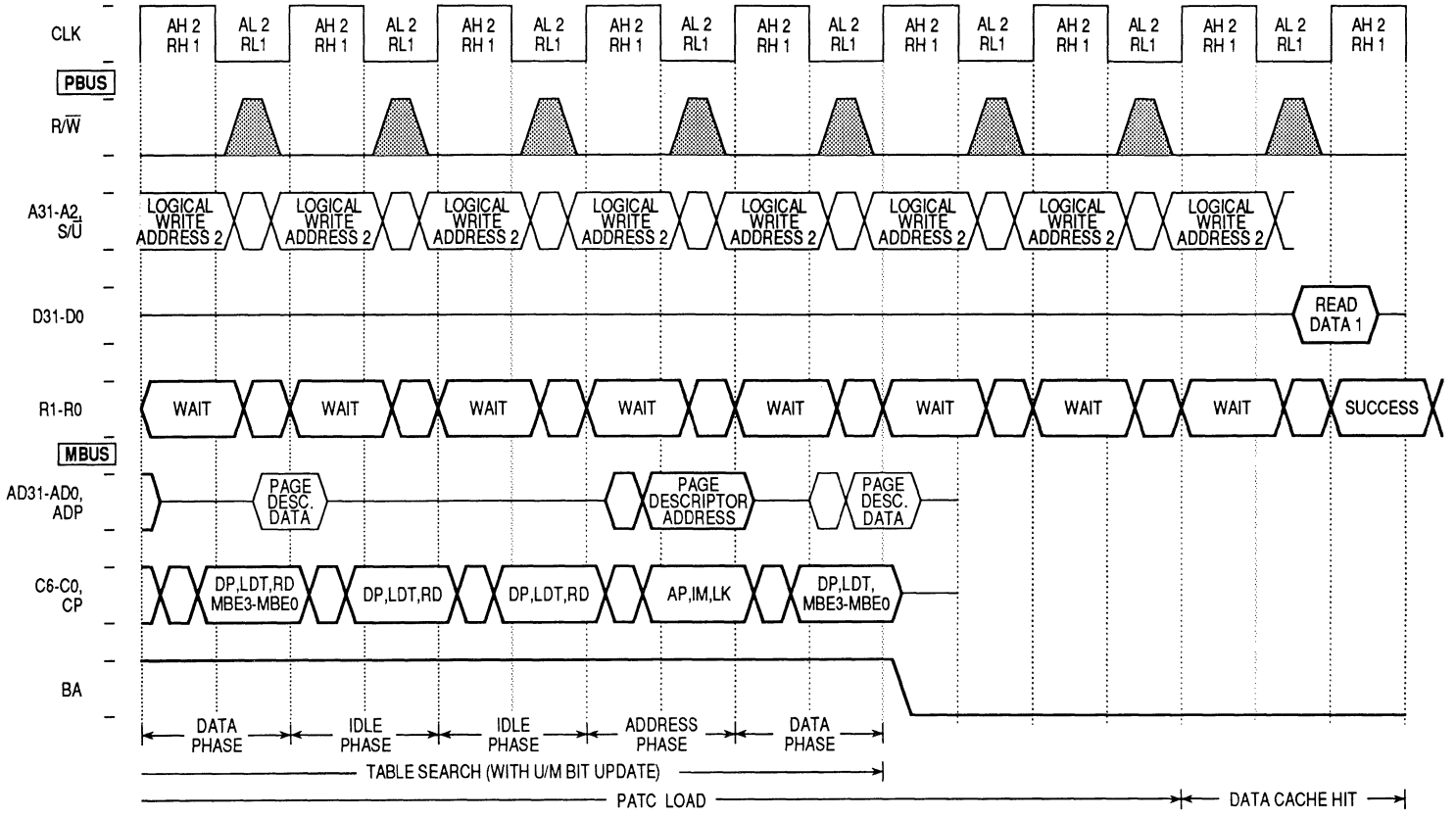


Figure 5-36. Data Cache Read Hit — ATC Miss (U/M Update) (Sheet 2 of 2)

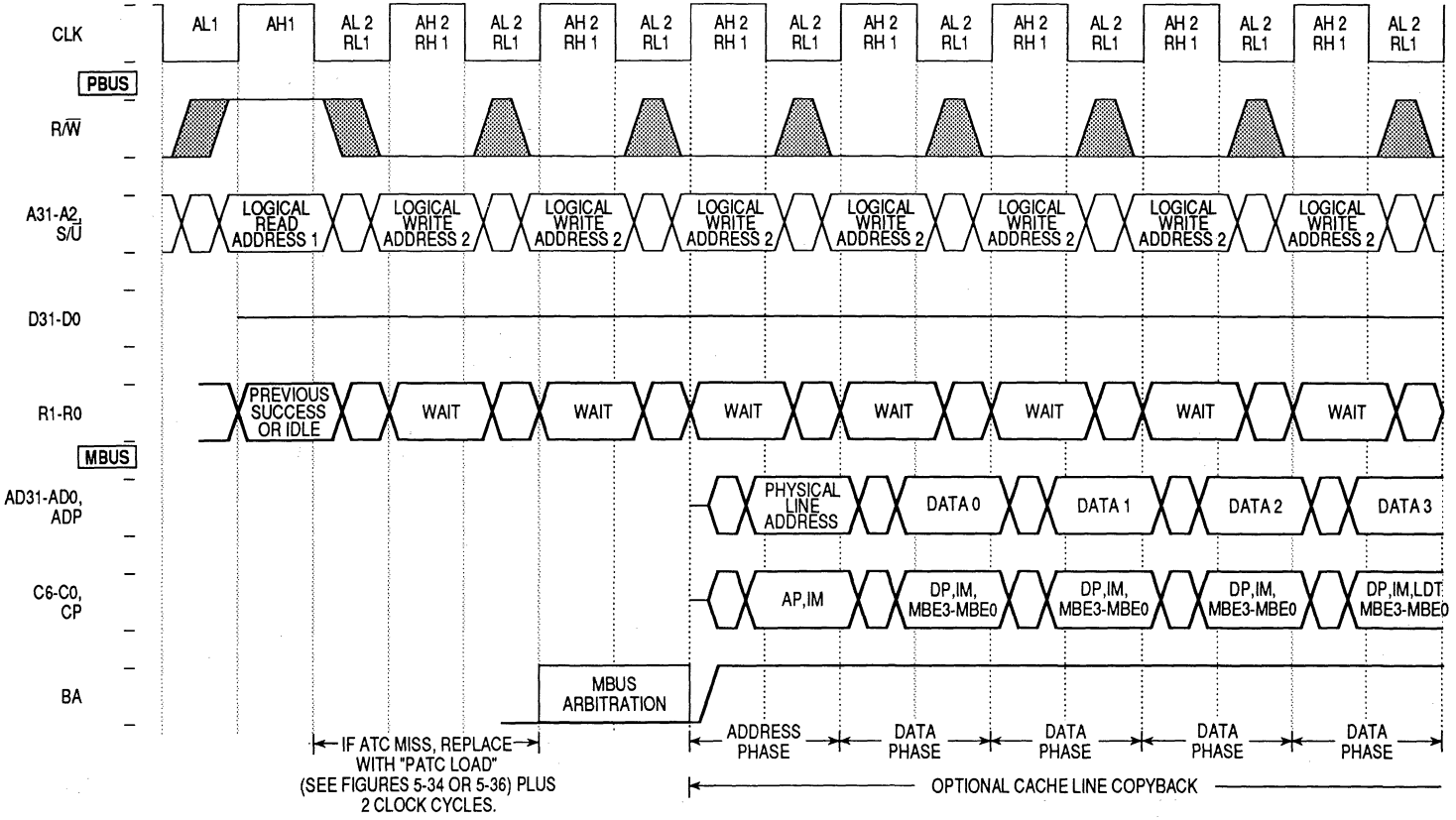


Figure 5-37. Data Cache Read Miss (Copyback and Line Fill) (Sheet 1 of 3)

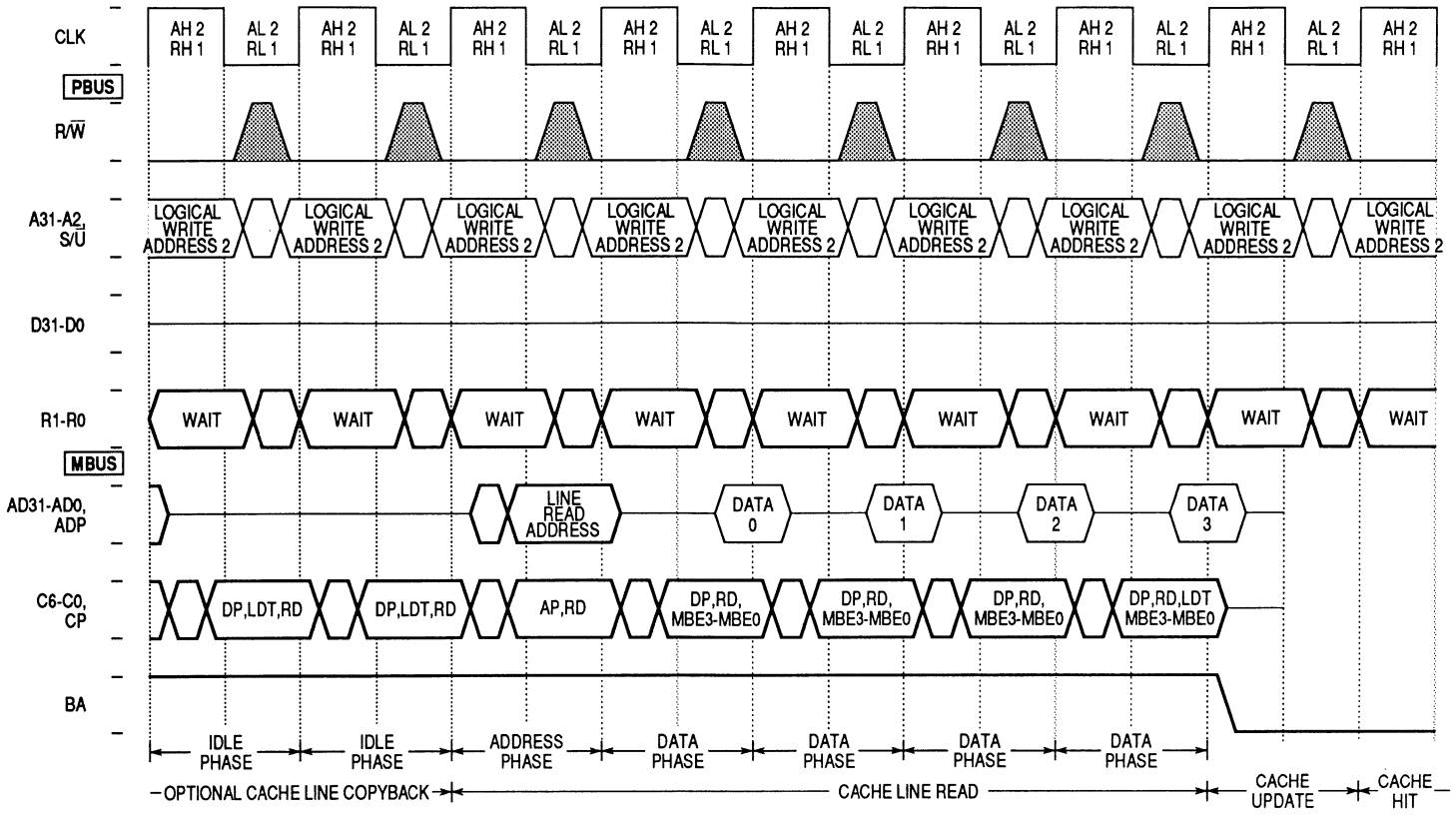


Figure 5-37. Data Cache Read Miss (Copyback and Line Fill) (Sheet 2 of 3)

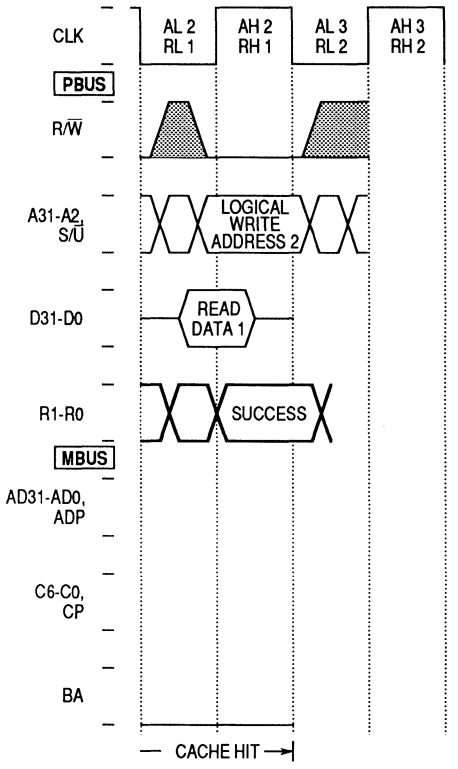


Figure 5-37. Data Cache Read Miss (Copyback and Line Fill) (Sheet 3 of 3)

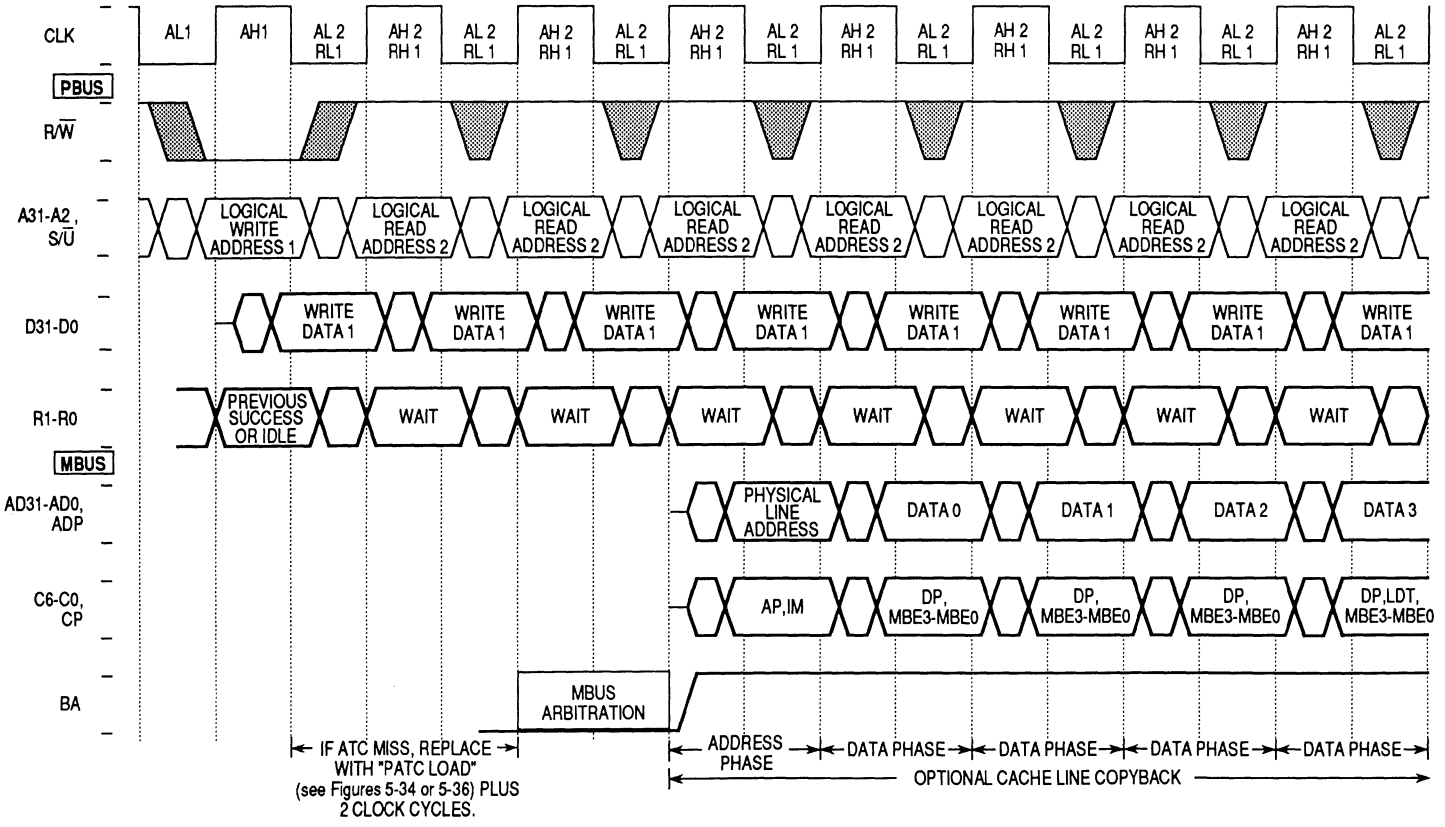


Figure 5-38. Data Cache Write Miss (Copyback, Line Fill, and Write-Once) (Sheet 1 of 3)

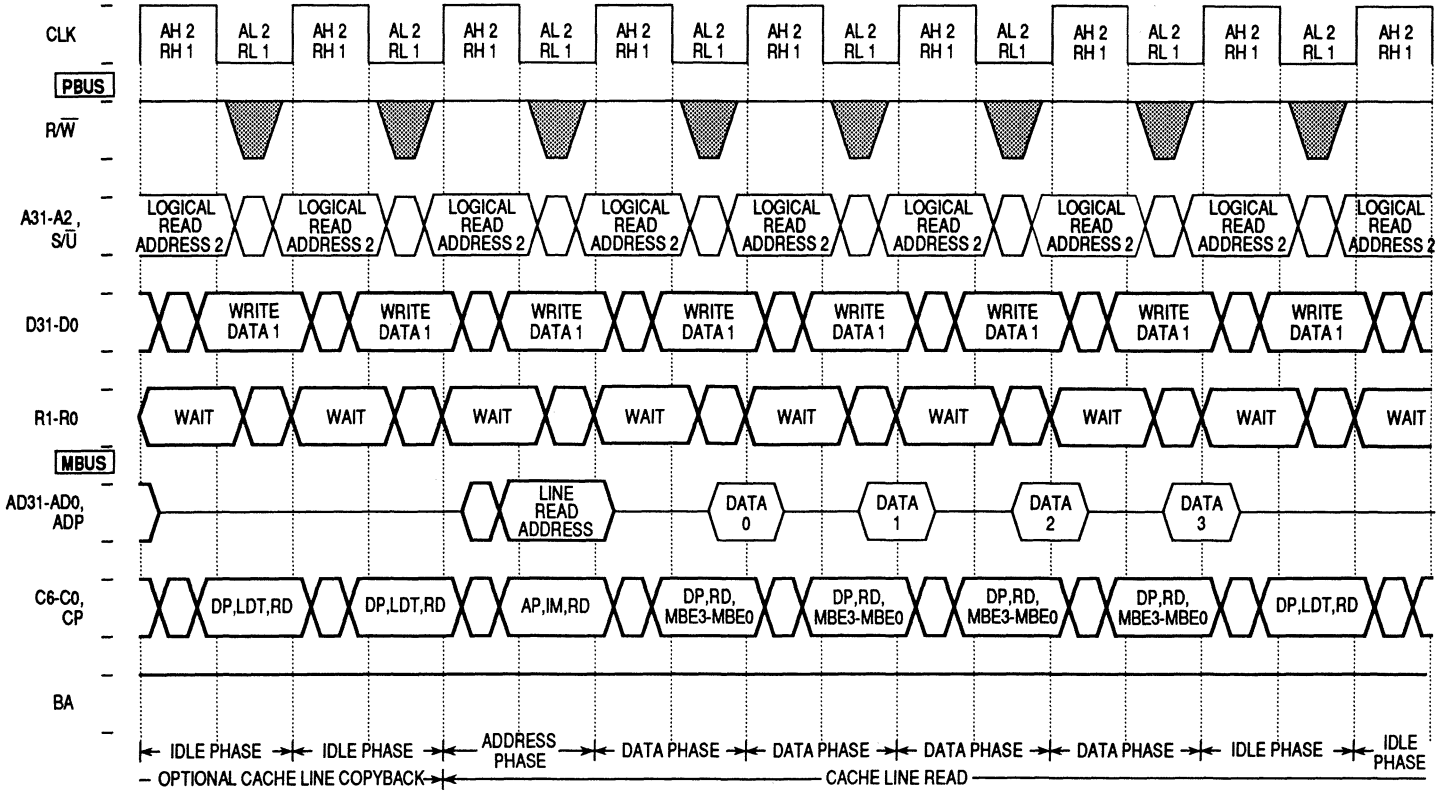


Figure 5-38. Data Cache Write Miss (Copyback, Line Fill, and Write-Once) (Sheet 2 of 3)

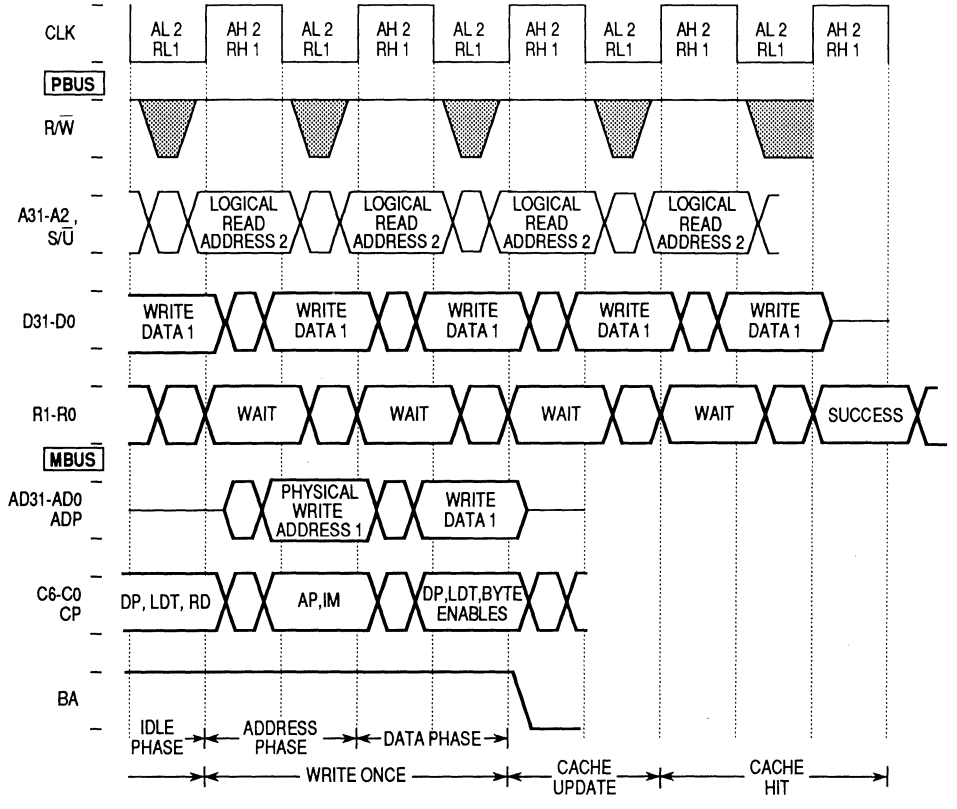


Figure 5-38. Data Cache Write Miss (Copyback, Line Fill, and Write-Once) (Sheet 3 of 3)

words of data. This line read has the intent to modify (IM) bit set on the M bus control (C1) to indicate that the read will be followed by a write-once to one of the four words. The write-once completes and the MC88200 relinquishes the bus tenure.

5.6.3 Data Cache Write-Once/Write-Through

Figure 5-39 shows a single-word write on the M bus performed during 1) the first write (write-once policy) which hits in a cache line when the transaction is mapped as copyback (WT bit clear in ATC entry), or 2) transactions mapped as write through (WT bit set in ATC entry), or 3) cache-inhibited writes. The last data transfer (LDT) signal is driven on control pin C1 during the data phase, indicating the last data phase in the transaction.

5.6.4 Data Cache Inhibit

Cache-inhibited accesses are performed when either $\overline{\text{DLOCK}}$ is asserted on the P bus or translated as cache inhibited in the BATC or PATC entry. The MC88200 asserts CI (C4) during the address phase of a cache-inhibited transaction. During the data phase, last data transfer (LDT) is asserted indicating that the data phase is the last one in the transaction. Cache-inhibited accesses never perform burst transactions. Figure 5-40 shows functional timing for a cache-inhibited read transaction. Figure 5-39, which shows functional timing for a write-once or write-through transaction, has the same timing as cache-inhibited write access, with the addition of CI = 1 on the M bus control.

5.6.5 Local CMMU Transactions

CMMU local register transactions are performed when the CPU accesses an internal CMMU register via the P bus. The MC88200 arbitrates for an M bus tenure and drives the register address during an address phase. For a read, the address phase is followed by an idle phase in which no data is driven on the M bus (see Figure 5-41). The data is instead driven back to the local processor or P bus.

For a write, the address phase is followed by an idle phase in which the write data is driven on the M bus (see Figure 5-42). However, since the M bus byte enables are not asserted on the M bus, the data is not qualified as being valid.

5.6.6 Remote CMMU Transactions

A remote CMMU register transaction is an access to control space (upper 1 Mbyte of supervisor space) that maps to a register in an MC88200 device that was not chip selected ($\overline{\text{PCS}}$ negated for that CMMU) by the P bus transaction. The master CMMU performing the remote register transaction acquires the M bus through arbitration and drives an address

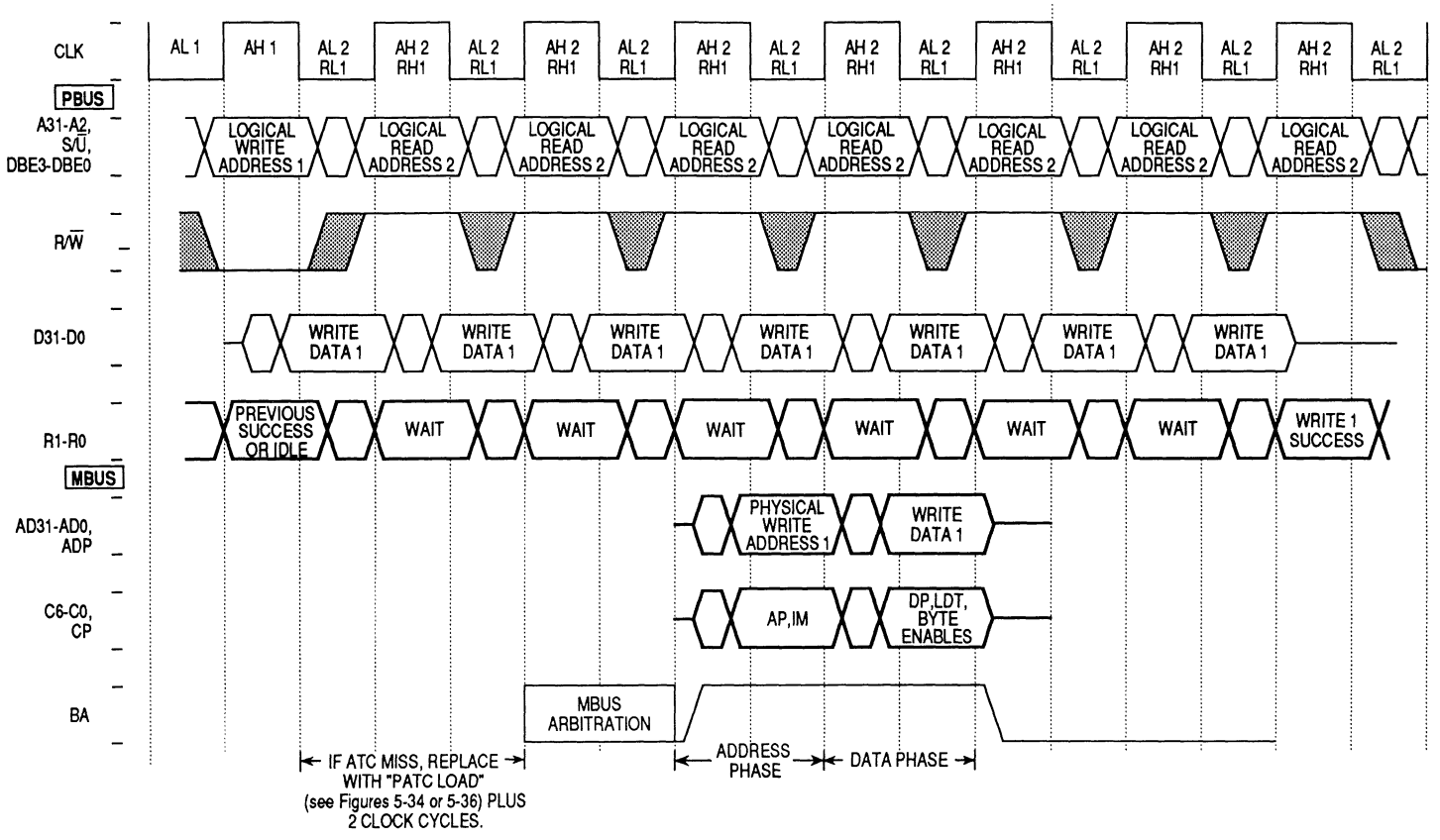


Figure 5-39. Write-Once or Write Through

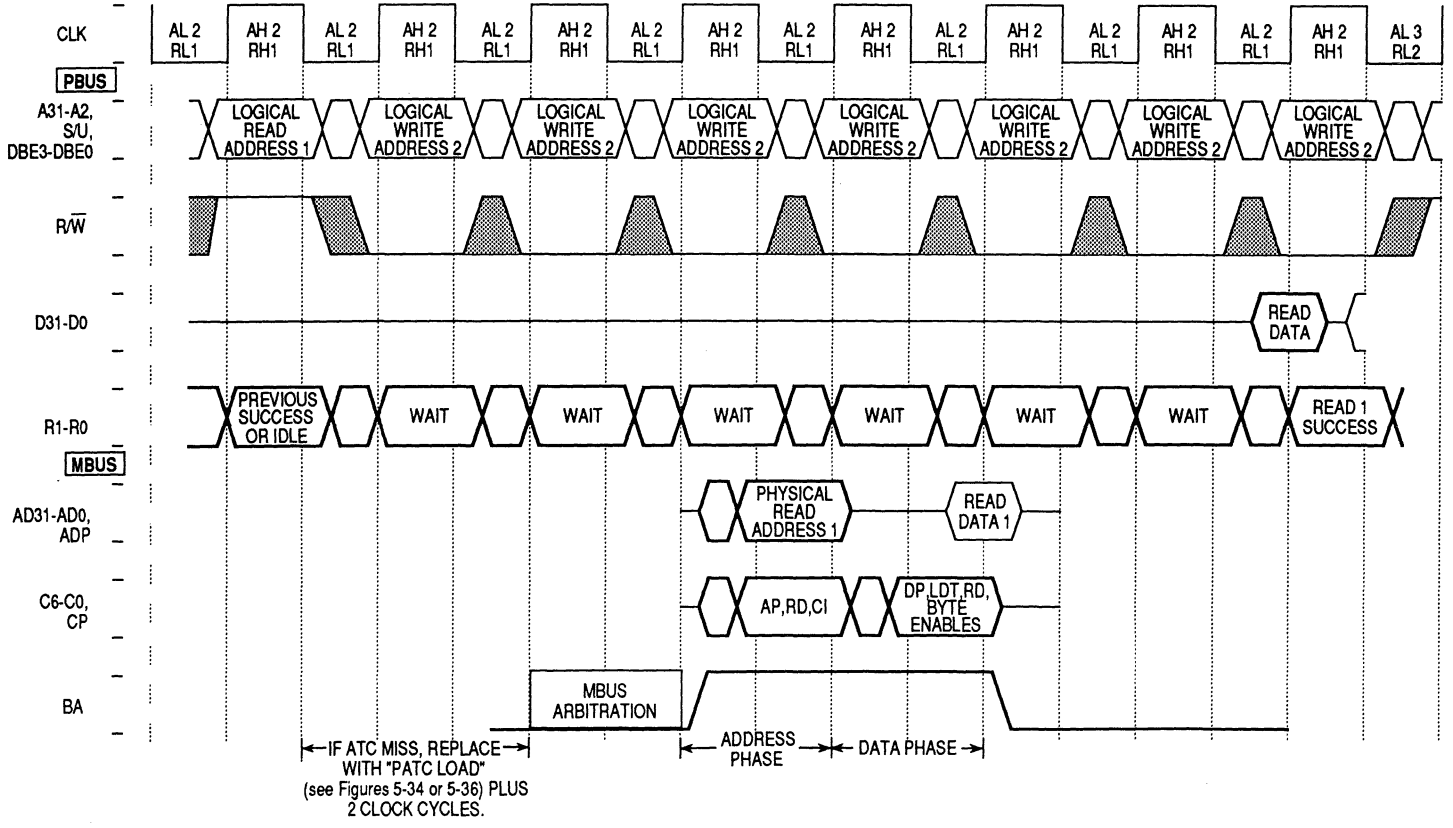


Figure 5-40. Data Cache Inhibit on Read

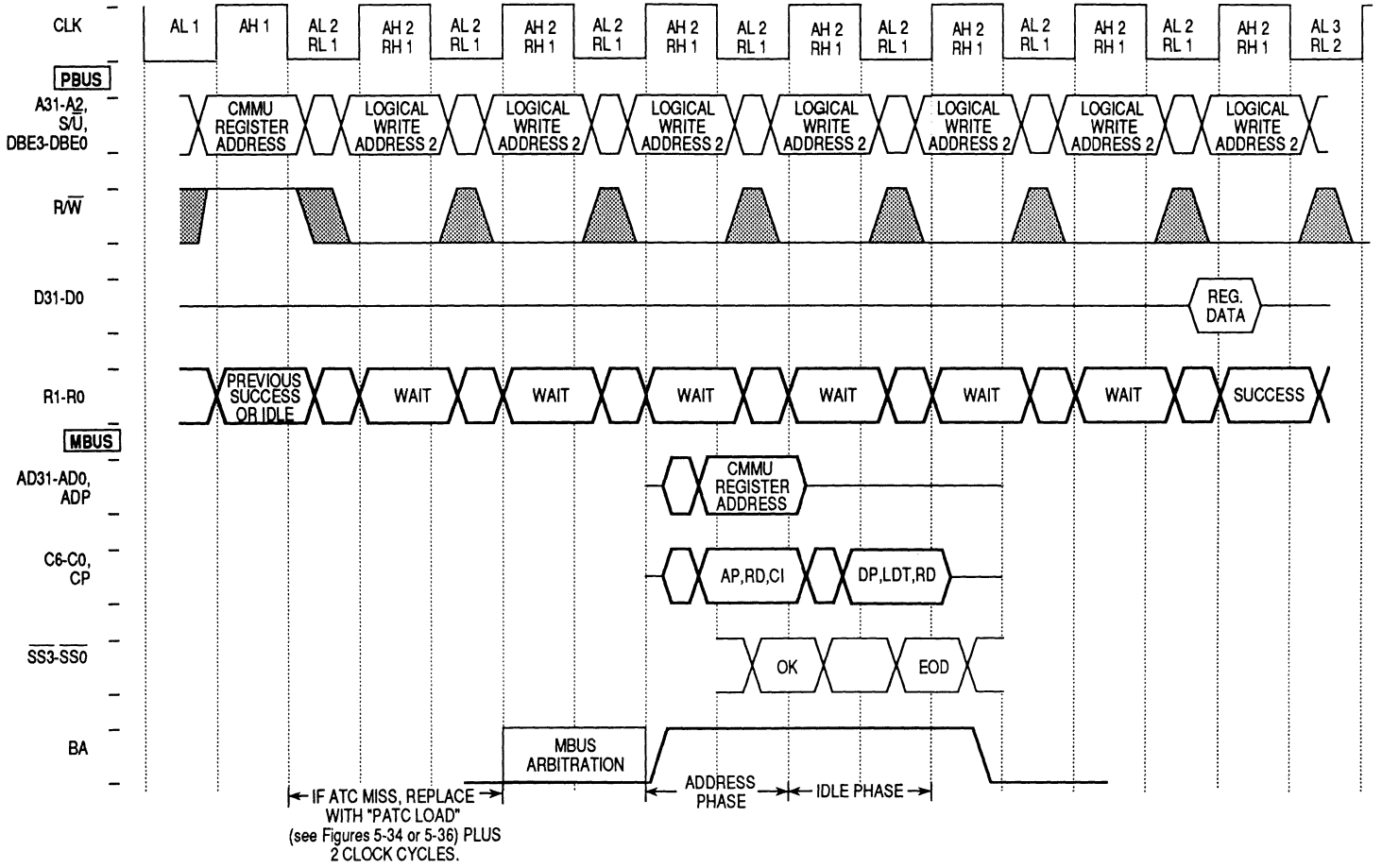


Figure 5-41. CMMU Local Register Read

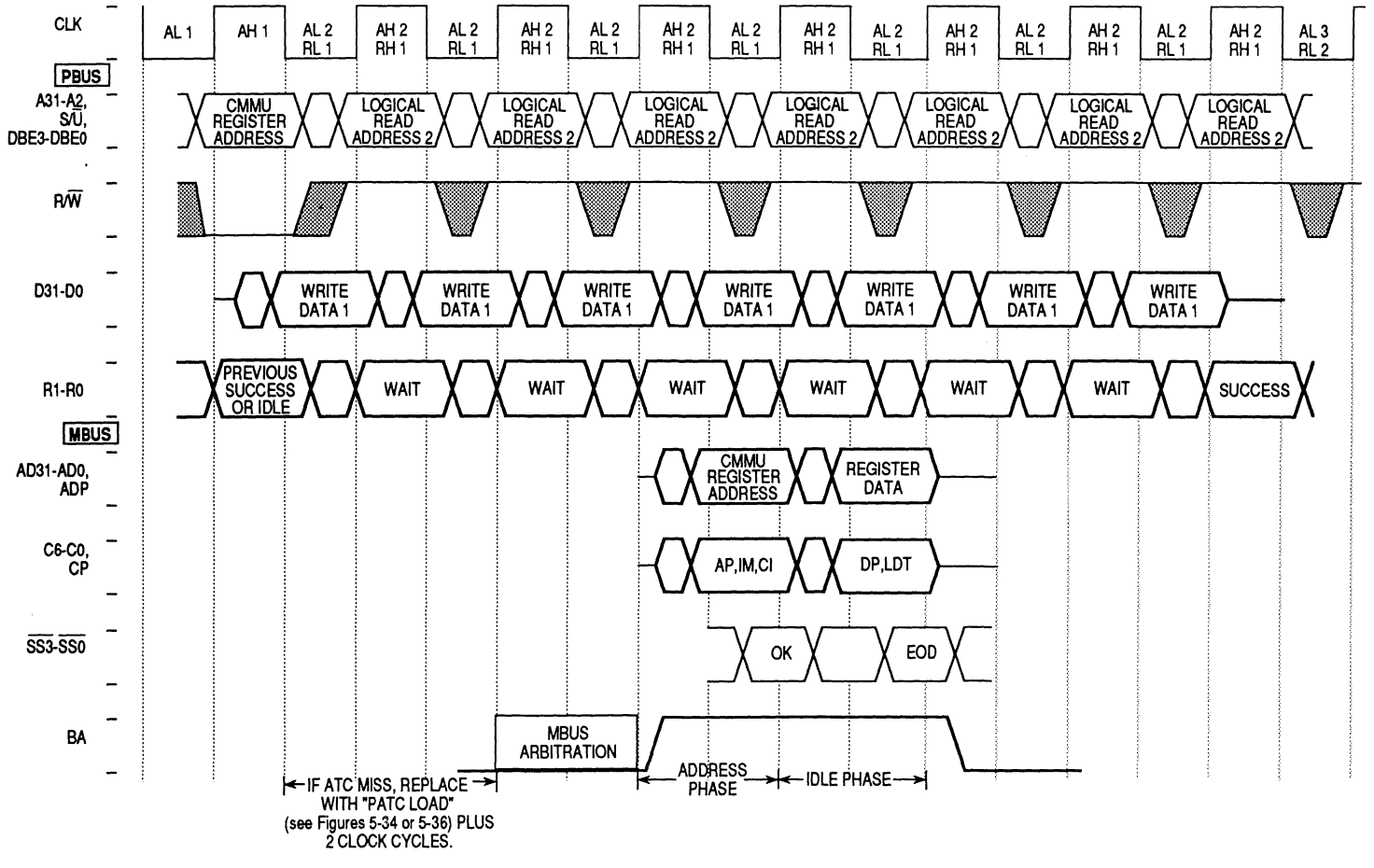


Figure 5-42. CMMU Local Register Write

in the control space during the address phase. All CMMUs on the M bus immediately latch the address to determine if they are the target (slave) of the M bus transaction. A CMMU is selected if bits A19–A12 of the M bus address match the 8-bit CMMU ID field of the ID register (IDR). The CMMU that is selected drives a wait on its local status for three clock cycles while it accesses the internal register. After the three wait clock cycles and after the master is in the data phase portion of the transaction, the remote CMMU slave drives data on the M bus for a read or latches data for a write. The CMMU drives EOD on its local status to terminate the transaction. If waits are driven on the system status by other non-CMMU devices, the slave CMMU keeps sourcing data (read) or sinking data (write) until it receives an OK or EOD on the system status. Timing for a remote register read is shown in Figure 5-43; timing for a remote register write is shown in Figure 5-44. If the slave MC88200 is selected from its P bus during the remote register access, the CMMU drives a wait P bus reply until the internal register access transaction is complete (see Figure 5-45).

Register accesses to a instruction CMMU go through a data CMMU and out onto the M bus; therefore, all instruction CMMU registers are accessed remotely. An M bus error or retry during a remote internal register access causes the master and slave CMMUs to abort the transaction.

5.6.7 xmem Sequence

Figure 5-46 illustrates a P bus and M bus sequence for the MC88100 exchange memory (**xmem**) instruction. The entire M bus operation, including the table search, occurs in one tenure to guarantee a locked access of the memory location. The following table is a functional chart of the **xmem** operation, indicating the signals of interest during the transaction.

Control Signals	M Bus Transactions					
	Read Segment Descriptor	**Read Page Descriptor**	**Write Page Descriptor**	**Copyback***	**Locked Read**	**Locked Write**
Lock (LK)	1	1	1	1	1	1
Intent to Modify (IM)	1	1	1	1	1	1
Cache Inhibit (CI)	0	0	0	0	1	1
Global (G)	0	0	0	0	—	—
Read (RD)	1	1	0	0	1	0
Address (AP)	[Segment	Page	[Page]]	[copybk]	**xmem**	**xmem**
Comments	Table Search (Optional)					

*Only if cache line exclusive modifed

NOTE:

Note 1 in Figure 5-46 indicates: If an ATC miss occurs, replace the bracketed area with "PATC LOAD" from "Read Hit, ATC Miss" (see Figure 5-34). If a copyback operation is necessary due to a data cache hit on modified data, also insert the cache copyback sequence from the read miss case (see Figure 5-37).

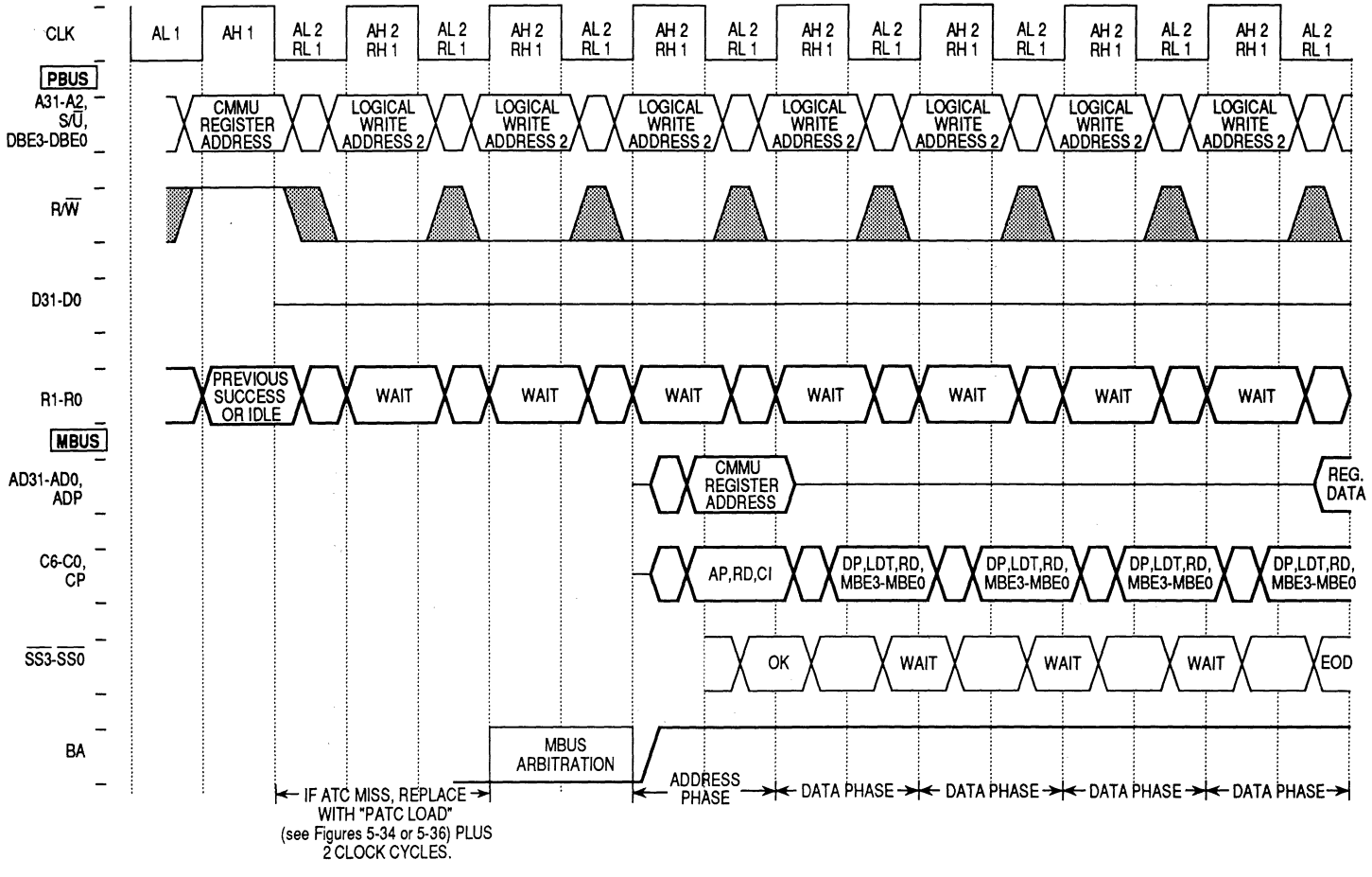


Figure 5-43. CMMU Remote Register Read (Sheet 1 of 2)

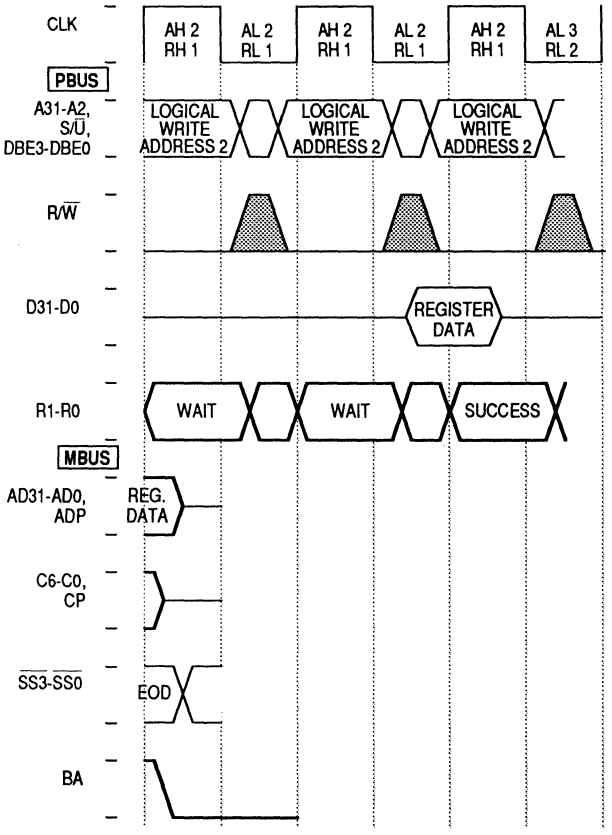


Figure 5-43. CMMU Remote Register Read (Sheet 2 of 2)

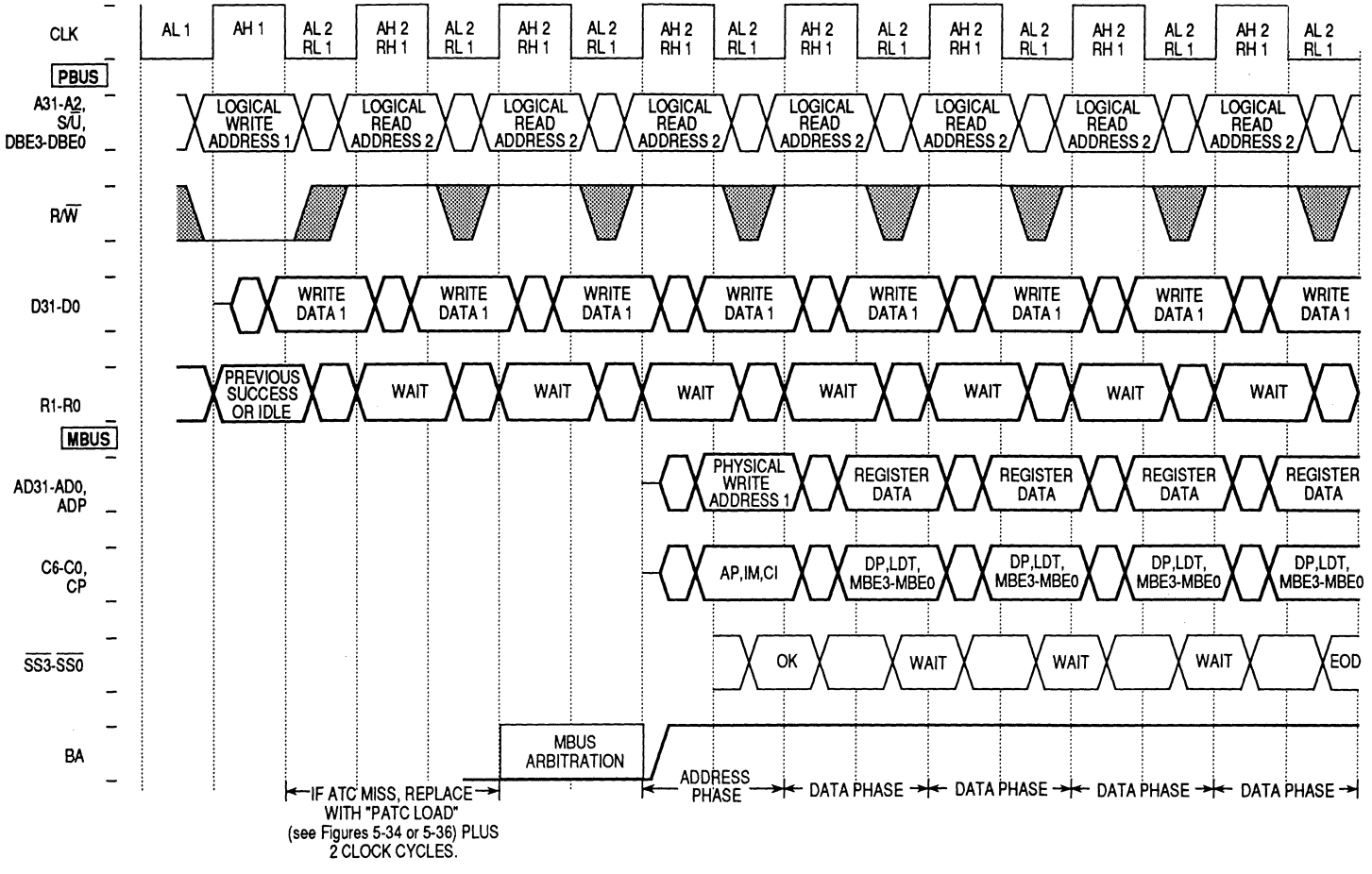


Figure 5-44. CMMU Remote Register Write (Sheet 1 of 2)

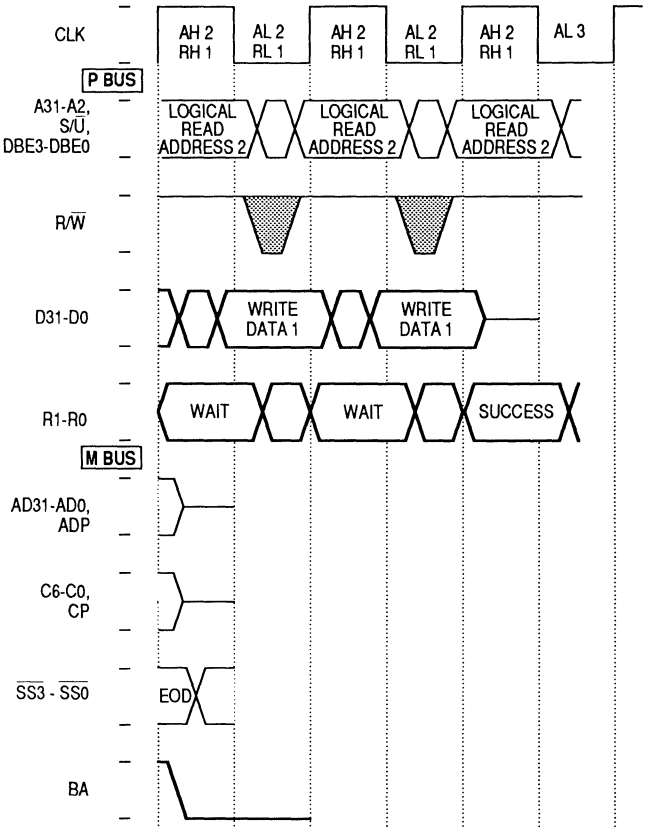


Figure 5-44. CMMU Remote Register Write (Sheet 2 of 2)

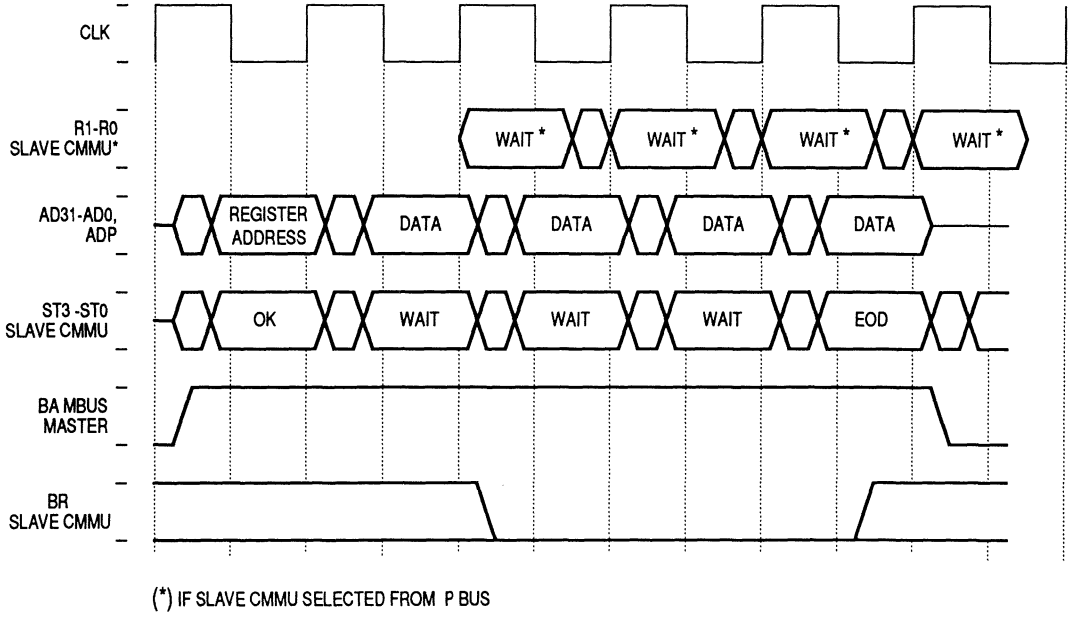


Figure 5-45. CMMU Slave Timing for Internal Register Access

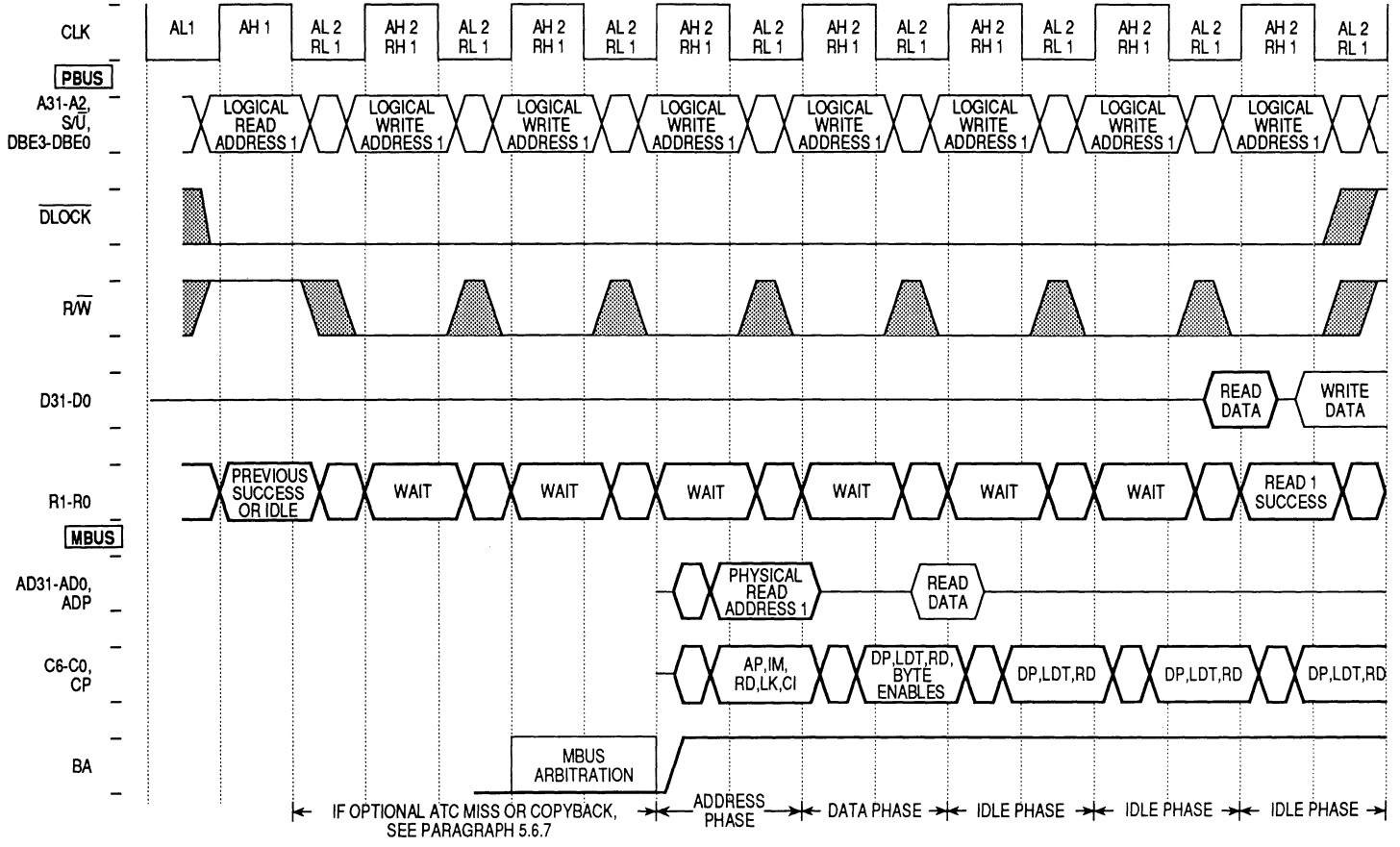


Figure 5-46. Processor xmem Sequence (Sheet 1 of 2)

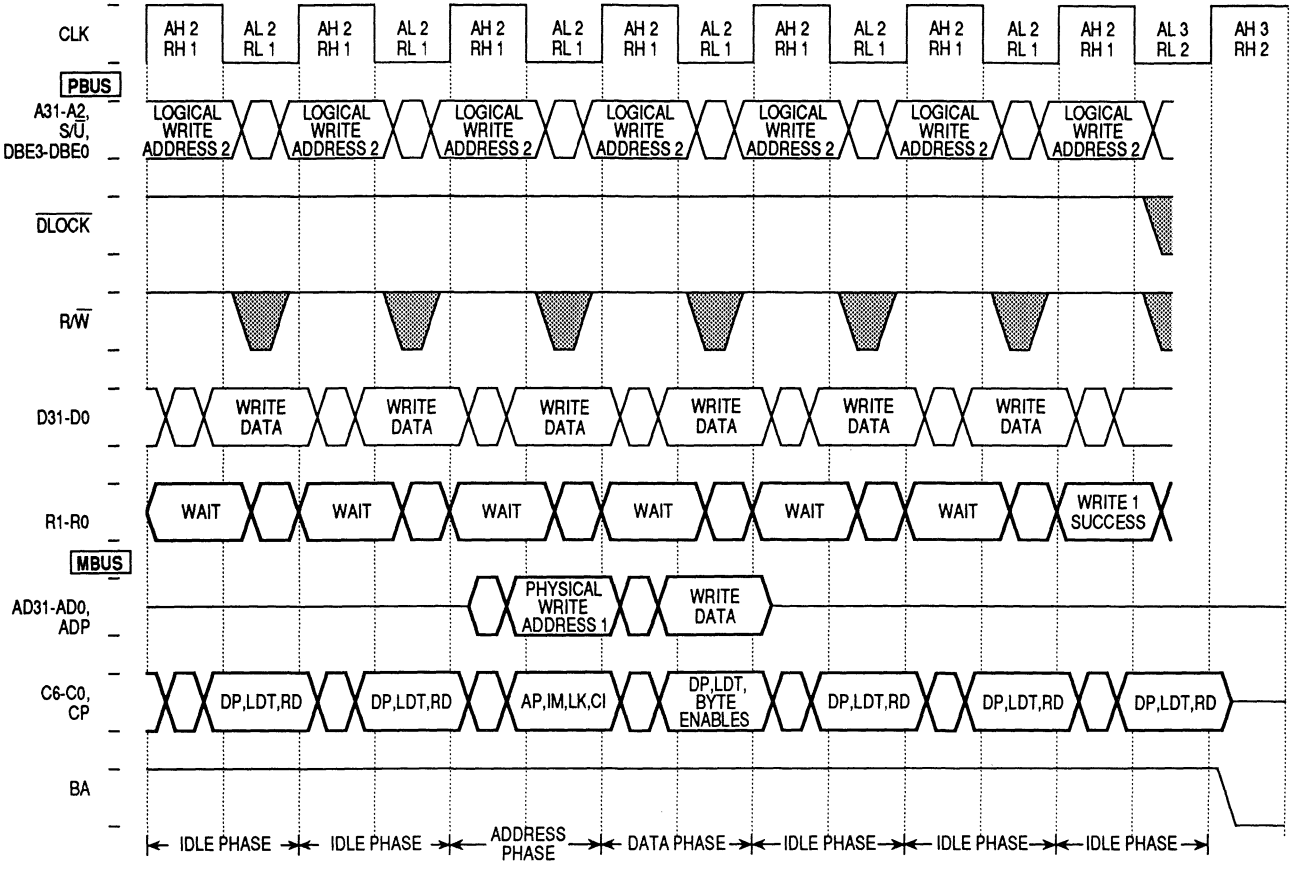


Figure 5-46. Processor xmem Sequence (Sheet 2 of 2)

5.6.8 M Bus Snooping

When a global transaction occurs on the M bus, all CMMUs (except the current M bus master) having snoop enabled (system control register, SE = 1) immediately suspend all other activity and snoop the transaction.

- P bus transactions in progress and any subsequent P bus transactions will be suspended with P bus WAITs.
- The snooping CMMU will assert two M bus waits beginning with the cycle following the first address phase. The WAITs will be issued in lieu of WAITs issued by any other M bus device.
- If the snooping CMMU "hits" on an exclusive modified entry, the CMMU responds with RETRY and writes back the dirty (exclusive modifed) entry. If an M bus error occurs, the copyback is terminated, and the line remains dirty. Functional timing is shown in Figure 5-47. The error is not reported to the P bus, but the CMMU's system status register (SSR) is updated (CE bit = 1, rest of register is not affected).
- The cache actions for other conditions (hits on shared unmodified or exclusive unmodified) remain as outlined in the cache state diagrams in **SECTION 3 DATA CACHE**. Functional timing for these cases is shown in Figure 5-48.

5

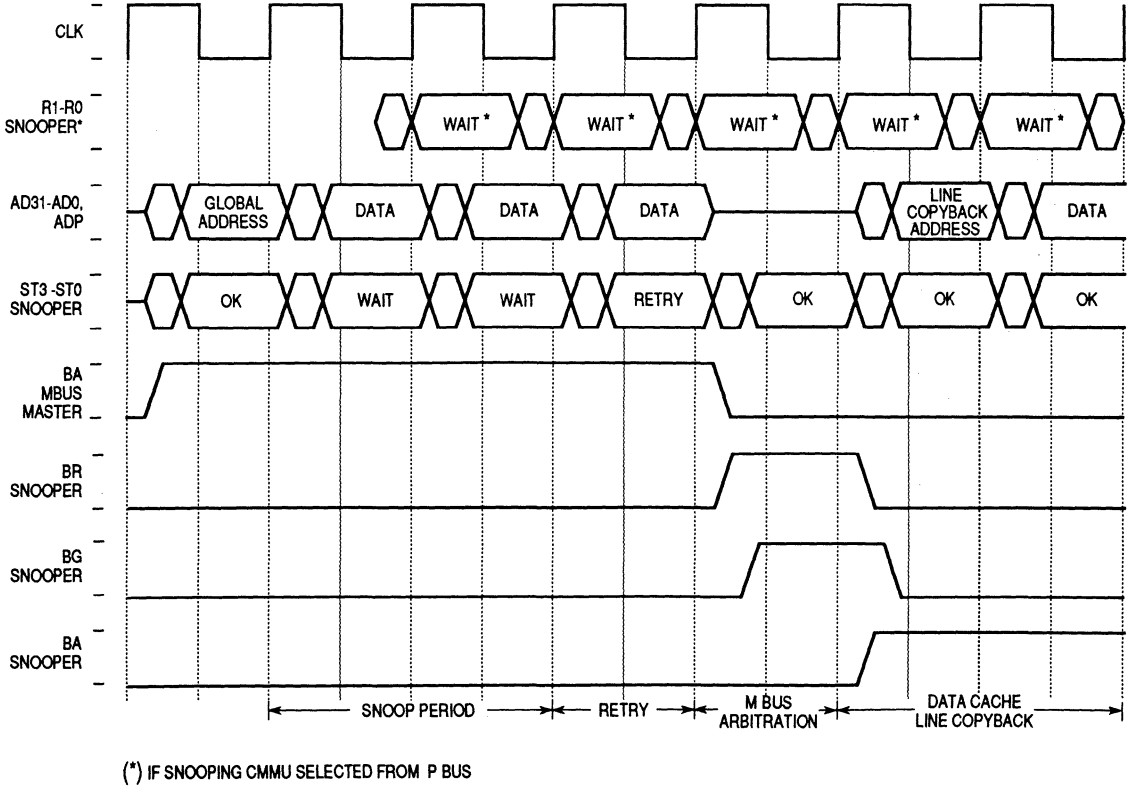
The following charts illustrate how waits generated from a snooping CMMU can be overlapped with waits generated from the M bus slave participating in the global transaction. The first case shows waits on the system status generated from the two snoop cycles, followed by the result of the snoop (either RETRY or OK). The second case shows the memory system initiating wait states starting on the address phase of the transaction. Since all M bus masters must drive a valid address on the address phase, the snooping CMMU latches the address and starts the snoop operation. The two snoop waits are overlapped with any additional waits needed by the memory system. At the end of the snoop period, the snooping CMMU signals a RETRY and forces the M bus master off the M bus to perform a snoop copyback. The third case shows the overlapped waits occurring on a snoop which does not hit on exclusive modified data. Figure 5-47 shows timing for a snoop hit with a copyback, and Figure 5-48 shows timing for a snoop miss or snoop hit or unmodified data (no copyback). If the processor selects a CMMU while it is snooping an M bus transaction, the CMMU drives wait replies on the P bus until the snoop operation completes.

Zero Waits from Memory

Phase	<addr>	<data1>	<data1>	<data1>
ST from Snooper	<OK>	<WAIT>	<WAIT>	<RETRY or OK>
$\overline{SS}$	<OK>	<WAIT>	<WAIT>	<RETRY or OK>

Snoop Hit with Memory WAIT on Address Phase

Phase	<addr>	<addr>	<addr>	<addr>
ST from Snooper	<OK>	<WAIT>	<WAIT>	<RETRY>
$\overline{SS}$	<WAIT>	<WAIT>	<WAIT>	<RETRY>



(*) IF SNOOPING CMMU SELECTED FROM P BUS

Figure 5-47. Snoop Hit on Exclusive Modified Data (Sheet 1 of 2)

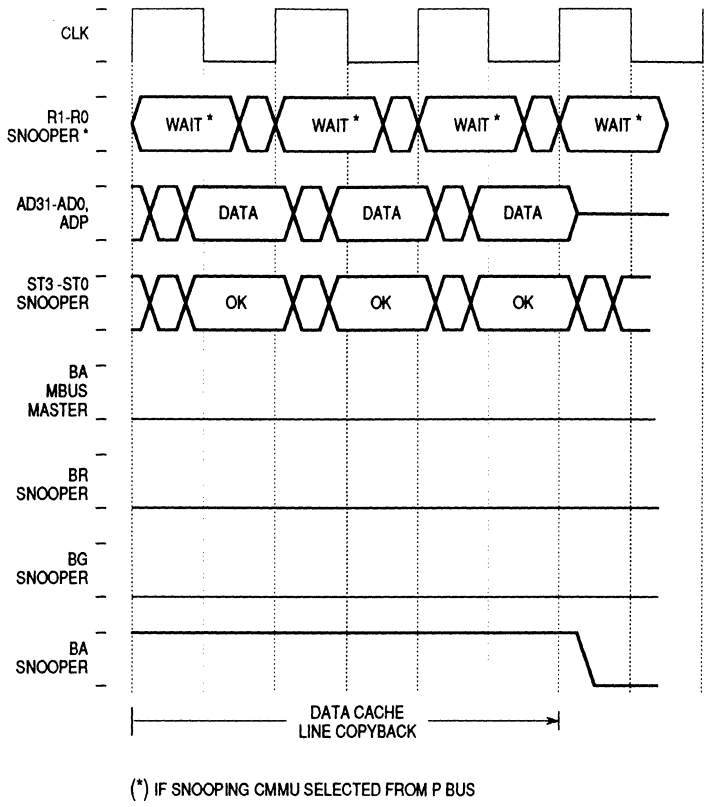
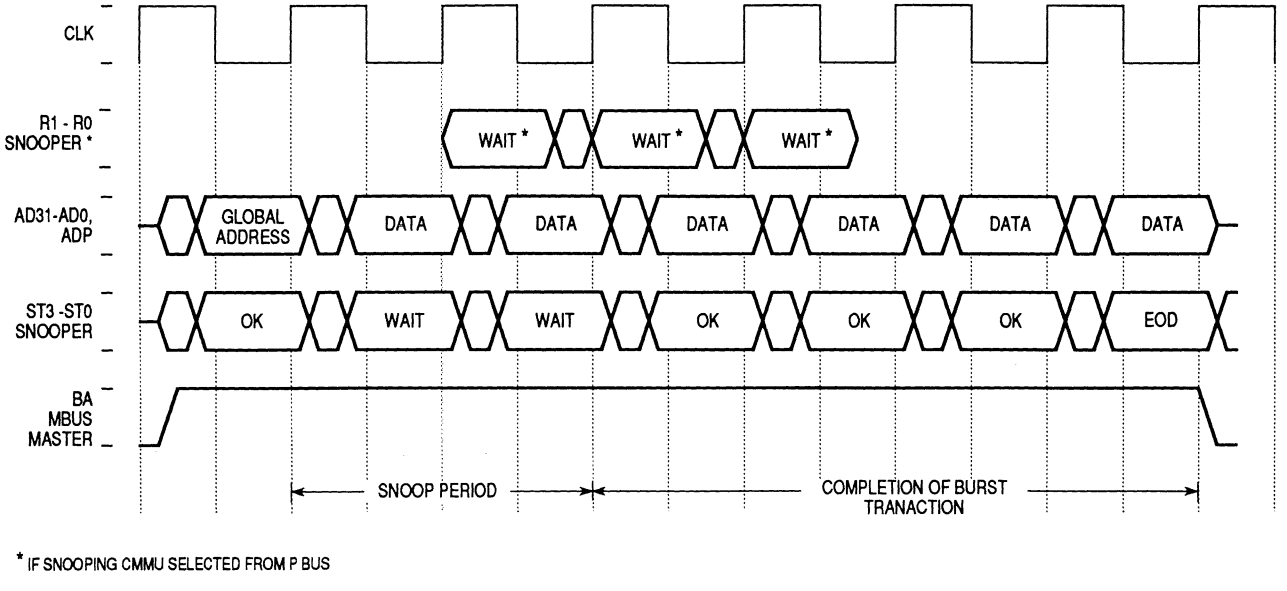


Figure 5-47. Snoop Hit on Exclusive Modified Data (Sheet 2 of 2)



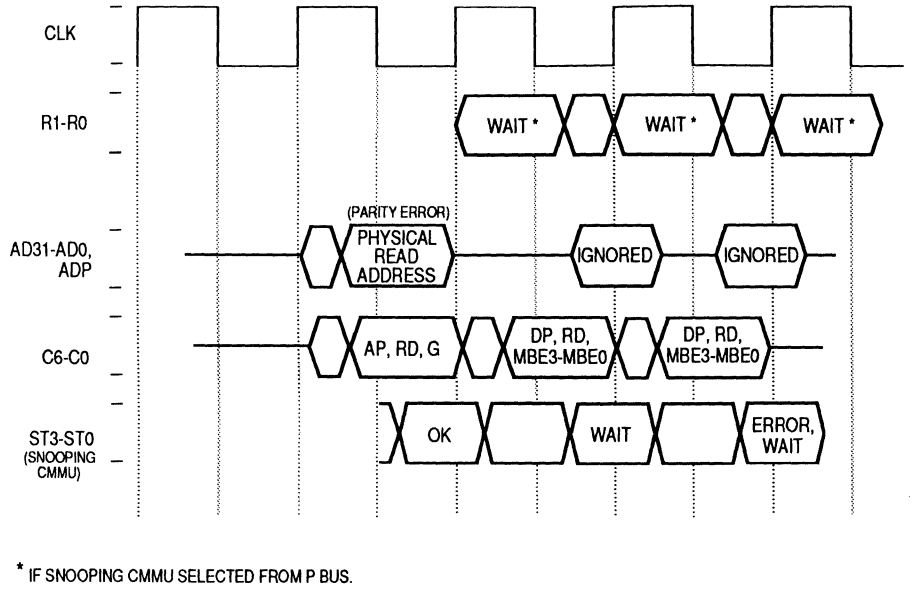
* IF SNOOPING CMMU SELECTED FROM P BUS

Figure 5-48. Snoop Miss or Snoop Hit on Unmodified Data

Snoop Miss with Memory WAIT on Address Phase

Phase	<addr>	<addr>	<addr>	<addr>	<data1>
ST from Snooper	<OK>	<WAIT>	<WAIT>	<OK>	<OK>
$\overline{SS}$	<WAIT>	<WAIT>	<WAIT>	<OK>	<OK>

Figure 5-49 illustrates a snooping CMMU detecting a parity error on the global address that hits in the internal data cache. M bus slaves (snoopers) only report parity errors by driving ST3 (ERROR) when they are the target of an M bus transaction.

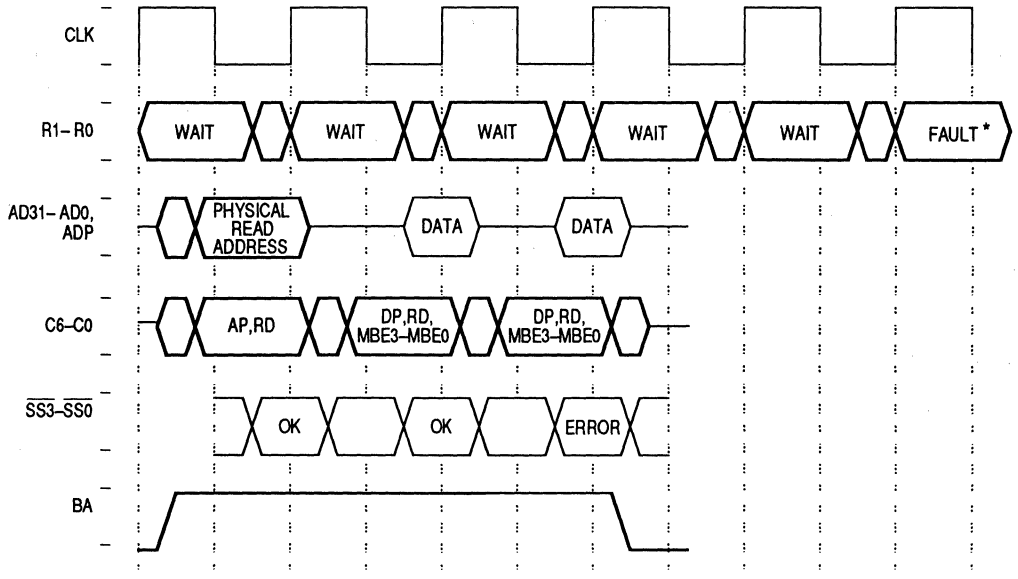


* IF SNOOPING CMMU SELECTED FROM P BUS.

Figure 5-49. Parity Error Detection by Snooping CMMU

5.6.9 M Bus Error/Retry

An M bus error or retry on the system status causes the current master and/or slave CMMU to immediately terminate the M bus transaction. The MC88200 master negates its bus acknowledge (BA) output and a slave MC88200 terminates its internal register access. Functional timing for a bus error or retry is shown in Figure 5-50. When a master MC88200 receives an M bus error that is a result of a P bus transaction (e.g., data cache miss, replacement copyback, write-once), the CMMU responds with a fault on the P bus reply. When the error is received as a result of a command (probe or data cache flush), the processor is not informed of the error. The BE bit of the SSR is set, and the processor must poll for this condition. When an error is received during a snoop copyback, the MC88200 leaves the cache line dirty and sets the copyback error (CE) bit of the SSR.



5

* IF M BUS ERROR NOT OCCURING ON DATA CACHE FLUSH, SNOOP COPYBACK, OR PROBE OPERATION

Figure 5-50. M Bus Error/Retry

Parity generation and checking (even parity) is controlled by the parity enable (PE) bit of the system control register (SCTR). Table 5-6 shows when parity generation and checking is finished by an MC88200 performing reads and writes in master or slave mode.

Table 5-6. Parity Generation and Checking

Parity Generation			
Mode	**Address**	**Data**	**Control**
Master-R	•		•
Master-W	•	•*	•
Slave-R		•*	
Slave-W			
Parity Checking			
Mode	**Address**	**Data**	**Control**
Master-R		•**	
Master-W			
Slave-R	•		•
Slave-W	•	•**	•

*Parity generated for all bytes
**Parity checked only for bytes with MBEx = 1

Two clock cycles after the offending address/data/control is detected, a parity error is reported by either the slave or snooping CMMU as a bus error. Figure 5-51 shows the functional timing for a parity error detected by a slave CMMU on the data phase. A snooping CMMU checks parity only on address and control. Figure 5-50 shows timing for a parity error on the address phase detected by a snooping CMMU.

A parity error is reported by the master CMMU in the same way as a bus error except for one case. When a parity error is detected by a master CMMU during a burst read transaction, the entire four words are read in before the fault is reported to the processor.

5.6.10 Cache Tag Monitoring

The MC88200 provides information about internal operations performed in the ATC and data cache as a result of P bus or M bus transactions. This information is provided via the tag monitoring (TM0–TM1) and trace (TR0–TR1) signals. The following paragraphs describe the cache tag monitoring functions.

5.6.10.1 HIT/MISS INFORMATION. The TM and TR signals can convey information about whether or not P bus transactions hit the address translation caches (ATCs) or data cache. When TM1, TM0 = 1 0, the TR pins indicate three cases of hit or miss information about the CMMU caches. Table 5-7 list the three cases of hit or miss information that the TR pins supply when TM1, TM0 = 1 0.

Table 5-7. Cache Hit/Miss Cases

Case	TM1	TM0	TR1	TR0	Hit/Miss Information
1	1	0	1	1	ATC Hit, Data Cache Hit
2	1	0	1	0	ATC Hit, Data Cache Miss
3	1	0	0	X	ATC Miss

The cache hit/miss signal status (TM1, TM0 = 1 0 and TR1, TR0 = x x) are only valid on the falling edge of the clock during the P bus reply phase of the transaction to which they refer. If the CMMU is busy servicing the M bus (due to snooping or having its internal registers accessed by the M bus master), the TM1, TM0 = 1 0 condition is delayed until the CMMU can service the P bus transaction. Figure 5-52 shows the functional timing for case 1, where both ATC and data cache contain the translation and data, respectively, for the P bus transaction. Figure 5-53 shows case 1 where the CMMU is busy snooping or servicing the M bus during the P bus transaction.

In case 2, where the translated address misses in the data cache, the TR pins indicate the ATC hit, data cache miss condition during the first wait reply to the processor. The CMMU arbitrates and acquires the M bus to fill the data cache. The transaction is then restarted

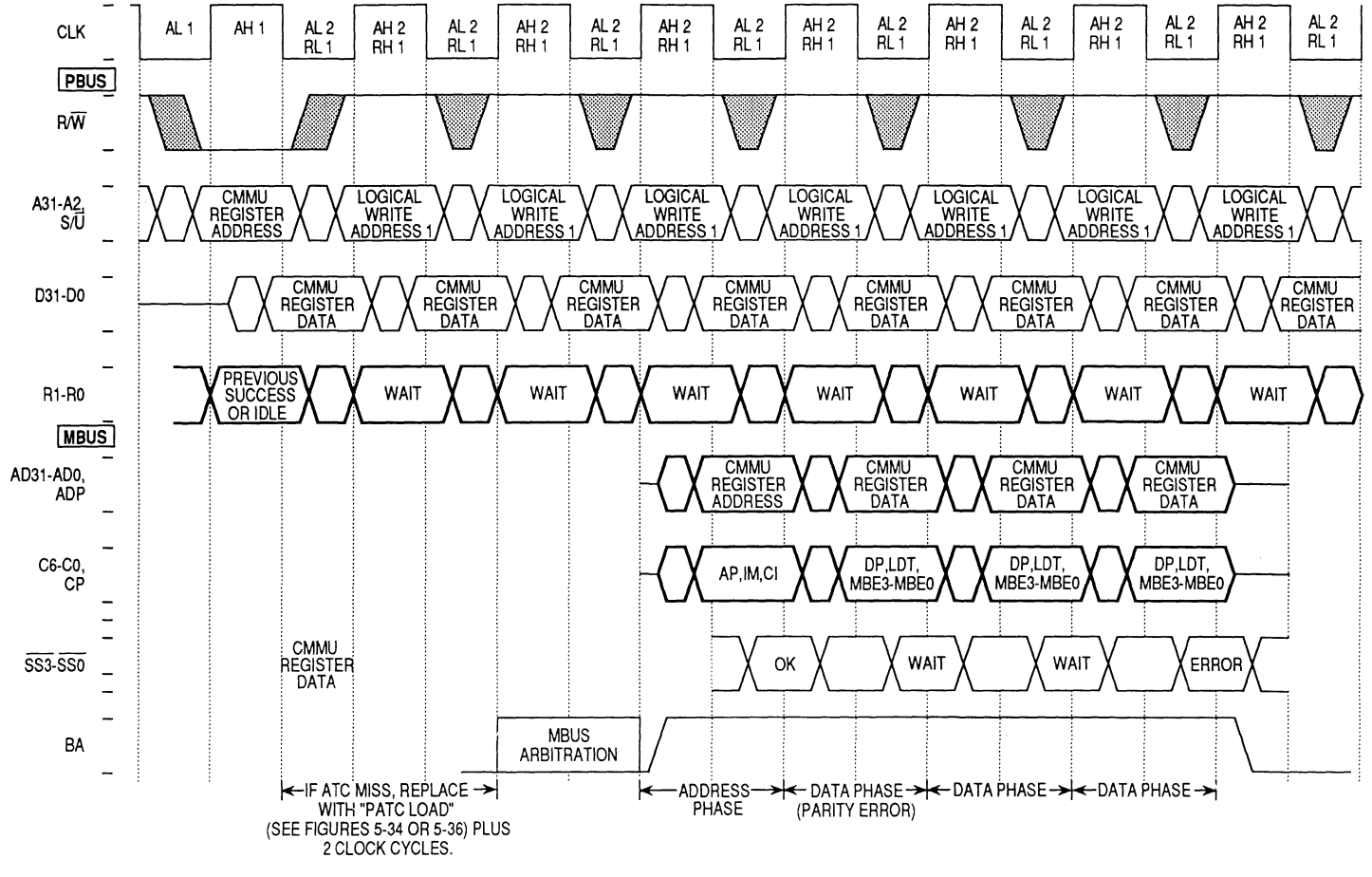


Figure 5-51. Parity Error Detection by Slave CMMU (Sheet 1 of 2)

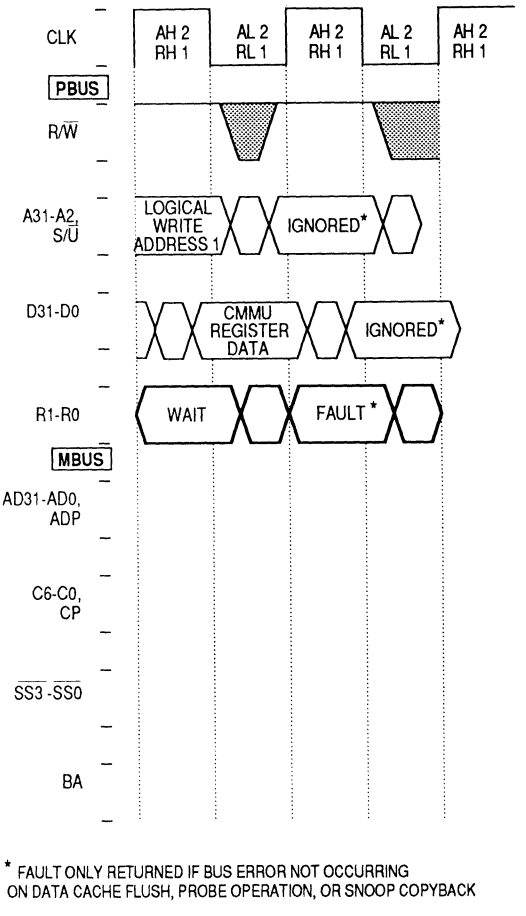


Figure 5-51. Parity Error Detection by Slave CMMU (Sheet 2 of 2)

from the latched logical address at the MC88200 P bus address inputs. This time, the data cache hits with the required data. Since the TM, TR signals have already identified this P bus transaction as an ATC hit cache miss case, no TM, TM0 = 1 0 strobe is driven. Functional timing for case 2 is given in Figure 5-54.

In case 3, where neither the BATC nor the PATC has the required logical to physical address translation (ATC miss), the CMMU must arbitrate and acquire the M bus to perform a table search operation. Following the table search, a new entry for the translation is created in the PATC. The data cache cannot be accessed when there is an ATC miss because no physical address is available for cache tag comparisons. Therefore, no hit/miss information can be given for the data cache when there is an ATC miss. After the table search operation,

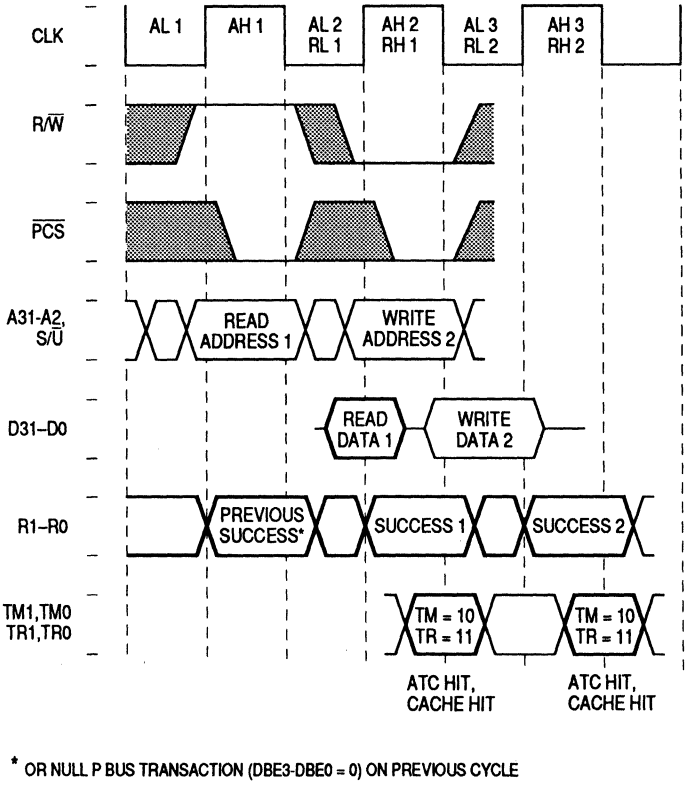


Figure 5-52. ATC Hit, Data Cache Hit Timing

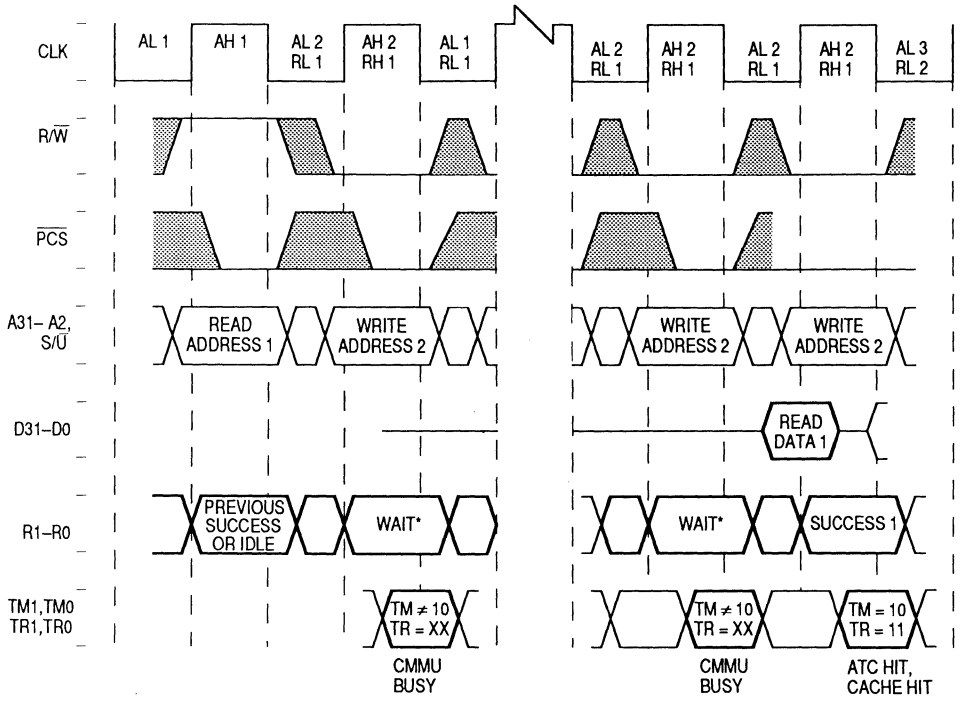
the transaction is restarted by the CMMU. This time the P bus transaction hits in the ATC, and the data cache is accessed with the physical address. If a cache hit occurs, the transaction is complete and no TM1, TM0 = 1 0 strobe occurs (see Figure 5-55).

If a cache miss occurs, then the CMMU continues according to case 2. Figure 5-56 shows functional timing associated with an ATC miss (case 3) followed by a data cache miss (case 2).

To accurately compute hit/miss ratios for the ATCs and data cache, one needs to count three events on the TM, TR, and P bus reply pins (see Table 5-8).

Transactions on the data P bus may be counted by observing the success reply on the R1, R0 pins as shown in Table 5-8.

Transactions on the instruction P bus are more difficult to count accurately. The instruction pipeline will stall if a data dependancy exists between two instructions (refer to the *MC88100*



* WAIT REPLY FROM CMMU WHEN BUSY ACTING AS AN M BUS SLAVE
• CMMU SNOOPING GLOBAL M BUS TRANSACTION
• CMMU SERVICING AN ACCESS TO ONE OF ITS INTERNAL REGISTERS BY THE M BUS MASTER

Figure 5-53. ATC Hit, Data Cache — CMMU Busy

Table 5-8. Hit/Miss Transactions

Event	TM1, TM0	TR1, TR0	R1, R0
P Bus Transactions (PBT)	XX	XX	10 (Success Reply)
ATC Misses (ATCM)	10	0X	XX
Data Cache Misses (DCM)	10	X0	XX

User's Manual). Stalling implies that the MC88100 will fetch the same instruction repetitively until the scoreboard bit associated with the destination register of previously decoded instruction is cleared. To reduce the affect of pipeline stalling, instructions should be mapped as cache inhibited (CI = 1) in the page descriptors or BATC entries used for instruction areas of memory. In this manner, every instruction fetch will take 7 + MW + clock cycles, where

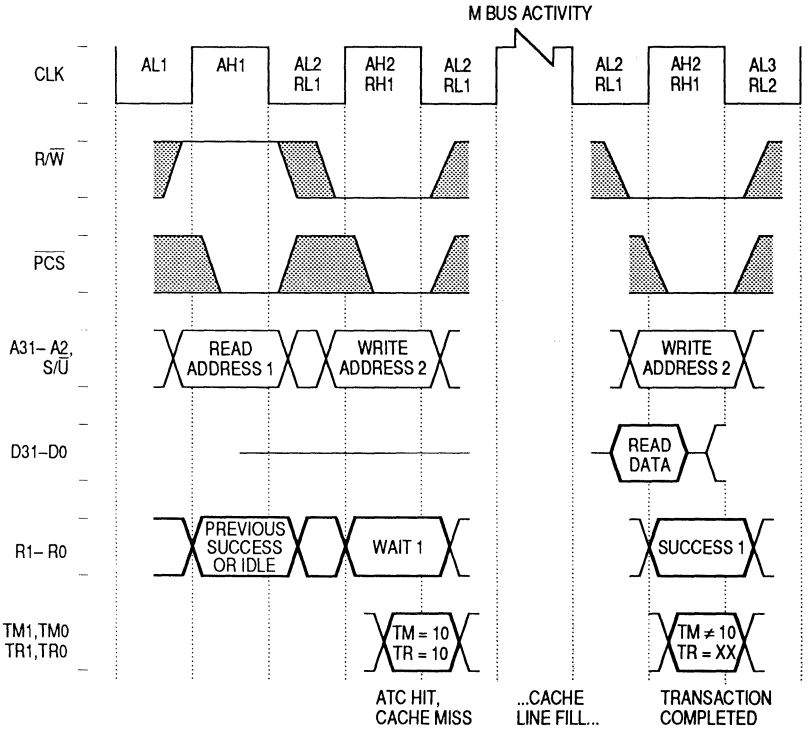


Figure 5-54. ATC Hit, Data Cache Miss Timing

MW = clocks for memory wait (refer to **6.1 OPERATION TIMING** for further information). Most pipeline stalls will have resolved by the time a cache inhibited instruction access to memory completes.

To eliminate the effects of instruction stalling, one compares P bus code addresses to detect two identical addresses. This condition indicates an instruction stall and should not be counted in the total number of P bus transactions.

Once the effects of pipeline stalls have been accounted for, hit/miss ratio statistics for the MC88200 caches are calculated as follows:

ATC Hit/Miss Ratio = (PBT — ATCM)/PBT
Data Cache Hit/Miss Ratio = (PBT — DCM)/PBT

5.6.10.2 TAG MONITORING INFORMATION. The TM and TR signals also provide information about the status of address tags in the MC88200 data cache during M bus transactions. The TM signals indicate the effect any bus address (from the CMMU) has on the tag. The TR signals indicate which line in the cache set is being accessed.

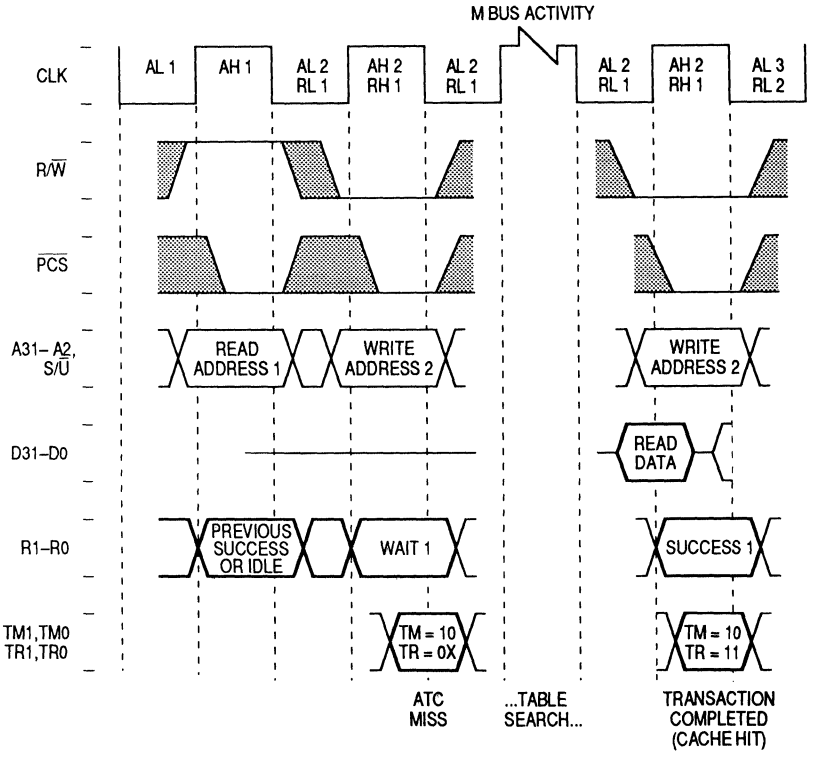


Figure 5-55. ATC Miss, Cache Hit Functional Timing

Tags **are loaded** when a data cache line is loaded as a result of a read or write miss in the CMMU data cache. The CMMU updates the status bits for the tag and line at the end of the M bus transaction, assuming no M bus error or retry condition aborts the transaction.

Tags **remain loaded** during certain M bus writes of data that are not cache inhibited. These cases are 1) an M bus write-once while in copyback mode (WT = 0), 2) an M bus write while in writethrough mode (WT = 1), and 3) a cache line copyback due to a cache flush operation (copyback-without-invalidation command).

Tags **become invalid (purged)** due to 1) a line copyback of exclusive modified data due to a snoop hit on a global address with intent to modify set (IM = 1) on the M bus control or 2) a line copyback of exclusive modified data due to a cache flush operation using the copyback with-invalidation command

Tags are **not affected** during 1) cache inhibited (CI = 1) accesses, including CMMU internal register accesses, or 2) table search operations

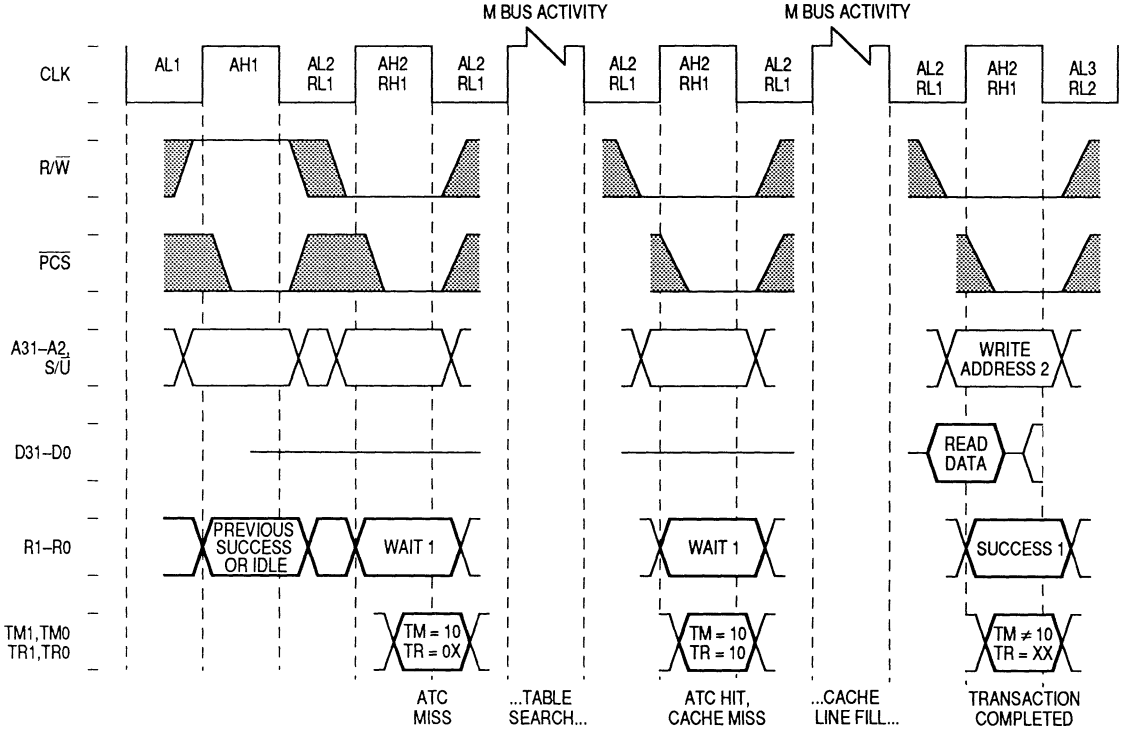


Figure 5-56. ATC Miss, Cache Miss Functional Timing

However, accesses mapped as cache inhibited that hit in the data cache will invalidate the tag that caused the hit. The tag invalidation (VV ⧫ 11) does not cause a copyback if the tag is marked exclusive modified.

All cases shown above are observable on the M bus as M bus transactions. During the address phase of a transaction, the TM signals indicate the effect of the transaction on the address tag, and the TR signals indicate which tag in the set is affected. Decoding of the TM and TR signals is shown in Table 4-6 and functional timing is shown with M bus master timing in **SECTION 8 ELECTRICAL CHARACTERISTICS**. For tag information the TM and TR signals are only guaranteed during the address phase of an M bus transaction.

5.7 RESET TIMING AND PHASE LOCKING

The following paragraphs describe reset and phase locking operations of the MC88200. Reset processing forces the MC88200 into a predefined initial state. No pending or partially completed transactions are retained, and signals enter predefined states. After the MC88200 is reset, P bus (logical) addresses are passed to the M bus without translation, and no data is cached. At this point, system software must initialize the on-chip data cache, initialize the translation tables in memory, and load the area descriptor registers.

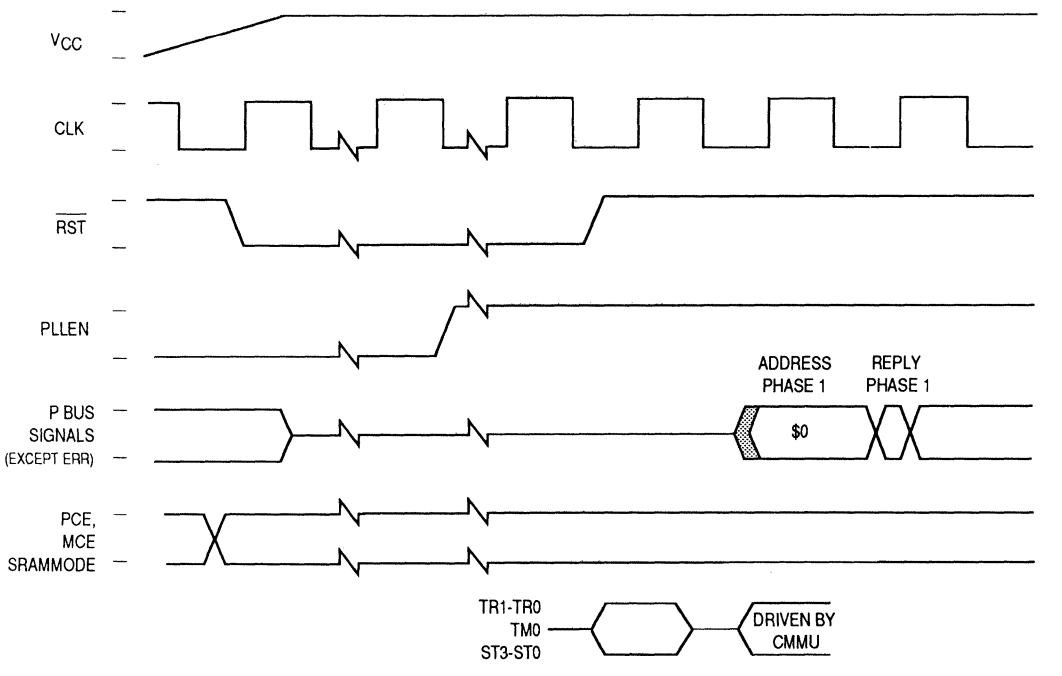
5.7.1 Reset Operation

To guarantee that the operation of multiple M88000 devices is completely synchronous, the reset ($\overline{RST}$) and phase locked loop enable (PLLEN) signals for all devices in the system must meet the setup and hold times specified in **SECTION 8 ELECTRICAL CHARACTERISTICS** for the same falling edge of the input clock signal (the clock signal must be shared). The same setup and hold time requirements must be met for both the assertion and negation of the $\overline{RST}$ and PLLEN signals for all devices in the system.

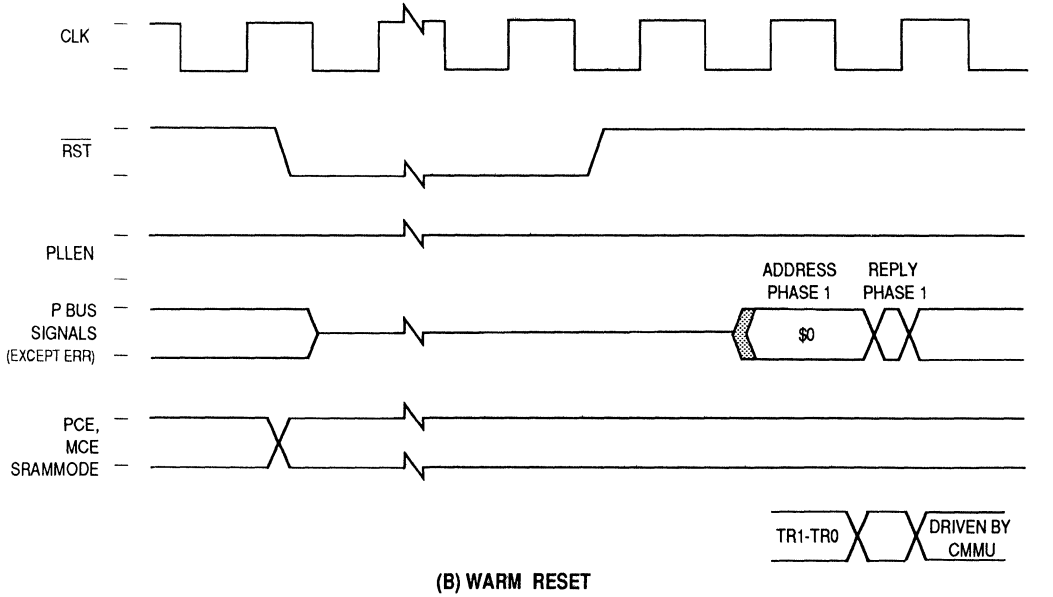
During the time that the $\overline{RST}$ signal is asserted on powerup, all output signals are placed in the high-impedance state except the error (ERR) signal, bus request (BR), and bus acknowledge (BA). The ERR signal is driven low (negated) while the MC88200 is in the reset state. Refer to **SECTION 4 SIGNAL DESCRIPTION** for a list of signal states during reset. For detailed information on the state of the MC88200 registers after reset, refer to **SECTION 6 OPERATIONS TIMING AND REGISTER DESCRIPTIONS**.

Figure 5-57 (a) shows the timing relationships on the P bus for the first instruction prefetch after the negation of reset on powerup. The bus signals remain in the high-impedance state for one clock after $\overline{RST}$ is detected as negated. The MC88200 CMMU automatically ignores the bus for one clock after the negation of $\overline{RST}$.

Figure 5-57 (b) shows the timing relationships for a reset after the system has powered up and has already been phase locked (warm reset). The only difference between a warm reset and powerup reset is the assertion length requirements of $\overline{RST}$. Because the phase-locked loop has presumably been properly initialized, $\overline{RST}$ must only be asserted for a



(A) POWERUP RESET



(B) WARM RESET

Figure 5-57. Reset Timing

minimum of eight clock cycles to ensure a full reset. The state of the internal registers and the operation of the buses after the warm reset is the same as that of the powerup reset.

Hardware reset begins when the $\overline{RST}$ signal is asserted. Upon detection of $\overline{RST}$, the MC88200 performs the following actions:

1. All address translation and data caching stops. Signals enter the states listed in Figure 5-1.
2. All PATC and BATC entries are invalidated.
3. The area pointer registers inhibit caching and address translation when the MC88200 begins operation. The P bus (logical) addresses are mapped directly to the M bus without translation.
4. As the $\overline{RST}$ signal is negated, the CMMU ID number is read to set the control register base address. (See **5.7.3 ID Initialization**). The PCE and MCE signals are also latched at this time.

5.7.2 Phase-Locked Loop Operation

The MC88200 is designed to operate completely synchronously with other devices in the M88000 system. In this way, all devices communicating on the P bus and M bus can make assumptions about the validity of signals with respect to the master clock. The signals can be used by the receiving device as soon as they are valid from the device driving them. To provide a tight tolerance on the relationship between the input clock and the output signals, the internal clock is derived on-chip by a digital phase-locked loop circuit that uses the input clock as its reference. Therefore, fabrication variations between multiple devices do not cause differences in timing delays that might otherwise be induced in the internal clock circuity. The phase-locked loop circuit of the MC88200 is also implemented on the MC88100 processor, which allows multiple M88000 devices to reside on the P bus with tightly coupled timing relationships between them. Since the clock signal is used as a reference for the phase-locked loop, care should be taken in the layout and routing of the clock signal to minimize propagation delays induced by transmission line effects of board traces.

To initialize the phase-locked loop circuit on powerup, the PLLEN signal must be asserted with the appropriate timing in relationship to the $\overline{RST}$ signal (see Figure 5-57 (a)). The combination of the $\overline{RST}$ and PLLEN requirements (see Figure 5-57) provide sufficient time for the phase-locked loop to be reset and for the internal clock to phase lock to the external clock on powerup. The PLLEN signal must subsequently remain stable and asserted to guarantee proper phase-locked operation. In addition, these requirements also ensure that the remainder of the MC88200 is properly reset.

NOTE

Unless the MC88200 is properly phase-locked, the AC specifications described in **SECTION 8 ELECTRICAL CHARACTERISTICS** cannot be guaranteed. Nonphase locked operation is **not** recommended.

5.7.3 ID Initialization

The ID field in the CMMU ID register defines the base address of the MC88200 registers in the control memory space. This field is initialized by hardware; the MC88200 reads seven M bus signals as the $\overline{RST}$ signal is negated; bit 31 is initialized to zero by the internal reset circuit:

ID Bit:	31	30	29	28	27	26	25	24
Signal:	(0)	ST0	ST1	ST2	ST3	TM0	TR0	TR1

These signals, which are the ID inputs when $\overline{RST}$ is asserted, have other uses when $\overline{RST}$ is negated. The entire 8-bit field can be reprogrammed in software.

SECTION 6
OPERATIONS TIMING AND REGISTER DESCRIPTIONS

This section provides information to compute clock cycle times for data cache flushes and memory bus (M bus) activity. A general overview of the MC88200 registers is also provided. Detailed register information can be found in the paragraphs that pertain to each register.

6.1 OPERATIONS TIMING

Tables 6-1 and 6-2 contain data for calculating bus cycle timing and data cache operation timing.

The data cache flush cycle counts in Table 6-1 refer to the minimum number of clocks required to flush the MC88200 data cache. This count does not include overhead associated with the operation, which may be up to 10 clock cycles. Data cache flushing is described in **SECTION 3 DATA CACHE**.

Table 6-1. Data Cache Flush Clock Count

Function	Clock Cycles
Data Cache Invalidation Process	
Line	1
Page	256
Segment	1024
All	256
Data Cache Copyback Process	
Line	1 + n(SCB) Where n = 0,1 (number of entries marked EM)
Page	256 + n(SCB) Where n = 0 – 256 (number of entries marked EM)
Segment	1024 + n(SCB) Where n = 0 – 1024 (number of entries marked EM)
All	1024 + n(SCB) Where n = 0 – 1024 (number of entries marked EM)

NOTE: SCB (simple copyback) = 7 + (memory waits) + (arbitration delays); EM = exclusive modified.

Table 6-2 contains cycle counts necessary to calculate M bus activity due to table searches, data cache line fills and copybacks, cache inhibited accesses, and probes.

6.2 MC88200 REGISTERS

The MC88200 contains four types of registers as shown in Figure 6-1. These are system interface, processor bus (P bus) fault, block address translation cache (BATC) write port, and cache diagnostic port registers. The area pointer registers are part of the system interface registers. The MC88200 registers occupy one page (4K bytes) of the control memory address space. The register base address is formed using a programmable 8-bit device identification (ID) field in the cache memory/management unit (CMMU) ID register. Since

Table 6-2. M Bus Activity Cycle Count

Function	Clock Cycles
Cache Write Miss	14 + MW + (SCB If Necessary)
Cache Read Miss	10 + MW + (SCB If Necessary)
Table Search (No U/M Update)	11 + (2*MW)
Table Search (With U/M Update)	15 + (2*MW)
Table Search (Violation in Segment Walk)	7 + MW
Table Search (Violation in Page Walk)	11 + (2*MW)
Cache Inhibited Read Access	7 + MW
Cache Inhibited Write Access	7
Write Once	7
SCR Write (No Op)	7
IRA Write	7
Probe ATC Hit	PIRA + 3
Probe ATC Miss	PIRA + 2 + Probe Table Search
Probe Table Search No U/M Update With U/M Update Invalid Segment Descriptor Invalid Page Descriptor	 11 + (2*MW) 14 + (2*MW) 6 + MW 10 + (2*MW)

NOTE: SCB (simple copyback) = 7
MW (memory wait) = 1 (this is common for all but fast static RAMs)
PIRA (normal processor IRA) = 6
IRA — internal control register access
ATC — address translation cache
U/M — used and/or modified bit in page descriptor
SCR — system command register

the device ID is programmable, MC88200s can be dynamically reconfigured by either P bus or (M bus) masters.

The M bus byte enable signals (MBE3–MBE0) are not asserted during local CMMU register accesses. During a register read, any reserved bits are returned to zeros. For exceptions to this, refer to **6.2.1.2 SYSTEM COMMAND REGISTER** and **6.2.1.5 SYSTEM CONTROL REGISTER**. Values written to reserved bits are ignored by the MC88200. Table 6-3 lists the status of the registers during reset.

6.2.1 System Interface Registers

The system interface registers provide control and status registers for various MC88200 operations (probe memory location, flush cache, etc.). In addition, they include the two area pointers that define the supervisor and user memory areas.

SYSTEM INTERFACE REGISTERS

BASE + $000	IDR	CMMU ID REGISTER
BASE + $004	SCR	SYSTEM COMMAND REGISTER
BASE + $008	SSR	SYSTEM STATUS REGISTER
BASE + $00C	SAR	SYSTEM ADDRESS REGISTER
BASE + $104	SCTR	SYSTEM CONTROL REGISTER

P BUS FAULT REGISTERS

BASE + $108	PFSR	P BUS FAULT STATUS REGISTER
BASE + $10C	PFAR	P BUS FAULT ADDRESS REGISTER

AREA POINTERS

BASE + $200	SAPR	SUPERVISOR AREA POINTER REGISTER
BASE + $204	UAPR	USER AREA POINTER REGISTER

BATC WRITE PORTS *

BASE + $400	BWP0	BLOCK ATC WRITE PORT 0
BASE + $404	BWP1	BLOCK ATC WRITE PORT 1
BASE + $408	BWP2	BLOCK ATC WRITE PORT 2
BASE + $40C	BWP3	BLOCK ATC WRITE PORT 3
BASE + $410	BWP4	BLOCK ATC WRITE PORT 4
BASE + $414	BWP5	BLOCK ATC WRITE PORT 5
BASE + $418	BWP6	BLOCK ATC WRITE PORT 6
BASE + $41C	BWP7	BLOCK ATC WRITE PORT 7

CACHE DIAGNOSTIC PORTS**

BASE + $800	CDP0	CACHE DATA PORT 0
BASE + $804	CDP1	CACHE DATA PORT 1
BASE + $808	CDP2	CACHE DATA PORT 2
BASE + $80C	CDP3	CACHE DATA PORT 3
BASE + $840	CTP0	CACHE TAG PORT 0
BASE + $844	CTP1	CACHE TAG PORT 1
BASE + $848	CTP2	CACHE TAG PORT 2
BASE + $84C	CTP3	CACHE TAG PORT 3
BASE + $880	CSSP	CACHE SET STATUS PORT

BASE = FFFii000, where ii = 8-bit CMMU ID from CMMU ID Register
* Address bit A5 is not decoded for BATC write ports
** Address bits A4 and A5 are not decoded for cache diaagnostic ports

Figure 6-1. MC88200 Registers

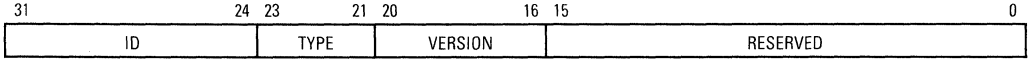
Table 6-3. Register Reset Status

Register	State During Reset
CMMU ID Register	Bits 31–24 receive the CMMU ID; bits 23–21 contain the CMMU type identified; bits 20–16 contain the version number; bits 15–0 clear.
System Command Register	Bits 5–0 (command field) clear; all other bits unused (clear).
System Status Register	Bits 15 and 14 (copyback error and bus error) clear; bits 9–0 (probe status) clear; all other bits unused (clear).
System Address Register	Undefined.
System Control Register	Bits 15–13 clear (parity disabled, snoop disabled, and fairness arbitration, respectively); all other bits unused (clear).
P Bus Fault Status Register	Bits 18–16 (fault code) clear; all other bits unused (clear).
P Bus Fault Address Register	Undefined.
Supervisor Area Pointer Register	Bit 6 set (supervisor addresses cache inhibited); bit 7 clear (addresses not global); all other bits clear (pointer invalid).
User Area Pointer Register	Bit 6 set (user address cache inhibited); bit 7 clear (addresses not global); all other bits clear (pointer invalid).
BATC Write Ports	Undefined.
Cache Tag Ports	Undefined.
Cache Data Ports	Undefined.
Cache Set Status Ports	Undefined.

The CMMU register decoding scheme does not decode bit A5 for the BATC write ports or bits A5 and A4 for the cache diagnostic ports. Also, since the MC88100 CPU does not provide A1 and A0, address bits A1 and A0 on the M bus are not used for decoding any CMMU internal registers.

The CMMU registers can be accessed by the local processor or by any M bus master. This allows both the processor and other M bus masters to configure the MC88200 dynamically as well as perform diagnostics and read the component status. Software semaphores must be used to synchronize register accesses among various M bus masters; these are discussed in **6.3 SEMAPHORES**.

6.2.1.1 ID REGISTER (IDR). This register defines the MC88200 device type and the MC88200 base address in the control memory address space.



Address: Base + $000

ID — Identification

Defines the device identification number of the MC88200 registers in control memory space. The ID field identifies a 4K-byte partition of the control memory space allocated to a particular MC88200. The least significant seven bits of this field must be initialized by hardware during reset. The most significant bit is always set to zero during reset.

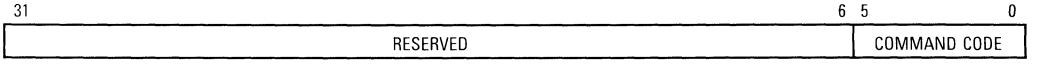
All eight bits can be reinitialized by system software to dynamically reconfigure the M bus peripherals.

TYPE — Component Type
This field is hardwired to 101_2 to specify that the component is an MC88200 CMMU. This field is read only.

VERSION — Mask Revision Number
This field is read only.

Bits 15–0 – Reserved
These bits are returned as zeros on a register read.

6.2.1.2 SYSTEM COMMAND REGISTER (SCR). This register is used to initiate various MC88200 functions such as probe transactions, address translation cache (ATC) flushes, and cache flushes. For several of the functions (probe transactions and certain flush operations), the processor writes the logical address to the system address register (SAR). The MC88200 places the results of a probe function in the system status register (SSR).



Address: Base + $004

Bits 31–6 — Reserved
These bits are undefined on a register read.

COMMAND CODE — Written by software to initiate the probe and cache control functions:

11X0gg	Invalidate User Page Address Translation Cache (PATC) Descriptors
11X1gg	Invalidate Supervisor PATC Descriptors
10X0XX	Probe User Address
10X1XX	Probe Supervisor Address
0100XX	No Operation
0101gg	Data Cache Invalidate
0110gg	Data Cache Copyback to Memory
0111gg	Data Cache Copyback and Invalidate
00XXXX	No Operation

where 'gg' is the desired granularity:
00 — line
01 — page
10 — segment
11 — all

6.2.1.3 SYSTEM STATUS REGISTER (SSR). The SSR provides probe results and M bus error information.

31 16	15	14	13 10	9	8	7	6	5	4	3	2	1	0
RESERVED	CE	BE	RESERVED	WT	SP	G	CI	R	M	U	WP	BH	V

Address: Base + $008

Bits 31–16, 13–10, 5 — Reserved
These bits are returned as zeros on a register read.

CE — Copyback Error
Indicates that an error occurred during a data cache line copyback initiated by a snoop transaction.

BE — Bus Error
Indicates that an M bus error occurred or that a data cache flush or probe operation was unsuccessful.

WT — Writethrough
1 = Data at the probed address is cached with the writethrough memory update policy.
0 = Data at the probed address is cached with the copyback memory update policy.

SP — Supervisor Privilege
1 = Probed address can only be accessed in the supervisor mode.
0 = Probed address can be accessed in the user or supervisor mode.

G — Global
1 = One or more of the descriptors for the probed address are marked global.
0 = All of the descriptors for the probed address have clear global bits.

CI — Cache Inhibit
1 = Data at the probed address cannot be cached.
0 = Data at the probed address can be cached.

M — Modified
1 = Data at the probed address has been modified in memory or with respect to memory (cached data).
0 = Data at the probed address has not been modified.

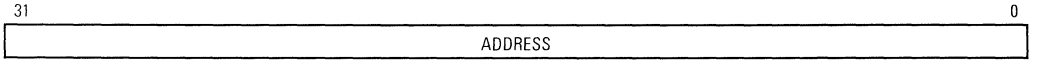
U — Used
This bit is always set since the probe transaction is interpreted as a use of the probed memory area.
1 = Data within the probed address has been used by the processor.

WP — Write Protection
1 = Probed address is write protected.
0 = Probed address can be read or written.

BH — BATC Hit
1 = Probed address resulted in a BATC hit instead of a PATC hit or table search.
0 = Probed address was found in the PATC or was generated by a table search.

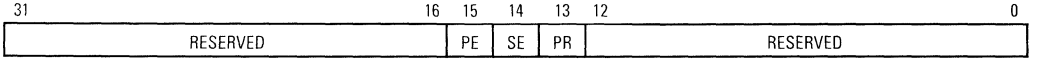
V — Valid
1 = All descriptors or ATC entries encountered by the probed address were marked valid. If this bit is set and no fault occurred, the probe results are valid.
0 = Probed address encountered an invalid table descriptor or unmapped transaction.

6.2.1.4 SYSTEM ADDRESS REGISTER (SAR). This register is written by software to pass addresses for the probe, invalidate PATC, and data cache flush commands. It is also used to pass the set number and word number for software access to cache tag, data, and set status bits. If an M bus error occurs during a flush, flush copyback, or probe command, the MC88200 writes the physical address of the faulted transaction into this register.



Address: Base + $00C

6.2.1.5 SYSTEM CONTROL REGISTER (SCTR). This register controls the MC88200 parity checking, M bus snooping, and bus arbitration scheme. Normally, system software writes this register during initialization to establish the operating mode of the MC88200. However, the local processor and other M bus masters can change this register as required to dynamically control the MC88200 operations.



Address: Base + $104

Bits 31–16, 12–0 — Reserved
These bits are undefined on a register read.

PE — Parity Enable
The MC88200 always generates parity on memory writes.
1 = Checks parity on memory reads.
0 = Does not check parity on memory reads.

SE — Snoop Enable

Snooping is discussed in **3.5 CACHE COHERENCY**.

1 = Snoop M bus transactions initiated by other M bus devices when those transactions are marked global.

0 = Do not snoop M bus transactions.

PR — Priority Arbitration

M bus arbitration is discussed in **SECTION 5 BUS OPERATIONS.**

1 = Arbitrate for M bus using priority protocol.

0 = Arbitrate for M bus using fairness protocol.

6.2.1.6 SUPERVISOR AND USER AREA POINTER REGISTER (SAPR, UAPR). These registers contain the area pointers (base address of the segment tables) used during translation table searches. They also contain control bits pertaining to all accesses in the selected area. When a table search begins, the supervisor/user bit received from the P bus specifies which area pointer to use.

31 — 12	11	10	9	8	7	6	5				1	0
SUPERVISOR SEGMENT TABLE BASE ADDRESS	0	0	WT	0	G	CI	0	0	0	0	0	TE

31 — 12	11	10	9	8	7	6	5				1	0
USER SEGMENT TABLE BASE ADDRESS	0	0	WT	0	G	CI	0	0	0	0	0	TE

Address: Base + $200 (Supervisor)
Base + $204 (User)

Bits 31–12 — Segment Table Base Address

Contains the most significant 20 bits of the physical address of the supervisor or user segment table, respectively. The segment table is always aligned on a page boundary (20-bit address).

Bits 11–10, 8, 5–1 — Reserved

These bits are returned as zeros on a register read.

WT — Writethrough

1 = Writethrough memory update policy

0 = Copyback memory update policy

G — Global

1 = Memory locations in this area are global locations.

0 = Memory locations in this area will be cached unless inhibited or segment or page level.

CI — Cache Inhibit

1 = Data and/or instructions mapped by this entry are not cached in the MC88200.
0 = Data and/or instructions mapped by this entry can be cached in the MC88200.

TE — Translation Enable

When the TE bit is clear and the CI bit (bit 6) is set, logical addresses are presented to the memory system.

1 = Enables address translation through the ATCs or by translation table search.
0 = Segment table base address is invalid, and a table search cannot be performed.

6.2.2 P Bus Fault Registers

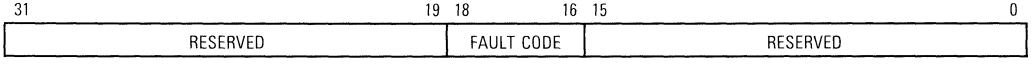
The P bus fault registers provide fault information to the local (P bus) processor. These faults may occur during the MC88100 execution of a load (**ld**), store (**st**), or exchange memory (**xmem**) instruction on the data P bus, or an instruction fetch on the instruction P bus. There are two fault registers: address and status. The fault information contained within these registers pertains strictly to the local processor.

NOTE

Any M bus master can access the P bus fault registers. However, register access should be restricted only to the P bus processor. If M bus devices access these registers, fault information could be lost and unpredictable behavior could occur.

Local exceptions are described in **2.4 FAULTS**.

6.2.2.1 P BUS FAULT STATUS REGISTER (PFSR). When a P bus transaction terminates with a fault, the PFSR indicates the type of exception.



Address: Base + $108

Bits 31–21, 15–0 — Reserved

These bits are returned as zeros on a register read.

FAULT CODE — P Bus Exception Type

The MC88200 writes this field to indicate the type of exception that occurred, then issues a fault reply on the P bus reply lines. When the processor receives the fault reply, it begins execution of an exception handler. This exception handler must read the fault code to determine the type of exception that occurred.

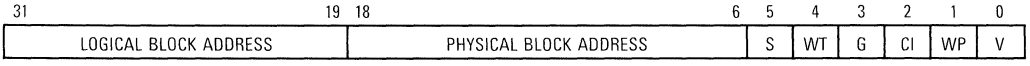
000 — Success (No Fault)
011 — Bus Error
100 — Segment Fault
101 — Page Fault
110 — Supervisor Violation
111 — Write Violation

6.2.2.2 P BUS FAULT ADDRESS REGISTER (PFAR). This register contains the physical address where the P bus exception occurred. The MC88200 updates this register whenever a P bus exception occurs (note: a write violation updates the PFAR with invalid data). If performing MC88200 diagnostics, software can write and read this register.

Address: Base + $10C

6.2.3 BATC Write Ports (BWP7–BWP0)

The BATC write ports are used by system software to write entries into the BATC.



Address: Base + $400 (BWP 0) through
Base + $41C (BWP 7)

6

Bits 31–19 — Logical Block Address
Contains the high-order 13 bits of the logical address for the mapped memory block.

Bits 18–6 — Physical Block Address
Contains the high-order 13 bits of the physical address for the mapped memory block.

S — Address Space
1 = Supervisor address space
0 = User address space

WT — Writethrough
1 = Writethrough memory update policy
0 = Copyback memory update policy

G — Global
1 = Memory mapped by this entry is global memory.
0 = Memory mapped by this entry is local memory.

CI — Cache Inhibit
1 = Data and/or instructions mapped by this entry are not cached in the MC88200.
0 = Data and/or instructions mapped by this entry can be cached in the MC88200.

WP — Write Protect
1 = Memory mapped by this entry is write protected by the MC88200.
0 = Memory mapped by this entry can be written by the MC88200.

V — Valid
1 = Descriptor is valid.
0 = Descriptor is not valid.

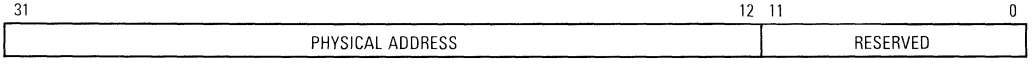
6.2.4 Cache Diagnostic Ports

The cache diagnostic ports provide a means for the local processor or for M bus devices to access the components of a data cache set: the data, the physical address tags, and the cache status bits. These registers are used for performing cache diagnostics as described in the following paragraphs.

6.2.4.1 CACHE DATA PORTS (CDP3–CDP0). These ports provide read/write access to each column in a cache set (e.g., word two of each line in the selected set). There are four cache data ports, corresponding to each line in a set. The set and word numbers to be accessed are specified by the value in the SAR: bits 11–4 specify the set number, and bits 3, 2 specify the word number.

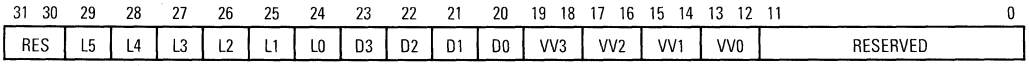
Address: Base + $800 (CDP0)
Base + $804 (CPD1)
Base + $808 (CPD2)
Base + $80C (CDP3)

6.2.4.2 CACHE TAG PORTS (CTP3–CTP0). These ports provide read/write access to the four address tags of a cache set. The cache tag contains the 20-bit physical page address of the data stored in each line of the cache set (the other 12 bits of the register are reserved). There are four cache tag ports, corresponding to each line in a set. The set to be accessed is selected by the value written to the SAR (bits 11–4).



Address: Base + $840 (CTP0)
Base + $844 (CTP1)
Base + $848 (CTP2)
Base + $84C (CTP3)

6.2.4.3 CACHE SET STATUS PORT (CSSP). This port provides information on the state of a cache set: least recently used (LRU) algorithm information, the control bits for disabling faulty lines, and the information on the state of each line. The set to be accessed is specified by the value in the SAR (bits 11–4).



Address: Base + $880

Bits 31–30, 11–0 — Reserved
These bits are returned as zeros on a register read.

L5 — Line 5
When set, line 3 is more recently used than line 2.

L4 — Line 4
When set, line 3 is more recently used than line 1.

L3 — Line 3
When set, line 3 is more recently used than line 0.

L2 — Line 2
When set, line 2 is more recently used than line 1.

L1 — Line 1
When set, line 2 is more recently used than line 0.

L0 — Line 0
When set, line 1 is more recently used than line 0.

D3–D0 — Line Disable
1 = Line disabled due to fault
0 = Line operational

VV3–VV0 — Line Valid
00 — Exclusive unmodified
01 — Exclusive modified
10 — Shared unmodified
11 — Invalid

6.3 SEMAPHORES

Some MC88200 commands require multiple register accesses for initiation and verification. For example, a cache copyback command requires an access to the SAR (for setting the

copyback granularity), an access to the SCR (for issuing the command), and multiple accesses to the SSR (for checking the copyback completion). In systems incorporating multiple M bus masters, these registers could be corrupted by another M bus master before the command completes execution.

To protect the MC88200 registers from corruption, each MC88200 in the system could be assigned an in-memory semaphore. When a particular M bus master requires an MC88200 for executing a command, then that master "claims" the MC88200 by taking ownership of the semaphore. Other masters then test the semaphore and do not access the MC88200 until the first M bus master has released ownership of the semaphore (i.e., completed its access of the MC88200).

6

SECTION 7
APPLICATIONS INFORMATION

This section describes the application of multiple MC88200 devices in a system, memory bus (M bus) connections, and power and ground considerations. The application information provided in this section is conceptual and is only provided as a guideline for system designers.

7.1 MULTIPLE MC88200 DEVICES

The following discussion provides guidelines when using two to eight MC88200 devices in a system. (Refer to the *MC88100 User's Manual* for a discussion of a basic system configuration.)

The MC88100 processor allows connections of multiple MC88200 cache memory management units (CMMUs) in one M88000 processing node. For this discussion, a processing node is characterized as containing one MC88100 processor and at least two MC88200 CMMUs, one connected to the data processor bus (P bus) and one connected to the instruction P bus. Each CMMU contains 16K bytes of cache memory organized as a four-way, set-associative, high-speed cache memory array. While a basic system containing one data CMMU and one instruction CMMU may provide adequate performance in most systems, it may be desirable to increase the number of CMMUs in one M88000 processing node.

In systems using one M88000 processing node, performance degradation caused by main memory latency may be alleviated by adding more MC88200 CMMUs. Each CMMU adds 16K bytes of fast cache memory and additional address translation cache (ATC) entries to the processing node, which reduces the amount of external data traffic on the M bus. This increases system throughput by reducing the number of accesses to main memory and the associated memory latency time.

Multiprocessing systems employing multiple M88000 processing nodes may also benefit by increasing the number of CMMUs. Total system throughput increases by reducing the dependency of each M88000 processing node on main memory. Less time is spent waiting to gain access to the main bus, alleviating a major bottleneck in system design. More of the main bus bandwidth becomes available for use by other bus masters such as direct memory access (DMA) devices.

Figure 7-1 shows a system block diagram using four MC88200 CMMUs in one processing node. Two CMMUs are connected to the instruction P bus and two to the data P bus. Each CMMU requests and receives ownership of the M bus through the M bus arbitration block. The system status interface block provides M bus slaves the ability to insert wait states, generate retry operations, control burst operations, and notify the current master of error conditions. The reset block provides powerup reset and warm reset information and initializes the phase locking of the system clock (see *MC88100 User's Manual* for a circuit implementing the reset function). The identification (ID) initialization block provides a unique identification number to each CMMU that allows software to independently address each CMMU internal register set.

Figure 7-2 shows a multiple M bus configuration with eight CMMUs sharing a common set of system resources. The CMMUs labeled A–D represent any combination of CMMUs configured either as instruction or data CMMUs (E–H also represent any combination). As the number of CMMUs connected to one M bus increases to greater than six, serious consideration should be given to creating multiple M buses.

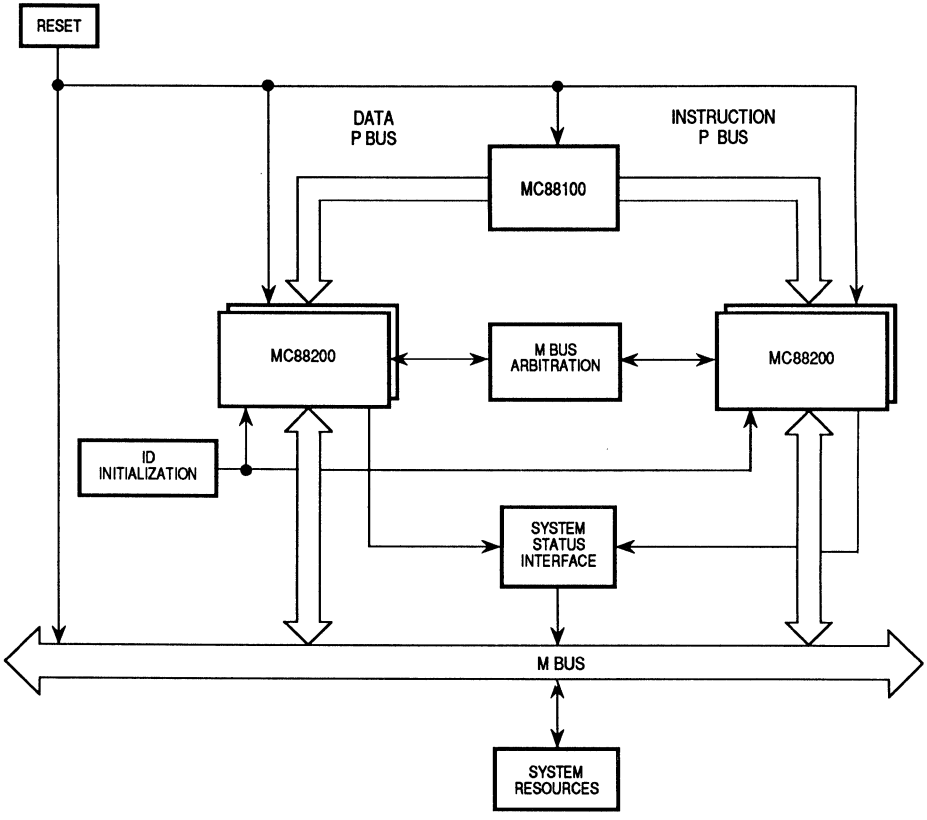


Figure 7-1. System Configuration

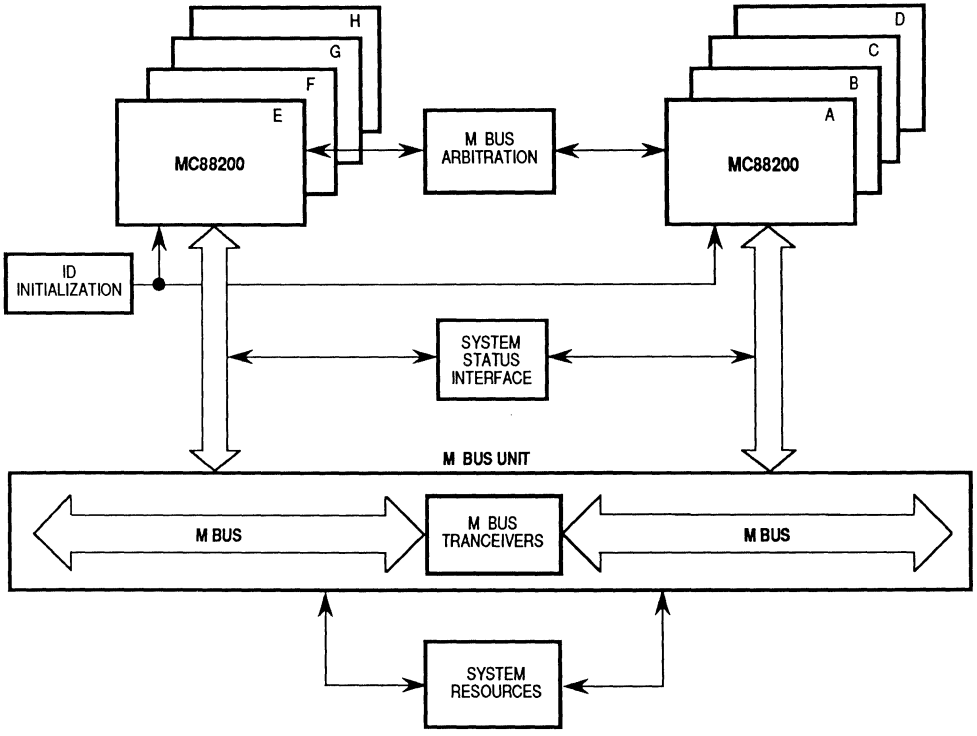


Figure 7-2. Multiple M Bus Configuration

Each CMMU on the M bus in a pin-grid array (PGA) package, adds 15 pF of capacitive loading to each signal on the M Bus. The output drivers of the MC88200 are specified and tested with an output load capacitance of 130 pF for M bus signals. Six CMMUs on an M bus represent a 75-pF load capacitance to any one CMMU driving bus. This leaves 55 pF for socket capacitance, circuit board trace capacitance, and capacitance associated with M bus interfaces to system resources. Eight CMMUs represent a 105-pF load capacitance (7 CMMU's $\times$ 15 pF), leaving 25 pF available from other capacitance sources. From a worst-case design standpoint, an eight CMMU system design using a single M bus should not be implemented without buffering or capacitive derating.

The two M bus configuration of Figure 7-2 provides a transceiver section that buffers one M bus from the other. The transceiver section allows the two M buses to be treated as one M bus unit. Only one of the eight CMMUs will be designated as M bus master at one time. In systems where A–D represent instruction CMMUs and E–F represent data CMMUs, the transceivers provide communication allowing the data CMMUs to access the internal registers of the instruction CMMUs. In systems in which CMMUs A–D represent one processing node and CMMUs E–H a second processing node, the transceivers provide snooping capability. The system resources section can be constructed using a single port connected to one or the other M bus, utilizing the transceivers for communication to the second

M bus. Dual porting of the system resources may be advantageous when trying to hide the propagation delay associated with the transceiver section (see **7.3 MULTIPLE M BUS CONNECTIONS**).

7.1.1 Instruction P Bus Connections

Figure 7-3 shows the connections between the MC88100 processor and the instruction CMMUs. Two CMMUs are shown connected to one P bus, providing 32K bytes of instruction cache memory. The code address bus (CA31–CA2), code bus (C31–C0), and the code supervisor/user (CS/$\overline{U}$) signals of the MC88100 connects to the address bus (A31–A2), data bus (D31–D0), and the supervisor/user (S/$\overline{U}$) signals of the instruction CMMU, respectively. Since the instruction P bus is a read-only bus, the read/write (R/$\overline{W}$) signal and the data bus lock ($\overline{DLOCK}$) signal of the instruction CMMUs are pulled to a logic high. When connecting more than one instruction CMMU, the code fetch (CFETCH) signal of the MC88100 processor must be buffered and then connected to all four data byte enable (DBE3–DBE0) signals. The 74F244 buffer used in this configuration prevents overloading of the CFETCH signal. The code reply (CR1–CR0) signals of the MC88100 are connected to the reply (R1–R0)

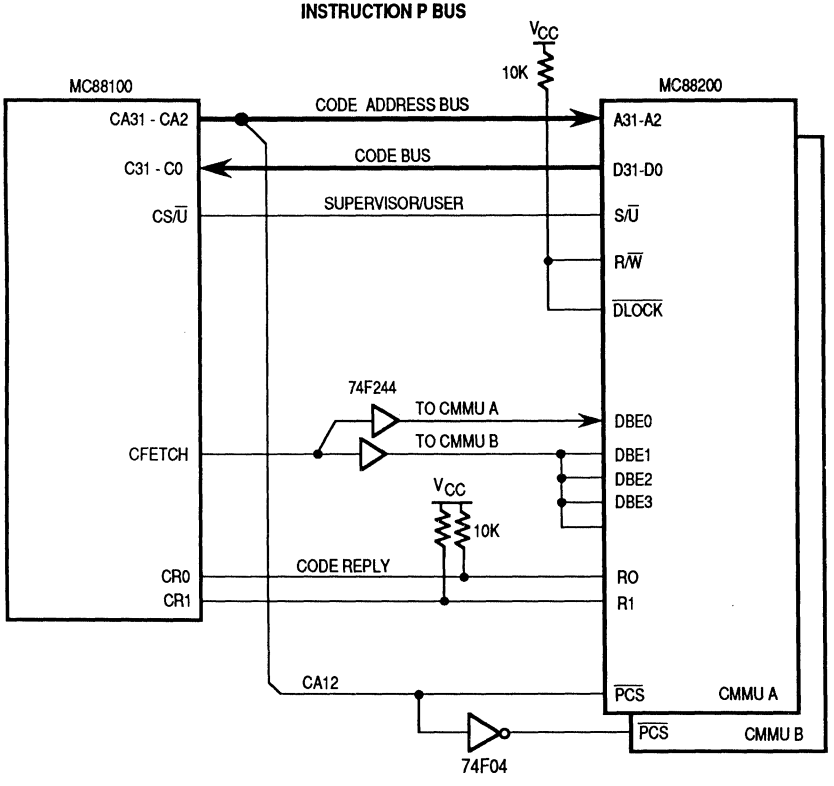


Figure 7-3. Instruction P Bus Connection

signals of the CMMU. In addition, the CR1–CR0 signals should be pulled high through pullup resistors, which defaults the reply bus to a fault condition.

The final connection to the instruction P bus provides a unique P bus chip select ($\overline{PCS}$) signal to each CMMU. The 74F04 inverter uses code address bit CA12 to select one CMMU for any given P bus transaction. Decoding addresses on a 4K-byte boundary provides the most efficient use of multiple MC88200 devices. Decoding a larger byte boundary, such as 8K or 16K, will increase the probability that the CMMU will have to replace an existing cache line. Each set in the four-way set-associative cache contains four entries. The 4K boundary attempts to fill one entry in each set before moving to the next CMMU. An 8K boundary fills two entries in each set before moving to the next CMMU; a 16K boundary attempts to fill all four entries. The higher the entry fill number, the higher the probability that a cache line will have to be replaced, which increases the chances for thrashing and results in poor system performance. Decoding a smaller byte boundary, such as 2K or 1K, does not efficiently utilize the cache sets before selecting another CMMU. This may result in performing a higher percentage of ATC misses, again reducing system performance. The $\overline{PCS}$ signal should be connected to ground when using only one CMMU.

7.1.2 Data P Bus Connections

Figure 7-4 shows the data P bus connections for two data CMMUs. The data address bus (DA31–DA2), data bus (D31–D0), data supervisor/user (DS/$\overline{U}$), data read/write (DR/$\overline{W}$), data lock ($\overline{DLOCK}$), and the data byte enable (DBE3–DBE0) signals on the MC88100 are connected to the A31–A2, D31–D0, S/$\overline{U}$, R/$\overline{W}$, $\overline{DLOCK}$, and the DBE3–DBE0 signals on the CMMU, respectively. The data reply (DR1–DR0) signals are connected to the reply lines (R1–R0) on the MC88200 in addition to being pulled high through pullup resistors. The pullups default the reply to a fault condition similar to the instruction P bus reply lines. The P bus $\overline{PCS}$ signal is generated by a 74F04 inverter using the data address bit DA12. When using one MC88200 configured as a data CMMU, the $\overline{PCS}$ signal is grounded.

7.1.3 ID Initialization

Figure 7-5 shows four CMMUs configured with unique identification numbers. After the reset signal negates, the MC88200 reads the logic levels on the local status (ST3–ST0) pins, the tag monitoring (TM0) pin, and the trace (TR1–TR0) pins. Figure 7-6 shows connections to eight CMMUs to initialize unique identification base addresses. These eight CMMUs are tied to one M bus unit, allowing only one CMMU to be designated as M bus master at one time. Table 7-1 lists the base address for each of the MC88200 CMMUs shown in Figures 7-5 and 7-6.

In the configuration shown in Figures 7-5 and 7-6, only the TM0 and TR1–TR0 pins use pulldown resistors since the ST3–ST0 signals are used by the system status interface logic. Pulldown resistors may reduce the effective fanout of these drivers to a great extent than do pullup resistors due to the high-impedance state during reset. When calculating pull-down resistor values, all reverse low-level input current from all devices connected to a

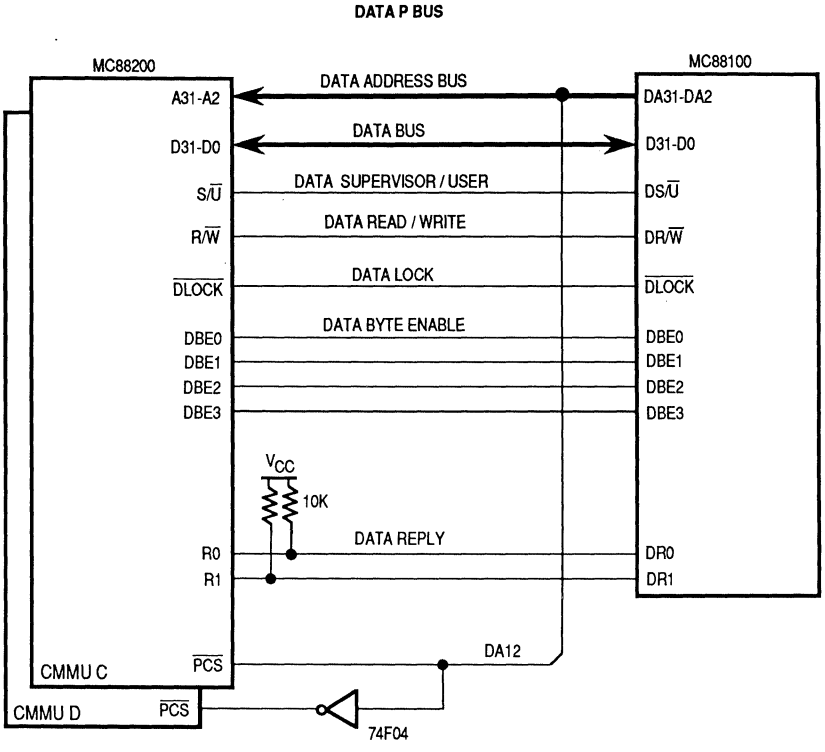


Figure 7-4. Data P Bus Connections

7

Table 7-1. ID Base Address

MC88200 CMMU	ID Base Address
A	FFF7F000
B	FFF7E000
C	FFF7D000
D	FFF7C000
E	FFF7B000
F	FFF7A000
G	FFF79000
H	FFF78000

particular signal must be added together. For the F, AS, and ALS family of logic devices, the resulting amount of current that the resistor must sink to ensure a logic-low voltage during ID initialization can be in the 0.5–2 mA range. A resistor value low enough to guarantee a logic low during reset may result in problems when trying to drive that signal high after reset.

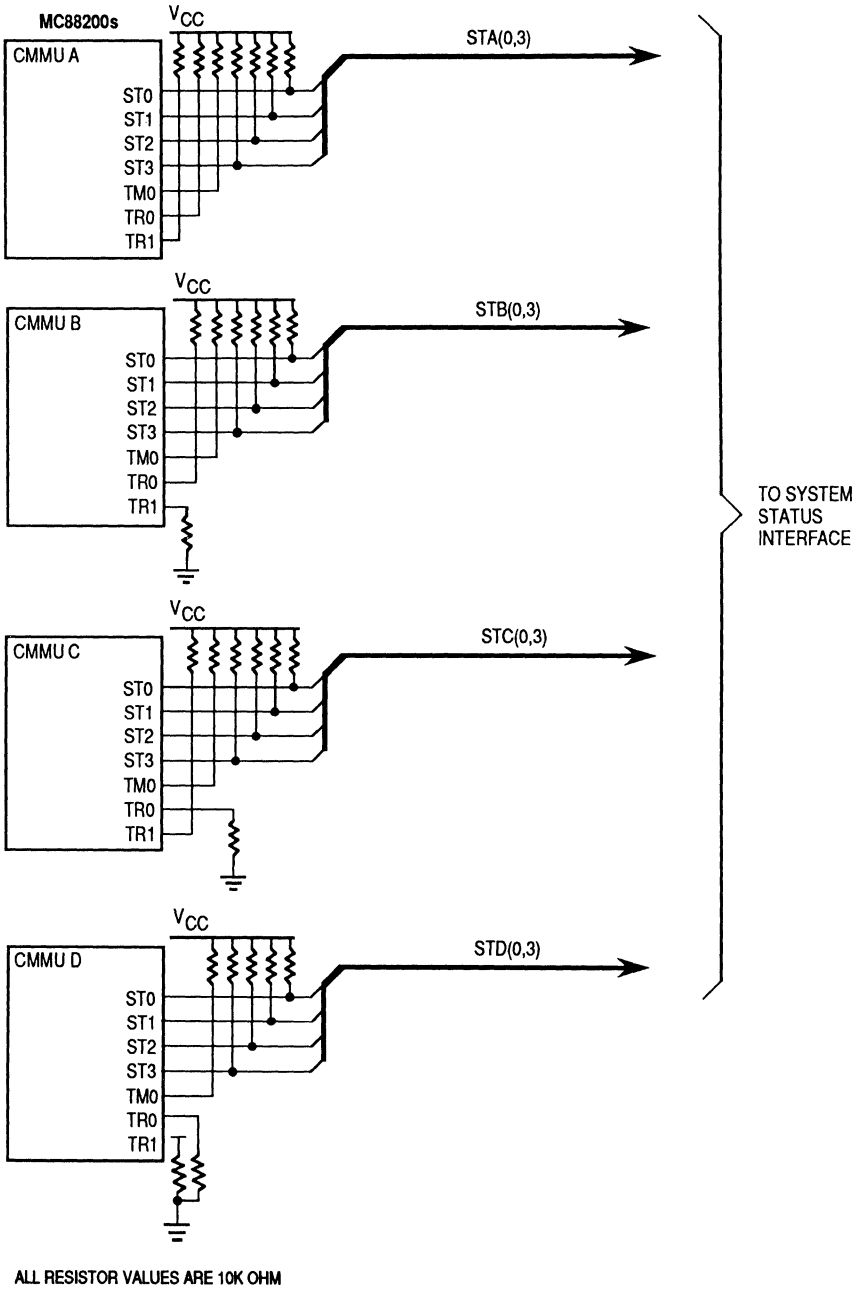


Figure 7-5. ID Initialization for Four CMMUs

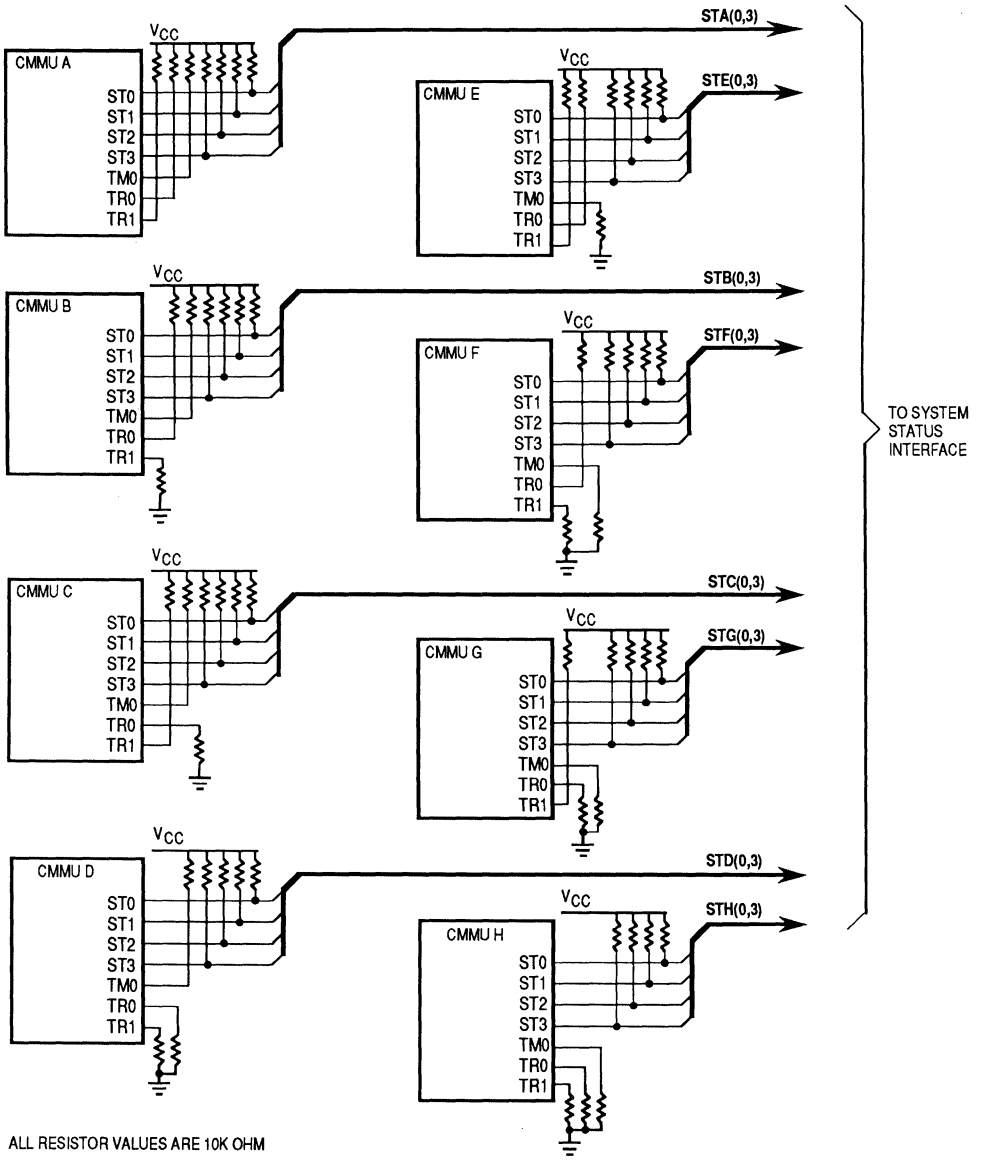


Figure 7-6. ID Initialization for Eight CMMUs

In the configuration shown in Figures 7-5 and 7-6, only the TM0 and TR1–TR0 pins use pulldown resistors since the ST3–ST0 signals are used by the system status interface logic. Pulldown resistors may reduce the effective fanout of these drivers to a great extent than do pullup resistors due to the high-impedance state during reset. When calculating pull-down resistor values, all reverse low-level input current from all devices connected to a particular signal must be added together. For the F, AS, and ALS family of logic devices,

the resulting amount of current that the resistor must sink to ensure a logic-low voltage during ID initialization can be in the 0.5–2 mA range. A resistor value low enough to guarantee a logic low during reset may result in problems when trying to drive that signal high after reset.

Figure 7-7 shows an alternate method of supplying ID information without the use of pulldown resistors. The 74F74 D-type flip-flop samples the reset ($\overline{RST}$) signal. When a low is detected, the clear input ($\overline{CD}$) of the second flip-flop is released. On the next clock edge, the output enable ($\overline{IDOE}$) of the 74F244 driver asserts, allowing one of the ID pins from each MC88200 to be driven to a logic-low level. The two flip-flops create a one clock delay from the time $\overline{RST}$ is sampled low to when the 74F244 starts driving. This delay allows the MC88200 to free its output drivers before ID information is driven on the bus. All signals used for ID initialization not shown are pulled up by 10K-ohm resistors as shown in Figure 7-6. When the 74F74 samples reset high, the $\overline{CD}$ input of the second 74F244 asserts. This in turn negates the $\overline{IDOE}$ signal, which three states the 74F244 driver outputs. The use of active drivers to initialize ID information allows the TM0 and the TR1–TR0 signals to be connected to other logic functions without sacrificing high-level output current from the MC88200.

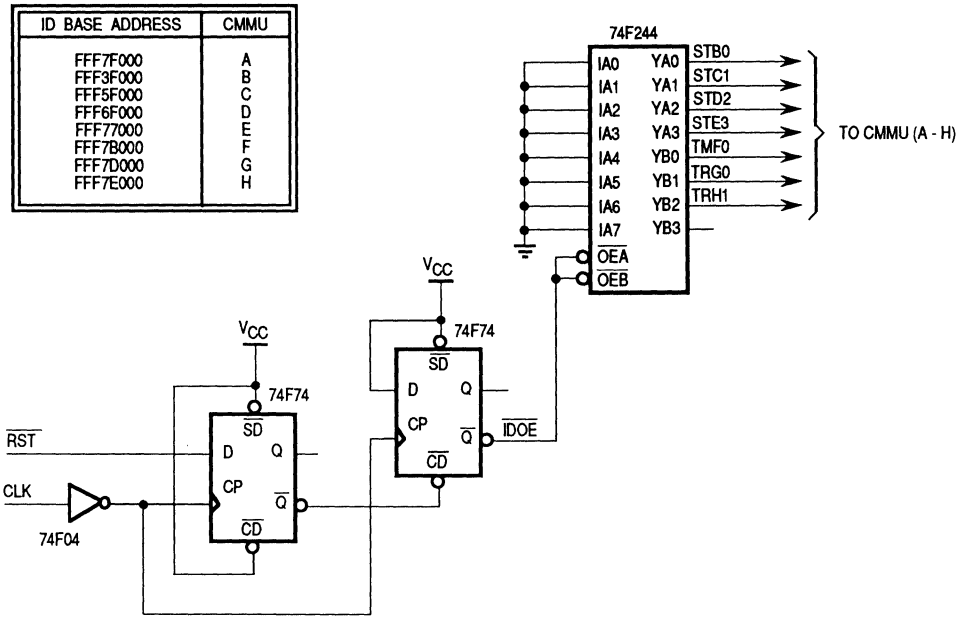


Figure 7-7. Alternate ID Initialization

7.1.4 System Status Connections

Figure 7-8 shows the ST3–ST0 signals interfacing to the $\overline{SS3}$–$\overline{SS0}$ signals for three CMMUs. The local status signals are inverted through open-collector inverting buffers (74F622) and wired together to form the system status signals. Due to capacitive loading, a maximum of three MC88200s in pin-grid array (PGA) packages can be connected using open-collector buffers. The 74F622 is tested using 50-pF capacitive loads, allowing connections to three CMMUs. The 10K-ohm resistor pullups on the local status signals are for ID initialization (see **7.1.3 ID Initialization**).

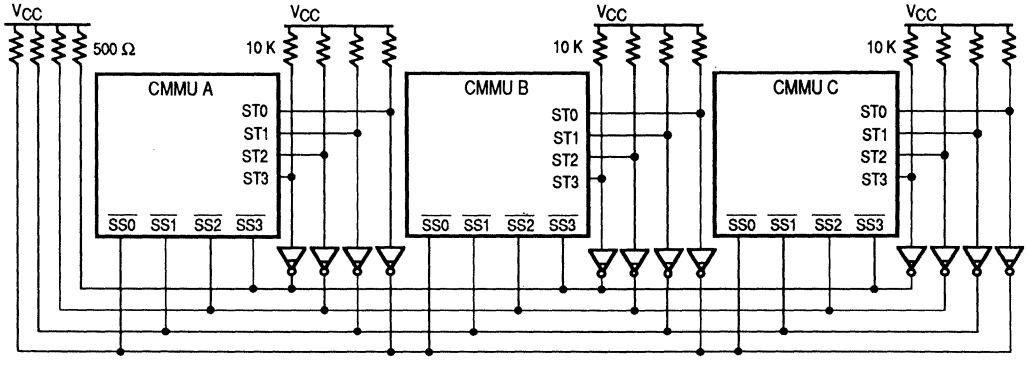


Figure 7-8. System Status Connections with Three CMMUs

Figure 7-9 shows the system status interface using four CMMUs. The local status signals from each CMMU are collected and NORed through a 74F260 five-input NOR gate. The output signal is then buffered through 74F244 buffers and sent to the CMMU $\overline{SS3}$–$\overline{SS0}$ signals. In addition to collecting the local status signals, the 74F260 includes a fifth signal to allow other M bus devices to signal ERROR, RETRY, WAIT, or EOD conditions.

Figure 7-10 shows the $\overline{SS3}$–$\overline{SS0}$ signals interfacing to eight CMMUs. The PALs® collect and combine the ST3–ST0 signals from each CMMU to generate the active-high system status signals. These active-high signals are then inverted through 74AS02 NOR gates before connection to the CMMU system status signals. The eight CMMUs have been divided into three groups to stay within the capacitive rating specification of the 74AS02 gates. The three groups are CMMU (A–C), CMMU (D–F), and CMMU (G–H). Each CMMU group has its own 74AS02 gate to drive the individual $\overline{SS3}$–$\overline{SS0}$ signals. In addition to inverting

PAL is a registered trademark of Monolithic Memories Inc.

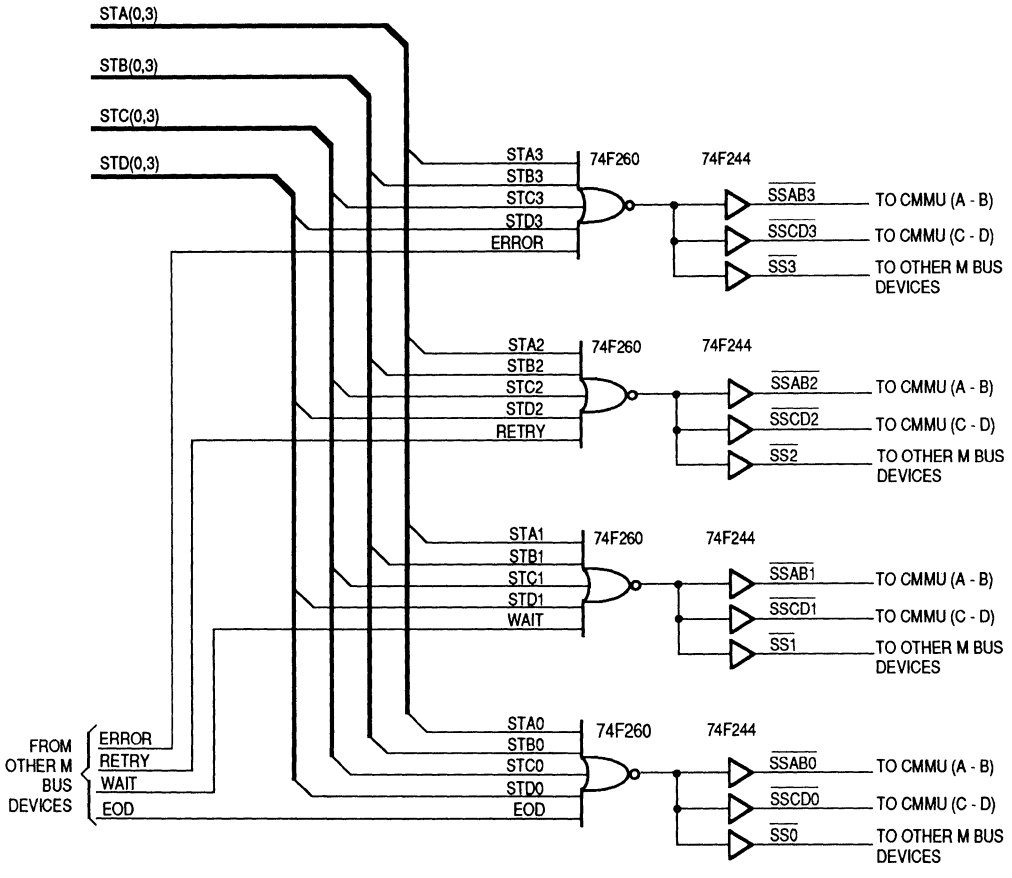


Figure 7-9. System Status Connections with Four CMMUs

the active-high SS3–SS0 signals from the PALs, the 74AS02 gates allow other M bus masters or slaves to incorporate their system status information (ERROR, RETRY, WAIT, and EOD). The PAL logic equations are shown in Figure 7-11.

7.1.5 M Bus Arbitration

Figure 7-12 depicts an arbitration scheme to control three MC88200 CMMUs connected to the same M bus. Arbitration begins when any of the CMMUs assert the bus request (BR) signal. The BR signal is then inverted through an open-collector inverting buffer (74F622) to generate the arbitration busy ($\overline{AB}$) signal used by the CMMU in the fairness protocol. In addition, the BR signal is also used by the arbitration logic to generate the bus grant (BG) signal. The arbitration logic prioritizes the bus request signals such that CMMU A has highest priority and CMMU C has lowest priority. To prevent CMMU A from blocking the

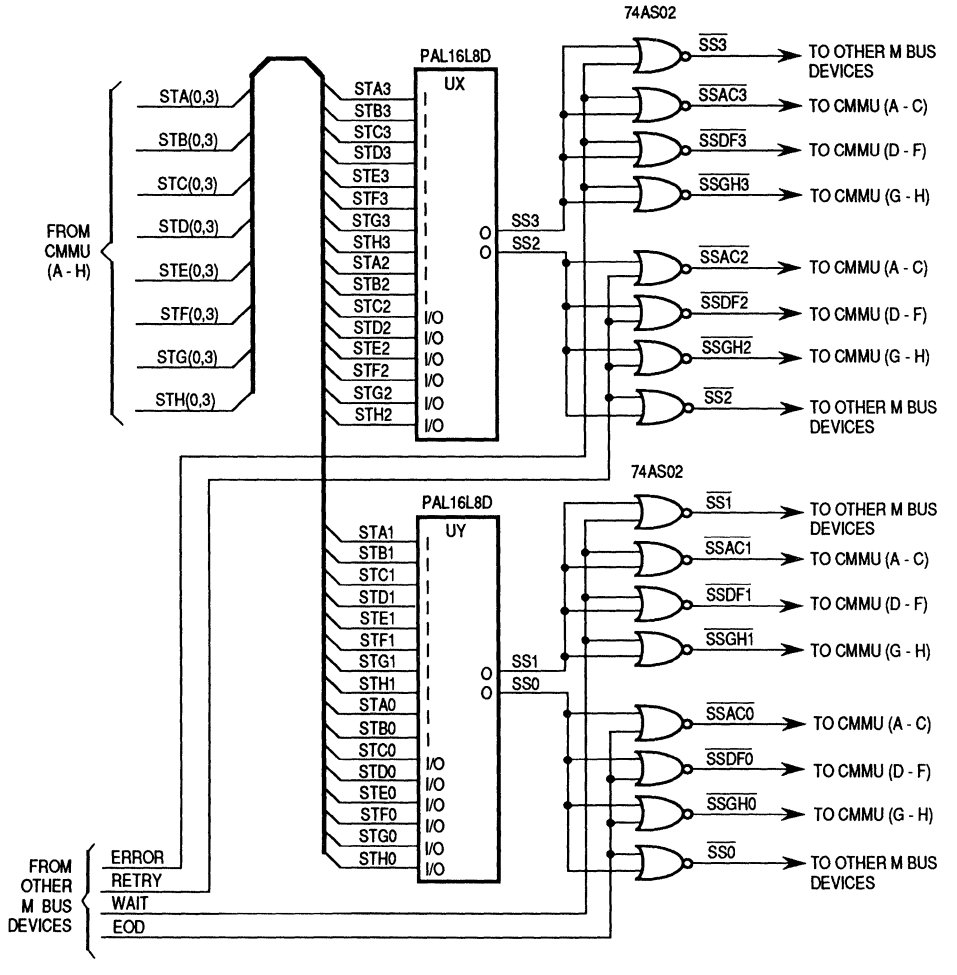


Figure 7-10. System Status Interface

M bus, the CMMUs must be initialized to use the fairness protocol. Generally, the highest priority CMMU should be an instruction CMMU, which helps to keep the instruction pipeline filled.

A maximum of three CMMUs in PGA packages may be connected using open-collector gates to generate the $\overline{AB}$ and the bus busy ($\overline{BB}$) signals. If more than three CMMUs are required in a system, then the $\overline{AB}$ and $\overline{BB}$ signals may be separated into groups where each group is within the rated capacitive loading of the driving gate. Figure 7-13 shows M bus arbitration for five CMMUs using one PAL16L8D device. The CMMUs have been divided into two groups, CMMU (A–C) and CMMU (D–E), to keep the capacitive loading

```
*************************************************
PAL16L8D      PAL DESIGN SPECIFICATION
UX            7/7/88
M88000 SYSTEM STATUS LOGIC - SS3, SS2
MOTOROLA INC., AUSTIN, TEXAS
STA3 STB3 STC3 STD3 STE3 STF3 STG3 STH3 STA2 GND
STB2 SS2  STC2 STD2 STE2 STF2 STG2 STH2 SS3 VCC

/SS3 = /STA3 * /STB3 * /STC3 * /STD3 * /STE3
    * /STF3 * /STG3 * /STH3;
/SS2 = /STA2 * /STB2 * /STC2 * /STD2 * /STE2
    * /STF2 * /STG2 * /STH2;


*************************************************
PAL16L8D      PAL DESIGN SPECIFICATION
UY            7/7/88
M88000 SYSTEM STATUS LOGIC - SS1, SS0
MOTOROLA INC., AUSTIN, TEXAS
STA1 STB1 STC1 STD1 STE1 STF1 STG1 STH1 STA0 GND
STB0 SS0  STC0 STD0 STE0 STF0 STG0 STH0 SS1 VCC

/SS1 = /STA1 * /STB1 * /STC1 * /STD1 * /STE1
    * /STF1 * /STG1 * /STH1;
/SS0 = /STA0 * /STB0 * /STC0 * /STD0 * /STE0
    * /STF0 * /STG0 * /STH0;
```

Figure 7-11. PAL Logic Equations for System Status Interface

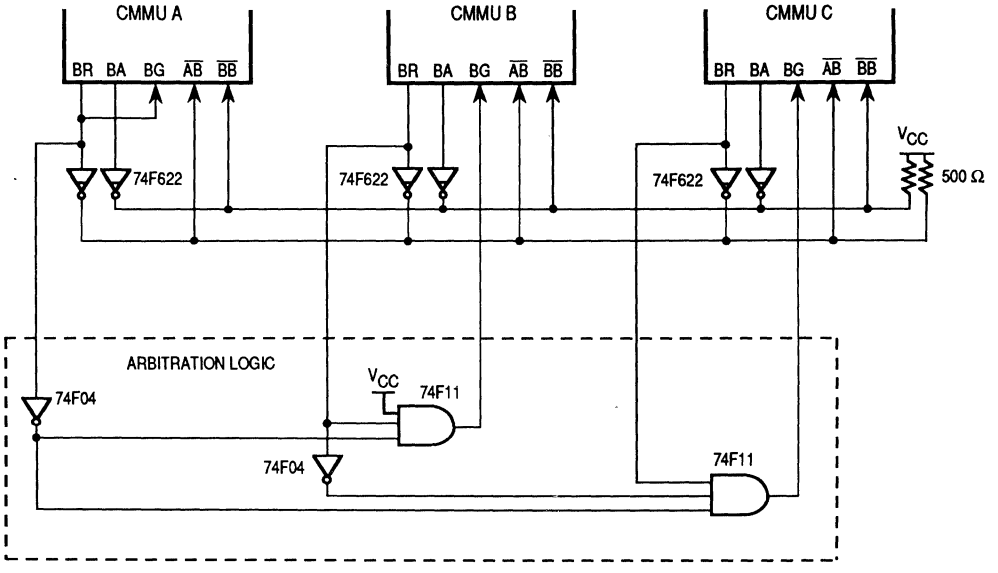


Figure 7-12. M Bus Arbitration with Three CMMUs

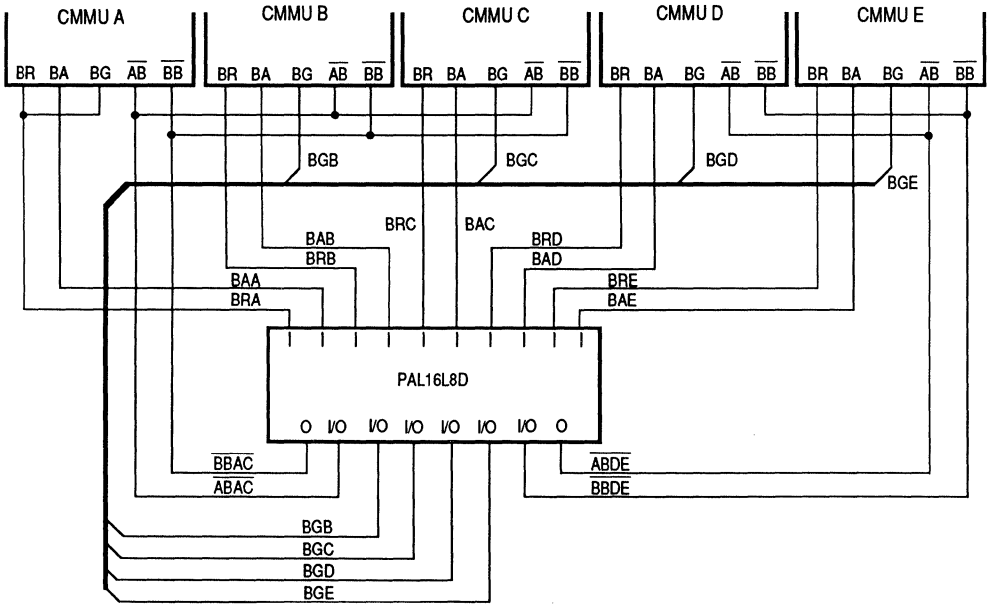


Figure 7-13. M Bus Arbitration with Five CMMUs

on the $\overline{AB}$ and $\overline{BB}$ signals below 50 pF. The PAL takes both the BR and bus arbitration (BA) signals as inputs from each individual CMMU and generates the BG, $\overline{AB}$, and $\overline{BB}$ signals. Figure 7-14 lists the PAL logic equations for this implementation. If only four CMMUs are being used, then one set of arbitration signals may be eliminated from the PAL equations. A four-CMMU implementation would also reduce the capacitive loading on each group of $\overline{AB}$ and $\overline{BB}$ signals to 30 pF.

Figure 7-15 shows arbitration logic for eight CMMUs connected to one M bus unit. The eight CMMUs have been split into three groups: CMMU (A–C), CMMU (D–F), and CMMU (G–H). Each group has its own driver for the $\overline{AB}$ and $\overline{BB}$ signals, which keeps the capacitive load within specification of the 74AS04 inverters used to drive these signals to the CMMU PGA packages.

The PAL at UX takes the individual BR signals from all eight CMMUs and generates the BG signals. Only one BG signal (BGA–BGH) asserts at any time, allowing only one CMMU to gain access to the M bus. The PAL logic equations to generate the BG signals are shown in Figure 7-16.

The PAL designated as UY in Figure 7-15 generates the active-high $\overline{AB}$ and $\overline{BB}$ signals. All of the individual BR and BA signals (BRA–BRH and BAA–BAH) are used as inputs. If any of the BR signals are active, then the $\overline{AB}$ signal asserts. Similarly, if any of the BA signals are active, then the $\overline{BB}$ signal asserts. The $\overline{AB}$ and $\overline{BB}$ signals are then inverted through

```
*********************************************
PAL16L8D    PAL DESIGN SPECIFICATION
UX      7/6/88
M88000 M BUS ARBITRATION LOGIC
MOTOROLA INC., AUSTIN TEXAS
BAE BRE BAD BRD BAC BRC BAB BRB BAA GND
BRA /BBAC /ABAC BGB BGC BGD BGE /BBDE /ABDE VCC

/BGB  = /BRB + BRA;
/BGC  = /BRC + BRA + BRB;
/BGD  = /BRD + BRA + BRB + BRC;
/BGE  = /BRE + BRA + BRB + BRC + BRD;

ABAC = BRA + BRB + BRC + BRD + BRE;
ABDE = BRA + BRB + BRC + BRD + BRE;

BBAC = BAA + BAB + BAC + BAD + BAE;
BBDE = BAA + BAB + BAC + BAD + BAE;
```

Figure 7-14. PAL Logic Equations for Arbitration with Five MC88200s

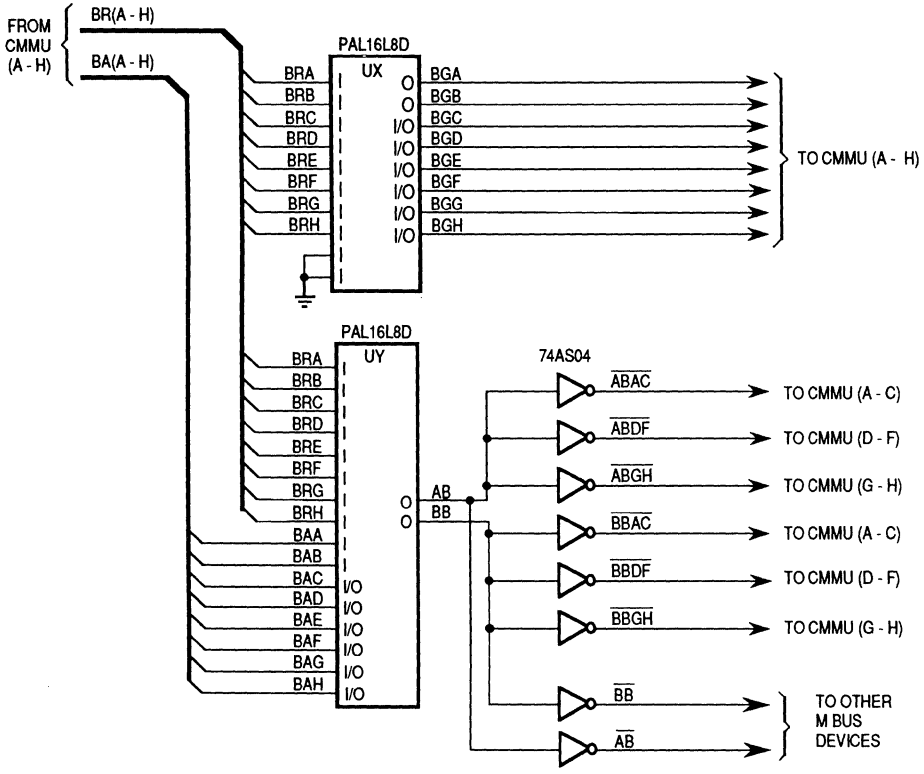


Figure 7-15. M Bus Arbitration with Eight CMMUs

```
*****************************************************
PAL16L8D      PAL DESIGN SPECIFICATION
UX          7/6/88
M88000 M BUS ARBITRATION LOGIC - BG GENERATION
MOTOROLA INC., AUSTIN, TEXAS
BRA BRB BRC BRD BRE BRF BRG BRH NC1 GND
NC2 BGB BGH BGG BGF BGE BGD BGC BGA VCC

/BGA = /BRA;
/BGB = /BRB + BRA;
/BGC = /BRC + BRA + BRB;
/BGD = /BRD + BRA + BRB + BRC;
/BGE = /BRE + BRA + BRB + BRC + BRD;
/BGF = /BRF + BRA + BRB + BRC + BRD + BRE;
/BGG = /BRG + BRA + BRB + BRC + BRD + BRE + BRF;
/BGH = /BRA + BRB + BRC + BRD + BRE + BRF + BRG;
```

Figure 7-16. PAL Logic Equations for BG Signal Generation

```
*****************************************************
PAL16L8D      PAL DESIGN SPECIFICATION
UY          7/6/88
M88000 M BUS ARBITRATION LOGIC - AB, BB GENERATION
MOTOROLA INC., AUSTIN, TEXAS
BRA BRB BRC BRD BRE BRF BRG BRH BAA GND
BAB BB  BAC BAD BAE BAF BAG BAH AB  VCC

/AB = /BRA * /BRB * /BRC * /BRD * /BRE * /BRF * /BRG * /BRH;
/BB = /BAA * /BAB * /BAC * /BAD * /BAE * /BAF * /BAG * /BAH;
```

Figure 7-17. PAL Logic Equations for AB and BB Signal Generation

74AS04 inverters and connected to the various CMMU groups. The $\overline{BB}$ and $\overline{AB}$ signals can also be used by other M bus devices to detect when an M bus tenure begins. Figure 7-17 shows the PAL logic equations to generate the $\overline{AB}$ and the $\overline{BB}$ signals.

7.2 M BUS CONNECTIONS

Figure 7-18 shows a block diagram of an M bus interface. The main functions of the interface are to translate the M bus signals into a bus structure that can directly connect to memory and peripheral devices. In addition, the interface keeps the capacitive loading on the M bus within specification by limiting the number of devices on the M bus itself. Common memory and peripheral devices use a nonmultiplexed address and data bus. The interface

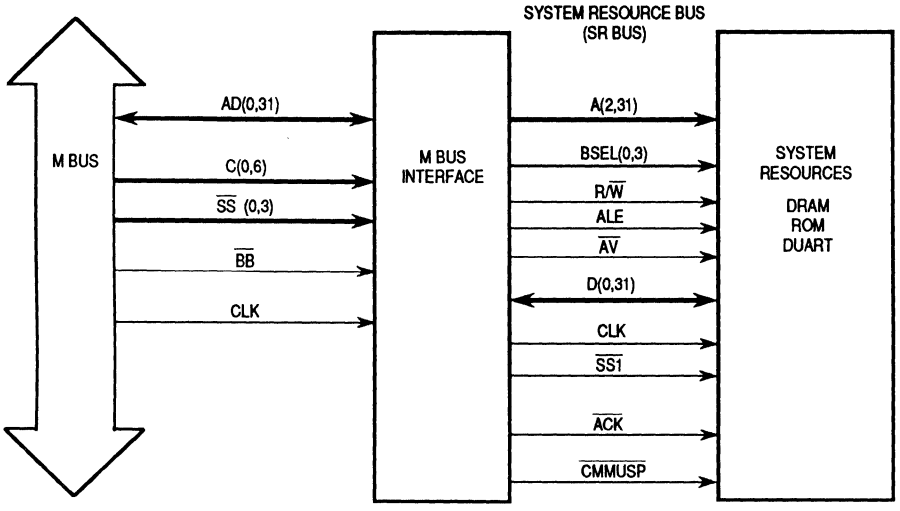


Figure 7-18. M Bus Interface Block Diagram

logic demultiplexes the AD31–AD0 signals of the M bus to generate separate A31–A2 and D31–D0 buses on the system resource bus (SR bus). The M bus control signals (C6–C0) are used in conjunction with the $\overline{BB}$ signal to generate the SR bus byte select (BSEL3–BSEL0), R/$\overline{W}$, address latch enable (ALE), and the address valid ($\overline{AV}$) signals. The $\overline{SS1}$–$\overline{SS0}$ signals are used by the interface logic to control wait states and system error conditions on the M bus. The $\overline{SS1}$ signal of the M bus is passed through to the SR bus to allow system resource devices to monitor wait conditions generated by other M bus devices. For example, a snooping CMMU will generate two wait cycles on a global data access, which the responding memory slave has to understand to remain in synchronous operation with the current M bus master. After the M bus requests a transaction, the interface logic drives a wait condition on the $\overline{SS3}$–$\overline{SS0}$ signals until a system resource responds by asserting the acknowledge signal ($\overline{ACK}$) on the SR bus. The $\overline{CMMUSP}$ signal indicates that the current transaction is to an M bus MC88200 device and prevents the interface logic from driving the wait code on the $\overline{SS3}$–$\overline{SS0}$ signals.

Limiting the number of devices on the M bus becomes a major concern with increased numbers of MC88200 devices due to capacitive loading specifications. When using eight CMMUs, the number of interface connections on the M bus may be limited to one. Therefore, grouping system resources on an alternate bus is one method of managing capacitive loading specifications. The instantaneous current requirements of the MC88200 increase with added capacitance on either the M bus or P bus. To manage the instantaneous current

requirements, fast, low-inductance, decoupling capacitors are required to maintain the localized voltage level on the power planes. If the instantaneous current requirements are not met, then local variations in supply voltage may adversely affect device operation. If the capacitive loading specification is violated, the maximum power dissipation specification (P_D) cannot be guaranteed.

7.2.1 M Bus Interface

Figure 7-19 shows the M bus interface logic. The PAL16R4D uses the M bus address/data phase signal (C0) and the $\overline{BB}$ signal to detect the address phase of a transaction. Upon detection of a valid address phase, the ALE signal asserts to latch the address in the 74F841 interface latch. The $\overline{AV}$ signal asserts later in the access to indicate that the A31–A2 signals the transaction to determine the last data transfer. The last data transfer state and the M bus $\overline{SS1}$ signal generate the internal PAL signal $\overline{RELEASE}$, which negates the ALE signal. The PAL generates the R/$\overline{W}$ signal from the M bus C2 signal. The data-enable-in ($\overline{DEIN}$) and data-enable-out (DEOUT) signals control the D31–D0 signal direction of the 74AS652 octal registered transceivers on read and write operations. Since the AD31–AD0 and C6–C0 signals are not guaranteed to remain valid immediately after a rising clock edge during wait cycles, both the 74AS652s and the 74F374 D-type flip-flops sample the data and control signals in the middle of each clock period, respectively. The ERROR and WAIT signals error conditions and to generate wait states. The PAL automatically asserts WAIT on the data phase of an M bus transaction if the transaction address is not an internal register of a CMMU indicated by the $\overline{CMMUSP}$ signal. If WAIT is asserted, then it remains asserted until the $\overline{ACK}$ signal is received or the transaction is terminated by the negation of $\overline{BB}$. If a system resource device does not acknowledge the access, then the 74AS867 8-bit synchronous counter generates the ERROR signal after 128 clock cycles. The ERROR signal terminates the M bus tenure, which in turn negates the WAIT signal and reloads the counter. The M bus C6–C3 signals contain byte select information on data phases and are used to generate the BSEL3–BSEL0 signals.

Three additional input control signals are included to allow the DRAM control logic to synchronize with M bus wait cycles. The row address ($\overline{RA}$) select signal allows the interface control to keep the address latched in the 74F841 during DRAM cycles. As long as the DRAM controller holds $\overline{RA}$ high, the interface logic should keep the current address latched.

The select B to A (SBA) signal connects directly to the 74AS652 and allows the DRAM controller to select real-time write M bus data or sampled M bus data. The data stall ($\overline{DSTALL}$) signal indicates that the DRAM controller sampled a wait status on the M bus and that stored data in the 74AS652 should be driven to stay in synchronous operation with M bus activity.

Figure 7-20 shows the PAL logic equations for the M bus interface. Figure 7- 21 shows the relative timing for the M bus to SR bus interface.

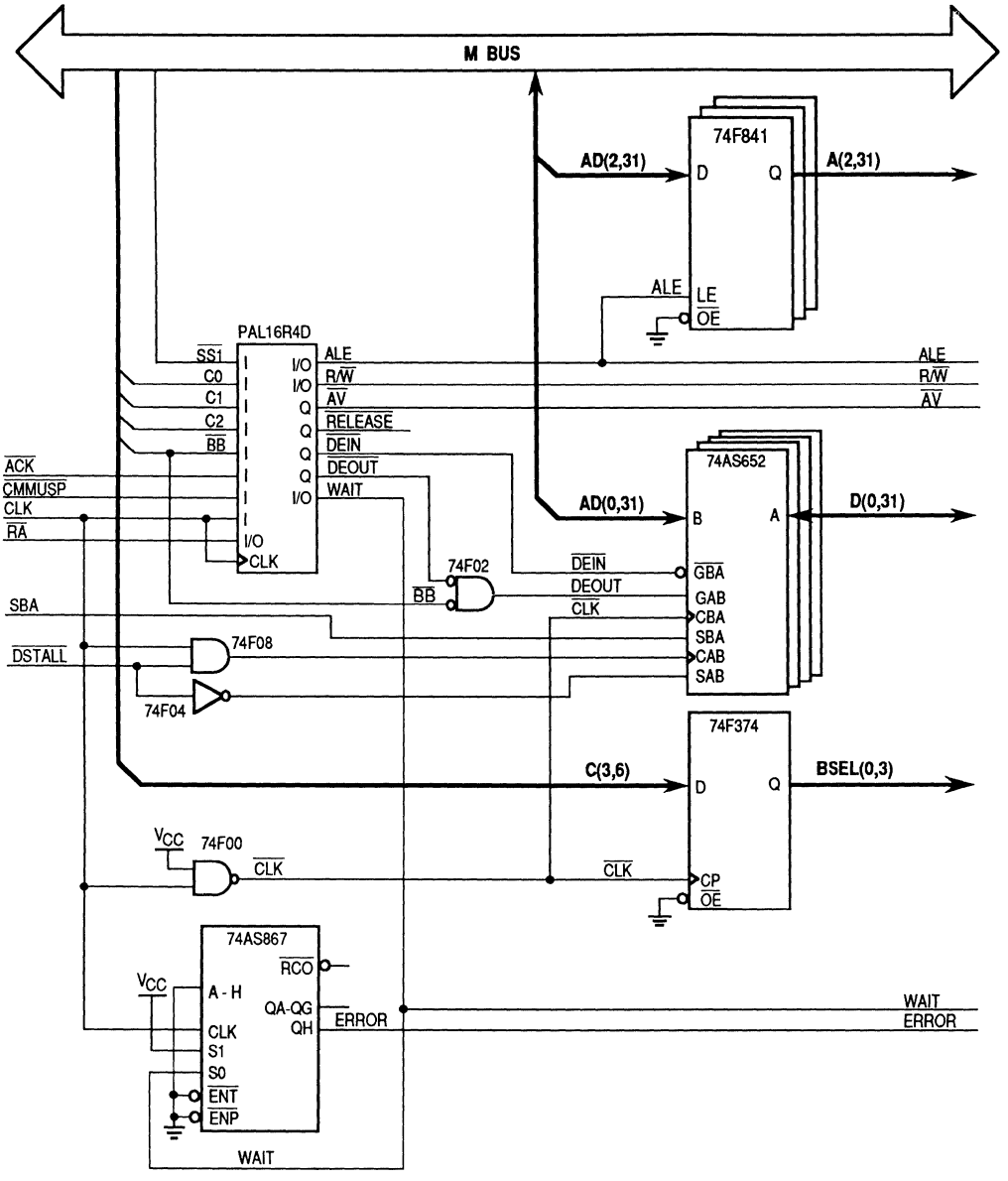


Figure 7-19. M Bus Interface Logic

Figure 7-22 shows the cycle acknowledge consolidation circuit and the address decode PAL. The 74F11 three-input AND gate collects the acknowledge signals from the ROM, MC68681 DUART, and the DRAM sections to generate the $\overline{ACK}$ signal. The PAL generates select signals for the various system resources. Figure 7-23 lists the PAL logic equations

```
PAL16R4D     PAL DESIGN SPECIFICATION
M BUS INTERFACE                                    10/27/88
M88000 M BUS CONTROL
MOTOROLA INC., AUSTIN, TEXAS
CLK /SS1 C0 C1 C2 /BB /ACK /CMMUSP CLK GND
/EN /RA WAIT /DEOUT /DEIN /RELEASE /AV RW ALE VCC

/ALE      =   BB * C0 * /CLK * /RELEASE +      ;Latch on a valid address phase
              /ALE * BB * /RELEASE +           ;Keep address until released
              /ALE * /RA                       ;Keep address if DRAM access

/RW       =   /C2 * BB * C0 +                  ;C2 indicates read/write
              /RW * /ALE                       ;latch read/write with ALE

AV        :=  BB * C0 * /SS1 +                 ;assert from end of A-phase
              AV * C1 * /SS1 *                 ;keep AV until LDT with no wait
               /C0 * BB                        ; on data phase of valid tenure

RELEASE   :=  C1 * /SS1 * /C0 * BB +           ;assert on LDT of valid tenure
              RELEASE * /RA                    ;keep release state if DRAM accessed

DEIN      :=  /RW * /ALE                       ;drive data onto SR Bus of
                                               ; write cycle on data phase

DEOUT     :=  ALE * /C0 * C2 * /CMMUSP +       ;assert on data phase of read
              DEOUT * BB * /C1 +               ;keep it for tenure until LDT
              DEOUT * BB * C1 * SS1            ;or until LDT without wait states

/WAIT     =   /AV +                            ;no waits during non-access
              CMMUSP +                         ;no waits for CMMU access
              ACK                              ;no wait after memory/peripheral responds
              C0 * /RELEASE +                  ;no waits for address phase unless waiting
                                               ; for release
```

Figure 7-20. PAL Logic Equations for M Bus Interface

for the address decode, which generates the $\overline{\text{ROM}}$, $\overline{\text{DUART}}$, $\overline{\text{DRAM}}$, and $\overline{\text{CMMUSP}}$ select signals. The $\overline{\text{CMMUSP}}$ signal notifies the interface control PAL, shown in Figure 7-19, that the current access is to an MC88200 and that the interface logic should not assert wait cycles. The interface to the MC68681 DUART is a cache-inhibited access. The DUART provides I/O capability for the system by transmitting and receiving characters over RS-232-C serial communication channels. Caching of DUART data causes stale data in the cache since the DUART updates its internal registers independent of M bus activity. The address decode for the $\overline{\text{DUART}}$ signal uses one of the hardwired BATC entries in the MC88200, which automatically inhibits caching of data.

7.2.2 ROM Interface

The ROM interface shown in Figure 7-24 uses four 27256 EPROMs to provide 128K bytes of data. The R/$\overline{\text{W}}$ signal keeps the outputs of the 27256's in high impedance during write operations. The $\overline{\text{ROM}}$ select signal is qualified with the $\overline{\text{AV}}$ signal using a 74F32 two-input OR gate to generate the $\overline{\text{ROMCS}}$ signal. After the $\overline{\text{ROMCS}}$ asserts, the 74F166 shift register

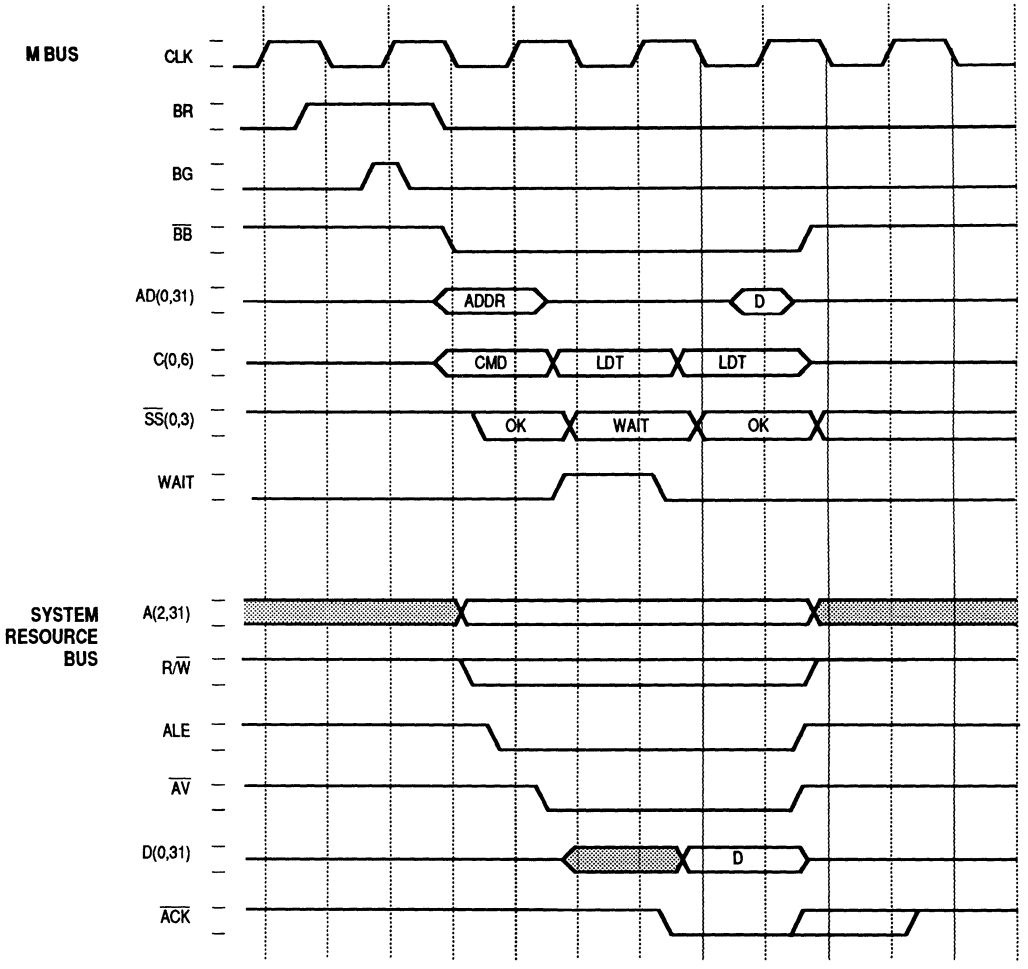


Figure 7-21. M Bus Interface Timing

begins shifting a logic zero through to the Q7 output. After eight clock cycles, the ROMACK signal asserts, which indicates to the M bus interface logic that the D31–D0 contains valid data. The ROM interface does not support cache line-fill operations of the MC88200 and therefore should be configured as cache inhibited when initializing the CMMU.

Figure 7-25 shows an alternate ROM interface supporting cache line-fill operations. The addition of the 74F191 4-bit binary counter allows the interface to increment the A2 and A3 lines connected to the 27256 EPROM. At the beginning of a transaction, the 74F191 is held in the parallel load mode, which passes the A2 and A3 lines from the SR bus directly to the EPROMs. This provides support for single-word read transactions that occur before

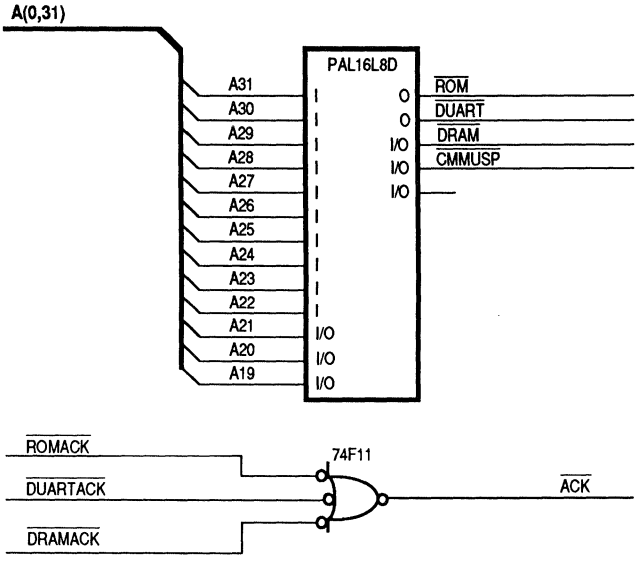


Figure 7-22. Address Decode and Cycle Acknowledge

```
PAL16L8D      PAL DESIGN SPECIFICATION
M BUS INTERFACE                                              7/26/88
M88000 ADDRESS DECODE - CHIP SELECT LOGIC
MOTOROLA INC., AUSTIN, TEXAS
A31 A30 A29 A28 A27 A26 A25 A24 A23 GND
A22 /ROM A21 A20 A19 NC1 /CMMUSP /DRAM /DUART VCC

ROM      =  /A31*/A30*/A29*/28*/A27*/A26*          ;ROM  00000000 - 001FFFFF
            /A25*/A24*/A23*/A22*/A21*/A20*/A19      ;

DUART    =  A31*A30*A29*A28*A27*A26*               ;DUART FFF80000
            A25*A24*A23*A22*A21*A20*A19             ;

DRAM     =  /A31*/A30*/A29*/A28*/A27*/A26*         ;DRAM  00200000 - 005FFFFF
            /A25*/A24*/A23*/A22*A21 +               ;(1st segment 200000 - 3FFFFF)
            /A31*/A30*/A29*/A27*/A26*               ;(2nd segment 400000 - 5FFFFF)
            /A25*/A24*/A23*A22*/A21                 ;

CMMUSP   =  A31*A30*A29*A27*A26*                   ;CMMU space FFF00000-FFF7FFFF
            A25*A24*A23*A22*A21*A20*/A19            ;
```

Figure 7-23. PAL Logic Equations for Address Decode

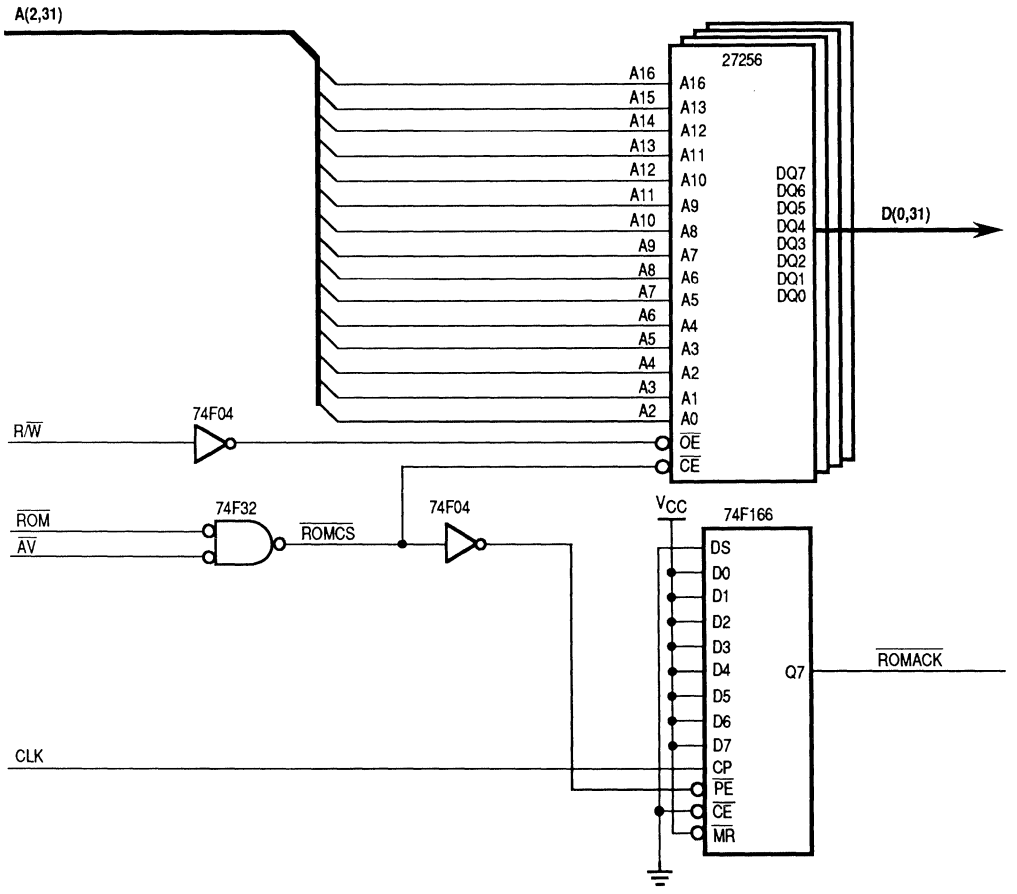


Figure 7-24. ROM Interface

the CMMU is initialized. During cache line-fill operations, the 74F191 is allowed to count to the next sequential address whenever the M bus receives a no-wait status as indicated by the $\overline{SS1}$ signal. The $\overline{SS1}$ signal also reloads the 74F166 shift register allowing access time for the 27256 EPROM. When designing with faster EPROMs, the access time may be adjusted by changing the preload value in the 74F166 shift register.

7.2.3 MC68681 DUART

The MC68681 DUART and its connections to the RS-232-C communication link are shown in Figure 7-26. The serial data from the DUART (TXDx and RXDx) is connected to an MC145406 RS-232-C driver/receiver, which translates TTL signal voltage levels into RS-232-C voltage levels. The clock source for the DUART uses a 3.6864 MHz oscillator, which is also used as a clock source for the DRAM refresh counter (see **7.2.4 DRAM Interface**).

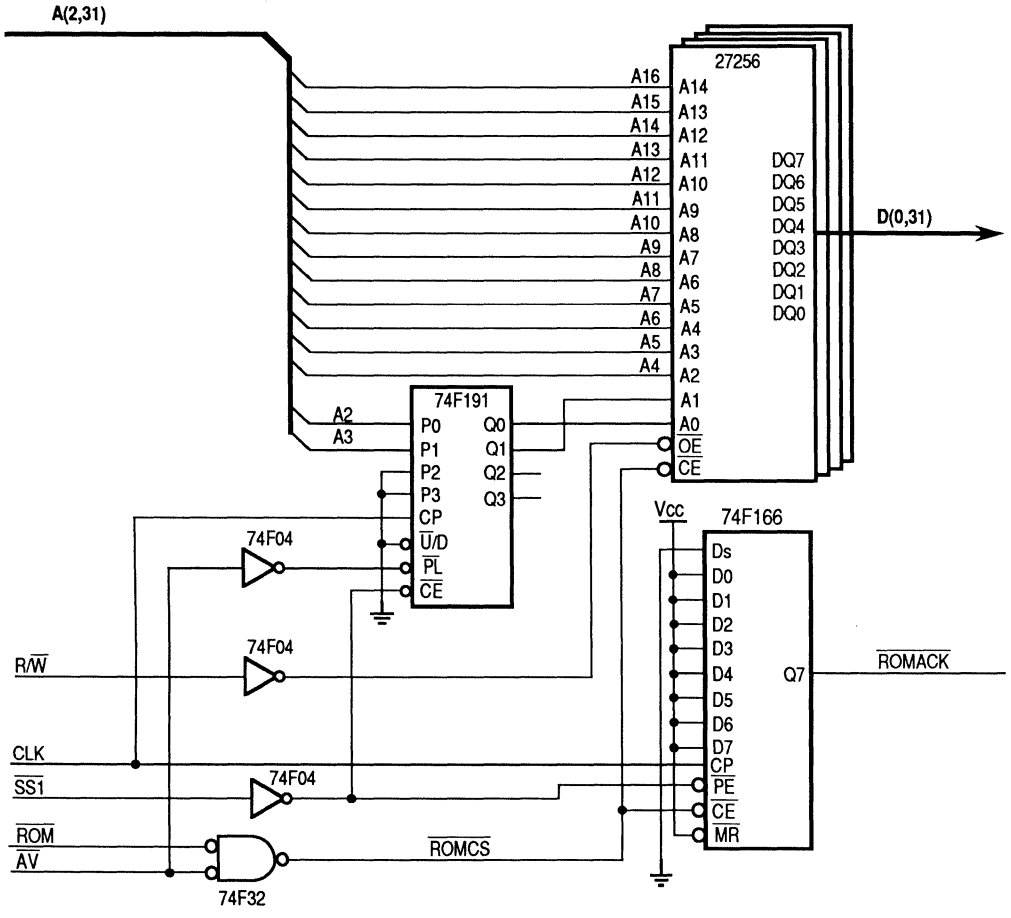


Figure 7-25. ROM Interfacing Supporting Line-Fill Operations

Figure 7-27 shows the interface logic to support the MC68681. A 74F245 transceiver is used to buffer the 8-bit data bus (BD7–BD0) to the DUART. The 74F245 ensures that the data bus (D7–D0) is placed in the high-impedance state after a DUART read operation and before a subsequent write operation can begin. The DUART, which operates at a slower clock speed, can take up to 100 ns to three state its data bus. The R/$\overline{W}$ signal controls the direction of the 74F245. The MC68681 also specifies that the chip select signal ($\overline{CS}$ on the DUART) must negate for at least 90 ns between accesses. To guarantee that a false DUART access does not start, the $\overline{DUART}$ select signal is qualified with the $\overline{AV}$ signal using a 74F32 two-input OR gate. The 74F74 D-type flip-flop and the 74F374 octal D-type flip-flop prevent the $\overline{DUARTCS}$ signal from asserting for four clock periods after a DUART access. The 74F74 catches the rising edge of the qualified $\overline{DUART}$ select signal, marking the end of one DUART access. The 74F374 then begins clocking a logic zero through to the Q3 output. After four rising edges of the clock input, the 74F74 is preset. This allows the $\overline{DUARTCS}$ signal to

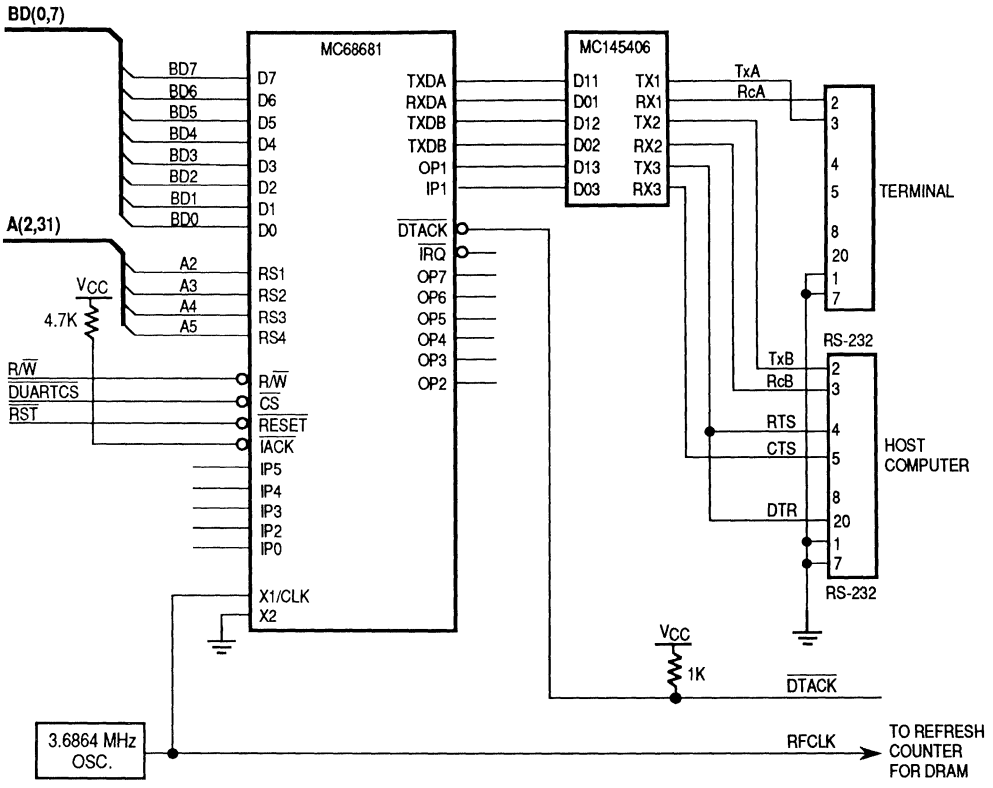


Figure 7-26. MC68681 DUART

7

assert if a qualified $\overline{\text{DUART}}$ select signal is present. The access delay mechanism created by the 74F74 and the 74F374 guarantees that the $\overline{\text{DUARTCS}}$ remains high for the specified 90 ns. This hardware delay mechanism may be eliminated if the MC88100/MC88200 processing node is the only M bus master that accesses the DUART. Since DUART accesses are not cached, the MC88100 cannot normally access the DUART fast enough to violate the chip-select-high time specification. For this case, qualifying the $\overline{\text{DUART}}$ select signal with the $\overline{\text{AV}}$ signal is sufficient to meet the specification.

The DUART acknowledges an access by asserting the $\overline{\text{DTACK}}$ signal. Since the DUART is operating asynchronous to the M88000, the $\overline{\text{DTACK}}$ response signal must be synchronized to the M bus activity. The 74F374 provides a two-level synchronizer for the $\overline{\text{DTACK}}$ signal. The first level synchronizes the assertion of $\overline{\text{DTACK}}$ to the system clock. The output of the first level may develop a metastable condition since the input may violate the input setup time of the 74F374. The second level allows any metastable conditions to settle out before acknowledging the access to the M88000. The synchronized acknowledge signal is then qualified with the $\overline{\text{DUARTCS}}$ signal to generate the $\overline{\text{DUARTACK}}$ signal.

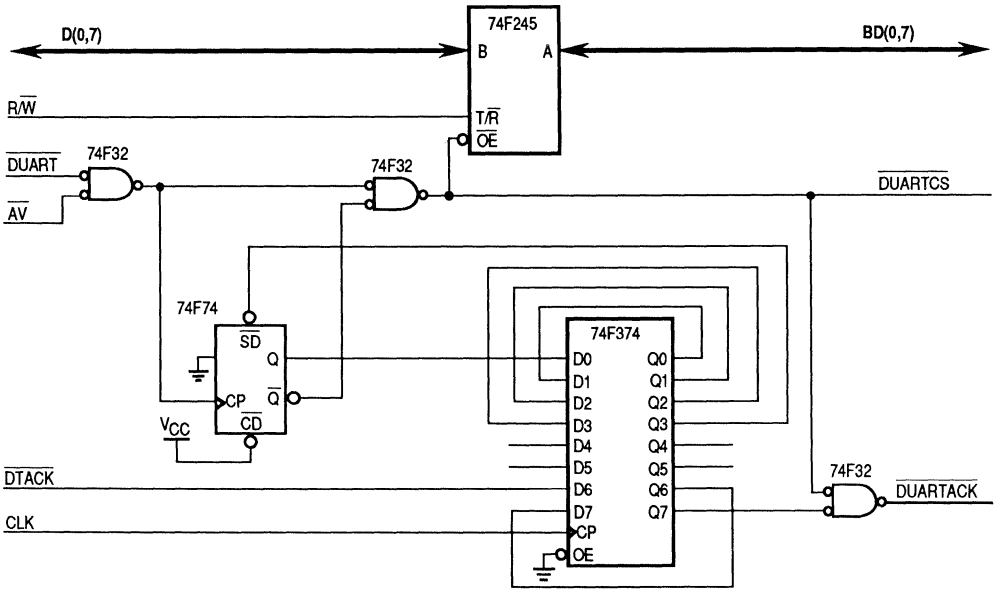


Figure 7-27. MC68681 Interface Logic

7.2.4 DRAM Interface

The 20-MHz nibble-mode DRAM controller design concept presented here incorporates a burst mode to increase system throughput for the M88000. The DRAM controller and memory system are implemented using PALs, discrete hardware, and 1M-bit nibble-mode DRAMs. While DRAMs typically do not support the maximum-possible performance with the M88000, they do offer the most cost-effective memory solution for system implementations. The effects of adding wait cycles to allow for DRAM access times can be partially offset by using burst-mode operations to support both cache line-fill and copyback operations. The DRAM-based memory system features automatic refresh, a write latch for early write cycle terminations, and support for burst-mode operations.

Figure 7-28 provides a system-level block diagram of the DRAM memory system. The design consists of four major sections: the data registers, the address multiplexers, the DRAM array, and the DRAM control logic.

The data registers use 74F574 octal D-type flip-flops and provide buffering and temporary data storage capability for both read and write operations to the DRAMs. Buffering of the DRAM data input and outputs is required since the D31–D0 signals on the SR bus are used by other system resource devices. The data storage capability allows the DRAMs to minimize cycle times during burst operations for both reads and writes. During burst read cycles, the data-out bus (DO31–DO0) is stored in the 74F574's to allow the DRAMs to cycle to the next sequential memory location without having to wait for the MC88200 to receive

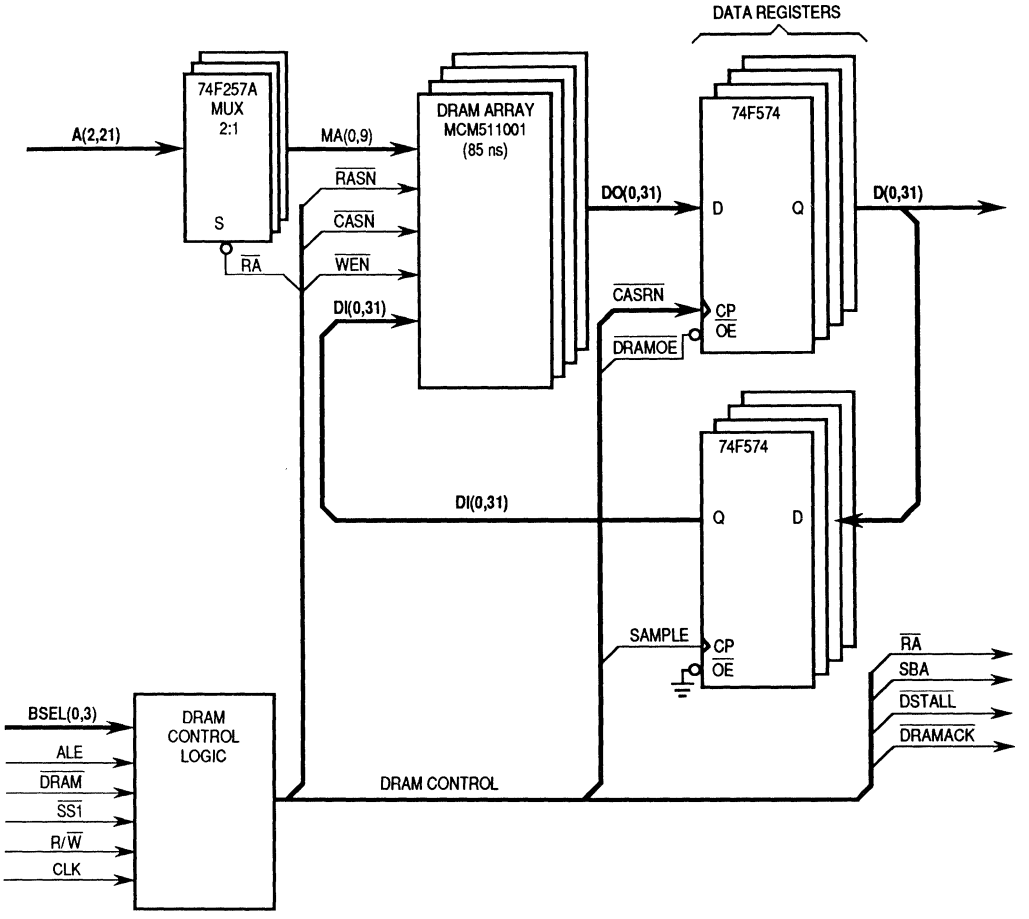


Figure 7-28. DRAM Memory Block Diagram (20 MHz)

the data. This minimizes the amount of time the DRAMs need to hold data valid since the 74F574 meets the specified data hold time for the MC88200. During write cycles, the 74F574 connected to the data-in bus (DI31–DI0) provides storage so that the MC88200 can complete a write operation in less clock periods than a comparable read operation. An entire burst write operation can complete without wait cycles on the M bus.

Address information (A21–A2) is multiplexed by three 74F257A two-line-to-one-line multiplexers. This generates the multiplexed address (MA9–MA0) signals connected to the DRAMs. The switching of row address to column address is controlled by the row address ($\overline{RA}$) signal generated by the DRAM control logic. After the row address information has

been latched by the DRAMs, the $\overline{RA}$ signal is switched to allow column information to propagate through.

The DRAM array consists of 32, 85-ns, 1-Mbit × 1, nibble-mode DRAMs (MCM511001). The MA9–MA0 signals provide row and column address information into the array. The row address strobe ($\overline{RASn}$) signals latch the row address; the column address strobe ($\overline{CASn}$) signals latch the column address. In addition, the $\overline{CASn}$ signals are toggled in the DRAM nibble mode to access the next sequential location during M bus burst operations. The write enable ($\overline{WEn}$) signals provide read/write information while the DI31–DI0 and DO31–DO0 connect to the D inputs and Q outputs of the DRAM, respectively.

Nibble-mode DRAMs are ideally suited to support the burst-mode requirements of the MC88200. When the MC88200 performs a line-fill or copyback transaction, it provides one address phase followed by four data phases. Each data phase represents an access to the next sequential memory location. Nibble-mode DRAMs support this type of access by allowing a fast access to the next memory cell by simply toggling the $\overline{CAS}$ signal. Figure 7-29 shows the functional timing of a nibble-mode DRAM. The first bit is accessed in the normal manner after the row address is latched with the $\overline{RAS}$ signal and the column address is latched with the $\overline{CAS}$ signal. The next sequential nibble bit is accessed by toggling $\overline{CAS}$. After the initial access, the external address signals are not used; therefore, it is important to understand which memory locations are being accessed by the toggling of $\overline{CAS}$.

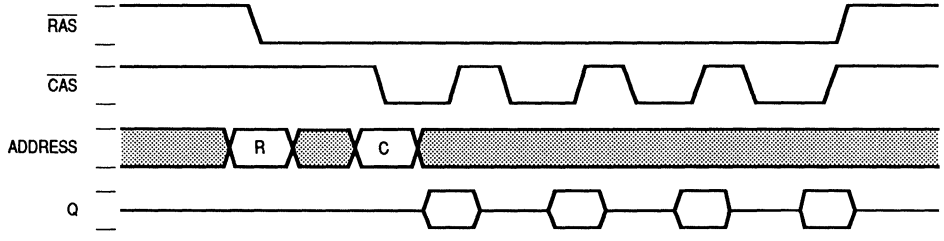


Figure 7-29. Nibble-Mode DRAM Functional Timing

For the 1M-bit DRAMs, the row A9 and column A9 address bits (RA9,CA9) are used for the initial selection, with row A9 being the least significant bit. Subsequently, the falling edge of $\overline{CAS}$ accesses the next bit of the circular 4-bit nibble. Figure 7-30 shows how the internal RA9 and CA9 address bits are incremented by the DRAM. Figure 7-31 shows the connections

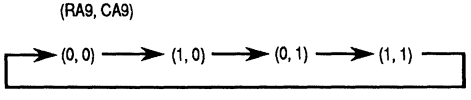


Figure 7-30. Nibble-Mode Bit Access

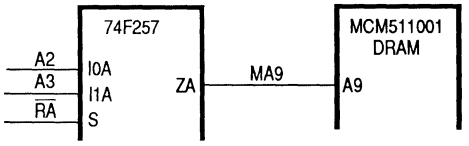


Figure 7-31. Address Multiplexer Connection to DRAM

of the low-order A2 and A3 bits on the SR bus through the multiplexer to the DRAM. The RA signal is initially low to allow the A2 bit to propagate through with the row address. Later in the cycle, the RA signal switches to a logic-high voltage to allow the A3 bit to propagate through with the column address.

Figures 7-32 and 7-33 show the functional DRAM timing for both burst read and write operations implemented by the DRAM control logic. The control state shown at the top of each figure represents the state of the DRAM controller as it progresses through the burst access. The burst read access returns data to the MC88200 in control states 2, 3, 4, and 5, taking a total of seven clock periods to complete. This represents two wait states on the M bus. If the SS1 signal indicates a wait condition, the data stall signal DSTALL in Figure 7-28 is asserted to allow the current M bus data to remain driven. In addition to generating the DSTALL signal, a wait cycle will also stall the state machine from advancing to the next state.

The burst write operation shown in Figure 7-33 operates without wait cycles, which is accomplished by storing write data in two sets of data registers during cache copyback operations. The first set of data registers, consisting of 74AS652 registered transceivers, is contained in the M bus interface (see Figure 7-19). The second set consists of 74F574 D-type flip-flops (see Figure 7-28). These two sets of registers allow the DRAM controller to hide the DRAM access by providing temporary storage of data.

Figure 7-34 shows the DRAM control logic implemented using two 10-ns PALs. The PALs start either a read or write access (indicated by the R/W signal) after sampling a valid DRAM select and ALE signal. In response to a DRAM start condition, the RAS signal is asserted. The CAS signal is asserted later in the access as the PAL advances the state machine defined by the state bits (SQ2–SQ0). The SS1 signal allows the control PAL to recognize M bus wait conditions and stall the state machine until a no-wait status is received. During stall states, the DSTALL signal is asserted to configure the M bus interface register to keep driving the current data on read cycles. The refresh request (RFRQ) signal indicates that a DRAM refresh operation must be performed to preserve memory data. The RFRQ signal is synchronized to generate the RFQ signal used as an input to the controller's state machine. In response to a synchronized refresh request, the refresh grant signal (RFSH) asserts. This allows the PAL to cycle through a CAS-before-RAS refresh operation as defined by the DRAM specification. If an access is attempted during a refresh operation, the access is held in wait cycles until the PAL finishes the refresh. Once the refresh operation begins, the LOAD signal asserts as a hardware handshake back to the refresh request logic. The DRAMACK signal asserts to indicate to the M bus interface logic that the current access is

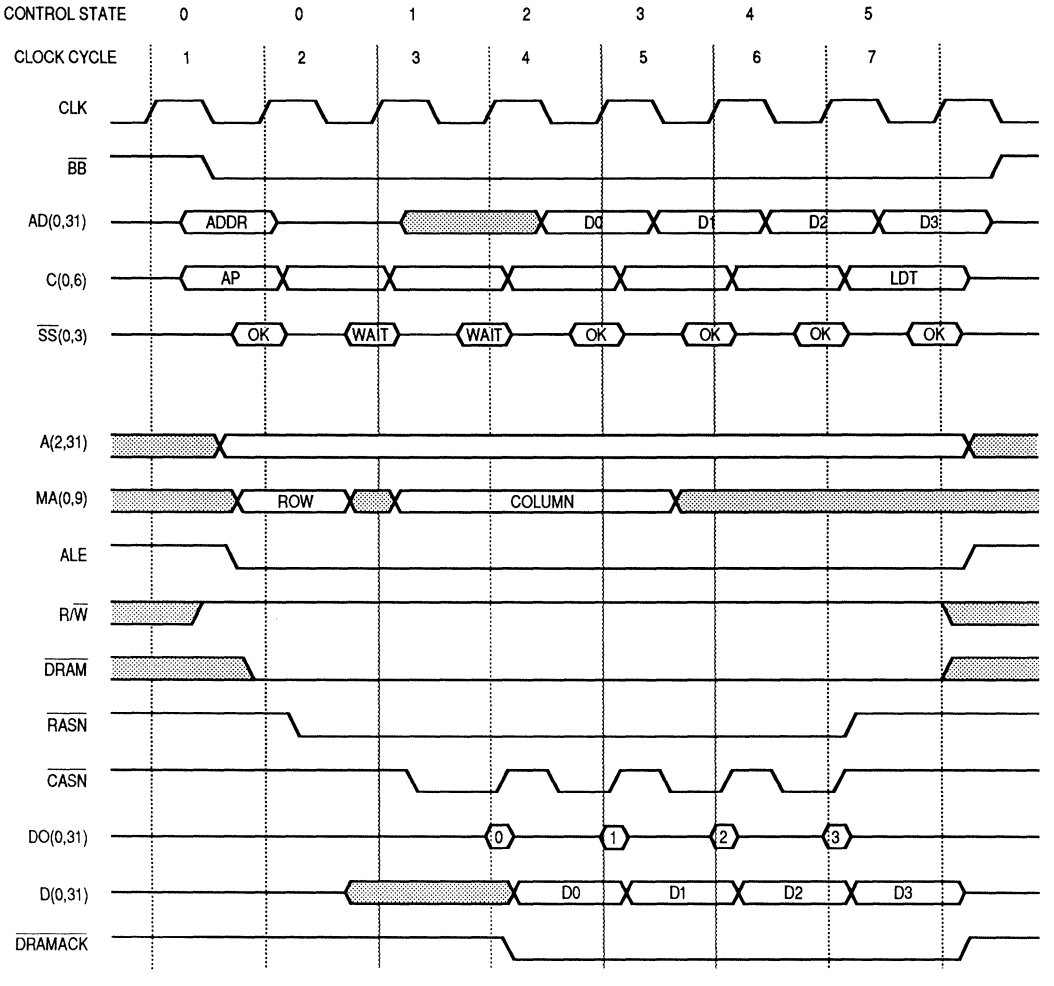


Figure 7-32. DRAM Burst Read Timing (20 MHz)

complete. If the ALE and $\overline{\text{DRAM}}$ signals remain asserted, then the DRAM controller advances to the next memory location. The $\overline{\text{DRAMOE}}$ signal controls the output enable of the data registers connected to the DO31–DO0 bus and is used during read accesses. When asserted, the row address ($\overline{\text{RA}}$) signal indicates that the multiplexers are driving the row address to the DRAMs. When negated, the column address is being driven. The SAMPLE signal allows the 74F574 to capture the BSEL3–BSEL0 signals and allows the data registers connected to the DI31–DI0 bus to capture write data on the SR bus. The BSEL3–BSEL0 signals are used to control the write enable signals ($\overline{\text{WE3}}$–$\overline{\text{WE0}}$) connected to the DRAMs. The logic equations for the PALs at reference designators UA and UB (Figures 7-35 and 7-36) provide more detailed information about each of the individual signals.

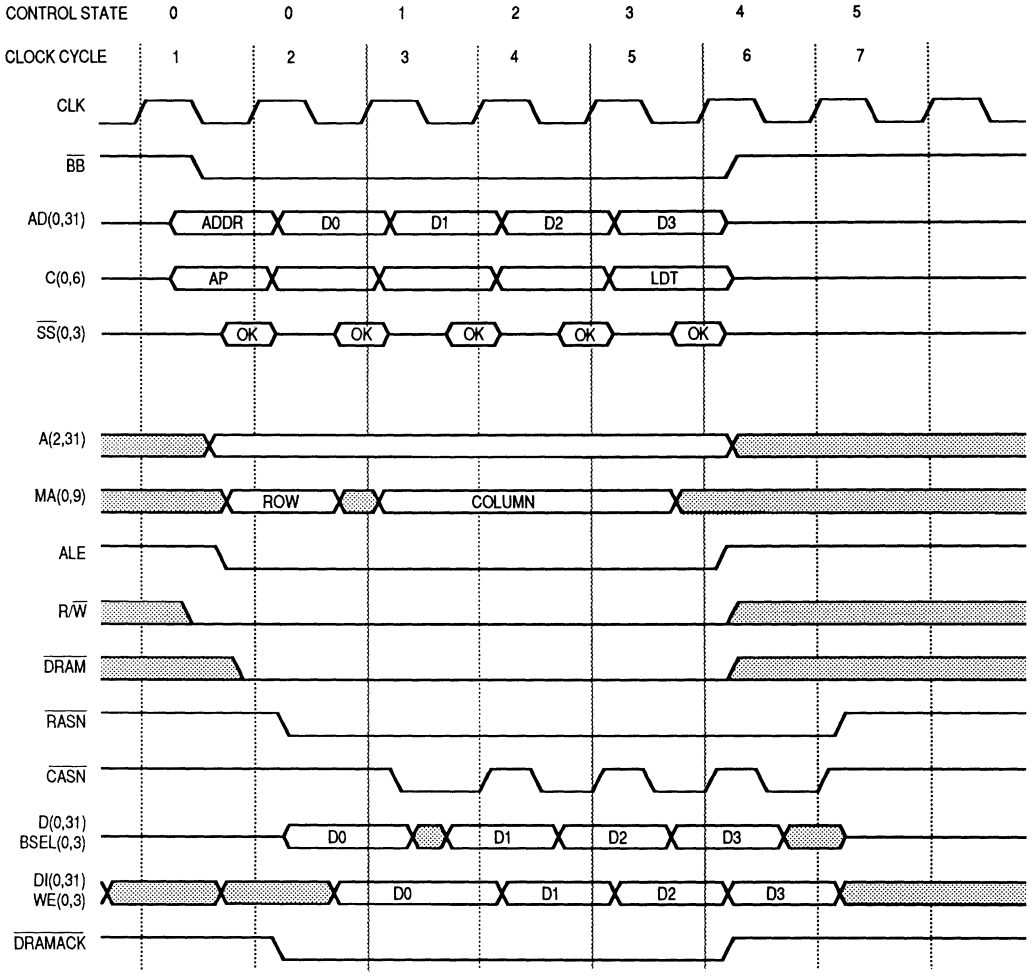


Figure 7-33. DRAM Burst Write Timing (20 MHz)

Figure 7-37 shows the $\overline{RAS}$ and $\overline{CAS}$ control signal drivers into the DRAM array. A 74AS241 buffer is used to drive a separate $\overline{RAS}$ and $\overline{CAS}$ signal to each byte of the memory array. Series terminating resistors are used to avoid ringing problems on these lines. The $\overline{CASR3}$–$\overline{CASR0}$ signals are also used to synchronize the DRAM access with the data registers connected to the output bus of the DRAMs.

Figure 7-38 depicts the refresh request logic. A 74AS867 8-bit counter is used to count clock cycles and generate the RFRQ signal. The $\overline{LOAD}$ handshake signal from the control PAL changes the mode of the counter to preload the value on the parallel inputs (A–H)

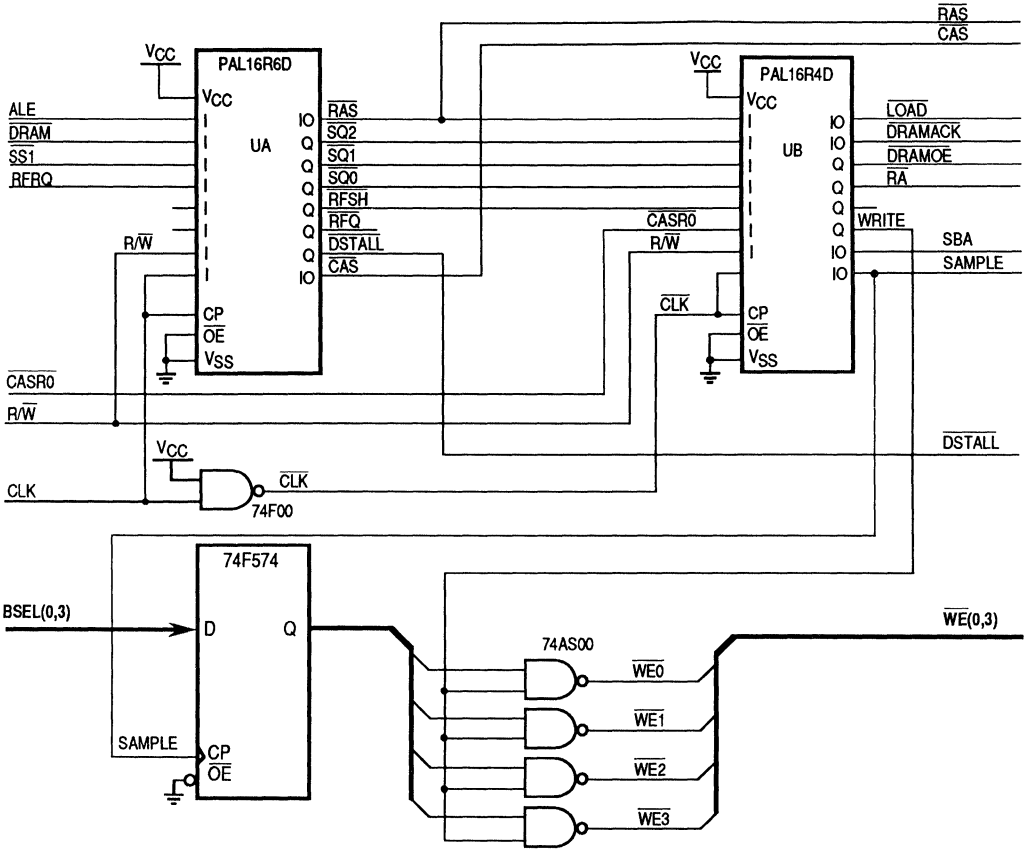


Figure 7-34. DRAM Control Logic

into the counter outputs (QA–QH). When $\overline{LOAD}$ negates, the counter is allowed to begin counting. The 1M-bit DRAM refresh specification requires the DRAM controller to perform 512 refresh cycles every 8 ms. The number of clock periods between refresh cycles is as follows:

$$RC = (\text{frequency}) \times \frac{8\text{ ms}}{512\text{ cycles}}$$

$$= (3.6864\text{ MHz}) \times \frac{8\text{ ms}}{512\text{ cycles}}$$

$$= 57.6\text{ clock periods}$$

where:

RC = the refresh count

```
PARTNO      DRAM CONTROL LOGIC;
NAME        RAS_CAS;
DATE        10/30/88;
REV         *;
DESIGNER    XXXX;
COMPANY     MOTOROLA;
ASSEMBLY    XXX ;
LOCATION    XXX ;

/*****************************************************************/
/* This device controls the RAS, CAS and Refresh operations      */
/* for nibble mode DRAM accesses to support the M88000           */
/*****************************************************************/
/* Allowable Target Device Types : PAL16R6D, TIBPAL16R6-10C      */
/*****************************************************************/
/** Inputs **/
PIN 1     =   clk          ;/* System Clock                      */
PIN 2     =   ALE          ;/* Address Latch Enable              */
PIN 3     =   !DRAM        ;/* DRAM Select signal                */
PIN 4     =   !SS1         ;/* M Bus System Status 1 - Wait signal */
PIN 5     =   RFRQ         ;/* Refresh counter reached 245 counts */
PIN 6     =   NC1          ;/* No Connect                        */
PIN 7     =   NC2          ;/* No Connect                        */
PIN 8     =   RW           ;/* Read - Write signal               */
PIN 9     =   CLK          ;/* System Clock - same as pin 1      */

/** Outputs **/

PIN 12    =   !CAS         ;/* CAS signal to DRAMs               */
PIN 13    =   !DSTALL      ;/* Data Stall signal                 */
PIN 14    =   !RFQ         ;/* Synchronized refresh request      */
PIN 15    =   !RFSH        ;/* DRAM Refresh operation in progress */
PIN 16    =   !SQ0         ;/* State Variable                    */
PIN 17    =   !SQ1         ;/* State Variable                    */
PIN 18    =   !SQ2         ;/* State Variable                    */
PIN 19    =   !RAS         ;/* RAS signal to DRAMs               */

/** Intermediate Equations **/
S0        =   !SQ2 & !SQ1 & !SQ0      ;/* State 0 - 000        */
S1        =   !SQ2 & !SQ1 &  SQ0      ;/* State 1 - 001        */
S2        =   !SQ2 &  SQ1 &  SQ0      ;/* State 2 - 011        */
S3        =   !SQ2 &  SQ1 & !SQ0      ;/* State 3 - 010        */
S4        =    SQ2 &  SQ1 & !SQ0      ;/* State 4 - 110        */
S5        =    SQ2 & !SQ1 & !SQ0      ;/* State 5 - 100        */

WAIT      =   SS1          ;/* M Bus Wait Status                 */
```

Figure 7-35. PAL Logic Equations for UA (Sheet 1 of 2)

```
/** Logic Equations **/
RAS       =  CLK & DRAM & !ALE &          /* assert on CLK high                */
             S0 & !RFSH #                 /*  of S0 if no refresh             */
             !RFSH & RAS & !SQ2 #         /* Keep for S0, S1, S2, S3          */
             !RFSH & RAS & SQ1 #          /* keep for S2, S3, S4              */
             RFSH & SQ1 & SQ0;            /* assert in S1 & S2 of             */
                                          /*   refresh cycle                  */

CAS       =  !RFSH & S1 #                 /* assert in state 1                */
             !RFSH & SQ1 &                /*  toggle in S2, S3, and S4        */
             !CLK & !DSTALL #             /*  if no data stall condition/
             RFSH & !SQ1;                 /* assert for refresh S0, S1        */

RFSH.d    =  RFQ & S0 & ALE #             /* refresh ok if no ALE or          */
             RFQ & S0 & !DRAM #           /*  DRAM not selected               */
             RFSH & SQ0;                  /* keep it for S1 and S2            */

DSTALL.d=  WAIT & RAS & !RFSH & SQ1;      /* record M Bus wait info           */

RFQ.d     =  RFRQ;                        /* synchronize refresh request*/

/* State Machine advances to next state if the request is still     */
/* active (as indicated by ALE) and a wait state is not forced on   */
/* the M Bus.  Refresh cycles only go through states S0, S1, S2, S3 */

SQ2.d     =  !RFSH & !WAIT & S3 & ALE #   /* look at wait to go to S4         */
             !RFSH & S4 & ALE #           /*                                  */
            !RFSH & S5 & RW & WAIT;       /* loop S5 if read waiting          */

SQ1.d     =  !RFSH & RW & S1 & ALE #      /* reads advance to S2              */
             !RFSH & !RW & !WAIT          /* writes look at wait to           */
                    S1 & ALE              /*    advance to S2                 */
             !RFSH & S2 & ALE #           /* keep SQ1 in S2                   */
             !RFSH & S3 & ALE #           /* keep SQ1 in S3                   */
             !RFSH & RW & WAIT &          /* reads depend on wait in          */
                    S4 & ALE              /*   state S4                       */
             RFSH & S1 #                  /* Refresh cycles through           */
             RFSH & S2;                   /*  states S0, S1, S2, S3           */

SQ0.d     =  !RFSH & !RW & !WAIT &        /* move to S1 on writes if          */
                    RAS & S0 & ALE #      /*   not waiting                    */
             !RFSH & RW &                 /* reads do not depend on           */
                    RAS & S0 & ALE #      /*   waits to go to S1              */
             !RFSH & S1 & ALE             /* keep SQ0 if state S1             */
             !RFSH & S2 & WAIT & ALE #    /* stay in S2 if waiting            */
             RFSH & S0 #                  /* Refresh cycles through           */
             RFSH & S1; /* states S0, S1, S2, S3  */
```

Figure 7-35. PAL Logic Equations for UA (Sheet 2 of 2)

```
PARTNO      DRAM CONTROL LOGIC;
NAME        DRAM_ACK;
DATE        10/30/88;
REV         *;
DESIGNER    XXXX;
COMPANY     MOTOROLA;
ASSEMBLY    XXX ;
LOCATION    XXX ;

/*****************************************************************/
/* This device generates control signals for the refresh counter,        */
/* data buffers, and address multiplexers to interface nibble    */
/* mode DRAMs to the M88000                                              */
/*****************************************************************/
/* Allowable Target Device Types : PAL16R4D, TIBPAL16R4-10C              */
/*****************************************************************/
/** Inputs **/
PIN 1       =   !clk            ;/* inverted system clock                */
PIN 2       =   !RAS            ;/* RAS signal to DRAMs                  */
PIN 3       =   !SQ2            ;/* State Variable                       */
PIN 4       =   !SQ1            ;/* State Variable                       */
PIN 5       =   !SQ0            ;/* State Variable                       */
PIN 6       =   !RFSH           ;/* Refresh cycle in progress            */
PIN 7       =   !CASR0          ;/* CAS signal to output buffer          */
PIN 8       =   RW              ;/* Read/Write signal                    */
PIN 9       =   !CLK            ;/* inverted system clock                */

/** Outputs **/

PIN 12      =   SAMPLE          ;/* Sample data and byte select terms    */
PIN 13      =   SBA             ;/* Select real time or stored data      */
PIN 14      =   WRITE           ;/* captured write signal                */
PIN 15      =   NC1             ;/* no connect                           */
PIN 16      =   !RA             ;/* Mux select - row/column address      */
PIN 17      =   !DRAMOE         ;/* Output enable from DRAM (reads)      */
PIN 18      =   !DRAMACK        ;/* DRAM Acknowledge signal              */
PIN 19      =   !LOAD           ;/* preload refresh counter              */

/** Intermediate Equations **/
S0          =   !SQ2 & !SQ1 & !SQ0      ;/* State 0 - 000               */
S1          =   !SQ2 & !SQ1 &  SQ0      ;/* State 1 - 001               */
S2          =   !SQ2 &  SQ1 &  SQ0      ;/* State 2 - 011               */
S3          =   !SQ2 &  SQ1 & !SQ0      ;/* State 3 - 010               */
S4          =    SQ2 &  SQ1 & !SQ0      ;/* State 4 - 110               */
S5          =    SQ2 & !SQ1 & !SQ0      ;/* State 5 - 100               */
```

Figure 7-36. PAL Logic Equations for UB (Sheet 1 of 2)

```
/** Logic Equations **/
!SAMPLE     =   CLK #                          /* toggle with clock info or          */
                CASR0;                         /* toggle with DRAM CAS signal/

!SBA        =   RFSH #                         /* real data during refresh           */
                S0 #                           /* real data in state S0              */
                S5 #                           /* switch back in S5                  */
                !WRITE;                        /* real data during reads             */

RA.d        =   RAS & !RFSH & S0               /* select row until RAS               */
            =   RA & S1 & !RFSH;               /* keep column until S2               */

DRAMOE.d    =   !RFSH & !WRITE & SQ0 #         /* S1,2 of Read access                */
                !RFSH & !WRITE & SQ1 #         /* S2,3,4 of Read access              */
                !RFSH & !WRITE & SQ2;          /* S4,5 of Read access                */

WRITE.d     =   !RFSH & S0 & !RW #             /* sample RW in S0                    */
                !RFSH & WRITE & S1 #           /* keep for S1 through S4             */
                !RFSH & WRITE & S2 #           /*                                    */
                !RFSH & WRITE & S3 #           /*                                    */
                !RFSH & WRITE & S4;            /*                                    */

DRAMACK     =   !RFSH & !WRITE & SQ1 #         /* S2,3,4 of read access              */
                !RFSH & !WRITE SQ2 #           /* S4,5 of read access                */
                !RFSH & !RW & RAS & !SQ2;      /* Ack from RAS on writes             */

LOAD.d      =   RFSH & S1; /* assert Load in S1 of refresh/
```

Figure 7-36. PAL Logic Equations for UB (Sheet 2 of 2)

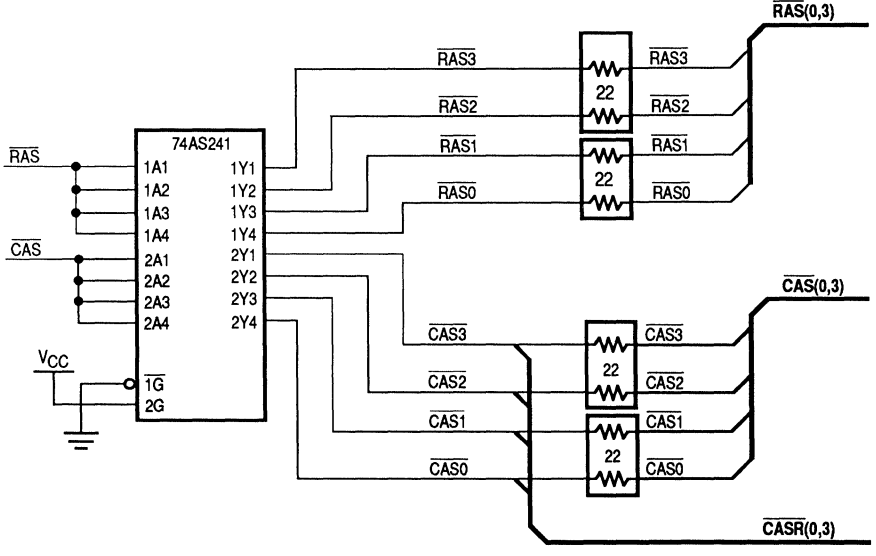


Figure 7-37. DRAM Control Signal Drivers

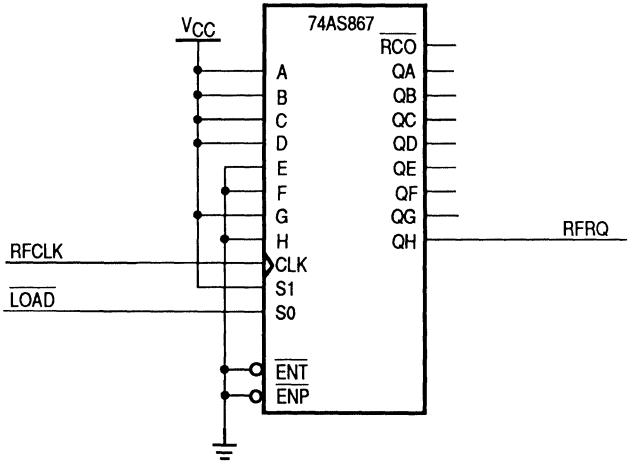


Figure 7-38. Refresh Request Counter

Any delay from the time that refresh is requested to the time a refresh actually occurs must be subtracted from the refresh count. If a refresh is requested just as the processor begins a burst operation, the refresh may be delayed by eight clock cycles. The preload value for the 74AS867 counter can then be calculated as follows:

$$\begin{aligned}\text{Preload Value} &= \text{QH Count} - (\text{RC} - \text{delay count}) \\ &= 128 - (57.6 - 8) \\ &= 78.4\end{aligned}$$

The preload value is then rounded up and converted to hexadecimal to give a value of $4F. This value is hardwired to the inputs of the 74AS867 refresh counter.

7

7.3 MULTIPLE M BUS CONNECTIONS

Figure 7-2 shows a multiple M bus configuration using eight CMMUs and an M bus transceiver section for communication between the two M buses. The transceivers allow the two M bus segments to be treated as one M bus unit. For 20-MHz operation, a single ported system resource section connected to one M bus segment may allow enough propagation time for the transceiver section. For higher frequency designs, dual porting of the system resources may be necessary to hide the propagation delay due to the transceivers.

7.3.1 Transceiver Interface

Figure 7-39 shows the M bus transceiver section. The AD31–AD0, C6–C0, control parity (CP), and address/data parity (ADP3–ADP0) signals from one M bus segment are connected through a 74AS652 transceiver to the other M bus segment. The 74AS652 provides equal

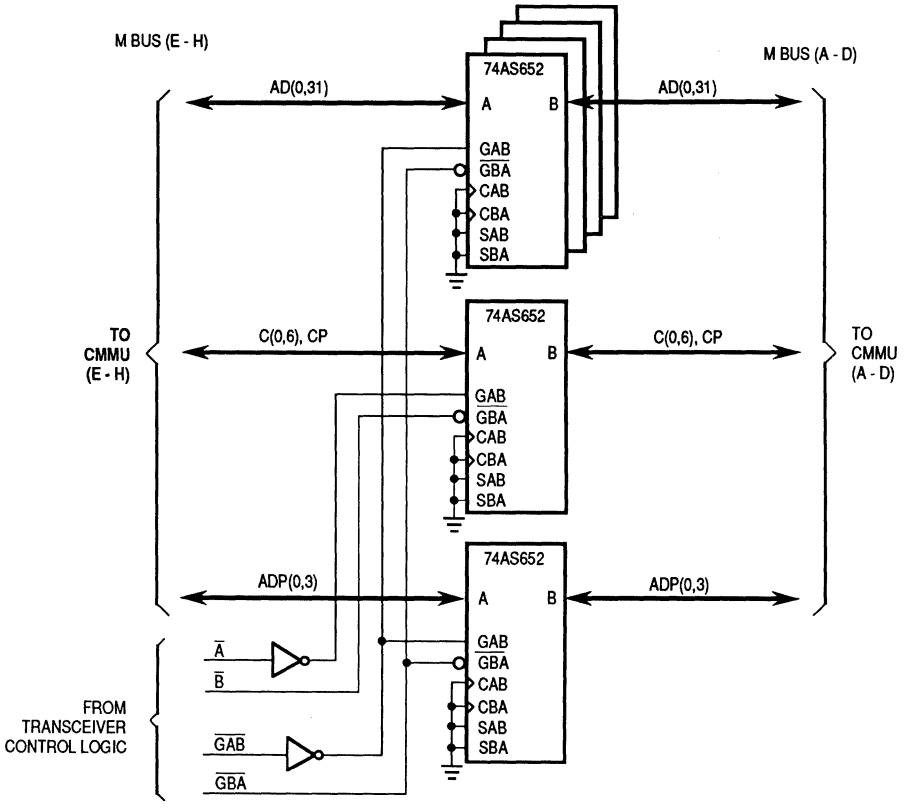


Figure 7-39. M Bus Transceivers

drive capability for both the A and the B output drivers, unlike a 74F245-type transceiver that has a reduced drive specification for the A drivers. In addition, the turnaround propagation time for the direction control signal (DIR) is not specified in the 245 series (74F245, 74AS245, 74F645, 74AS645, etc). The 74AS652 used in this design provides separate output enable controls and specifies output enable and disable times. The output enable signals ($\overline{A}$, $\overline{B}$, $\overline{GAB}$ and $\overline{GBA}$) connect to the transceiver control logic shown in Figure 7-40. A 10-ns PAL16R4D is used to control the direction of the transceivers. The BRA–BRH signals generate the two request signals ($\overline{REQB}$ and $\overline{REQA}$), which the control PAL uses in conjunction with the AB signal to determine which M bus segment is master for the current M bus tenure. The M bus C6–C0 and CP signals are always driven away from the M bus master side of the transceiver. Direction control for the AD31–AD0 and ADP3–ADP0 signals requires the control PAL to know which side of the transceiver the addressed M bus slave is connected to. Address information is decoded to produce a SLAVEA/$\overline{B}$ signal to notify the PAL about the physical location of the slave. The SLAVEA/$\overline{B}$ signal must indicate the slave side of the transceiver for both the CMMU control space and peripheral accesses.

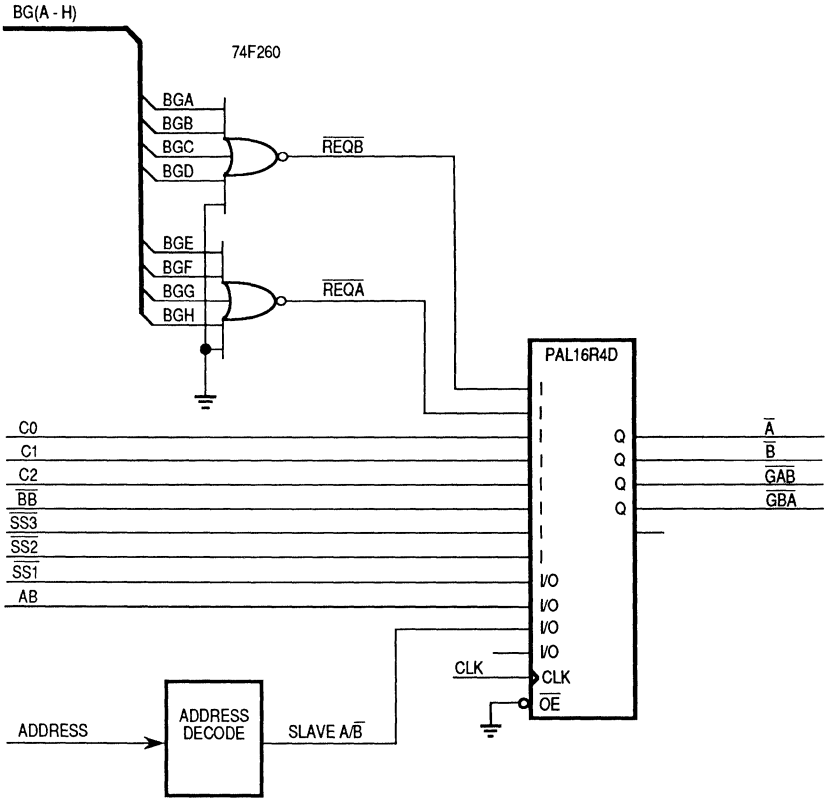


Figure 7-40. M Bus Tranceivers Control Logic

7

Direction control of the transceivers is then determined from state information inside the PAL. This design concept does not support an EOD response from an M bus slave.

Figure 7-41 shows the state diagram used by the PAL implementation to control the output enables of the transceivers. The state machine changes on the rising edge of the system clock signal and controls the $\overline{GAB}$ and $\overline{GBA}$ signals of the transceivers. State S0 corresponds to the no M bus master state; $\overline{BB}$ signal is negated. States S1A and S1B correspond to a address phase, data phase write transactions, M bus master and slave on the same side of the transceiver, or idle phases between transactions during the same M bus tenure. States S2A and S2B correspond to data phases for read transactions where the slave device is on the opposite side of the transceiver. Sample PAL equations to implement the state machine are shown in Figure 7-43.

7.3.2 Dual-Ported System Resources

Figure 7-43 shows a dual port to the system resource bus. The $\overline{A}$ and $\overline{B}$ signals from the M bus transceiver section (Figure 7-40) control which side is active. Both the $\overline{CMMUSP}$

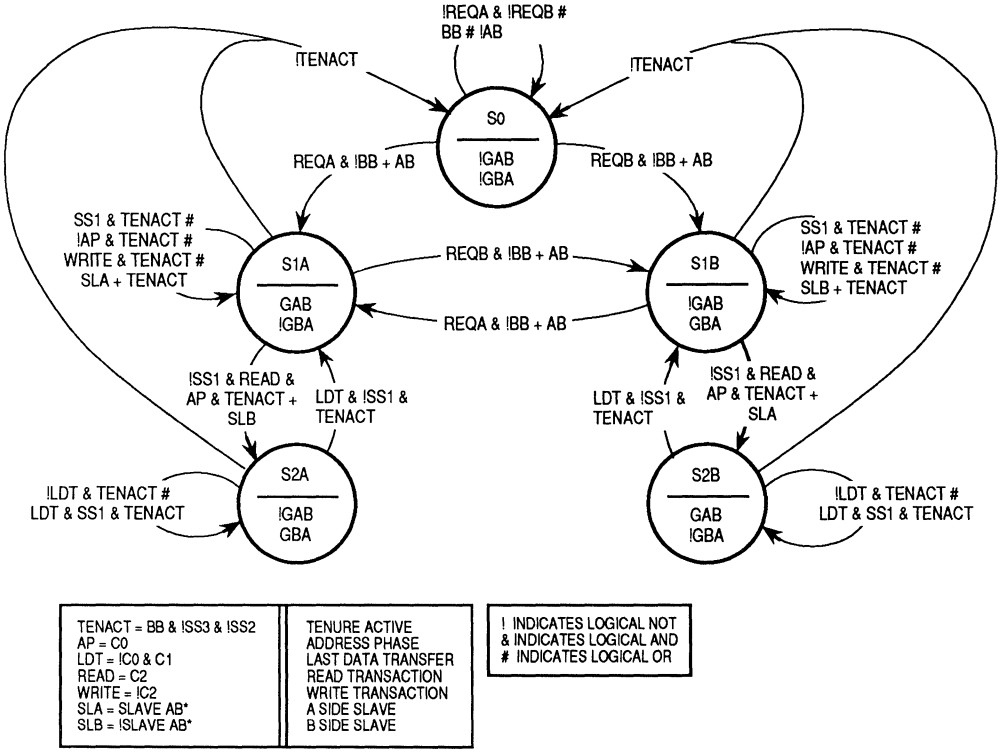


Figure 7-41. Transceiver Control State Diagram

7

and SLAVEA/$\overline{B}$ input signal are address decoded signals. The $\overline{CMMUSP}$ signal indicates that the current access is within the CMMU internal register space and that the system resources should stay of the M bus. The SLAVEA/$\overline{B}$ signal indicates which side of the M bus transceiver block connects to the currently addressed M bus slave. Since the system resources are dual ported, the SLAVEA/$\overline{B}$ signal only has meaning when the $\overline{CMMUSP}$ signal is asserted. The dual port allows the system resources to dynamically switch to the current M bus master side of the transceivers avoiding any extra propagation delay through the transceiver section. The PAL equations for the transceiver control (Figure 7-42) would have to be modified to prevent progression into the S2A or S2B states (Figure 7-41) unless there was an access within the CMMU control register (as indicated by the SLAVEA/$\overline{B}$ signal). The M bus transceivers would still progress into the S1A and S1B states to allow M bus snooping during read and write transactions.

The control signals developed for the single-port M bus interface shown in Figure 7-19 would still be utilized with the modifications shown in Figure 7-44. The $\overline{DEOUT}$ signal is used in conjunction with the $\overline{BB}$ signal, and the $\overline{A}$ or $\overline{B}$ signals to control which port drives

```
PARTNO      DRAM CONTROL LOGIC;
NAME        M_BUS_CONTROL;
DATE        11/04/88;
REV         *;
DESIGNER    XXXX;
COMPANY     MOTOROLA;
ASSEMBLY    XXX ;
LOCATION    XXX ;

/*****************************************************************/
/* This device controls the transceiver output enable signals    */
/* when connecting multiple M Buses together                     */
/*****************************************************************/
/* Allowable Target Device Types : PAL16R4D, TIBPAL1646-10C      */
/*****************************************************************/
/** Inputs **/
PIN 1       =   clk                 ;/* System Clock                    */
PIN 2       =   !REQB               ;/* Request from B side of 74AS652  */
PIN 3       =   !REQA               ;/* Request from A side of 74AS652  */
PIN 4       =   C0                  ;/* Address phase (AP)              */
PIN 5       =   C1                  ;/* Last Data Transfer (LDT)        */
PIN 6       =   C2                  ;/* Read/Write signal               */
PIN 7       =   !BB                 ;/* Bus Busy signal                 */
PIN 8       =   !SS3                ;/* System Status 3 (ERROR)         */
PIN 9       =   !SS2                ;/* System Status 2 (RETRY)         */
PIN 12      =   !SS1                ;/* System Status 1 (WAIT)          */

/** Outputs **/

PIN 13      =   AB                  ;/* Arbitration Busy Signal         */
PIN 14      =   !GBA                ;/* Enable B-to-A output            */
PIN 15      =   !GAB                ;/* Enable A-to-B output            */
PIN 16      =   !B                  ;/* B side of 74AS652 is M Bus master */
PIN 17      =   !A                  ;/* A side of 74AS652 is M Bus Master */
PIN 18      =   SLAVEAB*   ;/* Slave A/B* Signal                       */
PIN 19      =   NC1                 ;/* no connect                      */

/** Intermediate Equations **/
                                    /* Internal Positive Logic */
                                    /* State =   A   B  GAB  GBA  */
S0          =   !A & !B & !GAB & !GBA;/*    S0    0   0   0    0  */
S1A         =    A & !B &  GAB & !GBA;/*   S1A    1   0   1    0  */
S2A         =    A & !B & !GAB &  GBA;/*   S2A    1   0   0    1  */
S1B         =   !A &  B & !GAB &  GBA;/*   S1B    0   1   0    1  */
S2B         =   !A &  B &  GAB & !GBA;/*   S2B    0   1   1    0  */

ERROR       =   SS3;
RETRY       =   SS2;
WAIT        =   SS1;
AP          =   C0;
EOR         =   !C0 * C1;
READ        =   C2;
WRITE       =   !C2;
TENACT      =   BB & !ERROR & !RETRY;
SLA         =   SLAVEAB*
SLB         =   !SLAVEAB*
```

Figure 7-42. PAL Logic Equations for Transceiver Control (Sheet 1 of 2)

```
/** Logic Equations **/
A.D      =  REQA & !BB + AB #          /* assert on request and not busy and active arbitration */
            S1A & TENACT #             /* keep for states S1A and S2A                           */
            S2A & TENACT;              /*  unless tenure ends                                   */

B.D      =  REQB & !BB + AB #          /* assert on request and not busy and active arbitration */
            S1B & TENACT #             /* keep for states S1B and S2B                           */
            S2B & TENACT;              /*  unless tenure ends                                   */

GAB.D    =  REQA & !BB + AB #          /* assert on request and not busy and active arbitration */
            S1A & !AP & TENACT #       /* keep while waiting for address                        */
            S1A & WRITE & TENACT #     /* keep for write transactions                           */
            SIA + SLA + TENACT #       /* keep for master and slave on same side                */
            GAB & WAIT & TENACT #      /* keep while wait is asserted                           */
            S1B & !WAIT & READ & AP    /* assert after address phase                            */
             & TENACT #                /*                                                       */
            S2B & !LDT & TENACT;       /* keep until Last Data Transfer                         */

GBA.D    =  REQB & !BB + AB #          /* assert on request and not busy and active arbitration */
            S1B & !AP & TENACT #       /* keep while waiting for address                        */
            S1B & WRITE & TENACT #     /* keep for write transactions                           */
            SIA + SLB + TENACT #       /* keep for master and slave on same side                */
            GBA & WAIT & TENACT #      /* keep while wait is asserted                           */
            S1A & !WAIT & READ & AP    /* assert after address phase                            */
             & TENACT #                /*                                                       */
            S2A & !LDT & TENACT;       /* keep until last data transfer                         */
```

Figure 7-42. PAL Logic Equations for Transceiver Control (Sheet 2 of 2)

the M bus segment on read transactions. Similarly, the $\overline{DEIN}$ signal is used in conjunction with the $\overline{A}$ and $\overline{B}$ signals to control which data port is active on write transactions, thus avoiding bus contention on the system resource bus.

7

7.4 POWER AND GROUND CONSIDERATIONS

The MC88200 is fabricated in Motorola's advanced HCMOS process and offers significant reduced power consumption in comparison to an equivalent NMOS circuit. While the use of CMOS for a device reduces power consumption, the high clock speeds of the MC88200 make the characteristics of power supplied to the device quite important. The power supply must be able to supply large amounts of instantaneous current when the MC88200 is switching logic levels for many circuit nodes. These nodes include both the internal logic and the output drivers for both the P bus and the M bus. The power supply must remain within the rated specification at all times. To meet these requirements, more attention must be given to the power supply connection to the MC88200 than is required for other devices operating at slower clock rates.

To supply a solid power supply interface, 18 V_{CC} pins and 18 GND pins are provided. This provides 11 V_{CC} and 12 GND pins to supply the power for external signal drivers; the

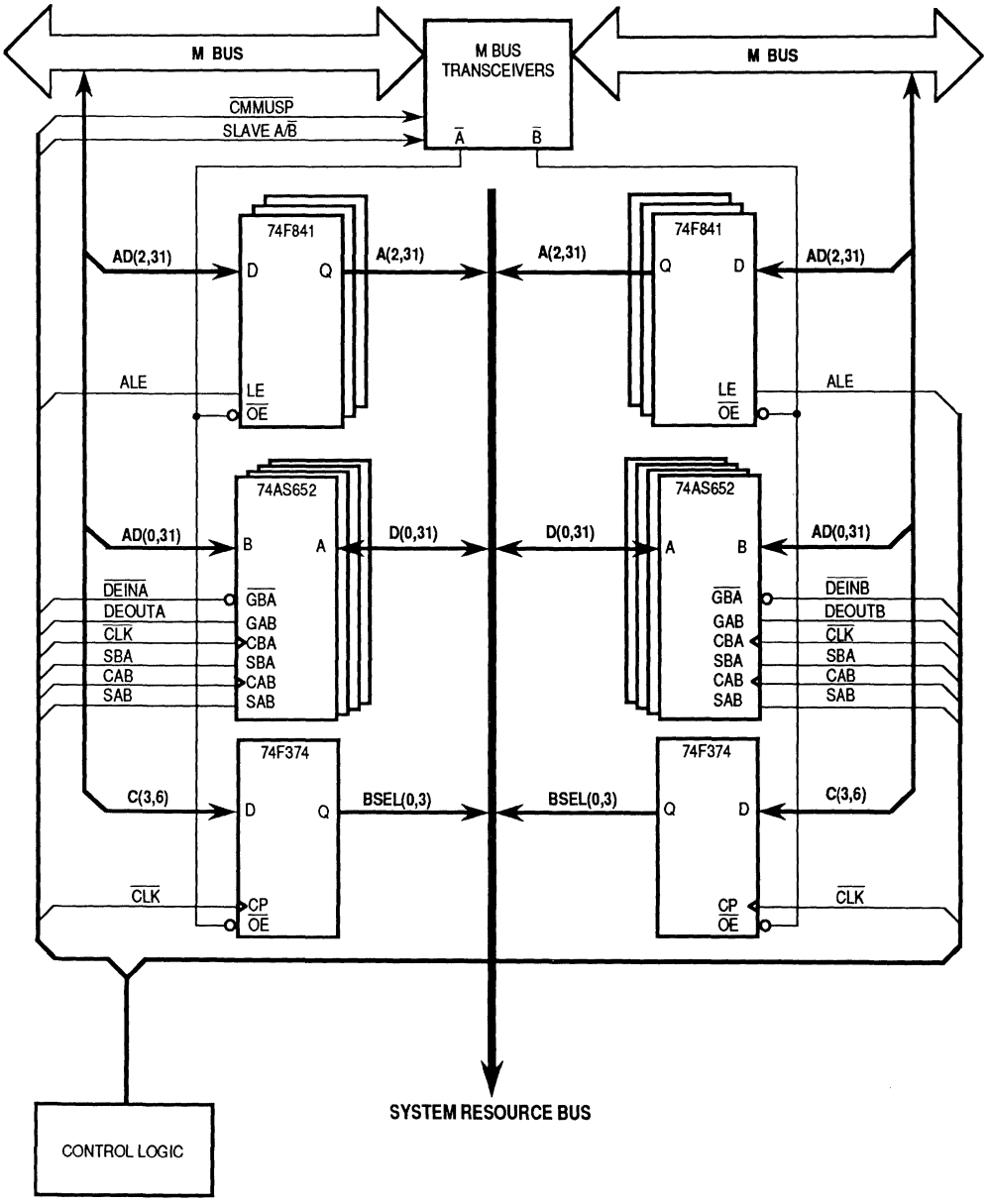


Figure 7-43. Dual-Ported M Bus Interface

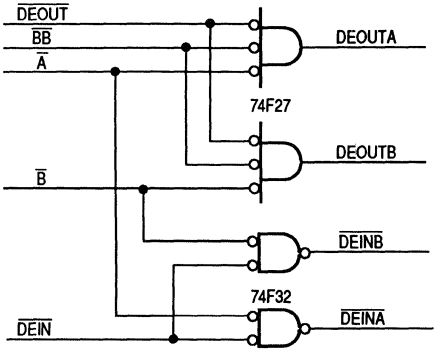


Figure 7-44. Dual-Port Control Signals

remaining seven V_{CC} and six GND pins are used by the internal logic and clock generation circuitry. Table 7-2 lists the V_{CC} and GND pin assignments.

To reduce the amount of noise in the power supplied to the MC88200 and to provide for instantaneous current requirements, common capacitive decoupling techniques should be observed. While there is no recommended layout for this capacitive decoupling, it is essential that the inductance between these devices and the MC88200 be minimized to provide sufficiently fast response time to satisfy momentary current demands and to maintain a constant supply voltage. A combination of low-, middle-, and high-frequency, high-quality capacitors should be placed as close as possible to the MC88200 (e.g., a set of 10 μF, 0.1 mF, and 330 pF capacitors in parallel provides filtering for most frequencies prevalent in a digital system). Decoupling capacitors should be placed as close as posible to the power pins of the MC88200. Consideration should be given to the use of under-socket decoupling capacitor arrays or the use of surface-mount capacitors attached to the solder side of a printed circuit board. Similar decoupling techniques should also be observed for other VLSI devices in the system.

In addition to the capacitive decoupling of the power supply, care must be taken to ensure a low-impedance connection between all MC88200 V_{CC} and GND pins and the system power supply planes. Failure to provide connections of sufficient quality between the MC88200 power supply pins and the system supplies will result in increased assertion delays for external signals, decreased voltage noise margins, and potential errors in internal logic.

Table 7-2. V_{CC} and GND Pin Assignments

Pin Group	V_{CC}	GND
Internal Logic	C7, C9, C11, D5, D13, P5, P13	C6, C8, C10, C12, N4, N14
External Signal Drivers	F3, F15, H3, H15, K3, K15, M3, M15 R7, R9, R11	E4, D14, G3, G15, J3, J15, L3, L15, R6, R8, R10, R12

7.5 CACHE DIAGNOSTICS

The MC88200 provides extensive data cache diagnostic facilities. The diagnostic facilities can determine if a particular cache line is faulty and can disable faulty cache lines to prevent cache data errors. Two levels of cache diagnostics can be performed:

1. Full cache testing using the cache diagnostic ports
2. Data RAM testing using address and control lines

The following paragraphs describe each diagnostic scheme.

7.5.1 Full Cache Diagnostics

The MC88200 register model includes cache diagnostic ports for accessing the cache components. Specifically, the diagnostic registers include:

1. Four ports for writing/reading data words (CDP3–CDP0). These ports are column oriented within a cache set. That is, when a set is selected, the four ports provide access to word 0 of each line of the set, word 1 of each line of the set, etc.
2. Four ports for writing/reading the four address tags in a cache set (CTP3–CTP0).
3. One port for testing the status bits in a cache set (CSSP).

The value written into the SAR (bits 11–2) specifies the cache set and data column. Figure 7-45 shows the relationship of the cache interface ports to the cache components.

Software (or firmware) performs the cache diagnostics by writing to the data, tag, or status ports, then reading the values for comparison. If a comparison error occurs, the software functionally removes the faulty line from the set. Each cache line has an associated disable bit that disables the line when set. When a cache error occurs, the diagnostic software sets this bit, and the faulty line is never used for data caching. The software specifies the faulting cache line by the value written in the system address register. The CSSP contains the disable bits for the cache line.

When performing cache diagnostics on a instruction CMMU, a comparison mismatch can occur with the CSSP. When caching is enabled, instruction fetches from the instruction CMMU cause the LRU algorithm to update the LRU field of the CSSP. Diagnostics should be aware that the LRU bits can change for cache sets used to service instruction fetches.

The cache is organized so that the SAR can be easily incremented to access all cache sets and data words. The sets are aligned on modulo 16 ($10) boundaries; the columns are aligned on modulo 4 boundaries within the set.

More information on the cache interface ports can be found in **3.9 CACHE DIAGNOSTIC PORTS**.

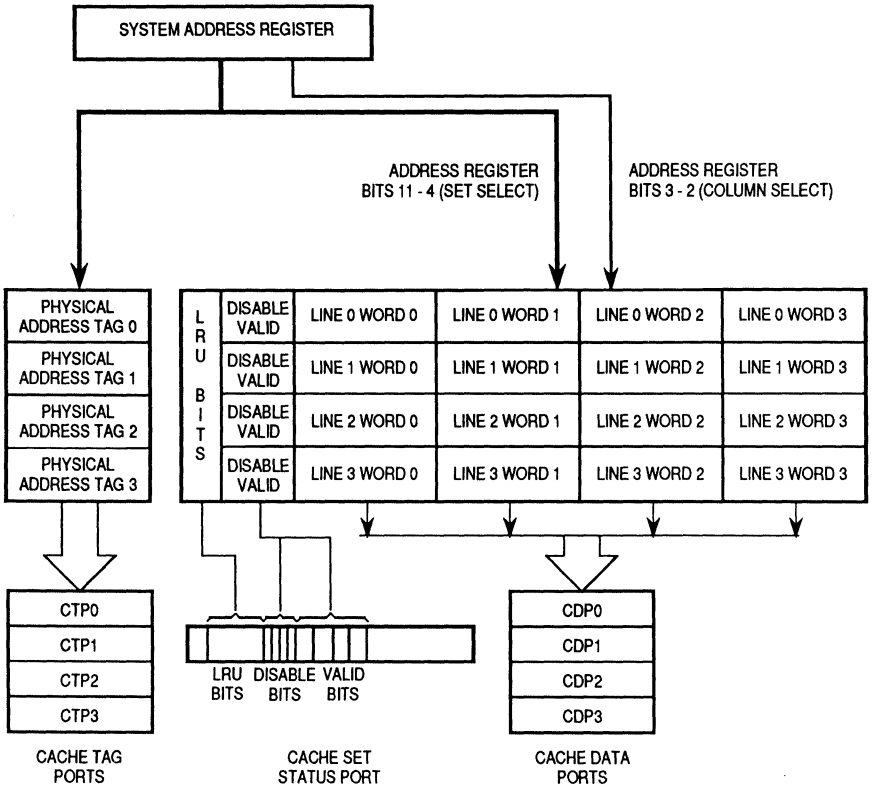


Figure 7-45. Cache Interface Ports

7

7.5.2 Cache Data RAM Diagnostics

Cache data RAM diagnostics test the data lines in the cache set; they ignore the address tags and the status bits. These diagnostics are initiated by the SRAMMODE input signal; when this signal is asserted, the MC88200 enters SRAM mode. The cache memory is addressed as random access memory on the P bus. Sixteen P bus inputs are used to access the cache:

A11–A4	set number
TR1–TR0	line number (TR1 is most significant bit)
A3–A2	word number (A3 is most significant bit)
DBE3–DBE0	byte enable

Like the full cache diagnostics, software or firmware performs the diagnostics by writing to the memory location, then reading the value for comparison. If a comparison error occurs, the software functionally removes the faulty line from the set by setting the associated disable bit. The disable bit must be set by a write to the CSSP port.

In the MC88100 processor, the TR1 and TR0 signals are not controllable by software. However, these inputs can be multiplexed with the A13 and A12 P bus signals, respectively, to access the cache through the address lines. Under this implementation, A13–A12 can be used to select one of the four 4K-byte banks of RAM. A11–A4 specify one of the 256 lines in the memory bank; A3–A2 and DBE3–DBE0 perform the word and byte selection. Figure 7-46 shows the selection of data locations for the cache data RAM diagnostics.

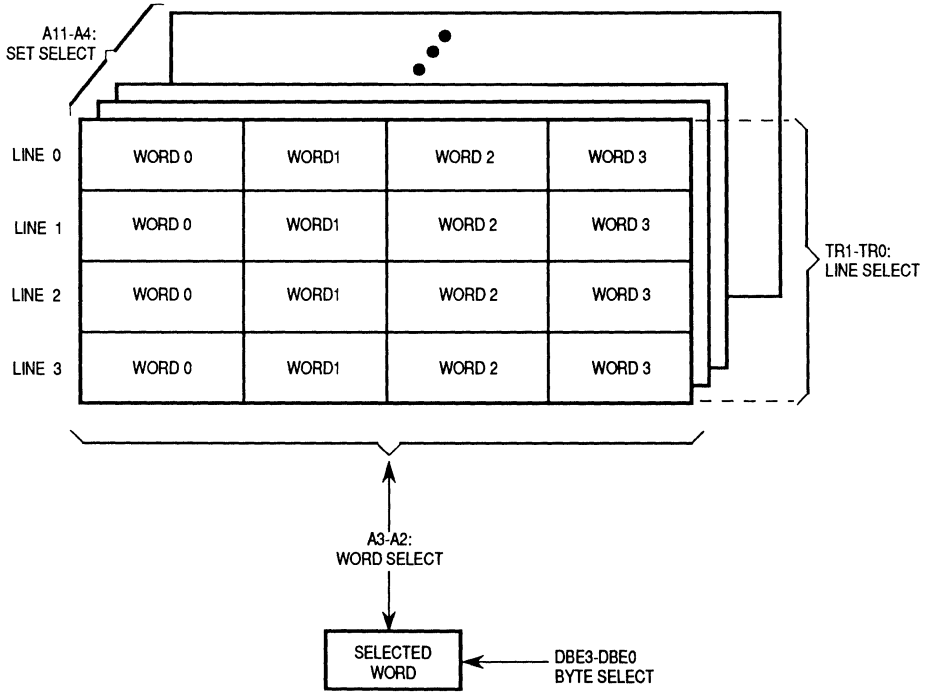


Figure 7-46. Cache Data RAM Diagnostic Memory Selection

7.6 SOFTWARE INITIALIZATION

After a reset, system software initializes the cache set status information and other information to bring the MC88200 into the normal operating state. The cache set status information must be initialized before any other steps are performed. The following list describes typical software initialization procedures:

1. The software initializes the set status information by performing the following steps for each set:
   a. Writes the set number to the system address register, bits 11–4.
   b. Writes $3F0FF000 to the cache set status port.

2. The software initializes the system control register to set the parity, snooping, and arbitration control bits (default: no parity, no snooping, fairness arbitration protocol).
3. If address translation will be enabled, the software can:
   a. Initialize the segment table(s) for the mapped address space(s). Each segment descriptor should be marked invalid or should be loaded with a pointer to an valid page table. If an invalid descriptor is encountered during the translation table search, a segment fault occurs.
   b. Initialize the page descriptors within each valid page table. Each page descriptor should be marked invalid or should be loaded with the address of a page frame in memory. If an invalid descriptor is encountered during the translation table search, a page fault occurs.
4. If address translation will be enabled, then the software must load one or both area pointers with valid descriptors (segment table base addresses and control and protection bits). The TE bit (bit 0) must be set in the area pointer(s) to enable caching, and the CI bit (bit 6) must be cleared to enable caching.

7.6.1 Data Cache Initialization

At reset, the MC88200 supervisor and user area pointers are initialized to put the CMMU in the following state:

- Data cache inhibited (CI = 1)
- Copyback mode (WT = 0)
- Memory mapped as local (G = 0)
- Translation through ATCs disabled (TE = 0)

Before the data cache can be enabled by setting the CI bit in either area pointer, the status of each set of the data cache must be initialized via software. To do this, software writes to the system address register (SAR) to specify which set will be initialized, and the cache set status port (CSSP) to initialize the state of the set.

The following is a code segment that initializes the data cache for one CMMU located at physical address 0xFFF7F000 (CMMU ID = 0 × 7F). For each set (0–255) the following events occur:

1. The line status bits are initialized to the invalid state (VV = 11).
2. The LRU bits are initialized to order the lines from least recently used to most recently used as line 0, line 1, line 2, line 3.
3. The disable bits for each line are cleared.

```
;***CMMU DATA CACHE INITIALIZATION***
;initialize data cache for one MC88200, id = 0x7F

;r21 contains CMMU internal register base address
        or      r21,r0,0x f000      ;init lower half word base adr
        or.u    r21,r21,0x fff7     ;init upper half word base adr

;r22 contains set count
        or      r22,r0,0x 1000      ;set count = 256

;r23 contains cache set status value: all lines enabled, all lines invalid,
;LRU = line0 . . . line3
        or      r23,r0,0x f000      ;init lower half of CSSP value
        or.u    r23,r23,0x 3f0f     ;init upper half of CSSP value

loop:   subu    r22,r22,0x 10       ;decrement loop/set count
        st      r22,r21,0x 00c      ;store set number in sys addr
        st      r23,r21,0x 880      ;store init val. in set status
        bcnd    ne0,r22,loop        ;test for set 0
```

Following initialization of the data cache, the cache inhibit (CI) bit of either area pointer is cleared, thus enabling the data cache.

7.6.2 Translation Table Initialization

Before enabling translation through the ATCs by setting the translation enable (TE) bit in either area pointer, translation tables must be initialized in memory. A minimum of one segment table and one page table must be allocated and initialized before translation is enabled. In the segment table, a minimum of one segment descriptor must be initialized to point to the page table(s). In the page table(s), a minimum of one page descriptor must be initialized to point to a page frame of available physical memory.

Segment Descriptor

Segment descriptor maps page table at 0x4000 as copyback, no supervisor protection (yet), local, caching enabled, and write unprotected.

Address	Data	Comments
0x00002000	0x00004001	WT = SP = G = CI = WP = 0,V = 1
0x00002004- 0x00002FFF	0	V = 0 (uninitialized, invalid descriptors)

Page Descriptors

First descriptor points to a page frame accessible in supervisor or user mode. Second descriptor points to a page frame accessible only in supervisor mode. Both page frames

are mapped as copyback, local, write unprotected, caching enabled, and their contents neither used nor modified.

Address	Data	Comments
0×00004000	0×00005001	WT×SP=G=CI=M=U=WP=0,V=1
0×00004004	0×00006101	WT=G=CI=M=U=WP=0,SP=1,V=1
0×00004008- 0×00004FFF	0	V=0 (uninitialized, invalid descriptors)

Page Frames

Address	Data	Comments
0×00005000- 0×00005FFF	—	4K byte page frame accessible in user or supervisor mode
0×00006000- 0×00006FFF	—	4K byte page frame accessible only in supervisor mode

Assume both the supervisor area pointer register (SAPR) and user area pointer register (UAPR) are initialized to 0×00002001. This causes both area pointers to point to the segment table at 0×00002000. All ensuing user and supervisor accesses to logical (virtual) address locations 0×2000–0×3FFF will map to physical address locations 0×5000–0×6FFF. User accesses to logical address locations 0×3000–0×3FFF will terminate in a supervisor privilege violation fault. The P bus fault address register (PFAR) is updated with the page descriptor at 0×00004004. The P bus fault status register (PFSR) is updated with the supervisor privilege violation fault code.

At some point during MC88200 initialization through software, cache diagnostics can be performed to test the integrity of the cache tags and cache data. Diagnostic software writes to the SAR to select a set and column in the set. Data is then written to the cache data ports (CDP0–CDP3) and cache tag ports (CTP0–CTP3), then read to verify the value read against the value written. More information about cache diagnostics may be found in **7.5 CACHE DIAGNOSTICS**.

SECTION 8
ELECTRICAL CHARACTERISTICS

This section contains electrical specifications and associated timing information for the MC88200.

8.1 MAXIMUM RATINGS

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

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

8.2 THERMAL CHARACTERISTICS — PGA PACKAGE

Characteristic	Symbol	Value	Rating
Thermal Resistance — Ceramic			°C/W
Junction to Ambient	θ_{JA}	25	
Junction to Case	θ_{JC}	10*	

*Estimated

8

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

Characteristic	Symbol	Min	Max	Unit
Clock Low Voltage	V_{CL}	−0.3	0.2 V_{CC}	V
Clock High Voltage	V_{CH}	0.8 V_{CC}	V_{CC} +0.3	V
Input Low Voltage (All Inputs Except CLK)	V_{IL}	−0.3	0.8	V
Input High Voltage (All Inputs Except CLK)	V_{IH}	2.0	V_{CC} +0.3	V
Output Low Voltage (« 8 mA I_{OL}	V_{OL}	—	0.5	V
Output High Voltage (« −4 mA I_{OH}	V_{OH}	2.4	—	V
Input Leakage Current	I_{in}	—	10	μA
High-Impedance Leakage Current	I_{TSI}	—	20	μA
Typical Power Dissipation (T_A = 0°C)	P_D	—	1.5	W
Input Capacitance (V_{in} = 0 V, T_A = 25°C, f = 1 MHz)	C_i	—	15	pF
Output Capacitance (V_{in} = 0 V, T_A = 25°C, f = 1 MHz)	C_o	—	15	pF
Output Load Capacitance — All P Bus Outputs All M Bus Outputs	C_L	—	70 130	pF
AC Output Delay Derating (See Note 1)	C_{LD}	—	1	ns/25 pF

NOTE 1: Only applies when exceeding the specified output load capacitance (C_L). Absolute output load capacitance, per output for correct device operation, must be less than or equal to:
120 pF for P bus
180 pF for M bus

8

8.4 AC ELECTRICAL SPECIFICATIONS — CLOCK INPUT (see Figure 8-1)

Num	Characteristic	20 MHz		25 MHz		Unit
		Min	Max	Min	Max	
	Frequency of Operation (See Note 1)	16.67	20	20.0	25	MHz
1	Clock Cycle Time (Measured at 0.5 V_{CC})	50	60	40	50	ns
2, 3	Clock Pulse Width (Measured at 0.5 V_{CC})	(CT* ÷ 2) − 1	(CT* ÷ 2) + 1	(CT* ÷ 2) − 1	(CT* ÷ 2) + 1	ns
4	Clock Rise Time (0.2 V_{CC} to 0.8 V_{CC})	—	5	—	4	ns
5	Clock Fall Time (0.8 V_{CC} to 0.2 V_{CC})	—	5	—	4	ns

*CT = Cycle Time

NOTE 1: The PLLEN and $\overline{RST}$ signals must be asserted as specified (phase-locked operation). Otherwise, correct device operation cannot be guaranteed.

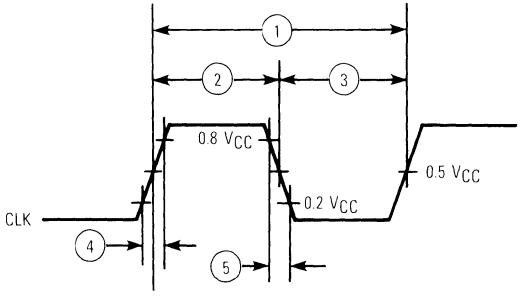


Figure 8-1. Clock Input Timing Diagram

8

8.5 P BUS AC SPECIFICATIONS (see Figure 8-2)

Num	Characteristic	20 MHz		25 MHz		Unit
		Min	Max	Min	Max	
7	Address (Axx), S/$\overline{U}$, R/$\overline{W}$, $\overline{DLOCK}$ to CLK Rising (Input Setup)	3	—	2	—	ns
8	CLK Falling to Address (Axx), S/$\overline{U}$, R/$\overline{W}$, $\overline{DLOCK}$ Invalid (Input Hold)	3	—	2	—	ns
9	Data Byte Enable (DBEx) Valid to CLK Falling (Input Setup)	15	—	12	—	ns
10	CLK Falling to Data Byte Enable (DBEx) Invalid (Input Hold)	3	—	2	—	ns
11	Chip Select ($\overline{PCS}$) Valid to CLK Falling (Input Setup)	15	—	12	—	ns
12	CLK Falling to Chip Select ($\overline{PCS}$) Invalid (Input Hold)	3	—	2	—	ns
13	Write Data In (Dxx) Valid (Write to CMMU) to CLK Falling (Input Setup)	3	—	2	—	ns
14	CLK Rising to Write Data In (Dxx) Invalid (Write to CMMU) (Input Hold)	3	—	2	—	ns
15	CLK Falling to Read Data Out (Dxx) Low-Impedance (Read from CMMU)	10	—	8	—	ns
16	CLK Falling to Read Data Out (Dxx) Valid (Read from CMMU)	10	20	8	16	ns
17	CLK Rising to Read Data Out (Dxx) Invalid (Output Hold) (Read from CMMU)	5	—	4	—	ns
18	CLK Rising to Read Data Out (Dxx) High-Impedance (Read from CMMU)	—	8	—	6	ns
19	CLK Falling to Reply (Rx) Low-Impedance	10	—	8	—	ns
20	CLK Falling to Reply (Rx) Valid	10	23	8	18	ns
21	CLK Falling to Reply (Rx) Invalid (Output Hold)	5	—	4	—	ns
22	CLK Falling to Reply (Rx) High-Impedance	—	8	—	6	ns
66	CLK Rising To TM1–TM0, TR1–TR0 Valid	—	20	—	15	ns
67	CLK Falling to TM1–TM0, TR1–TR0 Invalid (Output Hold)	5	—	3	—	ns

These waveforms should only be referenced to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.

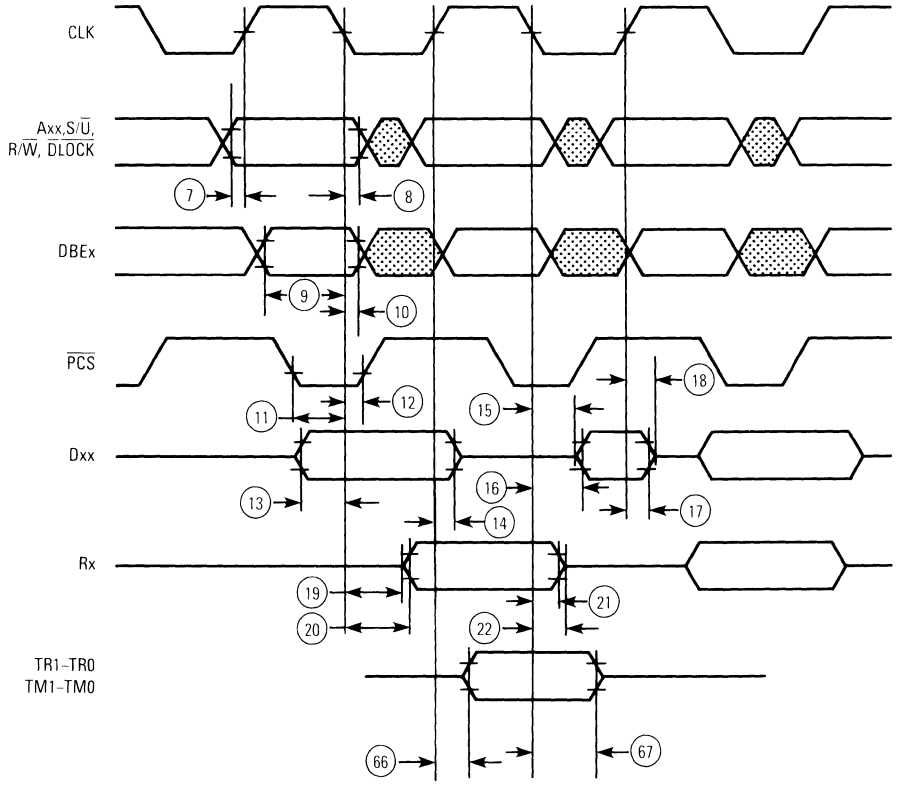


Figure 8-2. P Bus Timing Diagram

8.6 M BUS AC SPECIFICATIONS: M BUS ARBITRATION (see Figure 8-3)

Num	Characteristic	20 MHz		25 MHz		Unit
		Min	Max	Min	Max	
23	CLK Rising to BR Valid	—	20	—	15	ns
24	CLK Rising to BR Invalid (Output Hold)	5	—	4	—	ns
25	BG Valid to Clock Rising (Input Setup)	10	—	8	—	ns
26	Clock Rising to BG Invalid (Input Hold)	3	—	2	—	ns
27	Clock Rising to BA Valid	—	20	—	15	ns
28	Clock Rising to BA Invalid (Output Hold)	5	—	4	—	ns
29	$\overline{AB}$ Valid to Clock Rising (Input Setup)	10	—	8	—	ns
30	Clock Rising to $\overline{AB}$ Invalid (Input Hold)	3	—	2	—	ns
31	$\overline{BB}$ Valid to Clock Rising (Input Setup)	10	—	8	—	ns
32	Clock Rising to $\overline{BB}$ Invalid (Input Hold)	3	—	2	—	ns
33	BA Valid to $\overline{BB}$ Valid	0	20	0	15	ns

These waveforms should only be referenced to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.

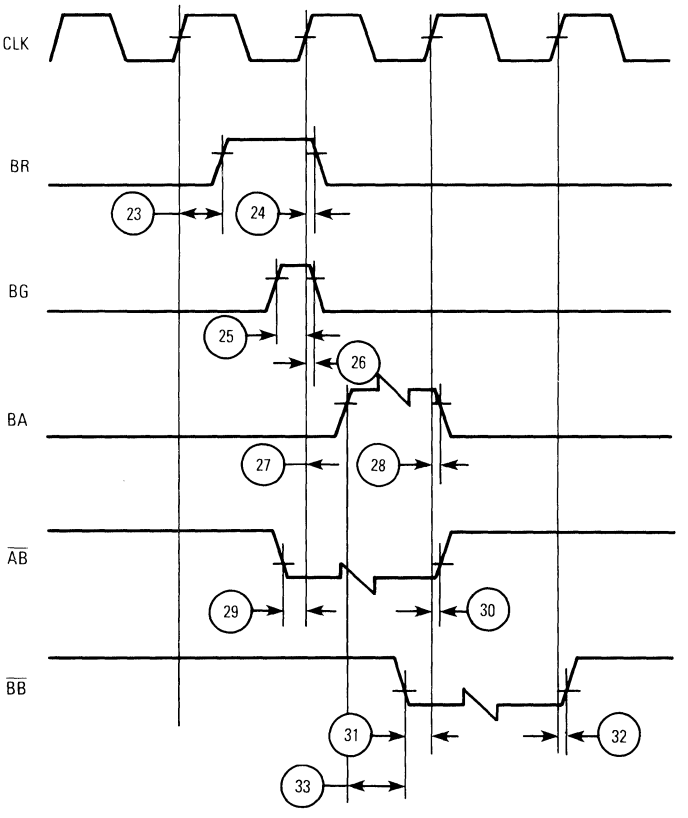


Figure 8-3. M Bus Arbitration Timing Diagram

8.7 M BUS AC SPECIFICATIONS: M BUS MASTER (see Figure 8-4)

Num	Characteristic	20 MHz		25 MHz		Unit
		Min	Max	Min	Max	
34	CLK Rising to AD Bus (ADxx, ADPx) Low-Impedance (Address Phase)	10	—	8	—	ns
35	CLK Rising to AD Bus (ADxx, ADPx) Valid (Address Phase)	—	20	—	15	ns
36	CLK Rising to AD Bus (ADxx, ADPx) Invalid (Output Hold) (Address Phase)	5	—	4	—	ns
37	CLK Rising to AD Bus (ADxx, ADPx) High-Impedance (Address Phase)	—	8	—	6	ns
38	AD Bus (ADxx, ADPx) Valid to CLK Rising (Input Setup) (Data Phase–Read)	10	—	8	—	ns
39	CLK Rising to AD Bus (ADxx, ADPx) Invalid (Input Hold) (Data Phase–Read)	3	—	2	—	ns
40	CLK Rising to AD Bus (ADxx, ADPx) Low-Impedance (Data Phase–Write)	10	—	8	—	ns
41	CLK Rising to AD Bus (ADxx, ADPx) Valid (Data Phase–Write)	—	20	—	15	ns
42	CLK Rising to AD Bus (ADxx, ADPx) Invalid (Output Hold) (Data Phase–Write)	5	—	4	—	ns
43	CLK Rising to AD Bus (ADxx, ADPx) High-Impedance (Data Phase–Write)	—	8	—	6	ns
44	CLK Rising to Control (Cx, CP) Low-Impedance	10	—	8	—	ns
45	CLK Rising to Control (Cx, CP) Valid	—	20	—	15	ns
46	CLK Rising to Control (Cx, CP) Invalid (Output Hold)	5	—	4	—	ns
47	CLK Rising to Control (Cx, CP) High-Impedance	—	8	—	6	ns
48	System Status Valid ($\overline{SSx}$) to CLK Rising (Input Setup)	10	—	8	—	ns
49	CLK Rising to System Status ($\overline{SSx}$) Invalid (Input Hold)	3	—	2	—	ns
66	CLK Rising to TM1–TM0, TR1–TR0 Valid	—	20	—	15	ns
67	CLK Falling to TM1–TM0, TR1–TR0 Invalid (Output Hold)	5	—	3	—	ns

These waveforms should only be referenced to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.

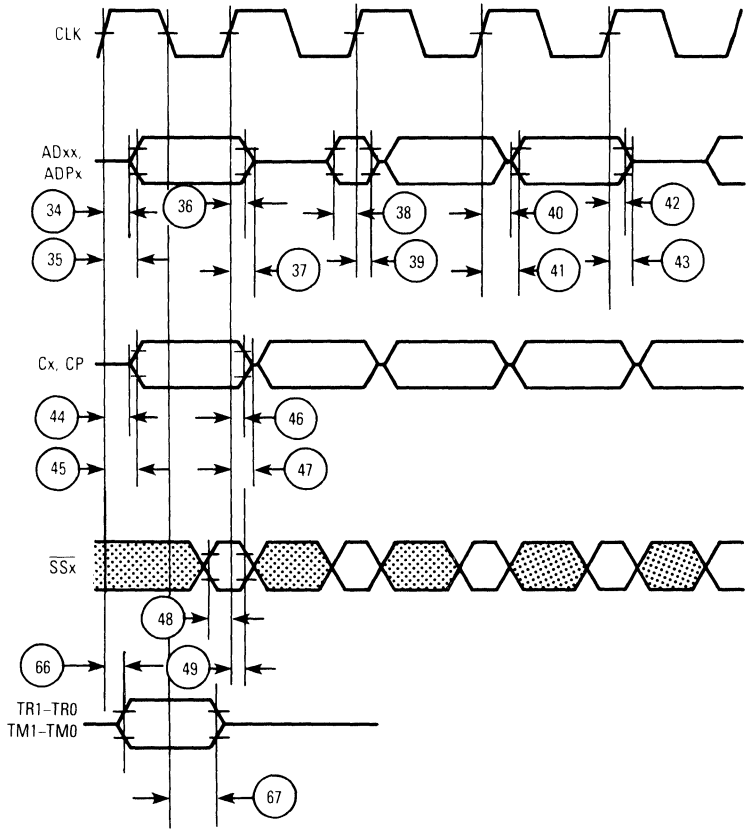


Figure 8-4. M Bus Master Timing Diagram

8

8.8 M BUS AC SPECIFICATIONS: M BUS SLAVE (see Figure 8-5)

Num	Characteristic	20 MHz		25 MHz		Unit
		Min	Max	Min	Max	
50	AD Bus (ADxx, ADPx) Valid to CLK Rising (Input Setup) (Address Phase)	10	—	8	—	ns
51	CLK Rising to AD Bus (ADxx, ADPx) Invalid (Input Hold) (Address Phase)	3	—	2	—	ns
52	CLK Rising to AD Bus (ADxx, ADPx) Low-Impedance (Data Phase–Read from CMMU)	10	—	8	—	ns
53	CLK Rising to AD Bus (ADxx, ADPx) Valid (Data Phase–Read from CMMU)	—	20	—	15	ns
54	CLK Rising to AD Bus (ADxx, ADPx) Invalid (Output Hold) (Data Phase–Read from CMMU)	5	—	4	—	ns
55	CLK Rising to AD Bus (ADxx, ADPx) High-Impedance (Data Phase–Read from CMMU)	—	8	—	6	ns
56	AD Bus (ADxx, ADPx) Valid to CLK Rising (Input Setup) (Data Phase–Write to CMMU)	10	—	8	—	ns
57	CLK Rising to AD Bus (ADxx, ADPx) Invalid (Input Hold) (Data Phase–Write to CMMU)	3	—	2	—	ns
58	Control (Cxx, CPx) Valid to CLK Rising (Input Setup)	10	—	8	—	ns
59	CLK Rising to Control (Cxx, CPx) Invalid (Input Hold)	3	—	2	—	ns
61	CLK Rising to Local Status (STx) Valid	—	20	—	15	ns
62	CLK Rising to Local Status (STx) Invalid (Output Hold)	5	—	4	—	ns
64	System Status ($\overline{SSx}$) Valid to CLK Rising (Input Setup)	10	—	8	—	ns
65	CLK Rising to System Status ($\overline{SSx}$) Invalid (Input Hold)	3	—	2	—	ns

These waveforms should only be referenced to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related diagrams for device operation.

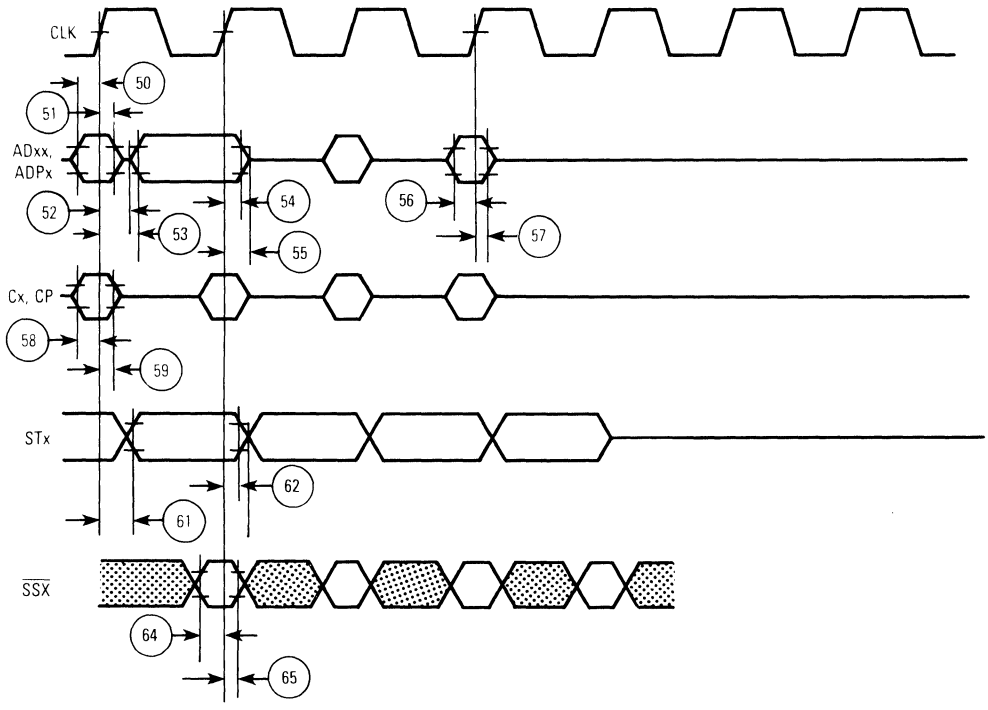


Figure 8-5. M Bus Slave Timing Diagram

8.9 MISCELLANEOUS SIGNAL AC SPECIFICATIONS (see Figures 8-6 and 8-7)

Num	Characteristic	20 MHz		25 MHz		Unit
		Min	Max	Min	Max	
68	$\overline{RST}$, PLLEN, Input Transition Time	—	10	—	10	ns
69	$\overline{RST}$, PCE, MCE, SRAMMODE Valid to Clock Falling (Input Setup) (see Note 1)	10	—	8	—	ns
70	TR1–TR0, TM0, ST3–ST0 Valid to CLK Falling (Input Setup)	10	—	8	—	ns
71	CLK Falling to TR1–TR0, TM0, ST3–ST0 (Input Hold) Invalid	10	—	8	—	ns
72	CLK Rising to TR1–TR0, TM0, ST3–ST0 Low Impedance	—	10	—	8	ns
73	TR1–TR0 Valid to CLK Falling (Input Setup—SRAMMODE)	10	—	8	—	ns
74	CLK Falling to TR1–TR0 Invalid (Input Hold—SRAMMODE)	10	—	8	—	ns
75	CLK Falling to ERR Valid	0	10	0	8	ns
76	CLK Rising to ERR Negated	5	15	4	12	ns

NOTE 1: This specification must be met for the same clock edge for all M88000 devices operating synchronously. PCE, MCE, and SRAMMODE may ony transition during reset.

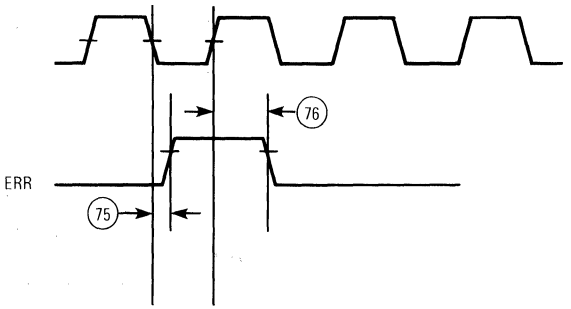
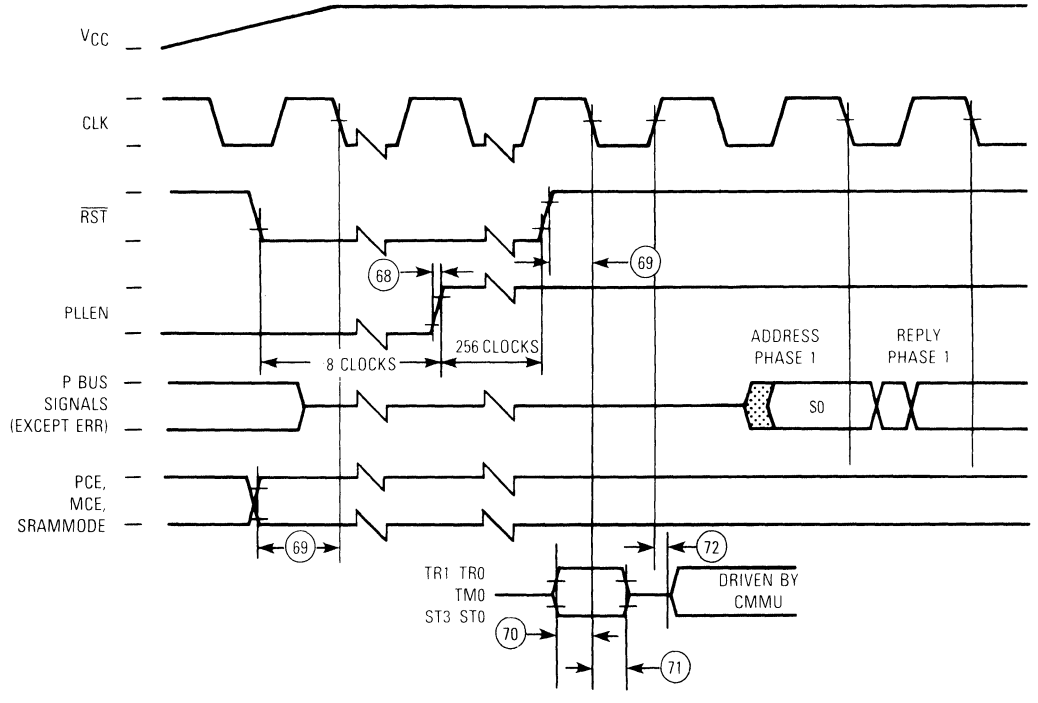
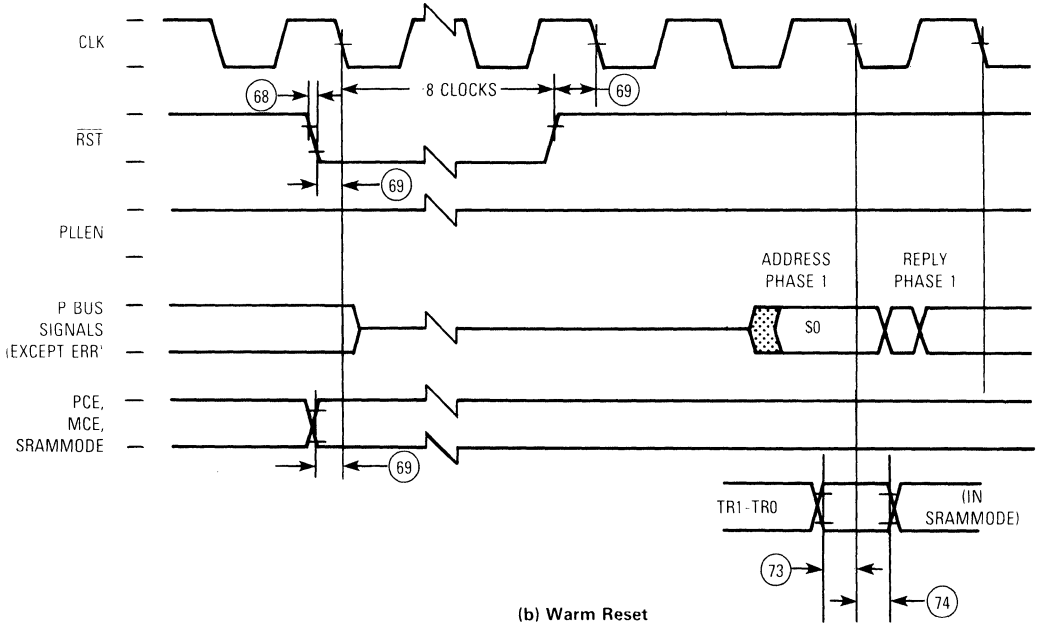


Figure 8-6. Error Signal Timing

8

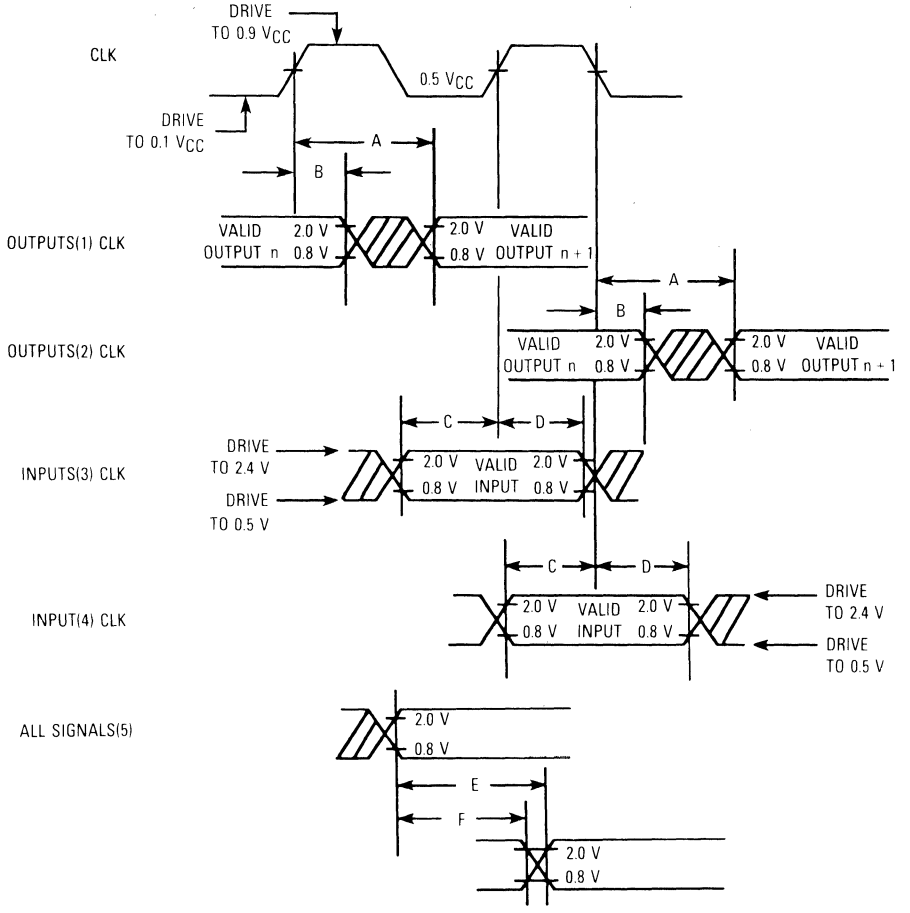


(a) Power-Up Reset



(b) Warm Reset

Figure 8-7. Miscellaneous Signal Timing Diagram



NOTES:

1. This output timing is applicable to all parameters specified relative to the rising edge of the clock.
2. This output timing is applicable to all parameters specified relative to the falling edge of the clock.
3. This input timing is applicable to all parameters specified relative to the rising edge of the clock.
4. This input timing is applicable to all parameters specified relative to the falling edge of the clock.
5. This timing is applicable to all parameters specified relative to the assertion negation of another signal.

LEGEND:

A Maximum output delay specification.
B Minimum output hold time.
C Minimum input setup time specification.
D Minimum input hold time specification.
E Signal valid to signal valid specification (maximum or minimum).
F Signal valid to signal invalid specification (maximum or minimum).

Figure 8-8. Drive Levels and Test Points for AC Specifications

SECTION 9
ORDERING INFORMATION AND MECHANICAL DATA

This section contains the pin assignments and package dimensions diagrams for the MC88200 and information to be used as a guide when ordering.

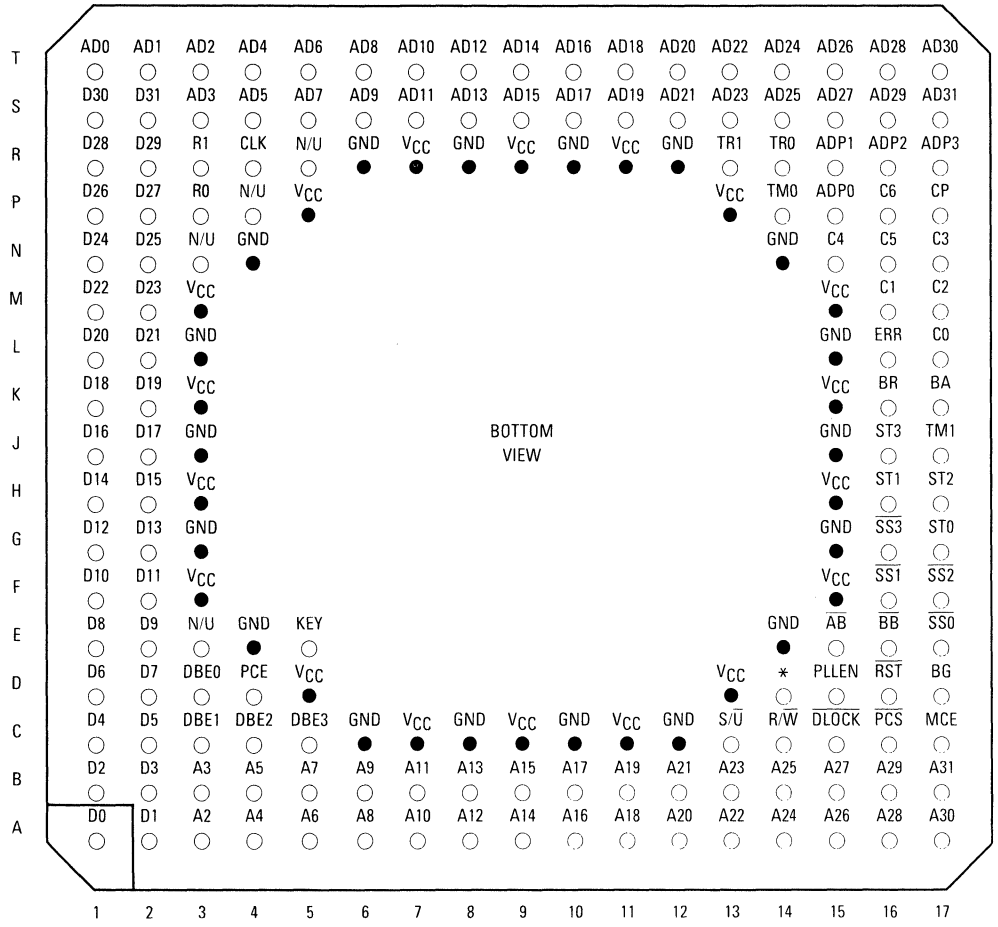
9.1 ORDERING INFORMATION

The following table provides ordering information pertaining to the package type, frequency, temperature, and Motorola order number for the MC88200.

Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic Pin Grid Array Package RC Suffix	20.0 25.0	0° to 70°C	MC88200RC20 MC88200RC25

9.2 PIN ASSIGNMENTS

The MC88200 is available in an 180-pin package. The following figure shows the pin assignments for the MC88200. Pin grouping for power and ground are also listed.



* = SRAMMODE

N/U — Do Not Use

Note: The "KEY" pin is an alignment key with no internal connection.

The V_{CC} and GND pins are separated into two groups to provide individual connections for output buffers and internal logic.

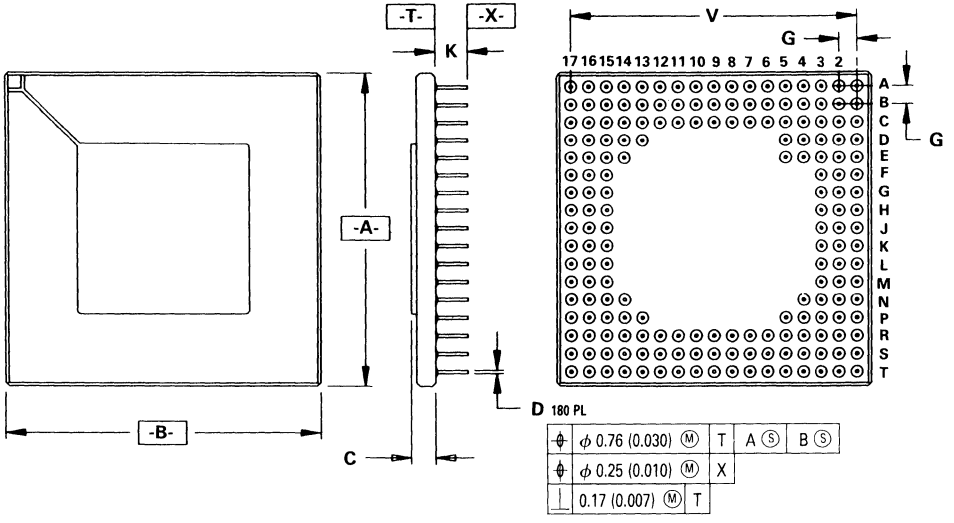
Pin Group	V_{CC}	GND
Internal Logic	C7, C9, C11, D5 D13, P5, P13	C6, C8, C10, C12 N4, N14
External Signals and Buses	F3, F15, H3, H15 K3, K15, M3, M15 R7, R9, R11	E4, E14, G3, G15 J3, J15, L3, L15 R6, R8, R10, R12

9.3 MECHANICAL DATA

The following figure provides the package dimensions for the MC88200.

PACKAGE DIMENSIONS

RC SUFFIX
CERAMIC PACKAGE
CASE 823-01



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	44.20	45.21	1.740	1.780
B	44.20	45.21	1.740	1.780
C	3.05	3.86	0.120	0.152
D	0.41	0.50	0.016	0.020
G	2.54 BSC		0.100 BSC	
K	4.07	5.08	0.160	0.200
V	40.64 BSC		1.600 BSC	

GLOSSARY

Block Address Translation Cache (BATC): A small cache memory in the MC88200 CMMU used for address translation. The BATC contains eight programmable entries for translating 512K-byte logical addresses into 512K-byte physical addresses. Two other BATC entries are hardwired for translating addresses in the control memory space.

Burst Mode: M bus transaction where an address is followed by up to four words of instruction or data. Allows data rates of up to four words/five clocks (64 Mbytes/sec). Used for performing data cache line reads (cache fills) and line writes (copyback).

Bus Snoop: CMMU operation in which M bus addresses marked global are compared to the data cache tags. If a tag matches the M bus address, the cache line is copied back to memory (if modified) and invalidated. Bus snooping is necessary to maintain cache coherency in a multimaster system.

Cache: Small, high-speed memory containing recently accessed data and/or instructions.

Cache Coherency: Caches are coherent if a processor performing a read from its cache is supplied with data corresponding to the most recent value written to memory or to another processor's cache.

Cache Set: A group of four lines of four words each that are tested collectively for a cache hit. A CMMU contains 256 sets, of which one is selected for use by bits 11–4 of the processor's logical address.

Checker Mode: An operating mode of both the MC88100 processor and the MC88200 CMMU. Checker mode is used in system configurations with parallel, redundant components, in which at least one component is the master: i.e., that component performs the meaningful processing. The checker component never drives the bus(es), but it executes all instructions, functions, etc., and monitors all signals as inputs. If the checker finds a mismatch between its calculated result and the result of the master, an error is signaled. See **Master Mode**.

Context: The internal state of the processor that defines the current execution. The context includes the contents of the various execution pipelines and the state of the register file.

Control Register: An MC88200 register that controls component execution and provides status, that provides BATC load ports, or that provides cache diagnostic information.

Copyback: A CMMU operation in which a cache line is copied back to memory to enforce cache coherency. A copyback is done either on a snoop that hits on modified cache data or as a result of a copyback command initiated by the processor.

Displacement: The offset of a field, register, data word, or instruction from a given base address.

Exception: An unusual or error condition encountered by the processor or the CMMU that results in special processing. Exceptions for the processor can be caused by arithmetic overflow or underflow, privilege violations, and other conditions; for the CMMU, they can be caused by bus errors and privilege violations, and other conditions. When exceptions occur, the processor stops the current execution and transfers control to the appropriate exception handler. When the CMMU encounters an exception condition, it stops processing and signals the processor that an exception occurred. Exceptions also occur when trap instructions are executed by the processor.

Exception Handler: A software routine that executes when an exception occurs. Normally, the exception handler will correct the condition that caused the exception or will perform some other meaningful task (such as aborting the program that caused the exception). Exception handlers are defined by a two-word exception vector, which is branched automatically when an exception occurs.

Fault: A logical memory error that causes an exception. Faults are caused by invalid memory descriptors or protection violations; both situations can be corrected by software.

Fully Associative: A memory system (cache) consisting of n entries, all of which are searched in parallel for the value desired. The BATC and PATC are fully associative of 10 and 56 entries, respectively.

Locality of Reference: The frame of reference spanning a memory location in which there is a high probability of future access.

M Bus: The interface between the MC88200 and system memory. The M bus provides 32-bit multiplexed address and data signals with parity protection, control signals with parity protection, and arbitration signals. See **P Bus**.

Master Mode: The normal operating mode of the MC88100 processor and the MC88200 CMMU. In systems with parallel, redundant components, the master components perform meaningful processing, while other components ("checkers") provide redundant error checking. See **Checker Mode**.

G

P Bus: The interface between the MC88100 processor and the MC88200 CMMU. The P bus provides dedicated 32-bit address and data signals as well as control signals. There is one P bus for data transfers and one P bus for instruction transfers. See **M Bus**.

Page: A 4K-byte area of memory aligned on a 4K-byte boundary.

Page Address Translation Cache (PATC): A cache memory in the MC88200 CMMU used for address translation. The PATC contains 56 entries for translating 4K-byte logical page addresses into 4K-byte physical page addresses. The PATC entries are loaded and updated automatically by the MC88200 hardware.

Scaled Indexing: A memory access with the address adjusted to the size of the access. For example, words are aligned on modulo 4 address boundaries; a scaled access to a word has the offset shifted left by two bits. The two least significant bits are filled with zero, making the offset a modulo 4 value. See **Unscaled Indexing**.

Set Associative: A memory system (cache), which consists of n memory (sets), of which one may be selected as being fully associative. Each set contains n entries. When a set is selected, the n entries are searched in parallel for the value (tag) desired. A cache of this type is termed an n-way set-associative cache consisting of n sets. The MC88200 data cache is a four-way set-associative cache consisting of 256 sets.

Unmapped Translation: A logical (virtual) to physical address translation in which the physical address is equal to the logical address (synonym: identity translation).

Unscaled Indexing: An indexed memory access to or from an address formed by addition. See **Scaled Indexing**.

Writethrough: A memory update policy in which all processor write cycles are written to both the data cache and memory.

G

INDEX

— A —

— B —

— C —

— D —

— E —

— F —

— G —

— I —

— L —

— M —

— P —

— R —

— S —

I

— T —

— U —

— W —

— X —

I

PRENTICE HALL
Englewood Cliffs, N.J. 07632

ISBN 0-13-567033-0