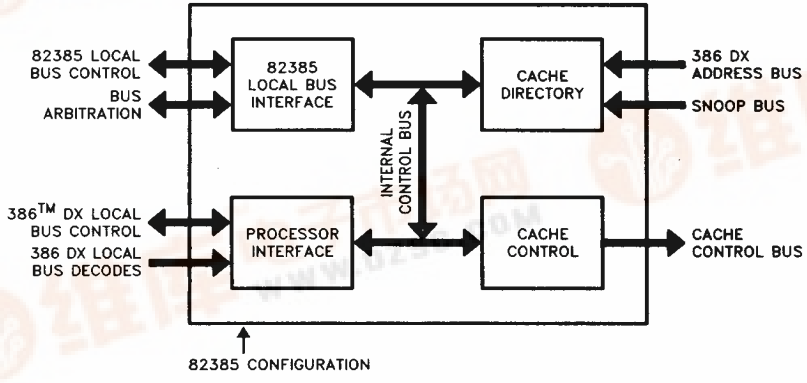
82385
HIGH PERFORMANCE 32-BIT CACHE CONTROLLER

- **Improves 386™ DX System Performance**
  - **— Reduces Average CPU Wait States to Nearly Zero**
  - **— Zero Wait State Read Hit**
  - **— Zero Wait State Posted Memory Writes**
  - **— Allows Other Masters to Access the System Bus More Readily**
- **Hit Rates up to 99%**
- **Optimized as 386 DX Companion**
  - **— Simple 386 DX Interface**
  - **— Part of 386 DX-Based Compute Engine Including 387™ DX Math Coprocessor and 82380 Integrated System Peripheral**
  - **— 20 MHz, 25 MHz, and 33 MHz Operation**
- **Software Transparent**
- **Synchronous Dual Bus Architecture**
  - **— Bus Watching Maintains Cache Coherency**
- **Maps Full 386 DX Address Space (4 Gigabytes)**
- **Flexible Cache Mapping Policies**
  - **— Direct Mapped or 2-Way Set Associative Cache Organization**
  - **— Supports Non-Cacheable Memory Space**
  - **— Unified Cache for Code and Data**
- **Integrates Cache Directory and Cache Management Logic**
- **High Speed CHMOS* IV Technology**
- **132-Pin PGA Package**
- **132-Lead Plastic Quad Flat Pack (PQFP)**

The 82385 Cache Controller is a high performance 32-bit peripheral for the Intel386 Microprocessor. It stores a copy of frequently accessed code and data from main memory in a zero wait state local cache memory. The 82385 enables the 386 DX to run at its full potential by reducing the average number of CPU wait states to nearly zero. The dual bus architecture of the 82385 allows other masters to access system resources while the 386 DX operates locally out of its cache. In this situation, the 82385's "bus watching" mechanism preserves cache coherency by monitoring the system bus address lines at no cost to system or local throughput.

The 82385 is completely software transparent, protecting the integrity of system software. High performance and board savings are achieved because the 82385 integrates a cache directory and all cache management logic on one chip.



82385 Internal Block Diagram

*CHMOS is a patented process of Intel Corporation.
Intel386™, 386™ DX, 387™ DX are trademarks of Intel Corporation.

December 1990
Order Number: 290143-006

CONTENTS

CONTENTS

CONTENTS

CONTENTS

5

1.0 82385 FUNCTIONAL OVERVIEW

The 82385 Cache Controller is a high performance 32-bit peripheral for the Intel386 microprocessor. This chapter provides an overview of the 82385, and of the basic architecture and operation of an 386 DX CPU/82385 system.

1.1 82385 OVERVIEW

The main function of a cache memory system is to provide fast local storage for frequently accessed code and data. The cache system intercepts 386 DX memory references to see if the required data resides in the cache. If the data resides in the cache (a hit), it is returned to the 386 DX without incurring wait states. If the data is not cached (a miss), the reference is forwarded to the system and the data retrieved from main memory. An efficient cache will yield a high "hit rate" (the ratio of cache hits to total 386 DX accesses), such that the majority of accesses are serviced with zero wait states. The net effect is that the wait states incurred in a relatively infrequent miss are averaged over a large number of accesses, resulting in an average of nearly zero wait states per access. Since cache hits are serviced locally, a processor operating out of its local cache has a much lower "bus utilization" which reduces system bus bandwidth requirements, making more bandwidth available to other bus masters.

The 82385 Cache Controller integrates a cache directory and all cache management logic required to support an external 32 Kbyte cache. The cache directory structure is such that the entire physical address range of the 386 DX (4 Gigabytes) is mapped into the cache. Provision is made to allow areas of memory to be set aside as non-cacheable. The user has two cache organization options: direct mapped and 2-way set associative. Both provide the high hit rates necessary to make a large, relatively slow main memory array look like a fast, zero wait state memory to the 386 DX.

1.2 SYSTEM OVERVIEW I: BUS STRUCTURE

A good grasp of the bus structure of a 386 DX CPU/82385 system is essential in understanding both the 82385 and its role in an 386 DX system. The following is a description of this structure.

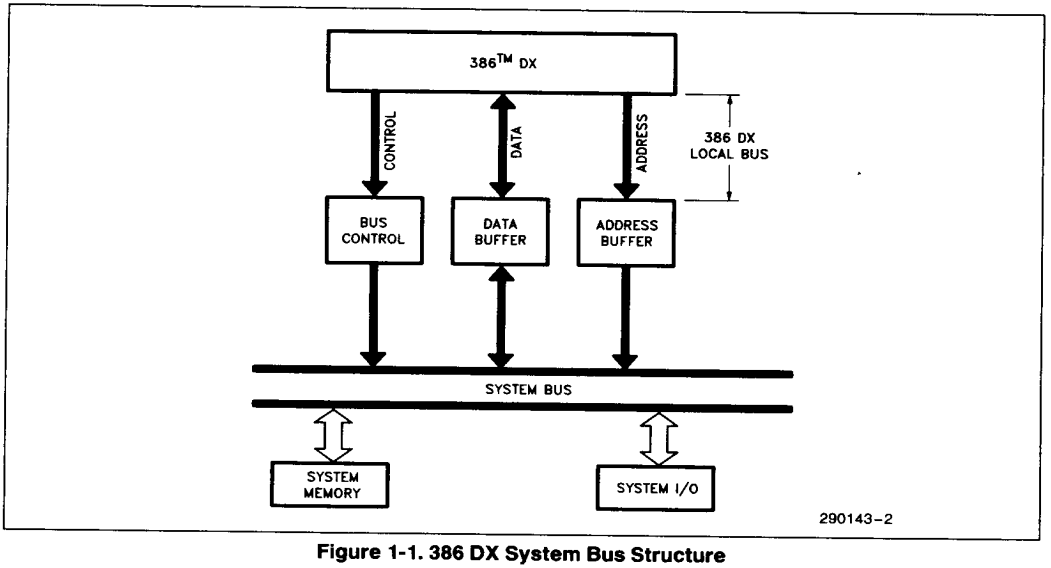


Figure 1-1. 386 DX System Bus Structure

1.2.1 386 DX Local Bus/82385 Local Bus/System Bus

Figure 1-1 depicts the bus structure of a typical 386 DX system. The "386 DX Local Bus" consists of the physical 386 DX address, data, and control busses. The local address and data busses are buffered and/or latched to become the "system" address and data busses. The local control bus is decoded by bus control logic to generate the various system bus read and write commands.

The addition of an 82385 Cache Controller causes a separation of the 386 DX bus into two distinct busses: the actual 386 DX local bus and the "82385 Local Bus" (Figure 1-2). The 82385 local bus is designed to look like the front end of an 386 DX by providing 82385 local bus equivalents to all appropriate 386 DX signals. The system ties to this "386 DX-like" front end just as it would to an actual 386 DX. The 386 DX simply sees a fast system bus, and the system sees a 386 DX front end with low bus bandwidth requirements. The cache subsystem is transparent to both. Note that the 82385 local bus is not simply a buffered version of the 386 DX bus, but rather is distinct from, and able to operate in parallel with the 386 DX bus. Other masters residing on either the 82385 local bus or system bus are free to manage system resources while the 386 DX operates out of its cache.

1.2.2 Bus Arbitration

The 82385 presents the "386 DX-like" interface which is called the 82385 local bus. Whereas the 386 DX provides a Hold Request/Hold Acknowledge bus arbitration mechanism via its HOLD and HLDA pins, the 82385 provides an equivalent mechanism via its BHOLD and BHLDA pins. (These signals are described in Section 3.7.) When another master requests the 82385 local bus, it issues the request to the 82385 via BHOLD. Typically, at the end of the current 82385 local bus cycle, the 82385 will release the 82385 local bus and acknowledge the request via BHLDA. The 386 DX is of course free to continue operating on the 386 DX local bus while another master owns the 82385 local bus.

1.2.3 Master/Slave Operation

The above 82385 local bus arbitration discussion is true when the 82385 is programmed for "Master" mode operation. The user can, however, configure the 82385 for "Slave" mode operation. (Programming is done via a hardware strap option.) The roles of BHOLD and BHLDA are reversed for an 82385 in slave mode; BHOLD is now an output indicating a request to control the bus, and BHLDA is an input indicating that a request has been granted. An 82385 programmed in slave mode drives the 82385 local bus only when it has requested and subsequently been granted bus control. This allows multiple 386 DX CPU/82385 subsystems to reside on the same 82385 local bus (Figure 1-3).

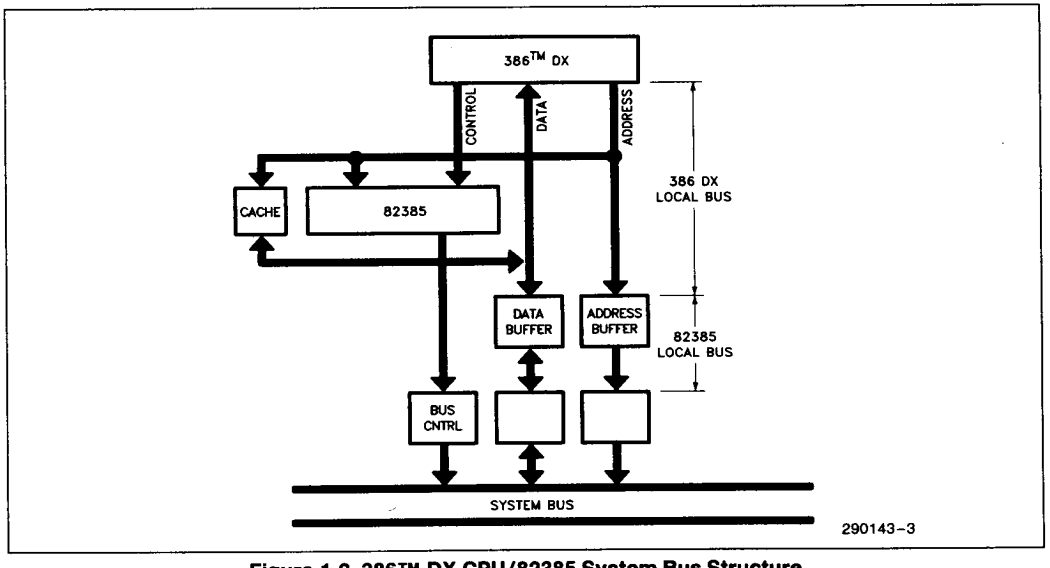


Figure 1-2. 386™ DX CPU/82385 System Bus Structure

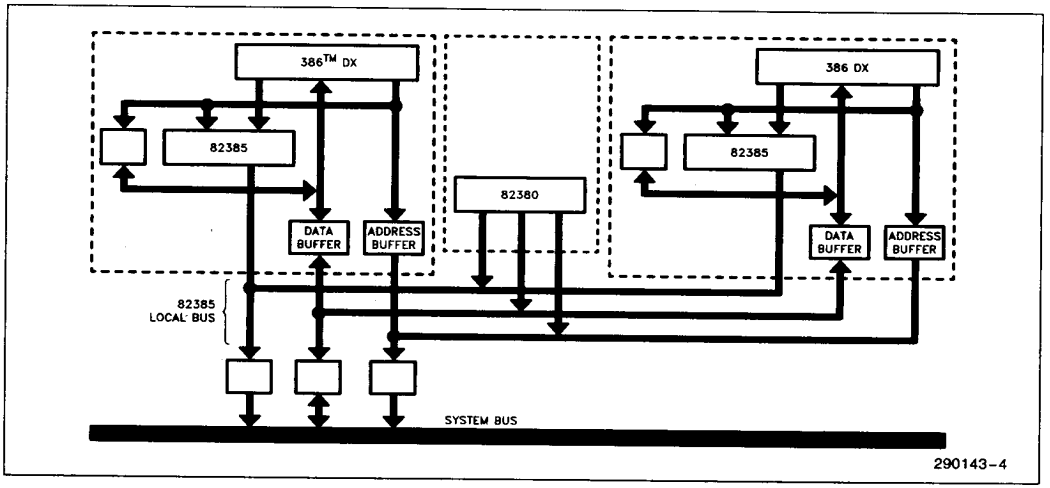


Figure 1-3. Multi-Master/Multi-Cache Environment

1.2.4 Cache Coherency

Ideally, a cache contains a copy of the most heavily used portions of main memory. To maintain cache "coherency" is to make sure that this local copy is identical to main memory. In a system where multiple masters can access the same memory, there is always a risk that one master will alter the contents of a memory location that is duplicated in the local cache of another master. (The cache is said to contain "stale" data.) One rather restrictive solution is to not allow cache subsystems to cache shared memory. Another simple solution is to flush the cache anytime another master writes to system memory. However, this can seriously degrade system performance as excessive cache flushing will reduce the hit rate of what may otherwise be a highly efficient cache.

The 82385 preserves cache coherency via "bus watching" (also called snooping), a technique that neither impacts performance nor restricts memory mapping. An 82385 that is not currently bus master monitors system bus cycles, and when a write cycle by another master is detected (a snoop), the system address is sampled and used to see if the referenced location is duplicated in the cache. If so (a snoop hit), the corresponding cache entry is invalidated, which will force the 386 DX to fetch the up-to-date data from main memory the next time it accesses this modified location. Figure 1-4 depicts the general form of bus watching.

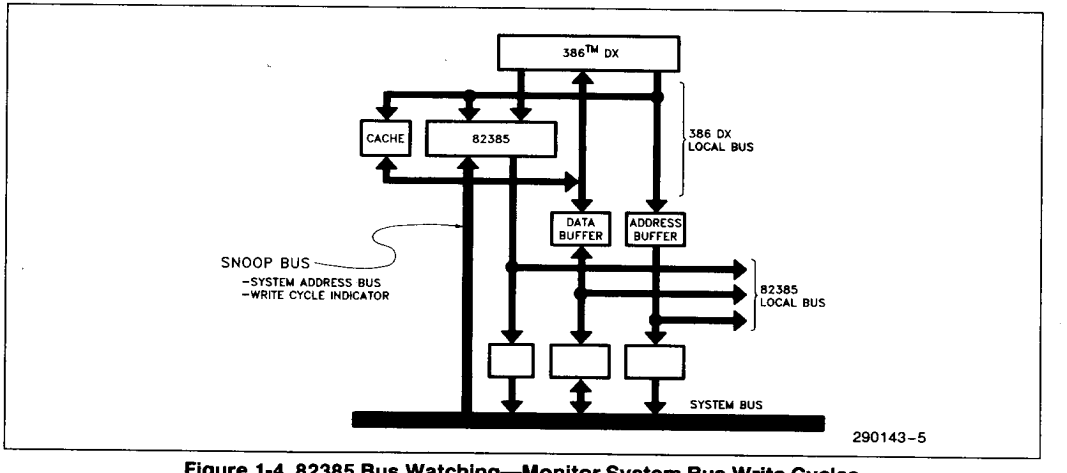


Figure 1-4. 82385 Bus Watching—Monitor System Bus Write Cycles

1.3 SYSTEM OVERVIEW II: BASIC OPERATION

This discussion is an overview of the basic operation of an 386 DX CPU/82385 system. Items discussed include the 82385's response to all 386 DX cycles, including interrupt acknowledges, halts, and shutdowns. Also discussed are non-cacheable and local accesses.

1.3.1 386 DX Memory Code and Data Read Cycles

1.3.1.1 READ HITS

When the 386 DX initiates a memory code or data read cycle, the 82385 compares the high order bits of the 386 DX address bus with the appropriate addresses (tags) stored in its on-chip directory. (The directory structure is described in Chapter 2.) If the 82385 determines that the requested data is in the cache, it issues the appropriate control signals that direct the cache to drive the requested data onto the 386 DX data bus, where it is read by the 386 DX. The 82385 terminates the 386 DX cycle without inserting any wait states.

1.3.1.2 READ MISSES

If the 82385 determines that the requested data is not in the cache, the request is forwarded to the 82385 local bus and the data retrieved from main memory. As the data returns from main memory, it is directed to the 386 DX and also written into the cache. Concurrently, the 82385 updates the cache directory such that the next time this particular piece of information is requested by the 386 DX, the 82385 will find it in the cache and return it with zero wait states.

The basic unit of transfer between main memory and cache memory in a cache subsystem is called the line size. In an 82385 system, the line size is one 32-bit aligned doubleword. During a read miss, all four 82385 local bus byte enables are active. This ensures that a full 32-bit entry is written into the cache. (The 386 DX simply ignores what it did not request.) In any other type of 386 DX cycle that is forwarded to the 82385 local bus, the logic levels of the 386 DX byte enables are duplicated on the 82385 local bus.

The 82385 does not actively fetch main memory data independently of the 386 DX. The 82385 is essentially a passive device which only monitors the address bus and activates control signals. The read miss is the only mechanism by which main memory data is copied into the cache and validated in the cache directory.

In an isolated read miss, the number of wait states seen by the 386 DX is that required by the system memory to respond with data plus the cache comparison cycle (hit/miss decision). The cache system must determine that the cycle is a miss before it can begin the system memory access. However, since misses most often occur consecutively, the 82385 will begin 386 DX address pipelined cycles to effectively "hide" the comparison cycle beyond the first miss (refer to Section 4.1.3).

The 82385 can execute a main memory access on the 82385 local bus only if it currently owns the bus. If not, an 82385 in master mode will run the cycle after the current master releases the bus. An 82385 in slave mode will issue a hold request, and will run the cycle as soon as the request is acknowledged. (This is true for any read or write cycle that needs to run on the 82385 local bus.)

1.3.2 386 DX Memory Write Cycles

The 82385's "posted write" capability allows the majority of 386 DX memory write cycles to run with zero wait states. The primary memory update policy implemented in a posted write is the traditional cache "write through" technique, which implies that main memory is always updated in any memory write cycle. If the referenced location also happens to reside in the cache (a write hit), the cache is updated as well.

Beyond this, a posted write latches the 386 DX address, data, and cycle definition signals, and the 386 DX local bus cycle is terminated without any wait states, even though the corresponding 82385 local bus cycle is not yet completed, or perhaps not even started. A posted write is possible because the 82385's bus state machine, which is almost identical to the 386 DX bus state machine, is able to run 82385 local bus cycles independently of the 386 DX. The only time the 386 DX sees write cycle wait states is when a previously latched (posted) write has not yet been completed on the 82385 local bus or during an I/O write (which is not posted). A 386 DX write can be posted even if the 82385 does not currently own the 82385 local bus. In this case, an 82385 in master mode will run the cycle as soon as the current master releases the bus, and an 82385 in slave mode will request the bus and run the cycle when the request is acknowledged. The 386 DX is free to continue operating out of its cache (on the 386 DX local bus) during this time.

1.3.3 Non-Cacheable Cycles

Non-cacheable cycles fall into one of two categories: cycles decoded as non-cacheable, and cycles

that are by default non-cacheable according to the 82385's design. All non-cacheable cycles are forwarded to the 82385 local bus. Non-cacheable cycles have no effect on the cache or cache directory.

The 82385 allows the system designer to define areas of main memory as non-cacheable. The 386 DX address bus is decoded and the decode output is connected to the 82385's non-cacheable access (NCA#) input. This decoding is done in the first 386 DX bus state in which the non-cacheable cycle address becomes available. Non-cacheable read cycles resemble cacheable read miss cycles, except that the cache and cache directory are unaffected. NCA defined non-cacheable writes, like most writes, are posted.

The 82385 defines certain cycles as non-cacheable without using its non-cacheable access input. These include I/O cycles, interrupt acknowledge cycles, and halt/shutdown cycles. I/O reads and interrupt acknowledge cycles execute as any other non-cacheable read. I/O write cycles are not posted. The 386 DX is not allowed to continue until a ready signal is returned from the system. Halt/Shutdown cycles are posted. During a halt/shutdown condition, the 82385 local bus duplicates the behavior of the 386 DX, including the ability to recognize and respond to a BHOLD request. (The 82385's bus watching mechanism is functional in this condition.)

1.3.3.1 16-BIT MEMORY SPACE

The 82385 does not cache 16-bit memory space (as decoded by the 386 DX BS16# input), but does make provisions to handle 16-bit space as non-cacheable. (There is no 82385 equivalent to the 386 DX BS16# input.) In a system without an 82385, the 386 DX BS16# input need not be asserted until the last state of a 16-bit cycle for the 386 DX to recognize it as such (unless NA# is sampled active earlier in the cycle.) The 82385, however, needs this information earlier, specifically at the end of the first 386 DX bus state in which the address of the 16-bit cycle becomes available. The result is that in a system without an 82385, 16-bit devices can inform the 386 DX that they are 16-bit devices "on the fly," while in a system with an 82385, devices decoded as 16-bit (using the 386 DX BS16#) must be located in address space set aside for 16-bit devices. If 16-bit space is decoded according to 82385 guidelines (as described later in the data sheet), then the 82385 will handle 16-bit cycles just like the 386 DX does, including effectively locking the two halves of a non-aligned 16-bit transfer from interruption by another master.

1.3.4 386 DX Local Bus Cycles

386 DX Local Bus Cycles are accesses to resources on the 386 DX local bus other than to the 82385 itself. The 82385 simply ignores these accesses: they are neither forwarded to the system nor do they affect the cache. The designer sets aside memory and/or I/O space for local resources by decoding the 386 DX address bus and feeding the decode to the 82385's local bus access (LBA#) input. The designer can also decode the 386 DX cycle definition signals to keep specific 386 DX cycles from being forwarded to the system. For example, a multi-processor design may wish to capture and remedy a 386 DX shutdown locally without having it detected by the rest of the system. Note that in such a design, the local shutdown cycle must be terminated by local bus control logic. The 387 Math Coprocessor is considered a 386 DX local bus resource, but it need not be decoded as such by the user since the 82385 is able to internally recognize 387 accesses via the M/IO# and A31 pins.

1.3.5 Summary of 82385 Response to All 386 DX Cycles

Table 1-1 summarizes the 82385 response to all 386 DX bus cycles, as conditioned by whether or not the cycle is decoded as local or non-cacheable. The table describes the impact of each cycle on the cache and on the cache directory, and whether or not the cycle is forwarded to the 82385 local bus. Whenever the 82385 local bus is marked "IDLE", it implies that this bus is available to other masters.

Table 1-1. 82385 Response to 386 DX Cycles

386 DX Bus Cycle Definition					82385 Response when Decoded as Cacheable			82385 Response when Decoded as Non-Cacheable			82385 Response when Decoded as an 386 DX Local Bus Access		
M/IO#	D/C#	W/R#	386 DX Cycle		Cache	Cache Directory	82385 Local Bus	Cache	Cache Directory	82385 Local Bus	Cache	Cache Directory	82385 Local Bus
0	0	0	INT ACK	N/A	—	—	INT ACK	—	—	INT ACK	—	—	IDLE
0	0	1	UNDEFINED	N/A			UNDEFINED			UNDEFINED			IDLE
0	1	0	I/O READ	N/A	—	—	I/O READ	—	—	I/O READ	—	—	IDLE
0	1	1	I/O WRITE	N/A	—	—	I/O WRITE	—	—	I/O WRITE	—	—	IDLE
1	0	0	MEM CODE READ	HIT	CACHE READ	—	IDLE	—	—	MEM CODE READ	—	—	IDLE
				MISS	CACHE WRITE	DATA VALIDATION	MEM CODE READ						
1	0	1	HALT/ SHUTDOWN	N/A	—	—	HALT/ SHUTDOWN	—	—	HALT/ SHUTDOWN	—	—	IDLE
1	1	0	MEM DATA READ	HIT	CACHE READ	—	IDLE	—	—	MEM DATA READ	—	—	IDLE
				MISS	CACHE WRITE	DATA VALIDATION	MEM DATA READ						
1	1	1	MEM DATA WRITE	HIT	CACHE WRITE	—	MEM DATA WRITE	—	—	MEM DATA WRITE	—	—	IDLE
				MISS	—	—	MEM DATA WRITE						

NOTES:
- A dash (—) indicates that the cache and cache directory are unaffected. This table does not reflect how an access affects the LRU bit.
- An "IDLE" 82385 Local Bus implies that this bus is available to other masters.
- The 82385's response to 80387 accesses is the same as when decoded as an 386 DX Local Bus access.
- The only other operations that affect the cache directory are:
  1. RESET or Cache Flush—all tag valid bits cleared.
  2. Snoop Hit—corresponding line valid bit cleared.

1.3.6 Bus Watching

As previously discussed, the 82385 "qualifies" an 386 DX bus cycle in the first bus state in which the address and cycle definition signals of the cycle become available. The cycle is qualified as read or write, cacheable or non-cacheable, etc. Cacheable cycles are further classified as hit or miss according to the results of the cache comparison, which accesses the 82385 directory and compares the appropriate directory location (tag) to the current 386 DX address. If the cycle turns out to be non-cacheable or a 386 DX local bus access, the hit/miss decision is ignored. The cycle qualification requires one 386 DX state. Since the fastest 386 DX access is two states, the second state can be used for bus watching.

When the 82385 does not own the system bus, it monitors system bus cycles. If another master writes into main memory, the 82385 latches the system address and executes a cache look-up to see if the altered main memory location resides in the cache. If so (a snoop hit), the cache entry is marked invalid in the cache directory. Since the directory is at most only being used every other state to qualify 386 DX accesses, snoop look-ups are interleaved between 386 DX local bus look-ups. The cache directory is time multiplexed between the 386 DX address and the latched system address. The result is that all snoops are caught and serviced without slowing down the 386 DX, even when running zero wait state hits on the 386 DX local bus.

1.3.7 Cache Flush

The 82385 offers a cache flush input. When activated, this signal causes the 82385 to invalidate all data which had previously been cached. Specifically, all tag valid bits are cleared. (Refer to the 82385 directory structure in Chapter 2.) Therefore, the cache is empty and subsequent cycles are misses until the 386 DX begins repeating the new accesses (hits). The primary use of the FLUSH input is for diagnostics and multi-processor support.

NOTE:
The use of this pin as a coherency mechanism may impact software transparency.

2.0 82385 CACHE ORGANIZATION

The 82385 supports two cache organizations: a simple direct mapped organization and a slightly more complex, higher performance two way set associative organization. The choice is made by strapping an 82385 input (2W/D#) either high or low. This chapter describes the structure and operation of both organizations.

2.1 DIRECT MAPPED CACHE

2.1.1 Direct Mapped Cache Structure and Terminology

Figure 2-1 depicts the relationship between the 82385's internal cache directory, the external cache memory, and the 386 DX's 4 Gigabyte physical address space. The 4 Gigabytes can conceptually be thought of as cache "pages" each being 8K doublewords (32 Kbytes) deep. The page size matches the cache size. The cache can be further divided into 1024 (0 thru 1023) sets of eight doublewords (8 x 32 bits). Each 32-bit doubleword is called a "line." The unit of transfer between the main memory and cache is one line.

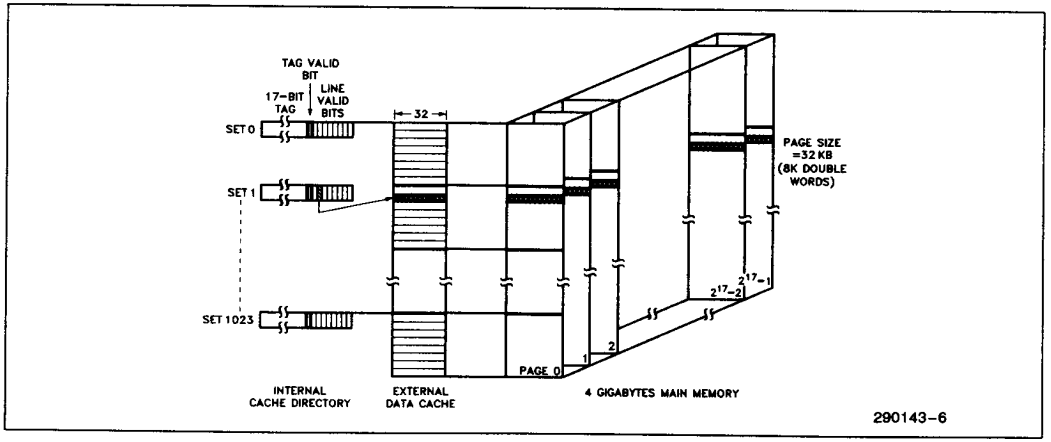


Figure 2-1. Direct Mapped Cache Organization

Each block in the external cache has an associated 26-bit entry in the 82385's internal cache directory. This entry has three components: a 17-bit "tag," a "tag valid" bit, and eight "line valid" bits. The tag acts as a main memory page number (17 tag bits support 2^{17} pages). For example, if line 9 of page 2 currently resides in the cache, then a binary 2 is stored in the Set 1 tag field. (For any 82385 direct mapped cache page in main memory, Set 0 consists of lines 0–7, Set 1 consists of lines 8–15, etc. Line 9 is shaded in Figure 2-1.) An important characteristic of a direct mapped cache is that line 9 of any page can only reside in line 9 of the cache. All identical page offsets map to a single cache location.

The data in a cache set is considered valid or invalid depending on the status of its tag valid bit. If clear, the entire set is considered invalid. If true, an individual line within the set is considered valid or invalid depending on the status of its line valid bit.

The 82385 sees the 386 DX address bus (A2–A31) as partitioned into three fields: a 17-bit "tag" field (A15–A31), a 10-bit "set-address" field (A5–A14), and a 3-bit "line select" field (A2–A4). (See Figure 2-2.) The lower 13 address bits (A2–A14) also serve as the "cache address" which directly selects one of 8K doublewords in the external cache.

2.1.2 Direct Mapped Cache Operation

The following is a description of the interaction between the 386 DX, cache, and cache directory.

2.1.2.1 READ HITS

When the 386 DX initiates a memory read cycle, the 82385 uses the 10-bit set address to select one of 1024 directory entries, and the 3-bit line select field to select one of eight line valid bits within the entry. The 13-bit cache address selects the corresponding doubleword in the cache. The 82385 compares the 17-bit tag field (A15–A31 of the 386 DX access) with the tag stored in the selected directory entry. If the tag and upper address bits match, and if both the tag and appropriate line valid bits are set, the result is a hit, and the 82385 directs the cache to drive the selected doubleword onto the 386 DX data bus. A read hit does not alter the contents of the cache or directory.

2.1.2.2 READ MISSES

A read miss can occur in two ways. The first is known as a "line" miss, and occurs when the tag and upper address bits match and the tag valid bit is set, but the line valid bit is clear. The second is called a "tag" miss, and occurs when either the tag and upper address bits do not match, or the tag valid bit is clear. (The line valid bit is a "don't care" in a tag miss.) In both cases, the 82385 forwards the 386 DX reference to the system, and as the returning data is fed to the 386 DX, it is written into the cache and validated in the cache directory.

In a line miss, the incoming data is validated simply by setting the previously clear line valid bit. In a tag miss, the upper address bits overwrite the previously stored tag, the tag valid bit is set, the appropriate line valid bit is set, and the other seven line valid bits are cleared. Subsequent tag hits with line misses will only set the appropriate line valid bit. (Any data associated with the previous tag is no longer considered resident in the cache.)

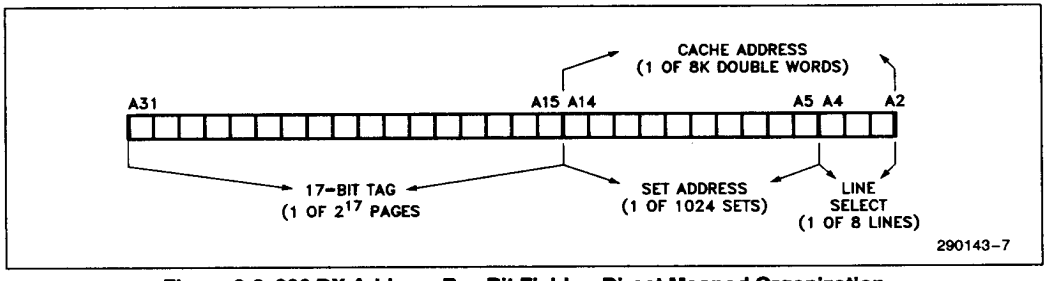


Figure 2-2. 386 DX Address Bus Bit Fields—Direct Mapped Organization

2.1.2.3 OTHER OPERATIONS THAT AFFECT THE CACHE AND CACHE DIRECTORY

The other operations that affect the cache and/or directory are write hits, snoop hits, cache flushes, and 82385 resets. In a write hit, the cache is updated along with main memory, but the directory is unaffected. In a snoop hit, the cache is unaffected, but the affected line is invalidated by clearing its line valid bit in the directory. Both an 82385 reset and cache flush clear all tag valid bits.

When an 386 DX CPU/82385 system "wakes up" upon reset, all tag valid bits are clear. At this point, a read miss is the only mechanism by which main memory data is copied into the cache and validated in the cache directory. Assume an early 386 DX code access seeks (for the first time) line 9 of page 2. Since the tag valid bit is clear, the access is a tag miss, and the data is fetched from main memory. Upon return, the data is fed to the 386 DX and simultaneously written into line 9 of the cache. The set directory entry is updated to show this line as valid. Specifically, the tag and appropriate line valid bits are set, the remaining seven line valid bits cleared, and a binary 2 written into the tag. Since code is sequential in nature, the 386 DX will likely next want line 10 of page 2, then line 11, and so on. If the 386 DX sequentially fetches the next six lines, these fetches will be line misses, and as each is fetched from main memory and written into the cache, its corresponding line valid bit is set. This is the basic flow of events that fills the cache with valid data. Only after a piece of data has been copied into the cache and validated can it be accessed in a zero wait state read hit. Also, a cache entry must have been validated before it can be subsequently altered by a write hit, or invalidated by a snoop hit.

An extreme example of "thrashing" is if line 9 of page two is an instruction to jump to line 9 of page one, which is an instruction to jump back to line 9 of page two. Thrashing results from the direct mapped cache characteristic that all identical page offsets map to a single cache location. In this example, the page one access overwrites the cached page two data, and the page two access overwrites the cached page one data. As long as the code jumps back and forth the hit rate is zero. This is of course an extreme case. The effect of thrashing is that a direct mapped cache exhibits a slightly reduced overall hit rate as compared to a set associative cache of the same size.

2.2 TWO WAY SET ASSOCIATIVE CACHE

2.2.1 Two Way Set Associative Cache Structure and Terminology

Figure 2-3 illustrates the relationship between the directory, cache, and 4 Gigabyte address space.

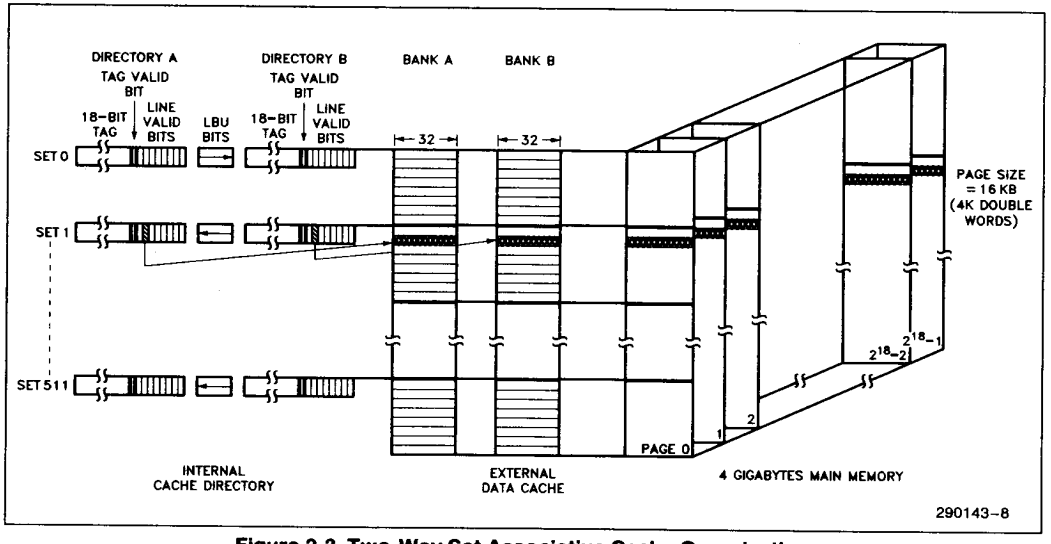


Figure 2-3. Two-Way Set Associative Cache Organization

Whereas the direct mapped cache is organized as one bank of 8K doublewords, the two way set associative cache is organized as two banks (A and B) of 4K doublewords each. The page size is halved, and the number of pages doubled. (Note the extra tag bit.) The cache now has 512 sets in each bank. (Two banks times 512 sets gives a total of 1024. The structure can be thought of as two half-sized direct mapped caches in parallel.) The performance advantage over a direct mapped cache is that all identical page offsets map to two cache locations instead of one, reducing the potential for thrashing. The 82385's partitioning of the 386 DX address bus is depicted in Figure 2-4.

2.2.2 LRU Replacement Algorithm

The two way set associative directory has an additional feature: the "least recently used" or LRU bit. In the event of a read miss, either bank A or bank B will be updated with new data. The LRU bit flags the candidate for replacement. Statistically, of two blocks of data, the block most recently used is the block most likely to be needed again in the near future. By flagging the least recently used block, the 82385 ensures that the cache block replaced is the least likely to have data needed by the CPU.

2.2.3 Two Way Set Associative Cache Operation

2.2.3.1 READ HITS

When the 386 DX initiates a memory read cycle, the 82385 uses the 9-bit set address to select one of 512 sets. The two tags of this set are simultaneously compared with A14–A31, both tag valid bits checked, and both appropriate line valid bits checked. If either comparison produces a hit, the corresponding cache bank is directed to drive the selected doubleword onto the 386 DX data bus. (Note that both banks will never concurrently cache the same main memory location.) If the requested data resides in bank A, the LRU bit is pointed toward B. If B produces the hit, the LRU bit is pointed toward A.

2.2.3.2 READ MISSES

As in direct mapped operation, a read miss can be either a line or tag miss. Let's start with a tag miss example. Assume the 386 DX seeks line 9 of page 2, and that neither the A or B directory produces a tag match. Assume also, as indicated in Figure 2-3, that the LRU bit points to A. As the data returns from main memory, it is loaded into offset 9 of bank A. Concurrently, this data is validated by updating the set 1 directory entry for bank A. Specifically, the upper address bits overwrite the previous tag, the tag valid bit is set, the appropriate line valid bit is set, and the other seven line valid bits cleared. Since this data is the most recently used, the LRU bit is turned toward B. No change to bank B occurs.

If the next 386 DX request is line 10 of page two, the result will be a line miss. As the data returns from main memory, it will be written into offset 10 of bank A (tag hit/line miss in bank A), and the appropriate line valid bit will be set. A line miss in one bank will cause the LRU bit to point to the other bank. In this example, however, the LRU bit has already been turned toward B.

2.2.3.3 OTHER OPERATIONS THAT AFFECT THE CACHE AND CACHE DIRECTORY

Other operations that affect the cache and cache directory are write hits, snoop hits, cache flushes, and 82385 resets. A write hit updates the cache along with main memory. If directory A detects the hit, bank A is updated. If directory B detects the hit, bank B is updated. If one bank is updated, the LRU bit is pointed toward the other.

If a snoop hit invalidates an entry, for example, in cache bank A, the corresponding LRU bit is pointed toward A. This ensures that invalid data is the prime candidate for replacement in a read miss. Finally, resets and flushes behave just as they do in a direct mapped cache, clearing all tag valid bits.

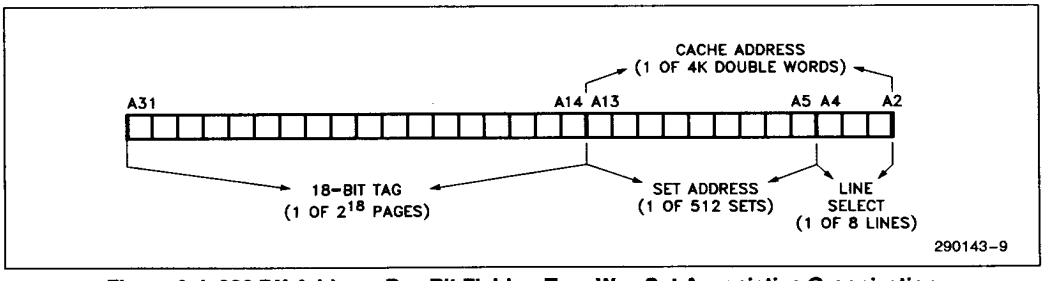


Figure 2-4. 386 DX Address Bus Bit Fields—Two-Way Set Associative Organization

3.0 82385 PIN DESCRIPTION

The 82385 creates the 82385 local bus, which is a functional 386 DX interface. To facilitate understanding, 82385 local bus signals go by the same name as their 386 DX equivalents, except that they are preceded by the letter "B". The 82385 local bus equivalent to ADS# is BADS#, the equivalent to NA# is BNA#, etc. This convention applies to bus states as well. For example, BT1P is the 82385 local bus state equivalent to the 386 DX T1P state.

3.1 386 DX CPU/82385 INTERFACE SIGNALS

These signals form the direct interface between the 386 DX and 82385.

3.1.1 386 DX CPU/82385 Clock (CLK2)

CLK2 provides the fundamental timing for an 386 DX CPU/82385 system, and is driven by the same source that drives the 386 DX CLK2 input. The 82385, like the 386 DX, divides CLK2 by two to generate an internal "phase indication" clock. (See Figure 3-1.) The CLK2 period whose rising edge drives the internal clock low is called PHI1, and the CLK2 period that drives the internal clock high is called PHI2. A PHI1-PHI2 combination (in that order) is known as a "T" state, and is the basis for 386 DX bus cycles.

3.1.2 386 DX CPU/82385 Reset (RESET)

This input resets the 82385, bringing it to an initial known state, and is driven by the same source that drives the 386 DX RESET input. A reset effectively flushes the cache by clearing all cache directory tag valid bits. The falling edge of RESET is synchronized to CLK2, and used by the 82385 to properly establish the phase of its internal clock. (See Figure 3-2.) Specifically, the second internal phase following the falling edge of RESET is PHI2.

3.1.3 386 DX CPU/82385 Address Bus (A2–A31), Byte Enables (BE0#–BE3#), and Cycle Definition Signals (M/IO#, D/C#, W/R#, LOCK#)

The 82385 directly connects to these 386 DX outputs. The 386 DX address bus is used in the cache directory comparison to see if data referenced by 386 DX resides in the cache, and the byte enables inform the 82385 as to which portions of the data bus are involved in an 386 DX cycle. The cycle definition signals are decoded by the 82385 to determine the type of cycle the 386 DX is executing.

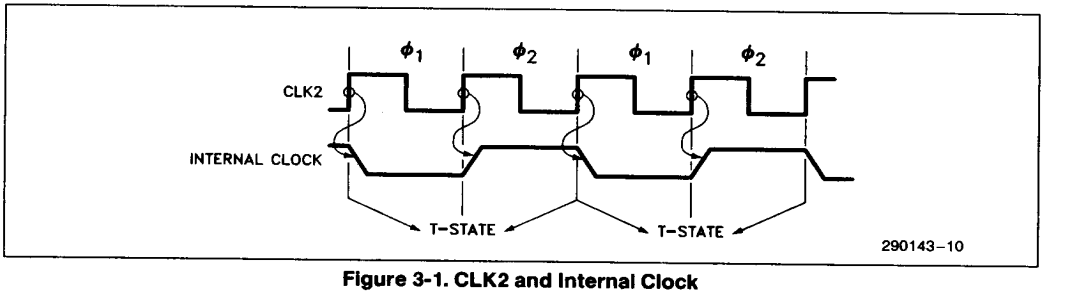


Figure 3-1. CLK2 and Internal Clock

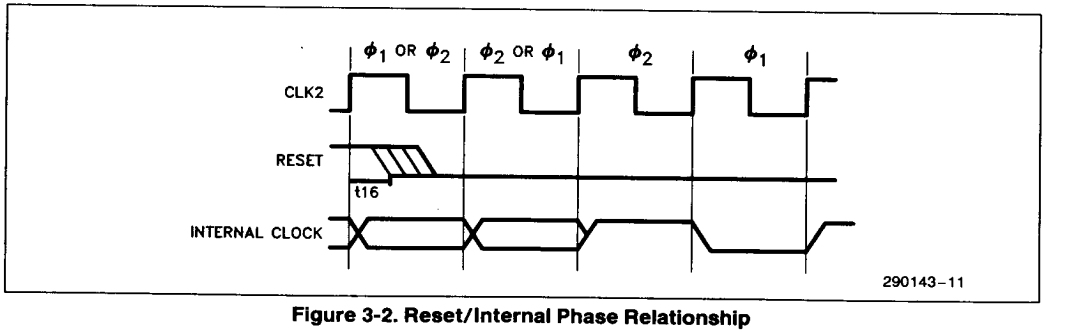


Figure 3-2. Reset/Internal Phase Relationship

3.1.4 386 DX CPU/82385 Address Status (ADS#) and Ready Input (READYI#)

ADS#, a 386 DX output, tells the 82385 that new address and cycle definition information is available. READYI#, an input to both the 386 DX (via the 386 DX READY# input pin) and 82385, indicates the completion of an 386 DX bus cycle. ADS# and READYI# are used to keep track of the 386 DX bus state.

3.1.5 386 DX Next Address Request (NA#)

This 82385 output controls 386 DX pipelining. It can be tied directly to the 386 DX NA# input, or it can be logically "AND"ed with other 386 DX local bus next address requests.

3.1.6 Ready Output (READYO#) and Bus Ready Enable (BRDYEN#)

The 82385 directly terminates all but two types of 386 DX bus cycles with its READYO# output. 386 DX local bus cycles must be terminated by the local device being accessed. This includes devices decoded using the 82385 LBA# signal and 80387 accesses. The other cycles not directly terminated by the 82385 are 82385 local bus reads, specifically cache read misses and non-cacheable reads. (Recall that the 82385 forwards and runs such cycles on the 82385 bus.) In these cycles the signal that terminates the 82385 local bus access is BREADY#, which is gated through to the 386 DX local bus such that the 386 DX and 82385 local bus cycles are concurrently terminated. BRDYEN# is used to gate the BREADY# signal to the 386 DX.

3.2 CACHE CONTROL SIGNALS

These 82385 outputs control the external 32 KB cache data memory.

3.2.1 Cache Address Latch Enable (CALEN)

This signal controls the latch (typically an F or AS series 74373) that resides between the low order 386 DX address bits and the cache SRAM address inputs. (The outputs of this latch are the "cache address" described in the previous chapter.) When CALEN is high the latch is transparent. The falling edge of CALEN latches the current inputs which remain applied to the cache data memory until CALEN returns to an active high state.

3.2.2 Cache Transmit/Receive (CT/R#)

This signal defines the direction of an optional data transceiver (typically an F or AS series 74245) between the cache and 386 DX data bus. When high, the transceiver is pointed towards the 386 DX local data bus (the SRAMs are output enabled). When low, the transceiver points towards the cache data memory. A transceiver is required if the cache is designed with SRAMs that lack an output enable control. A transceiver may also be desirable in a system that has a heavily loaded 386 DX local data bus. These devices are not necessary when using SRAMs which incorporate an output enable.

3.2.3 Cache Chip Selects (CS0#–CS3#)

These active low signals tie to the cache SRAM chip selects, and individually enable the four bytes of the 32-bit wide cache. CS0# enables D0–D7, CS1# enables D8–D15, CS2# enables D16–D23, and CS3# enables D24–D31. During read hits, all four bytes are enabled regardless of whether or not all four 386 DX byte enables are active. (The 386 DX ignores what it did not request.) Also, all four cache bytes are enabled in a read miss so as to update the cache with a complete line (double word). In a write hit, only those cache bytes that correspond to active byte enables are selected. This prevents cache data from being corrupted in a partial doubleword write.

5

3.2.4 Cache Output Enables (COEA#, COEB#) and Write Enables (CWEA#, CWEB#)

COEA# and COEB# are active low signals which tie to the cache SRAM or Transceiver output enables and respectively enable cache bank A or B. The state of DEFOE# (define cache output enable), an 82385 configuration input, determines the functional definition of COEA# and COEB#.

If DEFOE# = V_{IL}, in a two-way set associative cache, either COEA# or COEB# is active during read hit cycles only, depending on which bank is selected. In a direct mapped cache, both are activated during read hits, so the designer is free to use either one. This COEx# definition best suites cache SRAMs with output enables.

If DEFOE# = V_{IH}, COEx# is active during read hit, read miss (cache update) and write hit cycles only. This COEx# definition suites cache SRAMs without output enables. In such systems, transceivers are needed and their output enables must be active for writing, as well as reading, the cache SRAMs.

CWEA# and CWEB# are active low signals which tie to the cache SRAM write enables, and respectively enable cache bank A or B to receive data from the 386 DX data bus (386 DX write hit or read miss update). In a two-way set associative cache, one or the other is enabled in a read miss or write hit. In a direct mapped cache, both are activated, so the designer is free to use either one.

The various cache configurations supported by the 82385 are described in Chapter 4.

3.3 386 DX LOCAL BUS DECODE INPUTS

These 82385 inputs are generated by decoding the 386 DX address and cycle definition lines. These active low inputs are sampled at the end of the first state in which the address of a new 386 DX cycle becomes available (T1 or first T2P).

3.3.1 386 DX Local Bus Access (LBA#)

This input identifies an 386 DX access as directed to a resource (other than the cache) on the 386 DX local bus. (The 387 Numerics Coprocessor is considered a 386 DX local bus resource, but LBA# need not be generated as the 82385 internally decodes 387 accesses.) The 82385 simply ignores these cycles. They are neither forwarded to the system nor do they affect the cache or cache directory. Note that LBA# has priority over all other types of cycles. If LBA# is asserted, the cycle is interpreted as an 386 DX local bus access, regardless of the cycle type or status of NCA# or X16#. This allows any 386 DX cycle (memory, I/O, interrupt acknowledge, etc.) to be kept on the 386 local bus if desired.

3.3.2 Non-Cacheable Access (NCA#)

This active low input identifies a 386 DX cycle as non-cacheable. The 82385 forwards non-cacheable cycles to the 82385 local bus and runs them. The cache and cache directory are unaffected.

NCA# allows a designer to set aside a portion of main memory as non-cacheable. Potential applications include memory-mapped I/O and systems where multiple masters access dual ported memory via different busses. Another possibility makes use of the 386 DX D/C# output. The 82385 by default implements a unified code and data cache, but driving NCA# directly by D/C# creates a data only cache. If D/C# is inverted first, the result is a code only cache.

3.3.3 16-Bit Access (X16#)

X16# is an active low input which identifies 16-bit memory and/or I/O space, and the decoded signal that drives X16# should also drive the 386 DX BS16# input. 16-bit accesses are treated like non-cacheable accesses: they are forwarded to and executed on the 82385 local bus with no impact on the cache or cache directory. In addition, the 82385 locks the two halves of a non-aligned 16-bit transfer from interruption by another master, as does the 386 DX.

3.4 82385 LOCAL BUS INTERFACE SIGNALS

The 82385 presents a "386 DX-like" front end to the system, and the signals discussed in this section are 82385 local bus equivalents to actual 386 DX signals. These signals are named with respect to their 386 DX counterparts, but with the letter "B" appended to the front.

Note that the 82385 itself does not have equivalent output signals to the 386 DX data bus (D0–D31), address bus (A2–A31), and cycle definition signals (M/IO#, D/C#, W/R#). The 82385 data bus (BD0–BD31) is actually the system side of a latching transceiver, and the 82385 address bus and cycle definition signals (BA2–BA31, BM/IO#, BD/C#, BW/R#) are the outputs of an edge-triggered latch. The signals that control this data transceiver and address latch are discussed in Section 3.5.

3.4.1 82385 Bus Byte Enables (BBE0#–BBE3#)

BBE0#–BBE3# are the 82385 local bus equivalents to the 386 DX byte enables. In a cache read miss, the 82385 drives all four signals low, regardless of whether or not all four 386 DX byte enables are active. This ensures that a complete line (doubleword) is fetched from main memory for the cache update. In all other 82385 local bus cycles, the 82385 duplicates the logic levels of the 386 DX byte enables. The 82385 tri-states these outputs when it is not the current bus master.

3.4.2 82385 Bus Lock (BLOCK#)

BLOCK# is the 82385 local bus equivalent to the 386 DX LOCK# output, and distinguishes between locked and unlocked cycles. When the 386 DX runs a locked sequence of cycles (and LBA# is negated), the 82385 forwards and runs the sequence on the 82385 local bus, regardless of whether any locations

referenced in the sequence reside in the cache. A read hit will be run as if it is a read miss, but a write hit will update the cache as well as being completed to system memory. In keeping with 386 DX behavior, the 82385 does not allow another master to interrupt the sequence. BLOCK# is tri-stated when the 82385 is not the current bus master.

3.4.3 82385 Bus Address Status (BADS#)

BADS# is the 82385 local bus equivalent of ADS#, and indicates that a valid address (BA2–BA31, BBE0#–BBE3#) and cycle definition (BM/IO#, BW/R#, BD/C#) is available. It is asserted in BT1 and BT2P states, and is tri-stated when the 82385 does not own the bus.

3.4.4 82385 Bus Ready Input (BREADY#)

82385 local bus cycles are terminated by BREADY#, just as 386 DX cycles are terminated by the 386 DX READY# input. In 82385 local bus read cycles, BREADY# is gated by BRDYEN# onto the 386 DX local bus, such that it terminates both the 386 DX and 82385 local bus cycles.

3.4.5 82385 Bus Next Address Request (BNA#)

BNA# is the 82385 local bus equivalent to the 386 DX NA# input, and indicates that the system is prepared to accept a pipelined address and cycle definition. If BNA# is asserted and the new cycle information is available, the 82385 begins a pipelined cycle on the 82385 local bus.

3.5 82385 BUS DATA TRANSCEIVER AND ADDRESS LATCH CONTROL SIGNALS

The 82385 data bus is the system side of a latching transceiver (typically an F or AS series 74646), and the 82385 address bus and cycle definition signals are the outputs of an edge-triggered latch (F or AS series 74374). The following is a discussion of the 82385 outputs that control these devices. An important characteristic of these signals and the devices they control is that they ensure that BD0–BD31, BA2–BA31, BM/IO#, BD/C#, and BW/R# reproduce the functionality and timing behavior of their 386 DX equivalents.

3.5.1 Local Data Strobe (LDSTB), Data Output Enable (DOE#), and Bus Transmit/Receive (BT/R#)

These signals control the latching data transceiver. BT/R# defines the transceiver direction. When high, the transceiver drives the 82385 data bus in write cycles. When low, the transceiver drives the 386 DX data bus in 82385 local bus read cycles. DOE# enables the transceiver outputs.

The rising edge of LDSTB latches the 386 DX data bus in all write cycles. The interaction of this signal and the latching transceiver is used to perform the 82385's posted write capability.

3.5.2 Bus Address Clock Pulse (BACP) and Bus Address Output Enable (BAOE#)

These signals control the latch that drives BA2–BA31, BM/IO#, BW/R#, and BD/C#. In any 386 DX cycle that is forwarded to the 82385 local bus, the rising edge of BACP latches the 386 DX address and cycle definition signals. BAOE# enables the latch outputs when the 82385 is the current bus master and disables them otherwise.

3.6 STATUS AND CONTROL SIGNALS

5

3.6.1 Cache Miss Indication (MISS#)

This output accompanies cacheable read and write miss cycles. This signal transitions to its active low state when the 82385 determines that a cacheable 386 DX access is a miss. Its timing behavior follows that of the 82385 local bus cycle definition signals (BM/IO#, BD/C#, BW/R#) so that it becomes available with BADS# in BT1 or the first BT2P. MISS# is floated when the 82385 does not own the bus, such that multiple 82385's can share the same node in multi-cache systems. (As discussed in Chapter 7, this signal also serves a reserved function in testing the 82385.)

3.6.2 Write Buffer Status (WBS)

The latching data transceiver is also known as the "posted write buffer." WBS indicates that this buffer contains data that has not yet been written to the system even though the 386 DX may have begun its next cycle. It is activated when 386 DX data is latched, and deactivated when the corresponding

82385 local bus write cycle is completed (BREADY#). (As discussed in Chapter 7, this signal also serves a reserved function in testing the 82385.)

WBS can serve several functions. In multi-processor applications, it can act as a coherency mechanism by informing a bus arbiter that it should let a write cycle run on the system bus so that main memory has the latest data. If any other 82385 cache subsystems are on the bus, they will monitor the cycle via their bus watching mechanisms. Any 82385 that detects a snoop hit will invalidate the corresponding entry in its local cache.

3.6.3 Cache Flush (FLUSH)

When activated, this signal causes the 82385 to clear all of its directory tag valid bits, effectively flushing the cache. (As discussed in Chapter 7, this signal also serves a reserved function in testing the 82385.) The primary use of the FLUSH input is for diagnostics and multi-processor support. The use of this pin as a coherency mechanism may impact software transparency.

The FLUSH input must be held active for at least 4 CLK (8 CLK2) cycles to complete the flush sequence. If FLUSH is still active after 4 CLK cycles, any accesses to the cache will be misses and the cache will not be updated (since FLUSH is active).

3.7 BUS ARBITRATION SIGNALS (BHOLD AND BHLDA)

In master mode, BHOLD is an input that indicates a request by a slave device for bus ownership. The 82385 acknowledges this request via its BHLDA output. (These signals function identically to the 386 DX HOLD and HLDA signals.)

The roles of BHOLD and BHLDA are reversed for an 82385 in slave mode. BHOLD is now an output indicating a request for bus ownership, and BHLDA an input indicating that the request has been granted.

3.8 COHERENCY (BUS WATCHING) SUPPORT SIGNALS (SA2–SA31, SSTB#, SEN)

These signals form the 82385's bus watching interface. The Snoop Address Bus (SA2–SA31) connects to the system address lines if masters reside at both the system and 82385 local bus levels, or the 82385 local bus address lines if masters reside only at the 82385 local bus level. Snoop Strobe (SSTB#) indicates that a valid address is on the snoop address inputs. Snoop Enable (SEN) indicates that the cycle is a write. In a system with masters only at the 82385 local bus level, SA2–SA31, SSTB#, and SEN can be driven respectively by BA2–BA31, BADS#, and BW/R# without any support circuitry.

3.9 CONFIGURATION INPUTS (2W/D#, M/S#, DEFOE#)

These signals select the configurations supported by the 82385. They are hardware strap options and must not be changed dynamically. 2W/D# (2-Way/Direct Mapped Select) selects a two-way set associative cache when tied high, or a direct mapped cache when tied low. M/S# (Master/Slave Select) chooses between master mode (M/S# high) and slave mode (M/S# low). DEFOE# defines the functionality of the 82385 cache output enables (COEA# and COEB#). DEFOE# allows the 82385 to interface to SRAMs with output enables (DEFOE# low) or to SRAMs requiring transceivers (DEFOE# high).

4.0 386 DX LOCAL BUS INTERFACE

The following is a detailed description of how the 82385 interfaces to the 386 DX and to 386 DX local bus resources. Items specifically addressed are the interfaces to the 386 DX, the cache SRAMs, and the 387 Numerics Coprocessor.

The many timing diagrams in this and the next chapter provide insight into the dual pipelined bus structure of a 386 DX CPU/82385 system. It's important to realize, however, that one need not know every possible cycle combination to use the 82385. The interface is simple, and the dual bus operation invisible to the 386 DX and system. To facilitate discussion of the timing diagrams, several conventions have been adopted. Refer to Figure 4-2A, and note that 386 DX bus cycles, 386 DX bus states, and 82385 bus states are identified along the top. All states can be identified by the "frame numbers" along the bottom. The cycles in Figure 4-2A include a cache read hit (CRDH), a cache read miss (CRDM), and a write (WT). WT represents any write, cacheable or not. When necessary to distinguish cacheable writes, a write hit goes by CWTH and a write miss by CWTM. Non-cacheable system reads go by SBRD. Also, it is assumed that system bus pipelining occurs even though the BNA# signal is not shown. When the system pipeline begins is a function of the system bus controller.

386 DX bus cycles can be tracked by ADS# and READY!#, and 82385 cycles by BADS# and BREADY#. These four signals are thus a natural

choice to help track parallel bus activity. Note in the timing diagrams that 386 DX cycles are numbered using ADS# and READYI#, and 82385 cycles using BADS# and BREADY#. For example, when the address of the first 386 DX cycle becomes available, the corresponding assertion of ADS# is marked "1", and the READYI# pulse that terminates the cycle is marked "1" as well. Whenever a 386 DX cycle is forwarded to the system, its number is forwarded as well so that the corresponding 82385 bus cycle can be tracked by BADS# and BREADY#.

The "N" value in the timing diagrams is the assumed number of main memory wait states inserted in a non-pipelined 82385 bus cycle. For example, a non-pipelined access to N=2 memory requires a total of four bus states, while a pipelined access requires three. (The pipeline advantage effectively hides one main memory wait state.)

4.1 PROCESSOR INTERFACE

This section presents the 386 DX CPU /82385 hardware interface and discusses the interaction and timing of this interface. Also addressed is how to decode the 386 DX address bus to generate the 82385 inputs LBA#, NCA#, and X16#. (Recall that LBA# allows memory and/or I/O space to be set aside for 386 DX local bus resources; NCA# allows system memory to be set aside as non-cacheable; and X16# allows system memory and/or I/O space to be reserved for 16-bit resources.) Finally, the 82385's handling of 16-bit space is discussed.

4.1.1 Hardware Interface

Figure 4-1 is a diagram of an 386 DX CPU/82385 system, which can be thought of as three distinct interfaces. The first is the 386 DX CPU/82385 interface (including the Ready Logic). The second is the cache interface, as depicted by the cache control bus in the upper left corner of Figure 4-1. The third is the 82385 bus interface, which includes both direct connects and signals that control the 74374 address/cycle definition latch and 74646 latching data transceiver. (The 82385 bus interface is the subject of the next chapter.)

As seen in Figure 4-1, the 386 DX CPU/82385 interface is a straightforward connection. The only necessary support logic is that required to sum all ready sources.

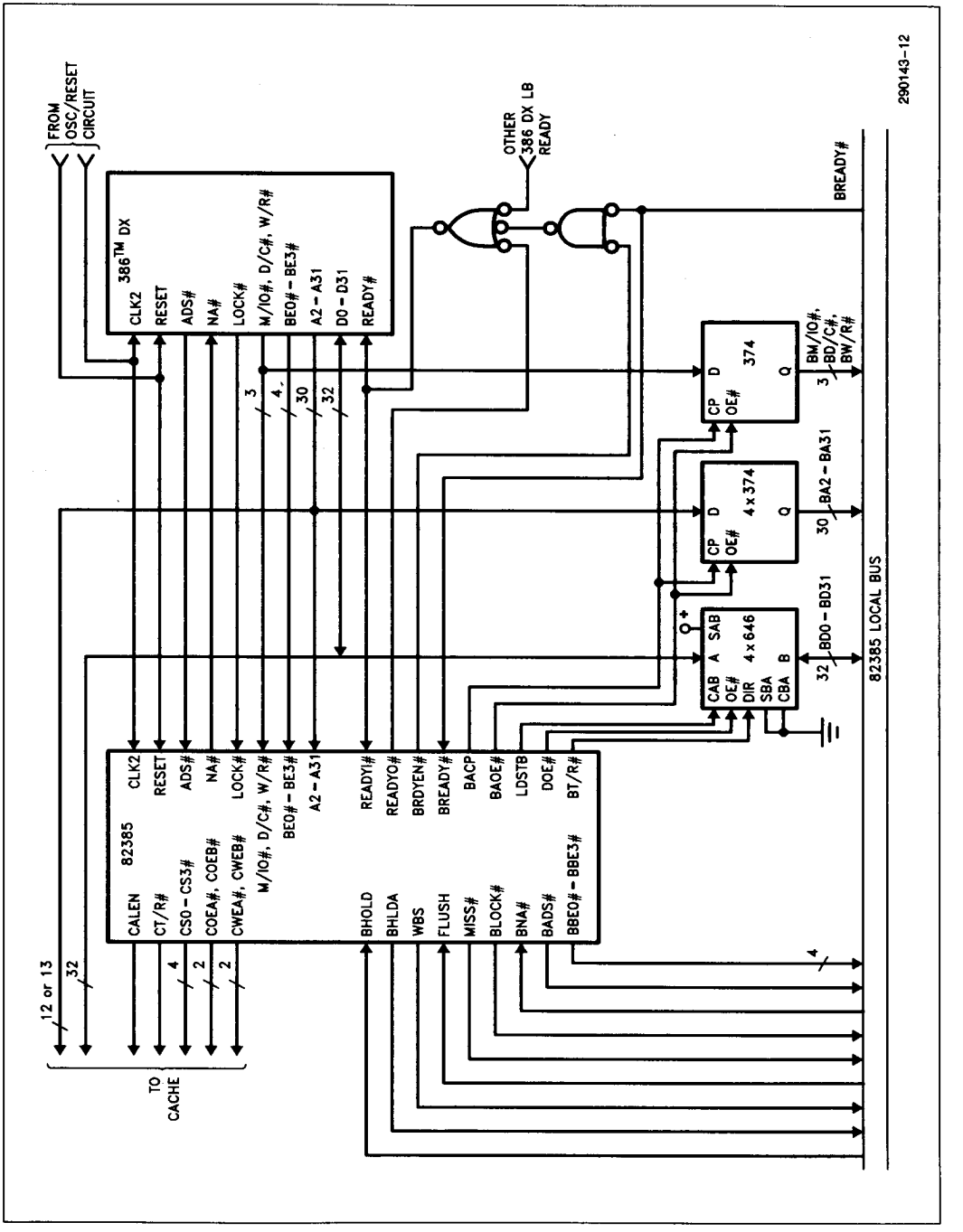


Figure 4-1. 386 DX CPU/82385 Interface

4.1.2 Ready Generation

Note in Figure 4-1 that the ready logic consists of two gates. The upper three-input AND gate (shown as a negative logic OR) sums all 386 DX local bus ready sources. One such source is the 82385 READYO# output, which terminates read hits and posted writes. The output of this gate drives the 386 DX READY# input and is monitored by the 82385 (via READYI#) to track the 386 DX bus state.

When the 82385 forwards a 386 DX read cycle to the 82385 bus (cache read miss or non-cacheable read), it does not directly terminate the cycle via READYO#. Instead, the 386 DX and 82385 bus cycles are concurrently terminated by a system ready source. This is the purpose of the additional two-input OR gate (negative logic AND) in Figure 4-1. When the 82385 forwards a read to the 82385 bus, it asserts BRDYEN# which enables the system ready signal (BREADY#) to directly terminate the 386 DX bus cycle.

Figures 4-2A and 4-2B illustrate the behavior of the signals involved in ready generation. Note in cycle 1 of Figure 4-2A that the 82385 READYO# directly terminates the hit cycle. In cycle 2, READYO# is not activated. Instead the 82385 BRDYEN# is activated in BT2, BT2P, or BT2I states such that BREADY# can concurrently terminate the 386 DX and 82385 bus cycles (frame 6). Cycle 3 is a posted write. The write data becomes available in T1P (frame 7), and

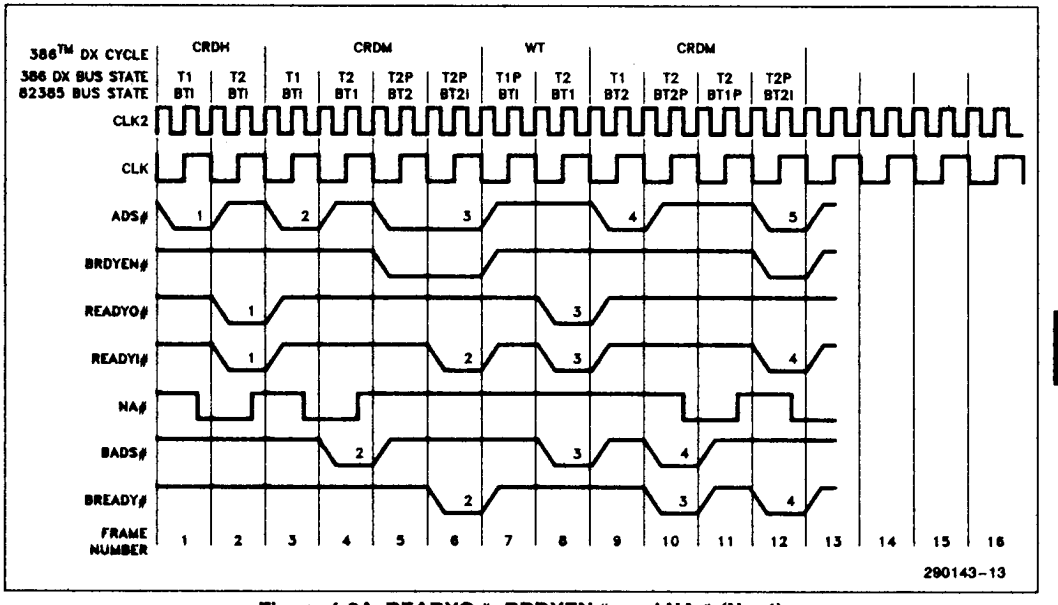


Figure 4-2A. READYO#, BRDYEN#, and NA# (N = 1)

the address, data, and cycle definition of the write are latched in T2 (frame 8). The 386 DX cycle is terminated by READYO# in frame 8 with no wait states. The 82385, however, sees the write cycle through to completion on the 82385 bus where it is terminated in frame 10 by BREADY#. In this case, the BREADY# signal is not gated through to the 386 DX . Refer to Figures 4-2A and 4-2B for clarification.

4.1.3. NA# and 386 DX Local Bus Pipelining

Cycle 1 of Figure 4-2A is a typical cache read hit. The 386 DX address becomes available in T1, and the 82385 uses this address to determine if the referenced data resides in the cache. The cache look-up is completed and the cycle qualified as a hit or miss in T1. If the data resides in the cache, the cache is directed to drive the 386 DX data bus, and the 82385 drives its READYO# output so the cycle can be terminated at the end of the first T2 with no wait states.

Although cycle 2 starts out like cycle 1, at the end of T1 (frame 3), it is qualified as a miss and forwarded to the 82385 bus. The 82385 bus cycle begins one state after the 386 DX bus cycle, implying a one wait state overhead associated with cycle 2 due to the look-up. When the 82385 encounters the miss, it immediately asserts NA#, which puts the 386 DX into pipelined mode. Once in pipelined mode, the 82385 is able to qualify an 386 DX cycle using the 386 DX pipelined address and control signals. The result is that the cache look-up state is hidden in all but the first of a contiguous sequence of read misses. This is shown in the first two cycles, both read misses, of Figure 4-2B. The CPU sees the look-up state in the first cycle, but not in the second. In fact, the second miss requires a total of only two states, as not only does 386 DX pipelining hide the look-up state, but system pipelining hides one of the main memory wait states. (System level pipelining via BNA# is discussed in the next chapter.) Several characteristics of the 82385's pipelining of the 386 DX are as follows:

— The above discussion applies to all system reads, not just cache read misses.

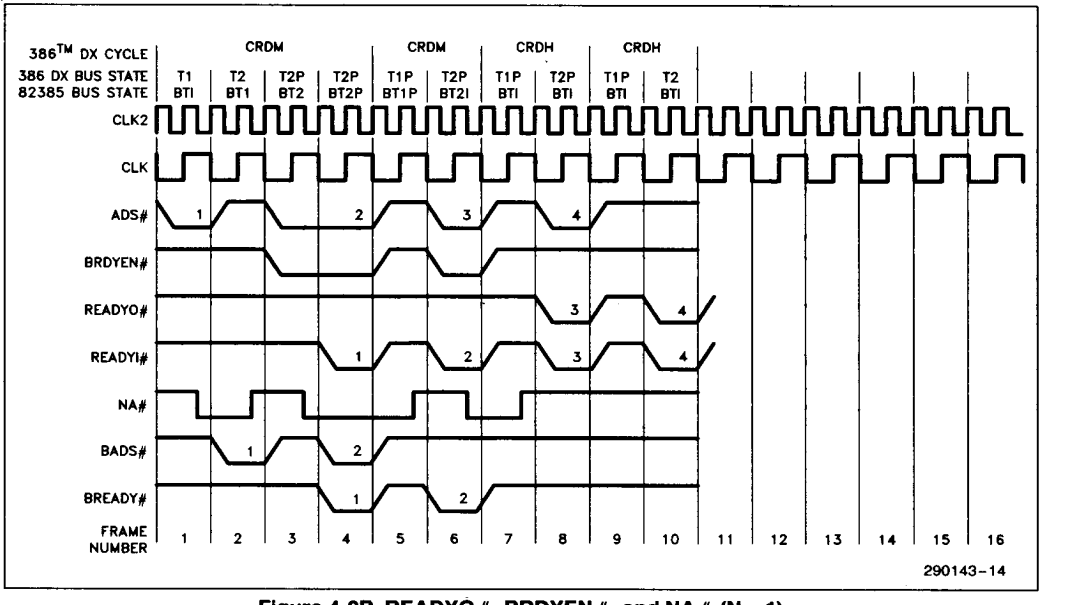


Figure 4-2B. READYO#, BRDYEN#, and NA# (N = 1)

— The 82385 provides the fastest possible switch to pipelining, T1-T2-T2P. The exception to this is when a system read follows a posted write. In this case, the sequence is T1-T2-T2-T2P. (Refer to cycle 4 of Figure 4-2A.) The number of T2 states is dependent on the number of main memory wait states.

— Refer to the read hit in Figure 4-2A (cycle 1), and note that NA# is actually asserted before the end of T1, before the hit/miss decision is made. This is of no consequence since even though NA# is sampled active in T2, the activation of READYO# in the same T2 renders NA# a "don't care". NA# is asserted in this manner to meet 386 DX timing requirements and to ensure the fastest possible switch to pipelined mode.

— All read hits and the majority of writes can be serviced by the 82385 with zero wait states in non-pipelined mode, and the 82385 accordingly attempts to run all such cycles in non-pipelined mode. An exception is seen in the hit cycles (cycles 3 and 4) of Figure 4-2B. The 82385 does not know soon enough that cycle 3 is a hit, and thus sustains the pipeline. The result is that three sequential hits are required before the 386 DX is totally out of pipelined mode. (The three hits look like T1P-T2P, T1P-T2, T1-T2.) Note that this does not occur if the number of main memory wait states is equal to or greater than two.

As far as the design is concerned, NA# is generally tied directly to the 386 DX NA# input. However, other local NA# sources may be logically "AND"ed with the 82385 NA# output if desired. It is essential, however, that no device other than the 82385 drive the 386 DX NA# input unless that device resides on the 386 DX local bus in space decoded via LBA#. If desired, the 82385 NA# output can be ignored and the 386 DX NA# input tied high. The 386 DX NA# input should never be tied low, which would always keep it active.

4.1.4 LBA#, NCA#, and X16# Generation

The 82385 input signals LBA#, NCA# and X16# are generated by decoding the 386 DX address (A2–A31) and cycle definition (W/R#, D/C#, M/IO#) lines. The 82385 samples them at the end of the first state in which they become available, which is either T1 or the first T2P cycle. The decode configuration and timings are illustrated respectively in Figures 4-3A and 4-3B.

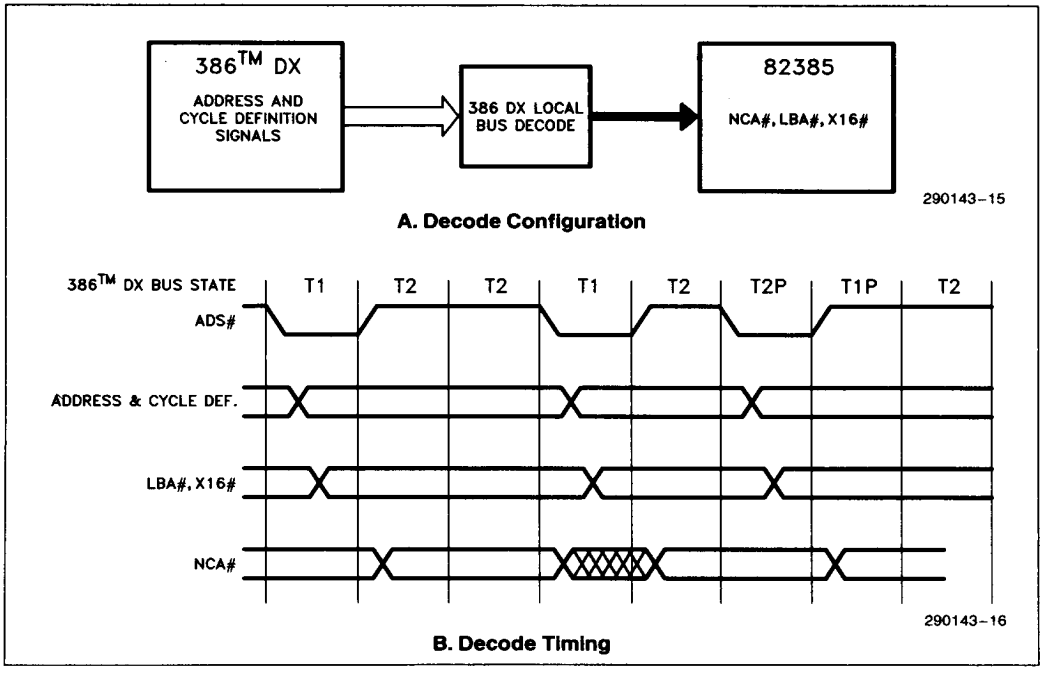


Figure 4-3. NCA#, LBA#, X16# Generation

4.1.5 82385 Handling of 16-Bit Space

As discussed previously, the 82385 does not cache devices decoded as 16-bit. Instead it makes provision to accommodate 16-bit space as non-cacheable via the X16# input. X16# is generated when the user decodes the 386 DX address and cycle definition lines for the BS16# input of the 386 DX (Figure 4-3). The decode output now drives both the 386 DX BS16# input and the 82385 X16# input. Cycles decoded this way are treated as non-cacheable. They are forwarded to and executed on the 82385 bus, but have no impact on the cache or cache directory. The 82385 also monitors the 386 DX byte enables in a 16-bit cycle to see if an additional cycle is required to complete the transfer. Specifically, a second cycle is required if (BE0# OR BE1#) AND (BE2# OR BE3#) is asserted in the current cycle. The 82385, like the 386 DX , will not allow the two halves of a 16-bit transfer to be interrupted by another master. There is an important distinction between the handling of 16-bit space in a 386 DX system with an 82385 as compared to a system without an 82385. The 386 DX BS16# input need not be asserted until the last state of a 16-bit cycle for the 386 DX to recognize it as such. The 82385, however, needs the information earlier, specifically at the end of the first 386 DX bus state (T1 or first T2P) in which the address of the 16-bit cycle becomes available. The result is that in a system without an 82385, 16-bit devices can define themselves as 16-bit devices "on the fly", while in a system with an 82385, 16-bit devices should be located in space set aside for 16-bit devices via the X16# decode.

4.2 CACHE INTERFACE

The following is a description of the external data cache and 82385 cache interface.

4.2.1 Cache Configurations

The 82385 controls the cache memory via the control signals shown in Figure 4-1. These signals drive one of four possible cache configurations, as depicted in Figures 4-4A through 4-4D. Figure 4-4A shows a direct mapped cache organized as 8K doublewords. The likely design choice is four 8K x 8 SRAMs. Figure 4-4B depicts the same cache memory but with a data transceiver between the cache and 386 DX data bus. In this configuration, CT/R# controls the transceiver direction, COEA# drives the transceiver output enable. (COEB# could also be used, and DEFOE# is strapped high.) A data buffer is required if the chosen SRAM does not have a separate output enable. Additionally, buffers may be used to ease SRAM timing requirements or in a system with a heavily loaded data bus. (Guidelines for SRAM selection are included in Chapter 6.)

Figure 4-4C depicts a two-way set associative cache organized as two banks (A and B) of 4K doublewords each. The likely design choice is sixteen 4K x 4 SRAM's. Finally, Figure 4-4D depicts the two-way organization with data buffers between the cache memory and data bus.

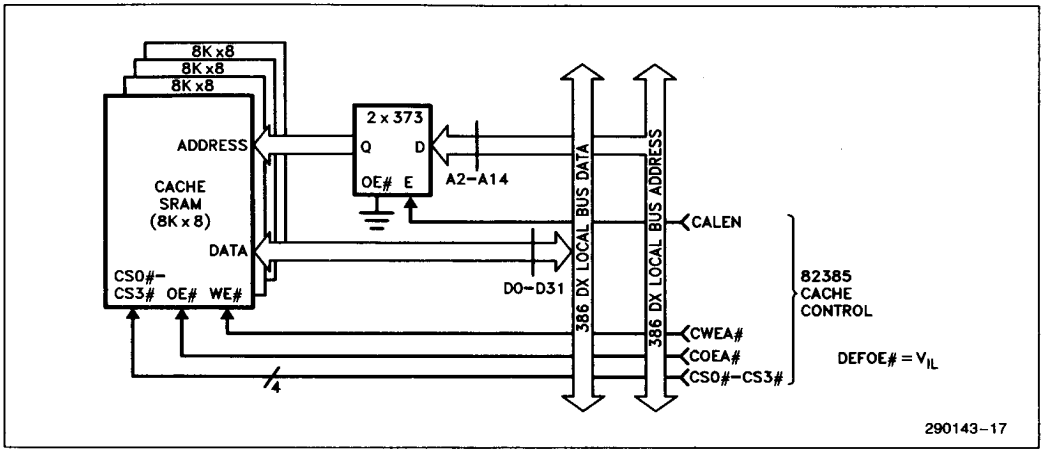


Figure 4-4A. Direct Mapped Cache without Data Buffers

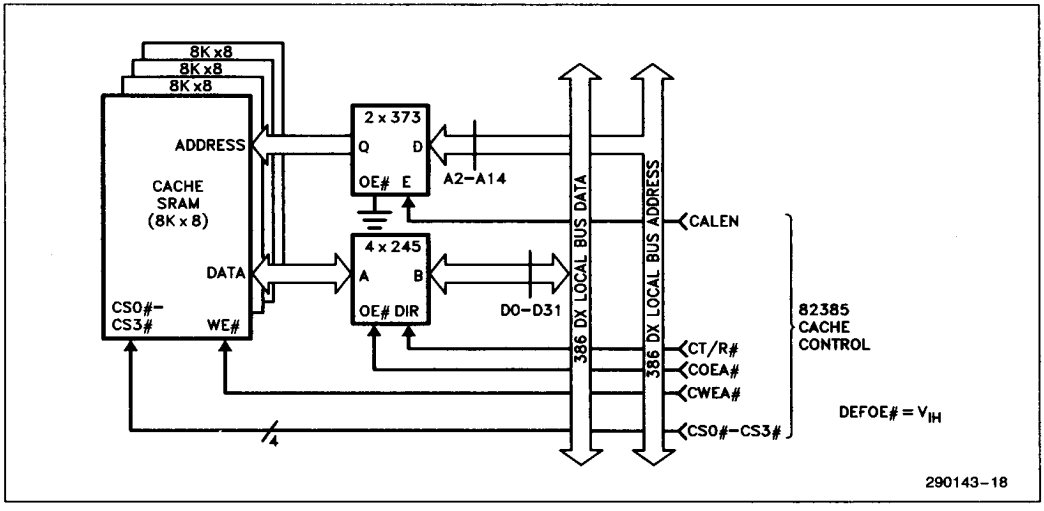


Figure 4-4B. Direct Mapped Cache with Data Buffers

5

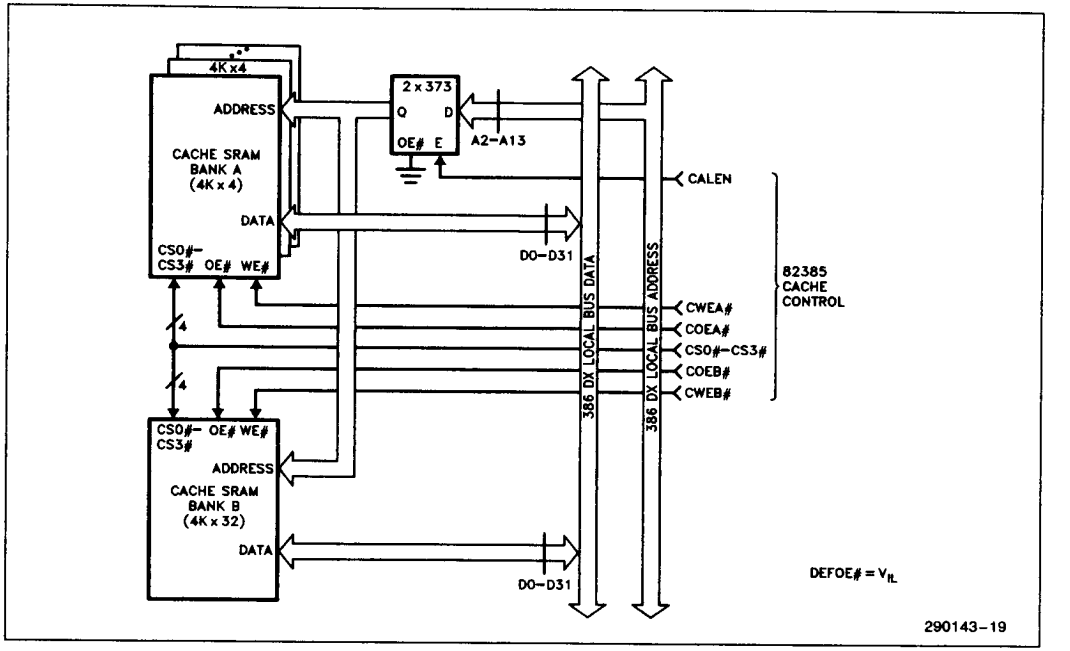


Figure 4-4C. Two-Way Set Associative Cache without Data Buffers

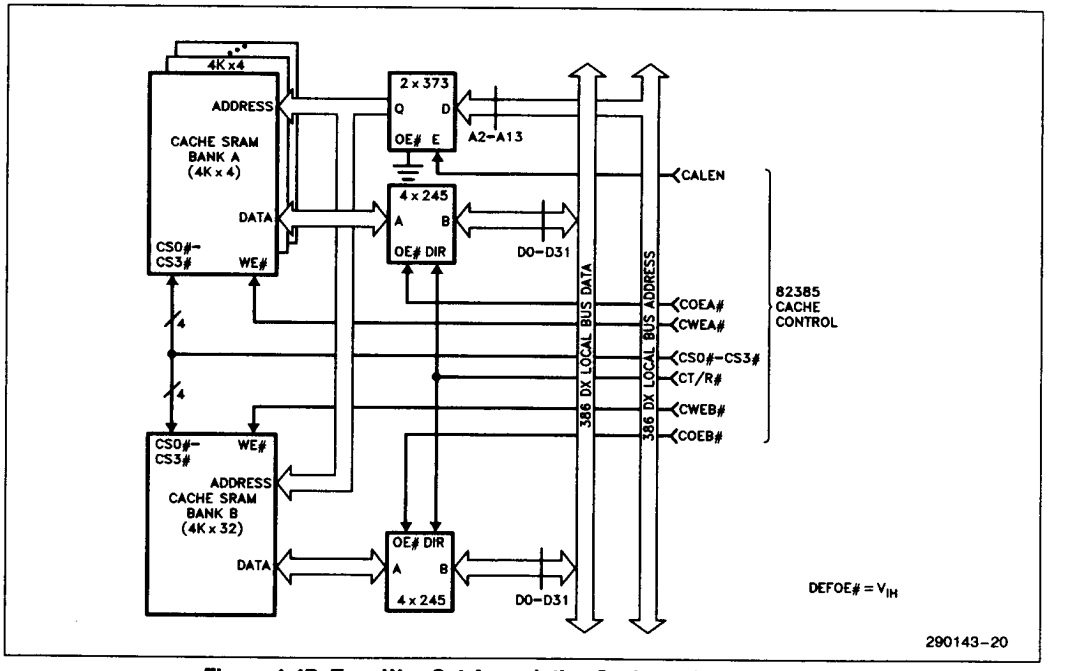


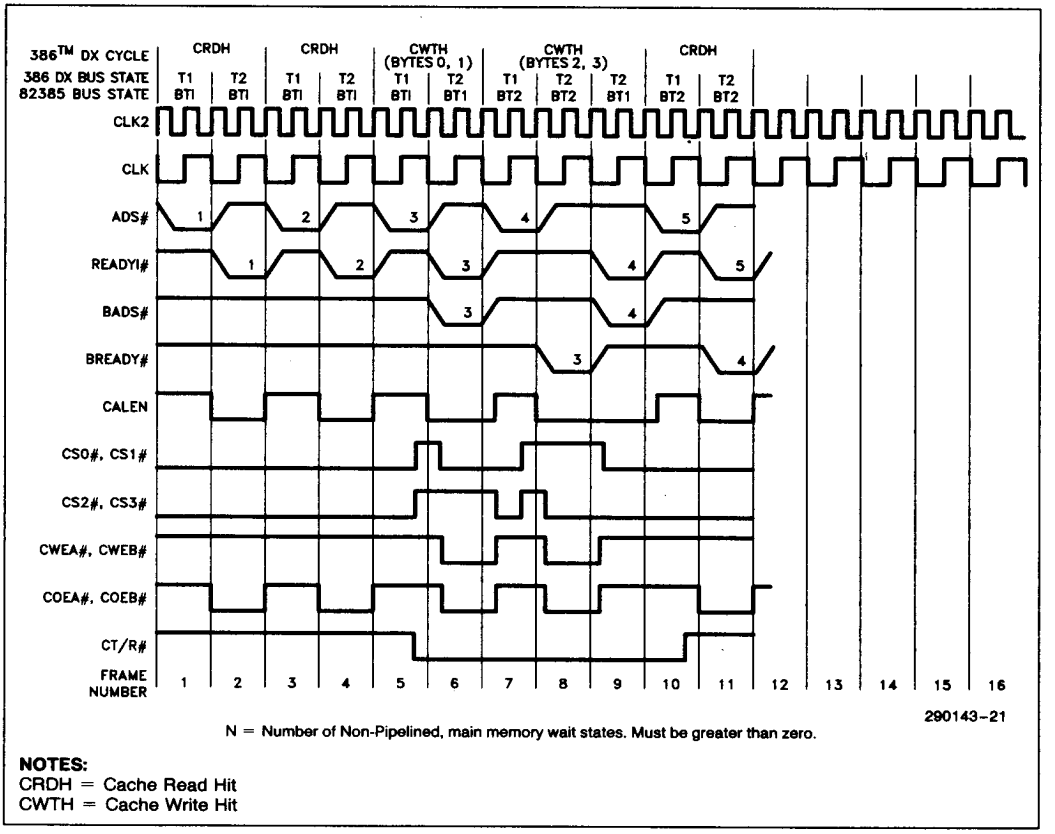
Figure 4-4D. Two-Way Set Associative Cache with Data Buffers

4.2.2 Cache Control—Direct Mapped

Figure 4-5A illustrates the timing of cache read and write hits, while Figure 4-5B illustrates cache updates. In a read hit, the cache output enables are driven from the beginning of T2 (cycle 1 of Figure 4-5A). If at the end of T1 the cycle is qualified as a cacheable read, the output enables are asserted on the assumption that the cycle will be a hit. (Driving the output enables before the actual hit/miss decision is made eases SRAM timing requirements.)

Cycle 1 of Figure 4-5B illustrates what happens when the assumption of a hit turns out to be wrong. Note that the output enables are asserted at the beginning of T2, but then disabled at the end of T2. Once the output enables are inactive, the 82385 turns the transceiver around (via CT/R#) and drives the write enables to begin the cache update cycle. Note in Figure 4-5B that once the 386 DX is in pipelined mode, the output enables need not be driven prior to a hit/miss decision, since the decision is made earlier via the pipelined address information.

One consequence of driving the output enables low in a miss before the hit/miss decision is made is that since the cache starts driving the 386 DX data bus,



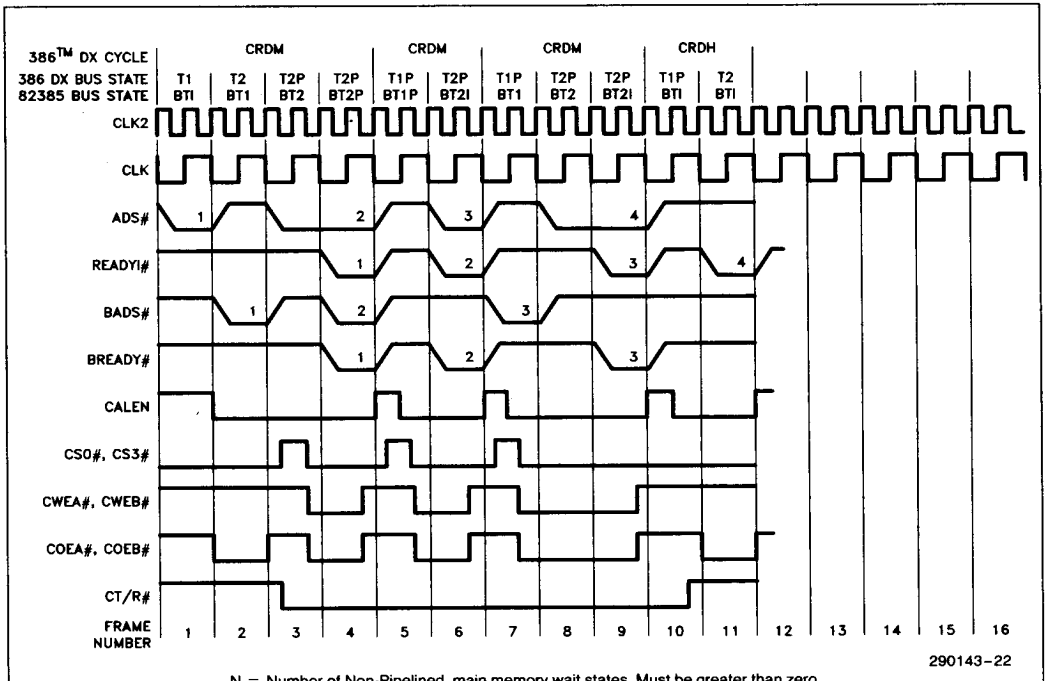
N = Number of Non-Pipelined, main memory wait states. Must be greater than zero.

NOTES:
CRDH = Cache Read Hit
CWTH = Cache Write Hit

Figure 4-5A. Cache Read and Write Cycles—Direct Mapped (N = 1)

the 82385 cannot enable the 74646 transceiver (Figure 4-1) until after the cache outputs are disabled. (The timing of the 74646 control signals is described in the next chapter.) The result is that the 74646 cannot be enabled soon enough to support N = 0 main memory ("N" was defined in section 4.0 as the number of non-pipelined main memory wait states). This means that memory which can run with zero wait states in a non-pipelined cycle should not be mapped into cacheable memory. This should not present a problem, however, as a main memory system built with N = 0 memory has no need of a cache. (The main memory is as fast as the cache.) Zero wait state memory can be supported if it is decoded as non-cacheable. The 82385 knows that a cycle is non-cacheable in time not to drive the cache output enables, and can thus enable the 74646 sooner.

In a write hit, the 82385 only updates the cache bytes that are meant to be updated as directed by the 386 DX byte enables. This prevents corrupting cache data in partial doubleword writes. Note in Figure 4-5A that the appropriate bytes are selected via the cache byte select lines CS0# – CS3#. In a read hit, all four select lines are driven as the 386 DX will simply ignore data it does not need. Also, in a cache update (read miss), all four selects are active in order to update the cache with a complete line (doubleword).



N = Number of Non-Pipelined, main memory wait states. Must be greater than zero.

NOTE:
CRDM = Cache Read Miss

Figure 4-5B. Cache Update Cycles—Direct Mapped (N = 1)

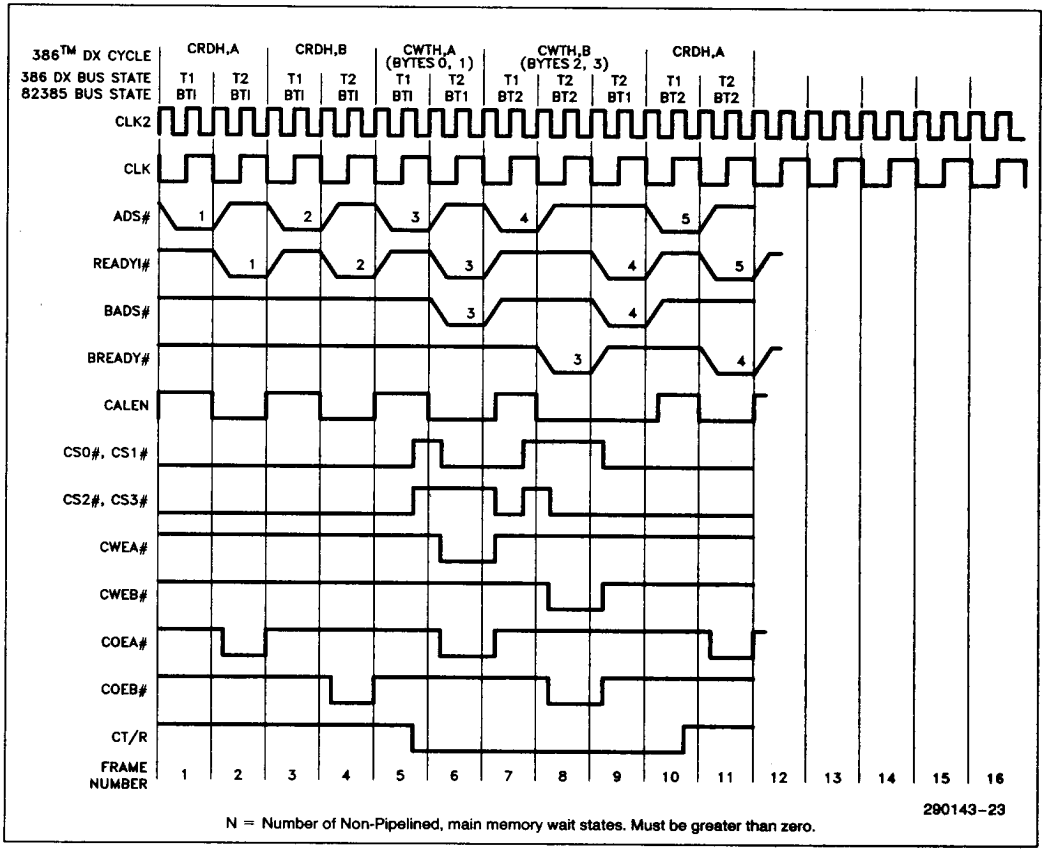
4.2.3 Cache Control—Two-Way Set Associative

Figures 4-6A and 4-6B illustrate the timing of cache read hits, write hits, and updates for a two-way set associative cache. (Note that the cycle sequences are the same as those in Figures 4-5A and 4-5B.) In a cache read hit, only one bank on the other is enabled to drive the 386 DX data bus, so unlike the control of a direct mapped cache, the appropriate cache output enable cannot be driven until the outcome of the hit/miss decision is known. (This implies stricter SRAM timing requirements for a two-way set associative cache.) In write hits and read misses, only one bank or the other is updated.

4.3 387™ DX INTERFACE

The 387 DX Math Coprocessor interfaces to the 386 DX just as it would in a system without an 82385. The 387 DX READYO# output is logically "AND"ed along with all other 386 DX local bus ready sources (Figure 4-1), and the output is fed to the 387 DX READY#, 82385 READYI#, and 386 DX READY# inputs.

The 386 DX uniquely addresses the 387 DX by driving M/IO# low and A31 high. The 82385 decodes this internally and treats 387 DX accesses in the same way it treats 386 DX cycles in which LBA# is asserted, it ignores them.



N = Number of Non-Pipelined, main memory wait states. Must be greater than zero.

Figure 4-6A. Cache Read and Write Cycles—Two Way Set Associative (N = 1)

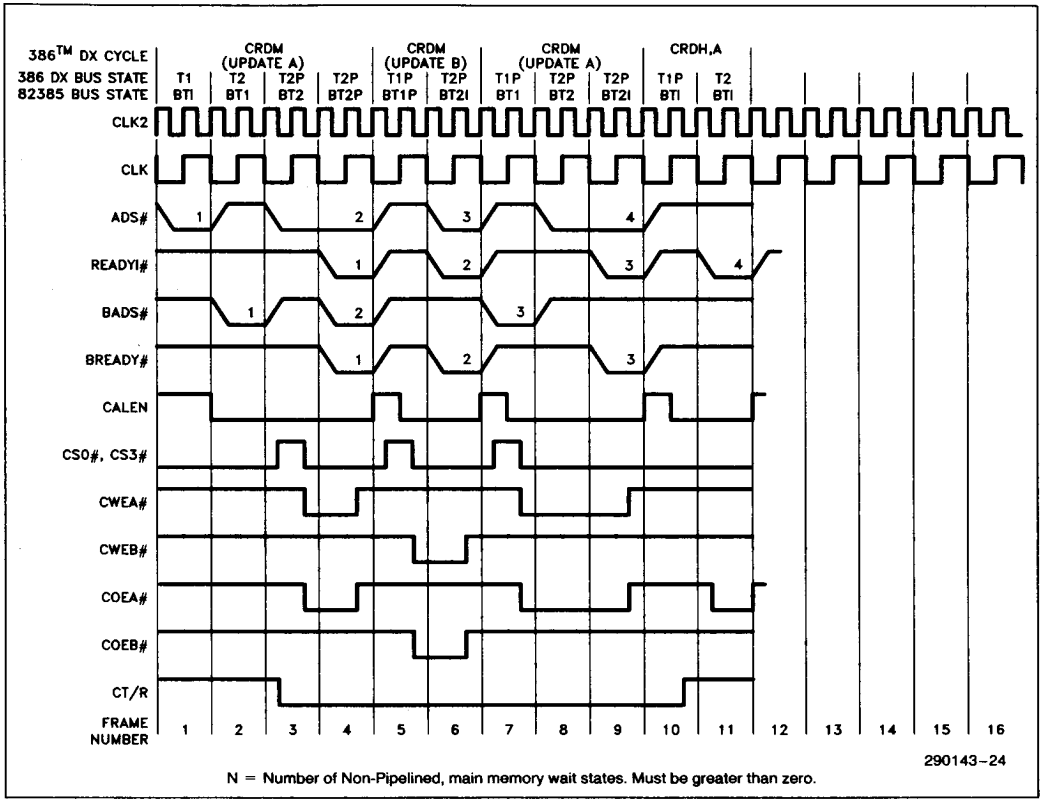


Figure 4-6B. Cache Update Cycles—Two Way Set Associative (N = 1)

5.0 82385 LOCAL BUS AND SYSTEM INTERFACE

The 82385 system interface is the 82385 Local Bus, which presents a "386 DX -like" front end to the system. The system ties to it just as it would to a 386 DX . Although this 386 DX -like front end is functionally equivalent to a 386 DX , there are timing differences which can easily be accounted for in a system design.

The following is a description of the 82385 system interface. After presenting the 82385 bus state machine, the 82385 bus signals are described, as are techniques for accommodating any differences between the 82385 bus and 386 DX bus. Following this is a discussion of the 82385's condition upon reset.

5.1 THE 82385 BUS STATE MACHINE

5.1.1 Master Mode

Figure 5-1A illustrates the 82385 bus state machine when the 82385 is programmed in master mode. Note that it is almost identical to the 386 DX bus state machine, only the bus states are 82385 bus states (BT1P, BTH, etc.) and the state transitions are conditioned by 82385 bus inputs (BNA#, BHOLD, etc.). Whereas a "pending request" to the 386 DX state machine indicates that the 386 DX execution or prefetch unit needs bus access, a pending request to the 82385 state machine indicates that a 386 DX bus cycle needs to be forwarded to the system (read miss, non-cacheable read, write,

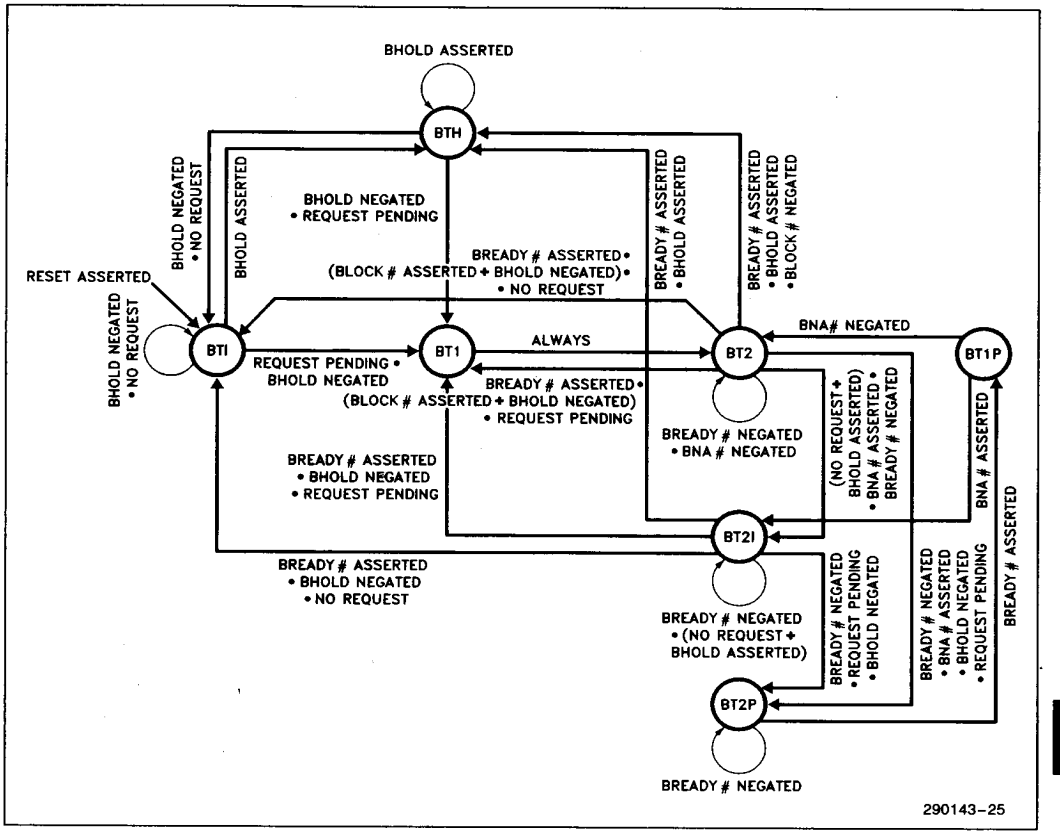


Figure 5-1A. 82385 Local Bus State Machine—Master Mode

5

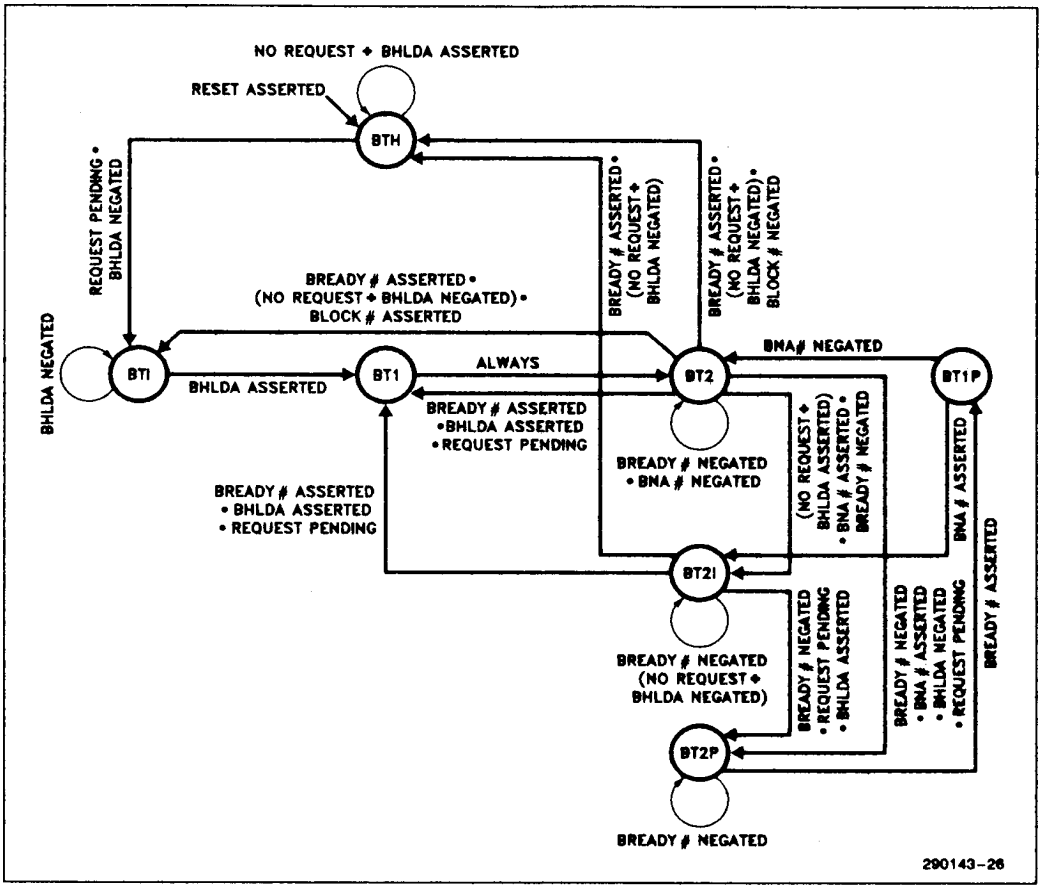


Figure 5-1B. 82385 Local Bus State Machine—Slave Mode

etc.). The only difference between the state machines is that the 82385 does not implement a direct BT1P–BT2P transition. If BNA# is asserted in BT1P, the resulting state sequence is BT1P-BT2I-BT2P. The 82385's ability to sustain a pipeline is not affected by the lack of this state transition.

5.1.2 Slave Mode

The 82385's slave mode state machine (Figure 5-1B) is similar to the master mode machine except that now transitions are conditioned by BHLDA rather than BHOLD. (Recall that in slave mode, the roles of BHOLD and BHLDA are reversed from their master mode roles.) Figure 5-2 clarifies slave mode state machine operation. Upon reset, a slave mode 82385 enters the BTH state. When the 386 DX of the slave 82385 subsystem has a cycle that needs to be forwarded to the system, the 82385 moves to BTI and issues a hold request via BHOLD. It is important to note that a slave mode 82385 does not drive the bus in a BTI state. When the master or bus arbiter returns BHLDA, the slave 82385 enters BT1 and runs the cycle. When the cycle is completed, and if no additional requests are pending, the 82385 moves back to BTH and disables BHOLD.

If, while a slave 82385 is running a cycle, the master or arbiter drops BHLDA (Figure 5-2B), the 82385 will complete the current cycle, move to BTH and remove the BHOLD request. If the 82385 still had cycles to run when it was kicked off the bus, it will immediately assert a new BHOLD and move to BTI to await bus acknowledgement. Note, however, that it will only move to BTI if BHLDA is negated, ensuring that the handshake sequence is completed.

There are several cases in which a slave 82385 will not immediately release the bus if BHLDA is dropped. For example, if BHLDA is dropped during a BT2P state, the 82385 has already committed to the next system bus pipelined cycle and will execute it before releasing the bus. Also, the 82385 will complete the second half of a two-cycle 16-bit transfer, or will complete a sequence of locked cycles before releasing the bus. This should not present any problems, as a properly designed arbiter will not assume that the 82385 has released the bus until it sees BHOLD become inactive.

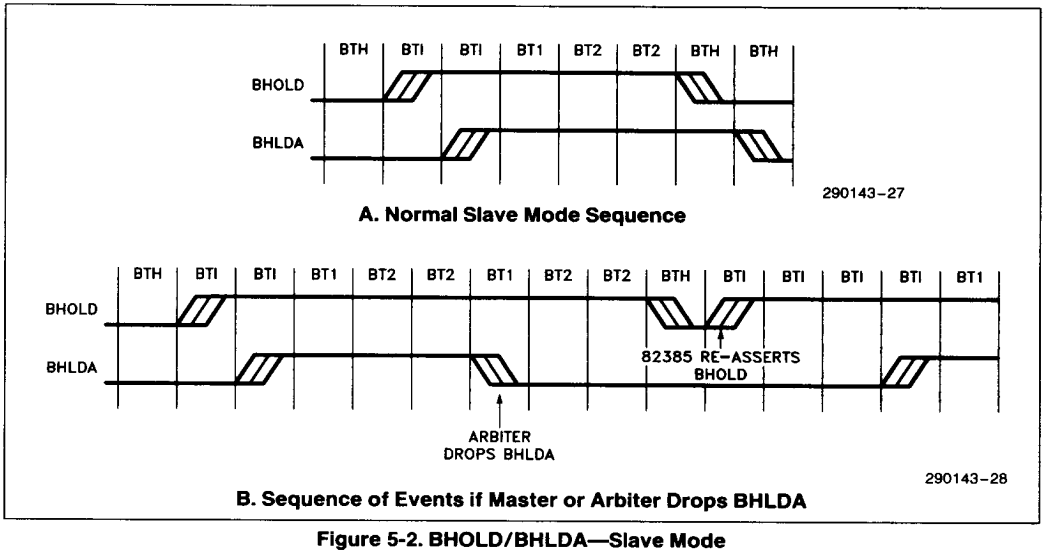


Figure 5-2. BHOLD/BHLDA—Slave Mode

5.2 The 82385 Local Bus

The 82385 bus can be broken up into two groups of signals: those which have direct 386 DX counterparts, and additional status and control signals provided by the 82385. The operation and interaction of all 82385 bus signals are depicted in Figures 5-3A through 5-3L for a wide variety of cycle sequences. These diagrams serve as a reference for the 82385 bus discussion and provide insight into the dual bus operation of the 82385.

5.2.1 82385 Bus Counterparts to 386 DX Signals

The following sections discuss the signals presented on the 82385 local bus which are functional equivalents to the signals present at the 386 DX local bus.

5.2.1.1 ADDRESS BUS (BA2–BA31) AND CYCLE DEFINITION SIGNALS (BM/IO#, BD/C#, BW/R#)

These signals are not driven directly by the 82385, but rather are the outputs of the 74374 address/cycle definition latch. (Refer to Figure 4-1 for the hardware interface.) This latch is controlled by the 82385 BACP and BAOE# outputs. The behavior and timing of these outputs and the latch they control (typically F or AS series TTL) ensure that BA2–BA31, BM/IO#, BW/R#, and BD/C# are compatible in timing and function to their 386 DX counterparts.

The behavior of BACP can be seen in Figure 5-3B, where the rising edge of BACP latches and forwards the 386 DX address and cycle definition signals in a BT1 or first BT2P state. However, the 82385 need not be the current bus master to latch the 386 DX address, as evidenced by cycle 4 of Figure 5-3A. In this case, the address is latched in frame 8, but not forwarded to the system (via BAOE#) until frame 10. (The latch and output enable functions of the 74374 are independent and invisible to one another.)

Note that in frames 2 and 6 the BACP pulses are marked "False." The reason is that BACP is issued and the address latched before the hit/miss determination is made. This ensures that should the cycle be a miss, the 82385 bus can move directly into BT1 without delay. In the case of a hit, the latched address is simply never qualified by the assertion of BADS#. The 82385 bus stays in BTI if there is no access pending (new cycle is a hit) and no bus activity. It will move to and stay in BT2I if the system has requested a pipelined cycle and the 82385 does not have a pending bus access (new cycle is a hit).

5.2.1.2 DATA BUS (BD0–BD31)

The 82385 data bus is the system side of the 74646 latching transceiver. (See Figure 4-1.) This device is controlled by the 82385 outputs LDSTB, DOE#, and BT/R#. LDSTB latches data in write cycles, DOE# enables the transceiver outputs, and BT/R# controls the transceiver direction. The interaction of these signals and the transceiver is such that BD0–BD31 behave just like their 386 DX counterparts. The transceiver is configured such that data flow in write cycles (A to B) is latched, and data flow in read cycles (B to A) is flow-through.

Although BD0–BD31 function just like their 386 DX counterparts, there is a timing difference that must be accommodated for in a system design. As mentioned above, the transceiver is transparent during read cycles, so the transceiver propagation delay must be added to the 386 DX data setup. In addition, the cache SRAM setup must be accommodated for in cache read miss cycles.

For non-cacheable reads the data setup is given by:

Min BD0–BD31 Read Data Setup	=	386 DX Min Data Setup	+	74646 B-to-A Max Propagation Delay

The required BD0–BD31 setup in a cache read miss is given by:

Min BD0–BD31 Read Data Setup	=	74646 B-to-A Max Propagation Delay	+	Cache SRAM Min Write Setup		
	+	One CLK2 Period	–	82385 CWEA# or CWEB# Min Delay		

If a data buffer is located between the 386 DX data bus and the cache SRAMs, then its maximum propagation delay must be added to the above formula as well. A design analysis should be completed for every new design to determine actual margins.

A design can accommodate the increased data setup by choosing appropriately fast main memory DRAMs and data buffers. Alternatively, a designer may deal with the longer setup by inserting an extra wait state into cache read miss cycles. If an additional state is to be inserted, the system bus controller should sample the 82385 MISS# output to distinguish read misses from cycles that do not require the longer setup. Tips on using the 82385 MISS# signal are presented later in this chapter.

The behavior of LDSTB, DOE#, and BT/R# can be understood via Figures 5-3A through 5-3L. Note that in cycle 1 of Figure 5-3A (a non-cacheable system read), DOE# is activated midway through BT1, but in cycle 1 of Figure 5-3B (a cache read miss), DOE# is not activated until midway through BT2. The reason is that in a cacheable read cycle, the cache SRAMs are enabled to drive the 386 DX data bus before the outcome of the hit/miss decision (in anticipation of a hit). In cycle 1 of Figure 5-3B, the assertion of DOE# must be delayed until after the 82385 has disabled the cache output buffers. The result is that N=0 main memory should not be mapped into the cache.

5.2.1.3 BYTE ENABLES (BBE0#–BBE3#)

These outputs are driven directly by the 82385, and are completely compatible in timing and function with their 386 DX counterparts. When a 386 DX cycle is forwarded to the 82385 bus, the 386 DX byte enables are duplicated on BBE0#–BBE3#. The one exception is a cache read miss, during which BBE0#–BBE3# are all active regardless of the status of the 386 DX byte enables. This ensures that the cache is updated with a valid 32-bit entry.

5.2.1.4 ADDRESS STATUS (BADS#)

BADS# is identical in function and timing to its 386 DX counterpart. It is asserted in BT1 and BT2P states, and indicates that valid address and cycle definition (BA2–BA31, BBE0#–BBE3#, BM/IO#, BW/R#, BD/C#) information is available on the 82385 bus.

5.2.1.5 READY (BREADY#)

The 82385 BREADY# input terminates 82385 bus cycles just as the 386 DX READY# input terminates 386 DX bus cycles. The behavior of BREADY# is the same as that of READY#, but note in the A.C. timing specifications that a cache read miss requires a longer BREADY# setup than do other cycles. This must be accommodated for in ready logic design.

5.2.1.6 NEXT ADDRESS (BNA#)

BNA# is identical in function and timing to its 386 DX counterpart. Note that in Figures 5-3A through 5-3L, BNA# is assumed asserted in every BT1P or first BT2 state. Along with the 82385's pipelining of the 386 DX , this ensures that the timing diagrams accurately reflect the full pipelined nature of the dual bus structure.

5.2.1.7 BUS LOCK (BLOCK#)

The 386 DX flags a locked sequence of cycles by asserting LOCK#. During a locked sequence, the 386 DX does not acknowledge hold requests, so the sequence executes without interruption by another master. The 82385 forces all locked 386 DX cycles to run on the 82385 bus (unless LBA# is active), regardless of whether or not the referenced location resides in the cache. In addition, a locked sequence of 386 DX cycles is run as a locked sequence on the 82385 bus; BLOCK# is asserted and the 82385 does not allow the sequence to be interrupted. Locked writes (hit or miss) and locked read misses affect the cache and cache directory just as their unlocked counterparts do. A locked read hit, however, is handled differently. The read is necessarily

forced to run on the 82385 local bus, and as the data returns from main memory, it is "re-copied" into the cache. (See Figure 5-3L.) The directory is not changed as it already indicates that this location exists in the cache. This activity is invisible to the system and ensures that semaphores are properly handled.

BLOCK# is asserted during locked 82385 bus cycles just as LOCK# is asserted during locked 386 DX cycles. The BLOCK# maximum valid delay, however, differs from that of LOCK#, and this must be accounted for in any circuitry that makes use of BLOCK#. The difference is due to the fact that LOCK#, unlike the other 386 DX cycle definition signals, is not pipelined. The situation is clarified in Figure 5-3K. In cycle 2 the state of LOCK# is not known before the corresponding system read starts (Frames 4 and 5). In this case, LOCK# is asserted at the beginning of T1P, and the delay for BLOCK# to become active is the delay of LOCK# from the 386 DX plus the propagation delay through the 82385. This occurs because T1P and the corresponding BT1P are concurrent (Frame 5). The result is that BLOCK# should not be sampled at the end of BT1P. The first appropriate sampling point is midway through the next state, as shown in Frame 6. In Figure 5-3L, the maximum delay for BLOCK# to become valid in Frame 4 is the same as the maximum delay for LOCK# to become valid from the 386 DX . This is true since the pipelining issue discussed above does not occur.

The 82385 should negate BLOCK# after BREADY# of the last 82385 Locked Cycle was asserted and Lock turns inactive. This means that in a sequence of cycles which begins with a 82385 Locked Cycle and goes on with all the possible Locked Cycles (other 82385 cycles, idles, and local cycles), while LOCK# is continuously active, the 82385 will maintain BLOCK# active continuously. Another implication is that in a Locked Posted Write Cycle followed by non-locked sequence, BLOCK# is negated one CLK after BREADY# of the write cycle. In other 82385 Locked Cycles, followed by non-locked sequences, BLOCK# is negated one CLK after LOCK# is negated, which occurs two CLKs after BREADY# is asserted. In the last case BLOCK# active moves by one CLK to the non-locked sequence.

The arbitration rules of Locked Cycles are:

MASTER MODE:

BHOLD input signal is ignored when BLOCK# or internal lock (16-bit non-aligned cycle) are active. BHLDA output signal remains inactive, and BAOE# output signal remains active at that time interval.

SLAVE MODE:

The 82385 does not relinquish the system bus if BLOCK# or internal lock are active. The BHOLD output signal remains active when BLOCK# or internal lock is active plus one CLK. The BHLDA input signal is ignored when BLOCK# or the internal lock is active plus one CLK. This means the 82385 slave does not respond to BHLDA inactivation. The BAOE# output signal remains active during the same time interval.

5.2.2 Additional 82385 Bus Signals

The 82385 bus provides two status outputs and one control input that are unique to cache operation and thus have no 386 DX counterparts. The outputs are MISS#, and WBS, and the input is FLUSH.

5.2.2.1 CACHE READ/WRITE MISS INDICATION (MISS#)

MISS# can be thought of as an extra 82385 bus cycle definition signal similar to BM/IO#, BW/R#, and BD/C#, that distinguishes cacheable read and write misses from other cycles. MISS#, like the other definition signals, becomes valid with BADS# (BT1 or first BT2P). The behavior of MISS# is illustrated in Figures 5-3B, 5-3C, and 5-3J. The 82385 floats MISS# when another master owns the bus, allowing multiple 82385s to share the same node in multi-cache systems. MISS# should thus be lightly pulled up ($\sim$20 KΩ) to keep it negated during hold (BTH) states.

MISS# can serve several purposes. As discussed previously, the BD0–BD31 and BREADY# setup times in a cache read miss are longer than in other cycles. A bus controller can distinguish these cycles by gating MISS# with BW/R#. MISS# may also prove useful in gathering 82385 system performance data.

5.2.2.2 WRITE BUFFER STATUS (WBS)

WBS is activated when 386 DX write cycle data is latched into the 74646 latching transceiver (via LDSTB). It is deactivated upon completion of the write cycle on the 82385 bus when the 82385 sees the BREADY# signal. WBS behavior is illustrated in Figures 5-3F through 5-3J, and potential applications are discussed in Chapter 3.

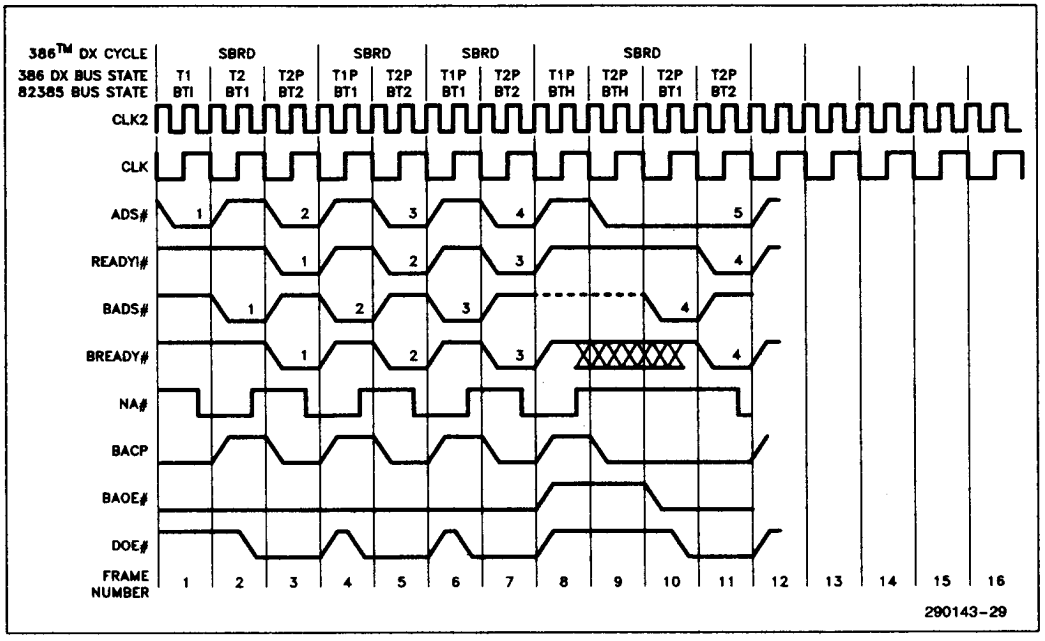


Figure 5-3A. Consecutive SBRD Cycles—(N = 0)

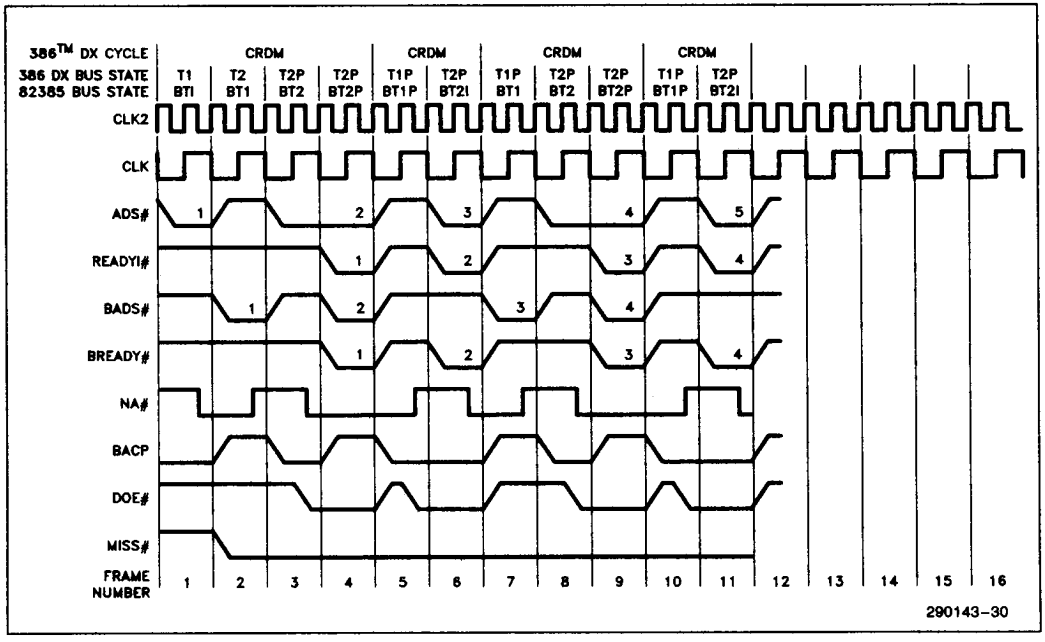


Figure 5-3B. Consecutive CRDM Cycles—(N = 1)

5

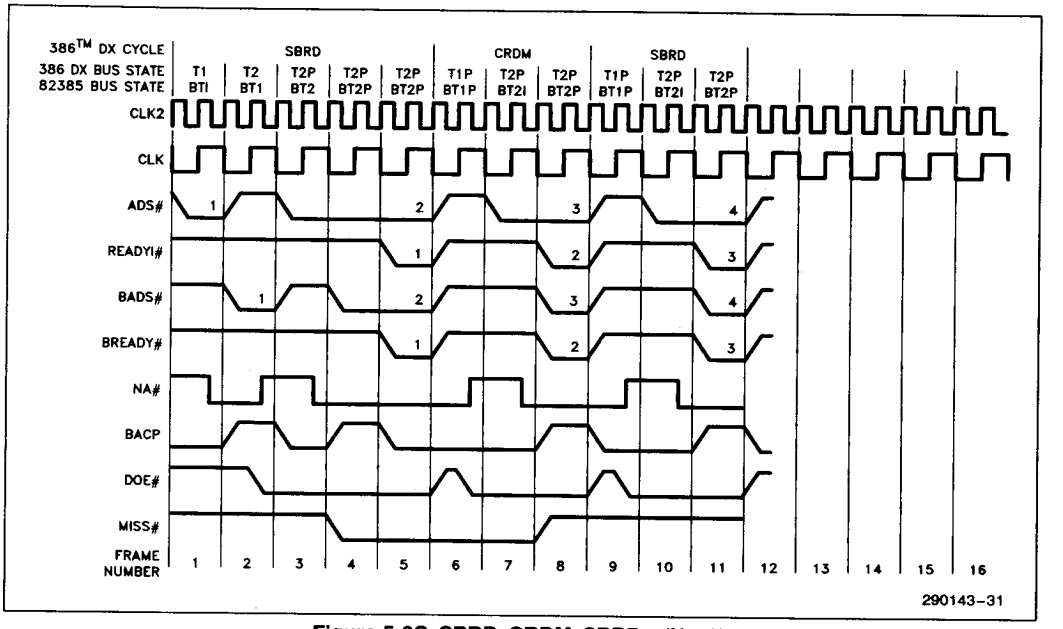


Figure 5-3C. SBRD, CRDM, SBRD—(N = 2)

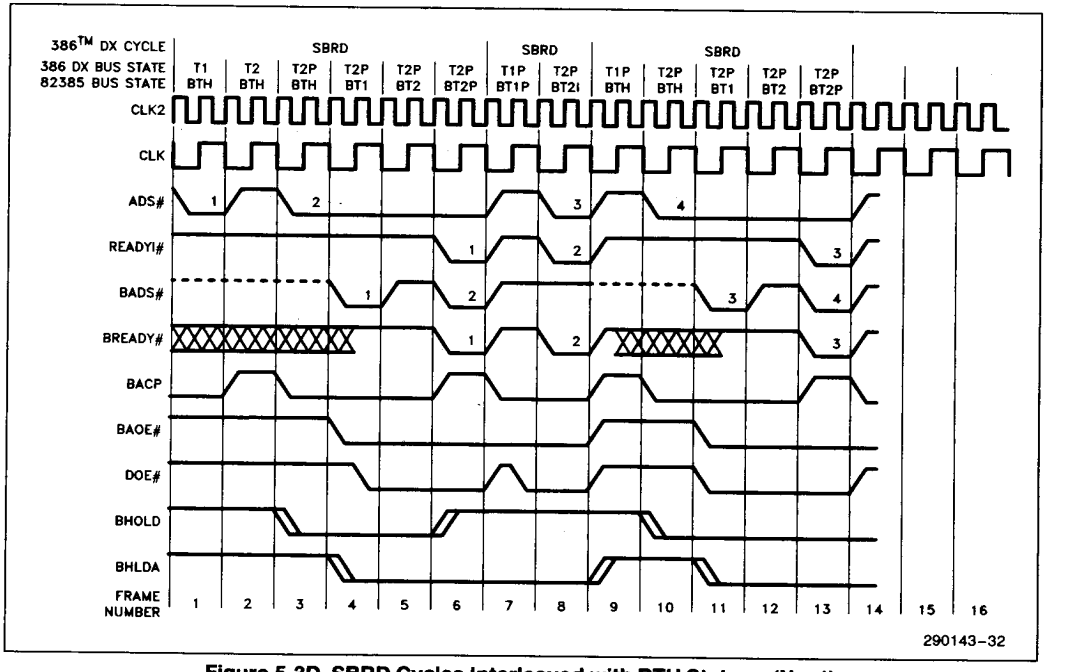


Figure 5-3D. SBRD Cycles Interleaved with BTH States—(N = 1)

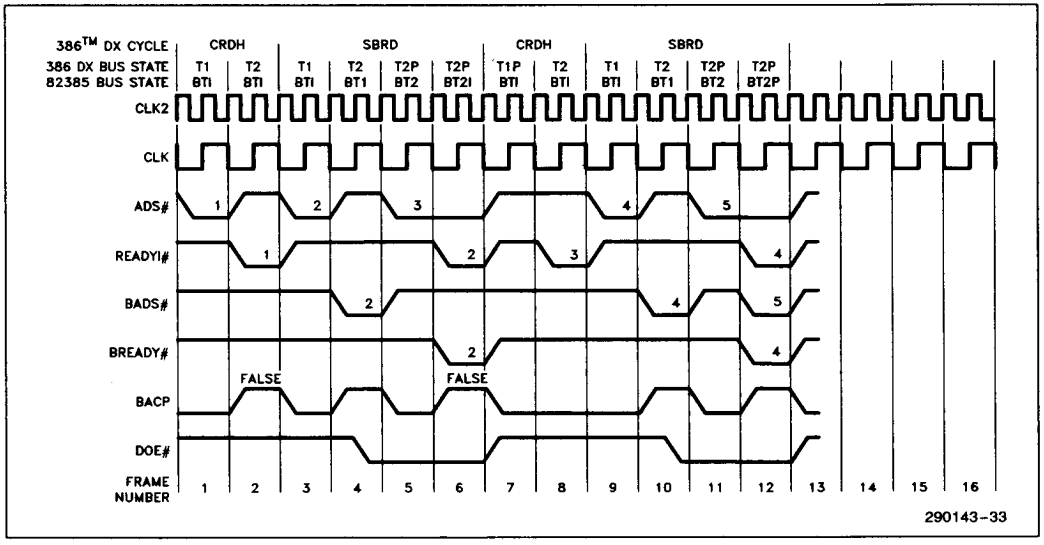


Figure 5-3E. Interleaved SBRD/CRDH Cycles—(N = 1)

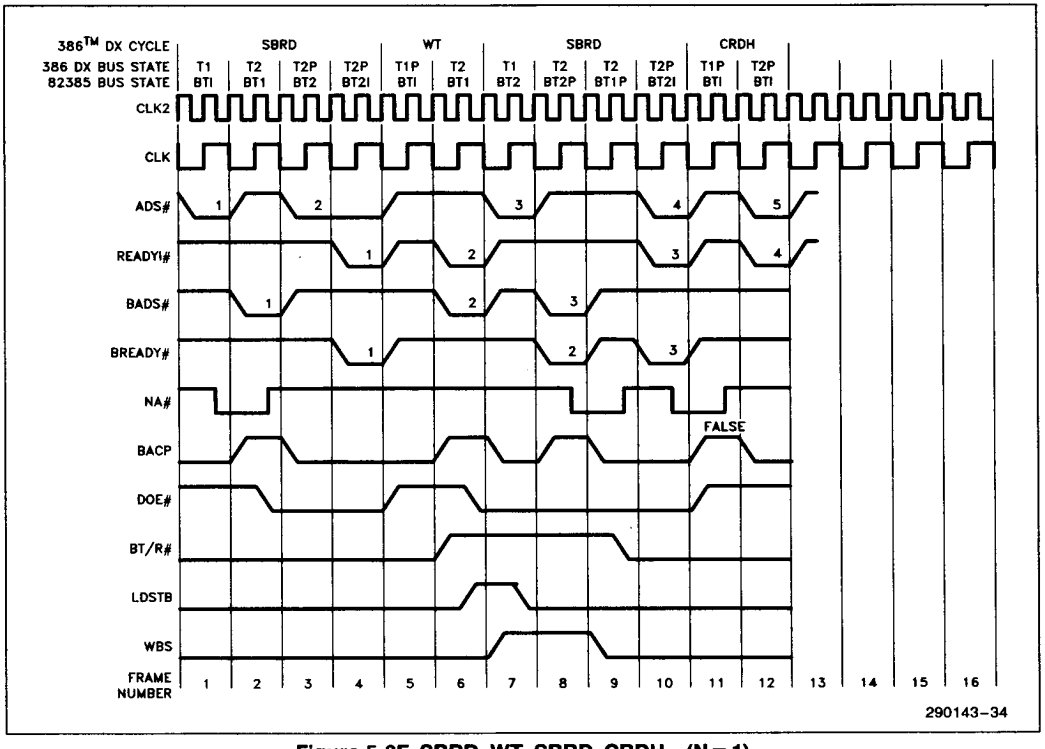


Figure 5-3F. SBRD, WT, SBRD, CRDH—(N = 1)

5

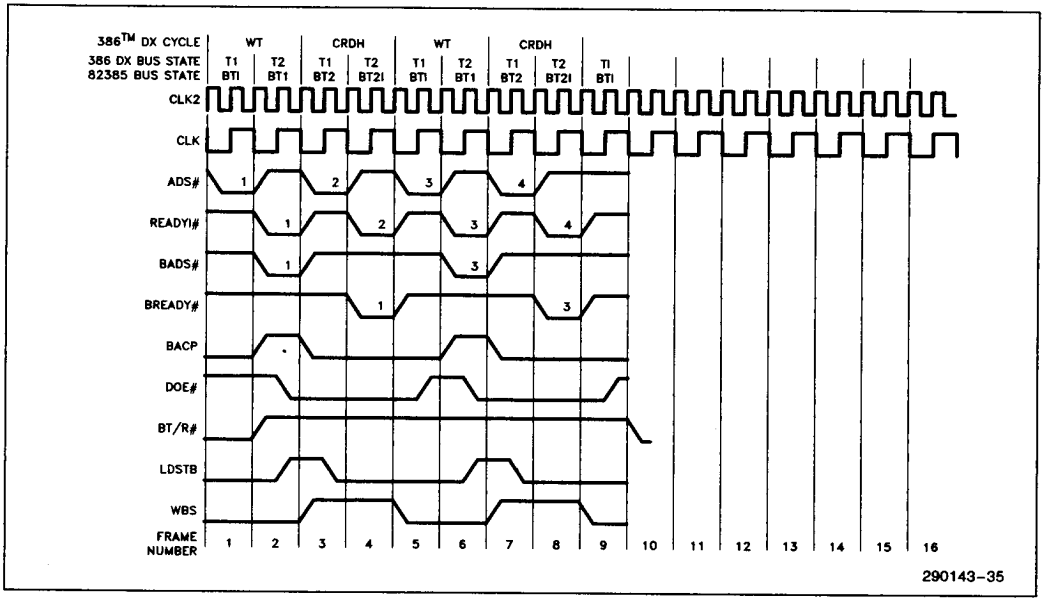


Figure 5-3G. Interleaved WT/CRDH Cycles—(N = 1)

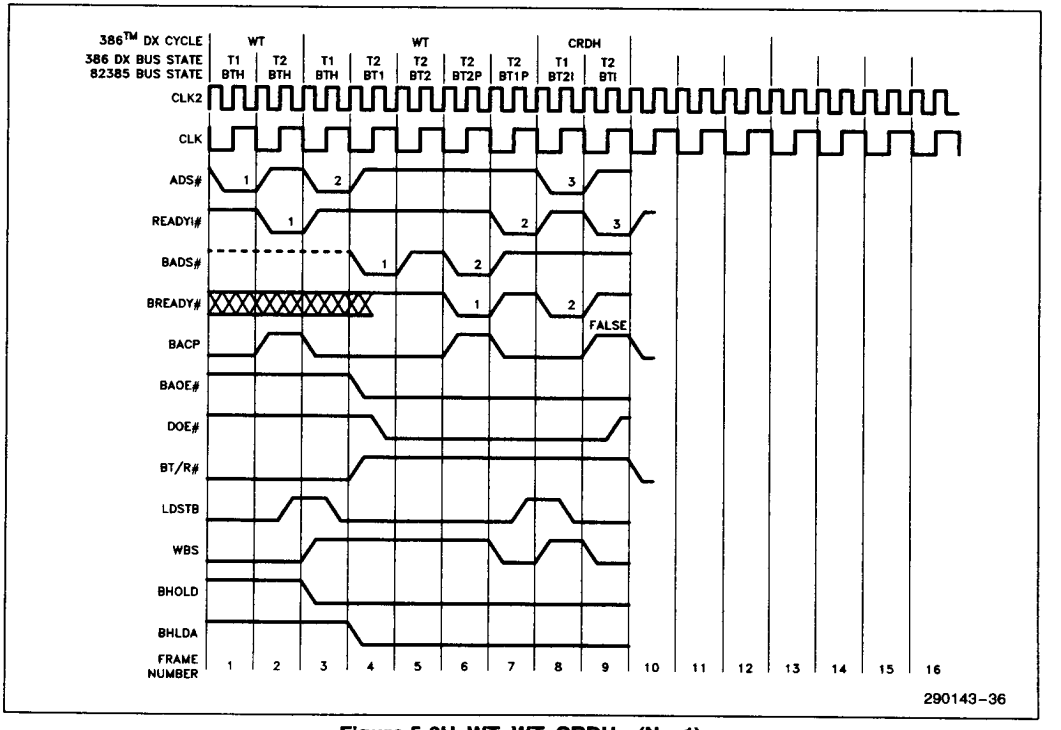


Figure 5-3H. WT, WT, CRDH—(N = 1)

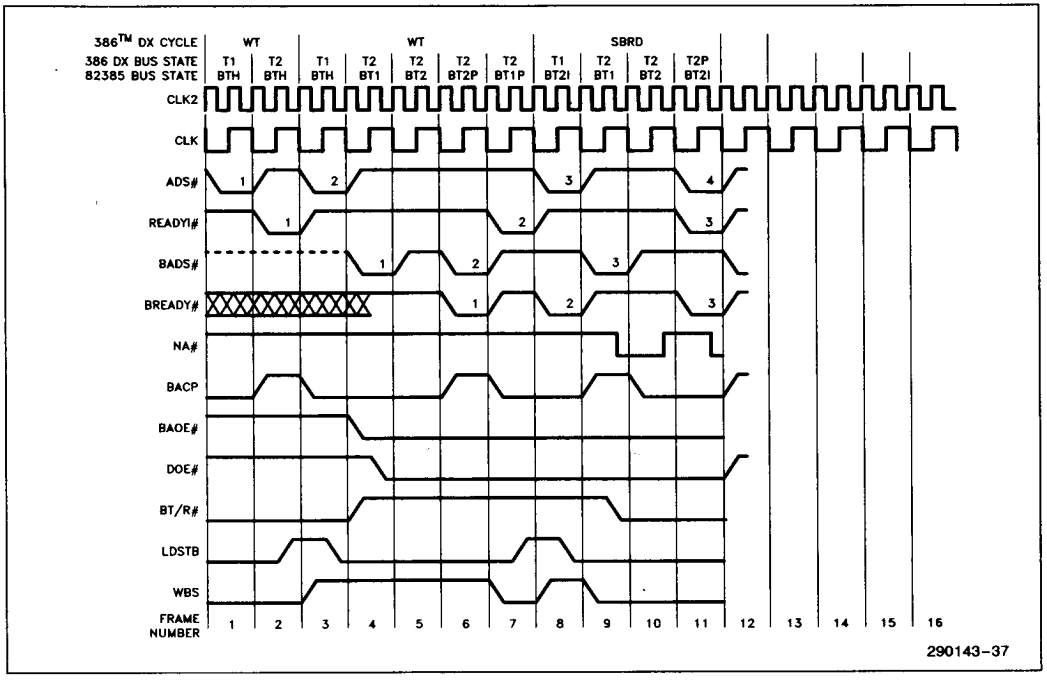


Figure 5-3I. WT, WT, SBRD—(N = 1)

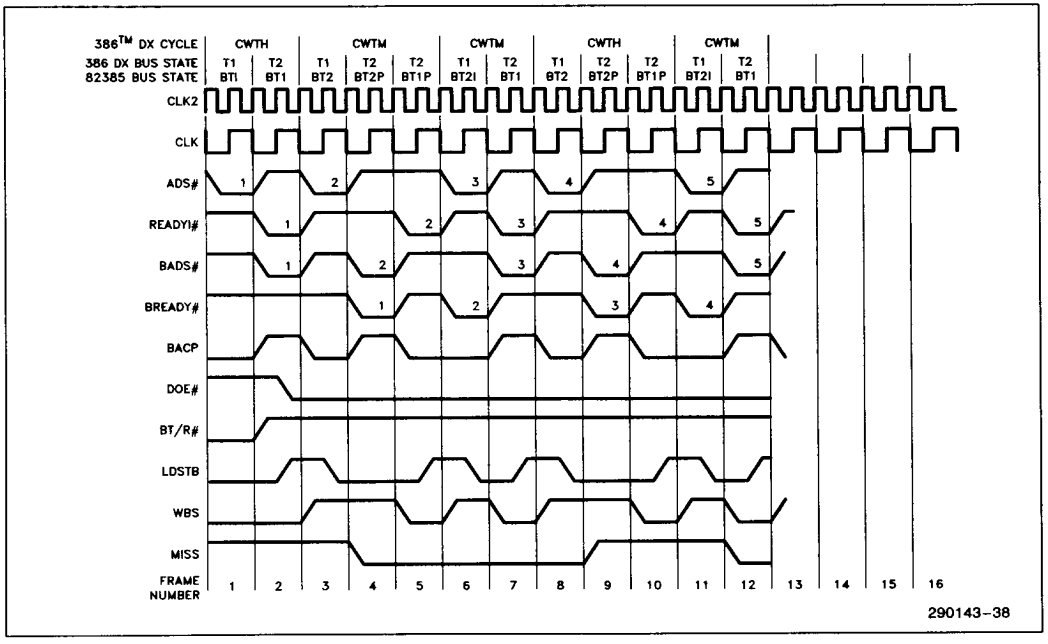


Figure 5-3J. Consecutive Write Cycles—(N = 1)

5

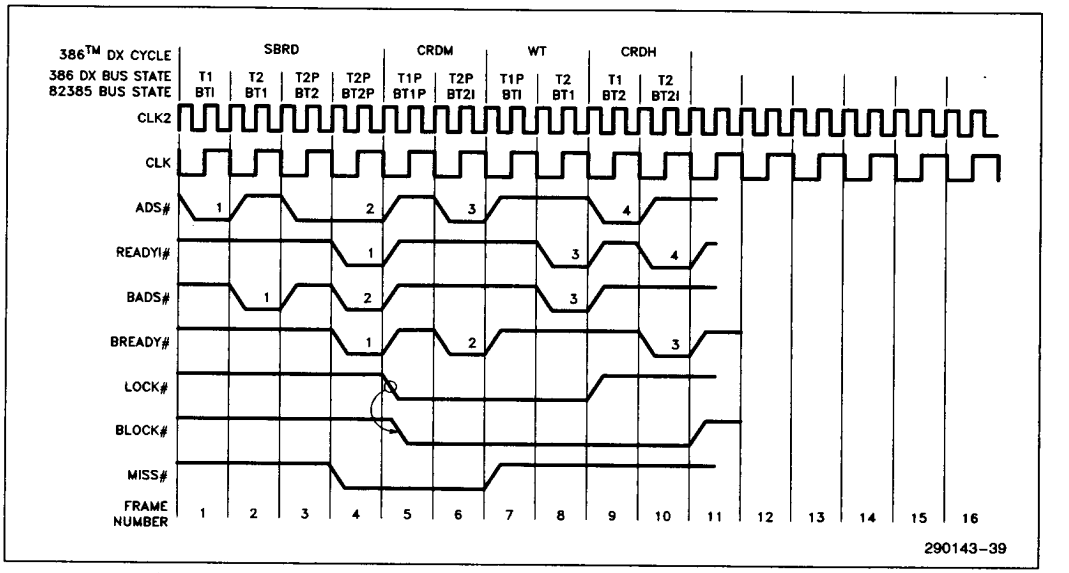


Figure 5-3K. LOCK#/BLOCK# in Non-Cacheable or Miss Cycles—(N = 1)

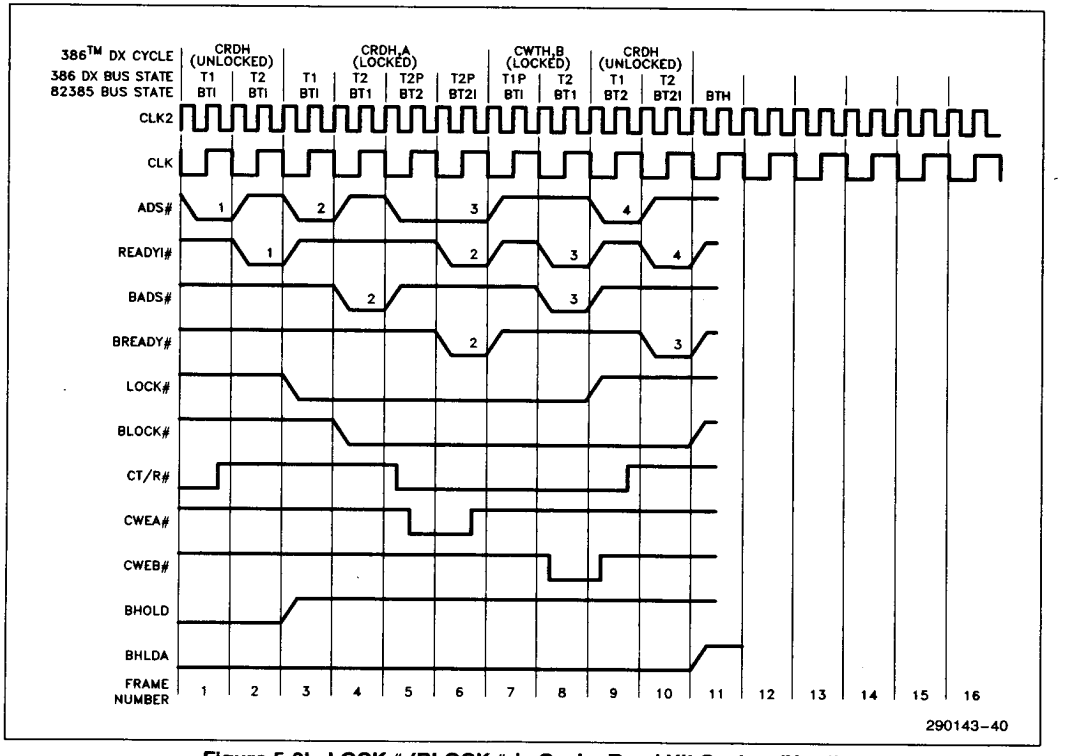


Figure 5-3L. LOCK#/BLOCK# in Cache Read Hit Cycle—(N = 1)

5.2.2.3 CACHE FLUSH (FLUSH)

FLUSH is an 82385 input which is used to reset all tag valid bits within the cache directory. The FLUSH input must be kept active for at least 4 CLK (8 CLK2) periods to complete the directory flush. Flush is generally used in diagnostics but can also be used in applications where snooping cannot guarantee coherency.

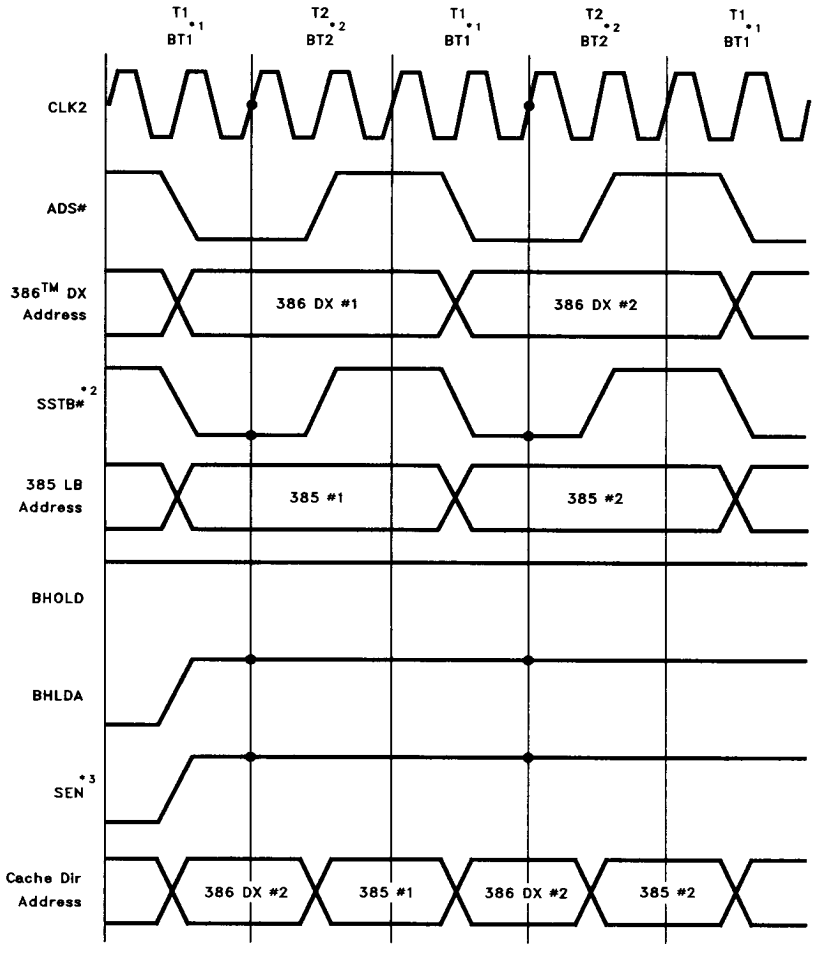
5.3 BUS WATCHING (SNOOP) INTERFACE

The 82385's bus watching interface consists of the snoop address (SA2–SA31), snoop strobe (SSTB#), and snoop enable (SEN) inputs. If masters reside at the system bus level, then the SA2–SA31 inputs are connected to the system address lines and SEN the system bus memory write command. SSTB# indicates that a valid address is present on the system bus. Note that the snoop bus inputs are synchronous, so care must be taken to ensure that they are stable during their sample windows. If no master resides beyond the 82385 bus level, then the 82385 inputs SA2–SA31, SEN, and SSTB# can respectively tie directly to BA2–BA31, BW/R#, and BADS# of the other system bus master (see Figure 5.5). However, it is recommended that SEN be driven by the logical "AND" of BW/R# and BM/IO# so as to prevent I/O writes from unnecessarily invalidating cache data.

When the 82385 detects a system write by another master and the conditions in Figure 5.4 are met: CLK2 PHI1 rising (CLK falling), BHLDA asserted, SEN asserted, SSTB# asserted, it internally latches SA2–SA31 and runs a cache look-up to see if the altered main memory location is duplicated in the cache. If yes (a snoop hit), the line valid bit associated with that cache entry is cleared. An important feature of the 82385 is that even if the 386 DX is running zero wait state hits out of the cache, all snoops are serviced. This is accomplished by time multiplexing the cache directory between the 386 DX address and the latched system address. If the SSTB# signal occurs during an 82385 comparison cycle (for the 386 DX), the 386 DX cycle has the highest priority in accessing the cache directory. This takes the first of the two 386 DX states. The other state is then used for the snoop comparison. This worst case example, depicted in Figure 5-4, shows the 386 DX running zero wait state hits on the 386 DX local bus, and another master running zero wait state writes on the 82385 bus. No snoops are missed, and no performance penalty incurred.

5.4 RESET DEFINITION

Table 5-1 summarizes the states of all 82385 outputs during reset and initialization. A slave mode 82385 tri-states its "386 DX-like" front end. A master mode 82385 emits a pulse stream on its BACP output. As the 386 DX address and cycle definition lines reach their reset values, this stream will latch the reset values through to the 82385 bus.



NOTES:
*1. These states are induced by another System Bus master.
*2. SSTB# on the 82385 is tied directly to BADS# of the System Bus master.
*3. SEN on the 82385 is tied directly to BW/R# of the System Bus master.

Figure 5.4. Interleaved Snoop and 386 DX Accesses to the Cache Directory

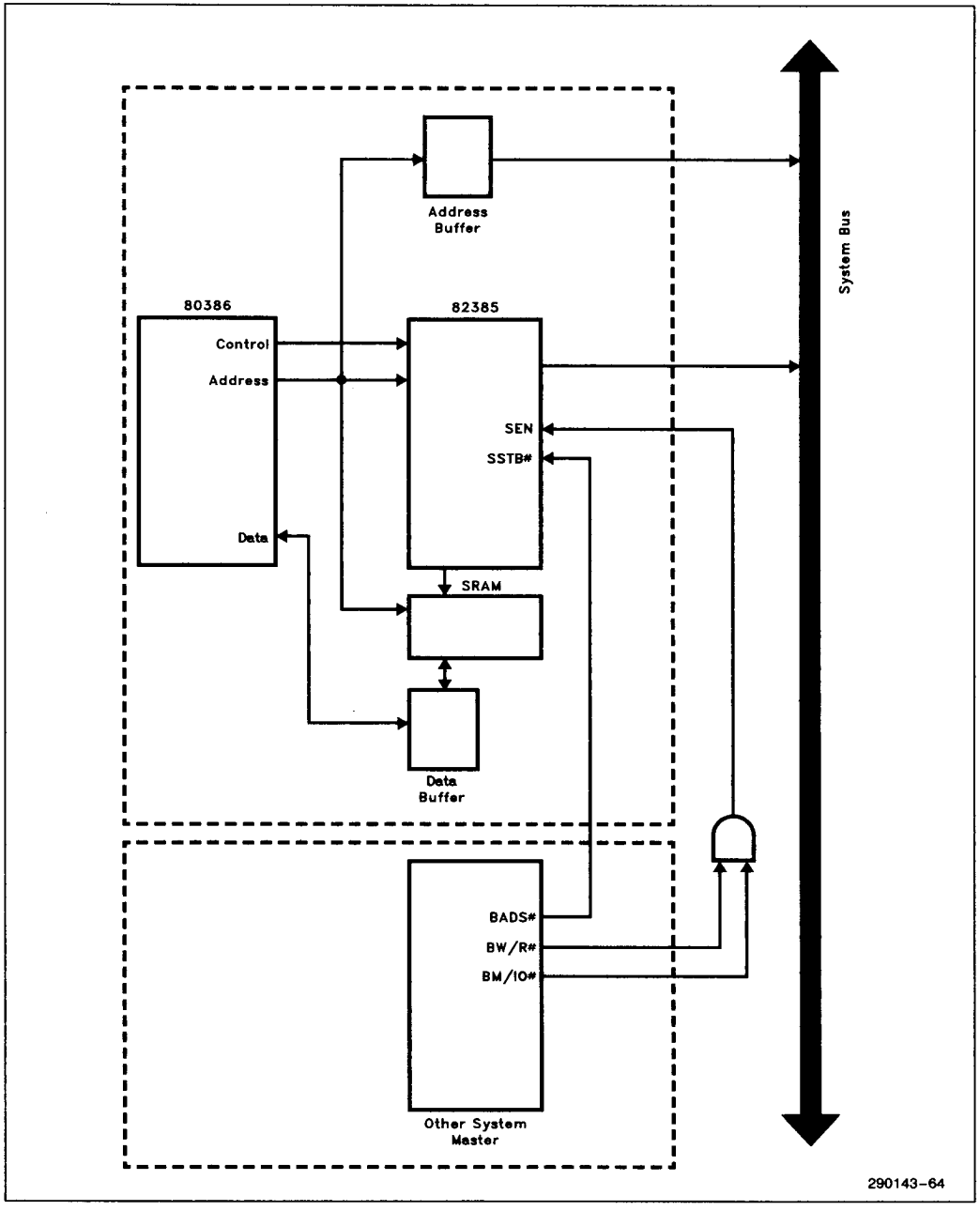


Figure 5.5. Snooping Connections in a Multi Master Environment

Table 5-1. Pin State During RESET and Initialization

Output Name	Signal Level During RESET and Initialization	
	Master Mode	Slave Mode
NA#	High	High
READY0#	High	High
BRDYEN#	High	High
CALEN	High	High
CWEA# – CWEB#	High	High
CS0# – CS3#	Low	Low
CT/R#	High	High
COEA# – COEB#	High	High
BADS#	High	High Z
BBE0# – BBE3#	386 DX BE#	High Z
BLOCK#	High	High Z
MISS#	High	High Z
BACP	Pulse(1)	Pulse
BAOE#	Low	High
BT/R#	Low	Low
DOE#	High	High
LDSTB	Low	Low
BHOLD	—	Low
BHLDA	Low	—
WBS	Low	Low

NOTE:
1. In Master Mode, BAOE# is low and BACP emits a pulse stream during reset. As the 386 DX address and cycle definition signals reach their reset values, the pulse stream on BACP will latch these values through to the 82385 local bus.

6.0 82385 SYSTEM DESIGN CONSIDERATIONS

6.1 INTRODUCTION

This chapter discusses techniques which should be implemented in an 82385 system. Because of the high frequencies and high performance nature of the 386 DX CPU/82385 system, good design and layout techniques are necessary. It is always recommended to perform a complete design analysis on new system designs.

6.2 POWER AND GROUNDING

6.2.1 Power Connections

The PGA 82385 utilizes 8 power (V_{CC}) and 10 ground (V_{SS}) pins. The PQFP 82385 has 9 power and 9 ground pins. All V_{CC} and V_{SS} pins must be connected to their appropriate plane. On a printed circuit board, all V_{CC} pins must be connected to the power plane and all V_{SS} pins must be connected to the ground plane.

6.2.2 Power Decoupling

Although the 82385 itself is generally a "passive" device in that it has few output signals, the cache

subsystem as a whole is quite active. Therefore, many decoupling capacitors should be placed around the 82385 cache subsystem.

Low inductance capacitors and interconnects are recommended for best high frequency electrical performance. Inductance can be reduced by shortening circuit board traces between the decoupling capacitors and their respective devices as much as possible. Capacitors specifically for PGA packages are also commercially available, for the lowest possible inductance.

6.2.3 Resistor Recommendations

Because of the dual bus structure of the 82385 subsystem (386 DX Local Bus and 82385 Local Bus), any signals which are recommended to be pulled up will be respective to one of the busses. The following sections will discuss signals for both busses.

6.2.3.1 386 DX LOCAL BUS

For typical designs, the pullup resistors shown in Table 6-1 are recommended. This table correlates to Chapter 7 of the 386 DX Data Sheet. However, particular designs may have a need to differ from the listed values. Design analysis is recommended to determine specific requirements.

6.2.3.2 82385 LOCAL BUS

Pullup resistor recommendations for the 82385 Local Bus signals are shown in Table 6-2. Design analysis is necessary to determine if deviations to the typical values given is needed.

Table 6-1. Recommended Resistor Pullups to V_{CC} (386 DX Local Bus)

Pin and Signal	Pullup Value	Purpose
ADS# PGA E13 PQFP 123	20 KΩ ±10%	Lightly Pull ADS# Negated for 386 DX Hold States
LOCK# PGA F13 PQFP 118	20 KΩ ±10%	Lightly Pull LOCK# Negated for 386 DX Hold States

Table 6-2. Recommended Resistor Pullups to V_{CC} (82385 Local Bus)

Signal and Pin	Pullup Value	Purpose
BADS# PGA N9 PQFP 89	20 KΩ ±10%	Lightly Pull BADS# Negated for 82385 Hold States
BLOCK# PGA P9 PQFP 86	20 KΩ ±10%	Lightly Pull BLOCK# Negated for 82385 Hold States
MISS# PGA N8 PQFP 85	20 KΩ ±10%	Lightly Pull MISS# Negated for 82385 Hold States

6.3 82385 SIGNAL CONNECTIONS

6.3.1 Configuration Inputs

The 82385 configuration signals (M/S#, 2W/D#, DEFOE#) must be connected (pulled up) to the appropriate logic level for the system design. There is also a reserved 82385 input which must be tied to the appropriate level. Refer to Table 6-3 for the signals and their required logic level.

Table 6-3. 82385 Configuration Inputs Logic Levels

Pin and Signal	Logic Level	Purpose
M/S# PGA B13 PQFP 129	High	Master Mode Operation
	Low	Slave Mode Operation
2W/D# PGA D12 PQFP 127	High	2-Way Set Associative
	Low	Direct Mapped
Resrved PGA L14 PQFP 102	High	Must be tied to V_{CC} via a pull-up for proper functionality
DEFOE# PGA A14 PQFP 128	N/A	Define Cache Output Enables. Allows use of any SRAM.

NOTE:
The listed 82385 pins which need to be tied high should use a pull-up resistor in the range of 5 KΩ to 20 KΩ.

6.3.2 CLK2 and RESET

The 82385 has two inputs to which the 386 DX CLK2 signal must be connected. One is labeled CLK2 (82385 PGA pin C13, PQFP lead 126) and the other is labeled BCLK2 (82385 PGA pin L13, PQFP lead 103). These two inputs must be tied together on the printed circuit board.

The 82385 also has two reset inputs. RESET (82385 PGA pin D13, PQFP lead 125) and BRESET (82385 PGA pin K12, PQFP lead 104) must be connected on the printed circuit board.

6.4 UNUSED PIN REQUIREMENTS

For reliable operation, ALWAYS connect unused inputs to a valid logic level. As is the case with most other CMOS processes, a floating input will increase the current consumption of the component and give an indeterminate state to the component.

6.5 CACHE SRAM REQUIREMENTS

The 82385 offers the option of using SRAMs with or without an output enable pin. This is possible by inserting a transceiver between the SRAMs and the 386 DX local data bus and strapping DEFOE# to the appropriate logic level for a given system configuration. This transceiver may also be desirable in a system which has a very heavily loaded 386 DX local data bus. The following sections discuss the SRAM requirements for all cache configurations.

6.5.1 Cache Memory without Transceivers

As discussed in Section 3.2, the 82385 presents all of the control signals necessary to access the cache memory. The SRAM chip selects, write enables, and output enables are driven directly by the 82385. Table 6-4 lists the required SRAM specifications. These specifications allow for zero margins. They should be used as guides for the actual system design.

6.5.2 Cache Memory With Transceivers

To implement an 82385 subsystem using cache memory transceivers, COEA# or COEB# must be used as output enable signals for the transceivers and DEFOE# must be appropriately strapped for proper COEx# functionality (since the cache SRAM transceivers must be enabled for writes as well as reads). DEFOE# must be tied high when using cache SRAM transceivers. In a 2-way set associative organization, COEA# enables the transceiver for bank A and COEB# enables the bank B transceiver. A direct mapped cache may use either COEA# or COEB# to enable the transceiver. Table 6-5 lists the required SRAM specifications. These specifications allow for zero margin. They should be used as guides for the actual system design.

Table 6-4. SRAM Specs for Non-Buffered Cache Memory

SRAM Spec Requirements						
	Direct Mapped			2-Way Set Associative		
	20	25	33	20	25	33
Read Cycle Requirements						
Address Access (MAX)	44	36	27	42	34	27
Chip Select Access (MAX)	56	44	35	56	41	35
OE# to Data Valid (MAX)	19	13	10	14	13	10
OE# to Data Float (MAX)	20	15	10	20	15	10
Write Cycle Requirements						
Chip Select to End of Write (MIN)	30	25	20	30	25	20
Address Valid to End of Write (MIN)	42	37	29	40	37	29
Write Pulse Width (MIN)	30	25	20	30	25	20
Data Setup (MAX)	—	—	—	—	—	—
Data Hold (MIN)	4	4	2	4	4	2

Table 6-5. SRAM Specs for Buffered Cache Memory

SRAM Spec Requirements						
	Direct Mapped			2-Way Set Associative		
	20	25	33	20	25	33
Read Cycle Requirements						
Address Access (MAX)	37	29	20	35	29	20
Chip Select Access (MAX)	48	36	27	48	36	27
OE# to Data Valid (MAX)	N/A	N/A	N/A	N/A	N/A	N/A
OE# to Data Float (MAX)	N/A	N/A	N/A	N/A	N/A	N/A
Write Cycle Requirements						
Chip Select to End of Write (MIN)	30	25	20	30	23	20
Address Valid to End of Write (MIN)	42	37	29	40	36	27
Write Pulse Width (MIN)	30	25	20	30	25	20
Data Setup (MAX)	15	10	10	15	10	10
Data Hold (MIN)	3	3	3	3	3	3

7.0 SYSTEM TEST CONSIDERATIONS

7.1 INTRODUCTION

Power On Self Testing (POST) is performed by most systems after a reset. This chapter discusses the requirements for properly testing an 82385 based system after power up.

7.2 MAIN MEMORY (DRAM) TESTING

Most systems perform a memory test by writing a data pattern and then reading and comparing the data. This test may also be used to determine the total available memory within the system. Without properly taking into account the 82385 cache memory, the memory test can give erroneous results. This will occur if the cache responds with read hits during the memory test routine.

7.2.1 Memory Testing Routine

In order to properly test main memory, the test routine must not read from the same block consecutively. For instance, if the test routine writes a data pattern to the first 32 kbytes of memory (0000–7FFFH), reads from the same block, writes a new pattern to the same locations (0000–7FFFH), and reads the new pattern, the second pattern tested would have had data returned from the 82385 cache memory. Therefore, it is recommended that the test routine work with a memory block of at least 64 kbytes. This will guarantee that no 32 kbyte block will be read twice consecutively.

7.3 82385 CACHE MEMORY TESTING

With the addition of SRAMs for the cache memory, it may be desirable for the system to be able to test the cache SRAMs during system diagnostics. This requires the test routine to access only the cache memory. The requirements for this routine are based on where it resides within the memory map. This can be broken into two areas: the routine residing in cacheable memory space or the routine residing in either non-cacheable memory or on the 386 DX local bus (using the LBA# input).

7.3.1 Test Routine in the NCA# or LBA# Memory Map

In this configuration, the test routine will never be cached. The recommended method is code which will access a single 32 kbyte block during the test. Initially, a 32 kbyte read (assume 0000–7FFFH) must be executed. This will fill the cache directory with the address information which will be used in the diagnostic procedure. Then, a 32 kbyte write to the same address locations (0000–7FFFH) will load the cache with the desired test pattern (due to write hits). The comparison can be made by completing another 32 kbyte read (same locations, 0000–7FFFH), which will be cache read hits. Subsequent writes and reads to the same addresses will enable various patterns to be tested.

7.3.2 Test Routine in Cacheable Memory

In this case, it must be understood that the diagnostic routine must reside in the cache memory before the actual data testing can begin. Otherwise, when the 386 DX performs a code fetch, a location within the cache memory which is to be tested will be altered due to the read miss (code fetch) update.

The first task is to load the diagnostic routine into the top of the cache memory. It must be known how much memory is required for the code as the rest of the cache memory will be tested as in the earlier method. Once the diagnostics have been cached (via read updates), the code will perform the same type of read/write/read/compare as in the routine explained in the above section. The difference is that now the amount of cache memory to be tested is 32 kbytes minus the length of the test routine.

7.4 82385 CACHE DIRECTORY TESTING

Since the 82385 does not directly access the data bus, it is not possible to easily complete a comparison of the cache directory. (The 82385 can serially transmit its directory contents. See Section 7.5.) However, the cache memory tests described in Section 7.3 will indicate if the directory is working properly. Otherwise, the data comparison within the diagnostics will show locations which fail.

There is a slight possibility that the cache memory comparison could pass even if locations within the directory gave false hit/miss results. This could cause the comparison to always be performed to main memory instead of the cache and give a proper comparison to the 386 DX . The solution here is to use the MISS# output of the 82385 as an indicator to a diagnostic port which can be read by the 386 DX . It could also be used to flag an interrupt if a failure occurs.

The implementation of these techniques in the diagnostics will assure proper functionality of the 82385 subsystem.

7.5 SPECIAL FUNCTION PINS

As mentioned in Chapter 3, there are three 82385 pins which have reserved functions in addition to their normal operational functions. These pins are MISS#, WBS, and FLUSH.

As discussed previously, the 82385 performs a directory flush when the FLUSH input is held active for at least 4 CLK (8 CLK2) cycles. However, the FLUSH pin also serves as a diagnostic input to the 82385. The 82385 will enter a reserved mode if the FLUSH pin is high at the falling edge of RESET.

If, during normal operation, the FLUSH input is active for only one CLK (2 CLK2) cycle/s, the 82385 will enter another reserved mode. Therefore it must be guaranteed that FLUSH is active for at least the 4 CLK (8 CLK2) cycle specification.

WBS and MISS# serve as outputs in the 82385 reserved modes.

8.0 MECHANICAL DATA

8.1 INTRODUCTION

This chapter discusses the physical package and its connections in detail.

8.2 PIN ASSIGNMENT

The 82385 pinout as viewed from the top side of the component is shown by Figure 8-1. Its pinout as viewed from the Pin side of the component is shown in Figure 8-2.

V_{CC} and V_{SS} connections must be made to multiple V_{CC} and V_{SS} (GND) pins. Each V_{CC} and V_{SS} must be connected to the appropriate voltage level. The circuit board should include V_{CC} and GND planes for power distribution and all V_{CC} and V_{SS} pins must be connected to the appropriate plane.

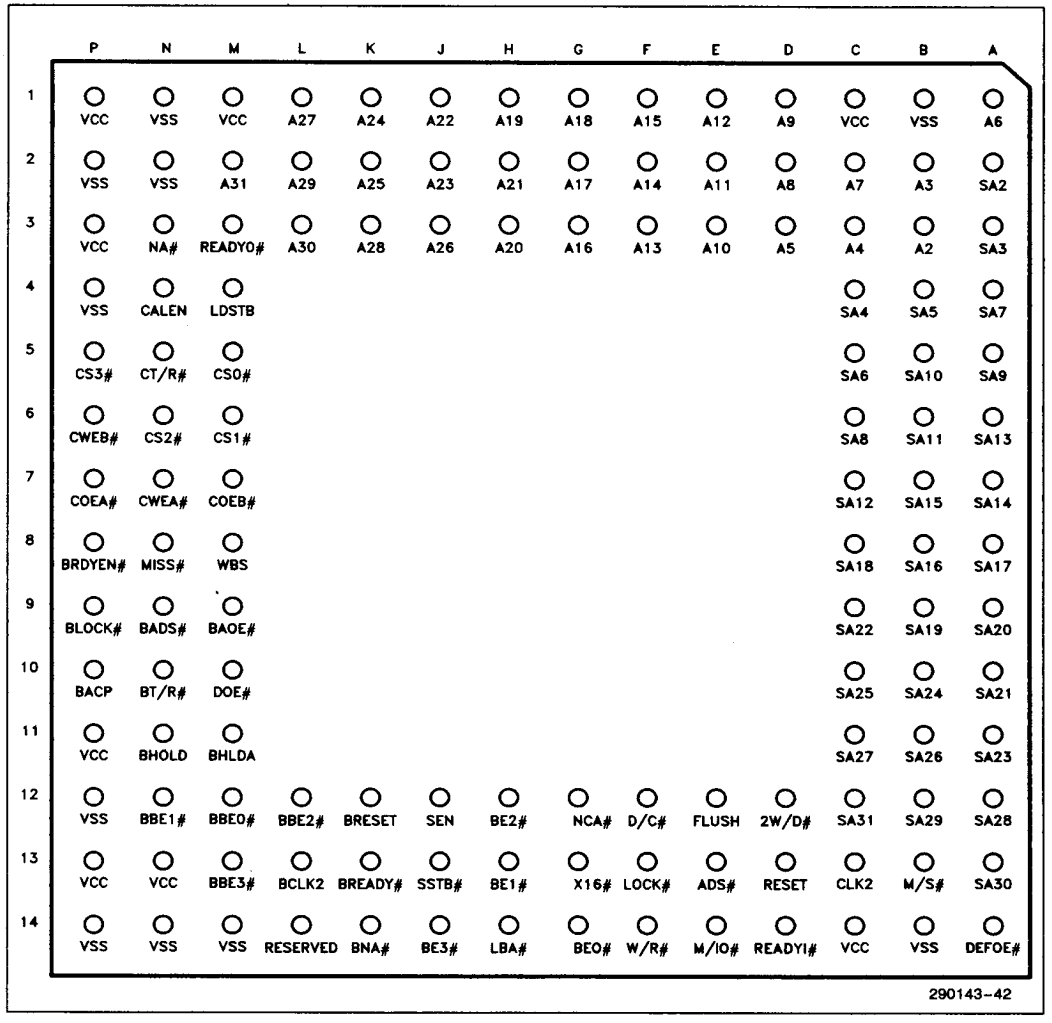


Figure 8-1. 82385 PGA Pinout—View from TOP Side

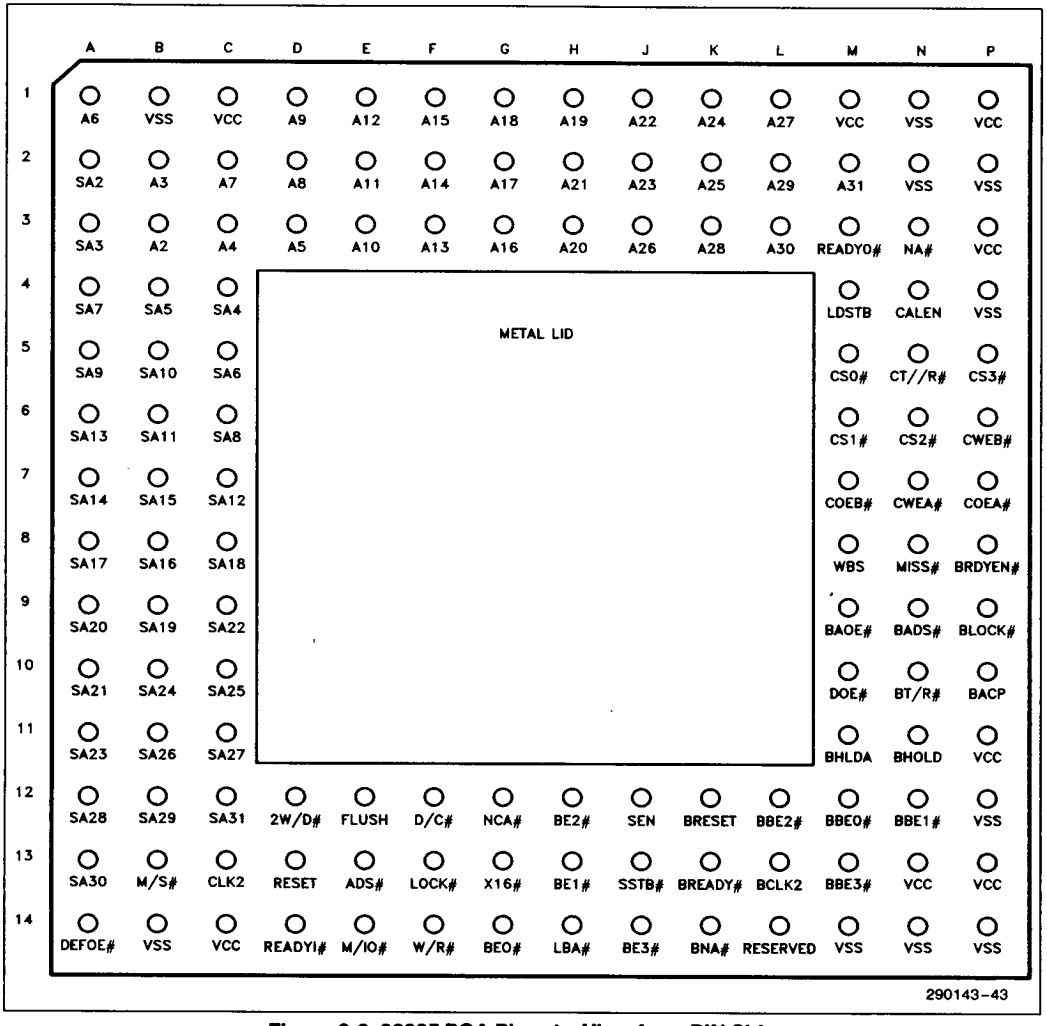


Figure 8-2. 82385 PGA Pinout—View from PIN Side

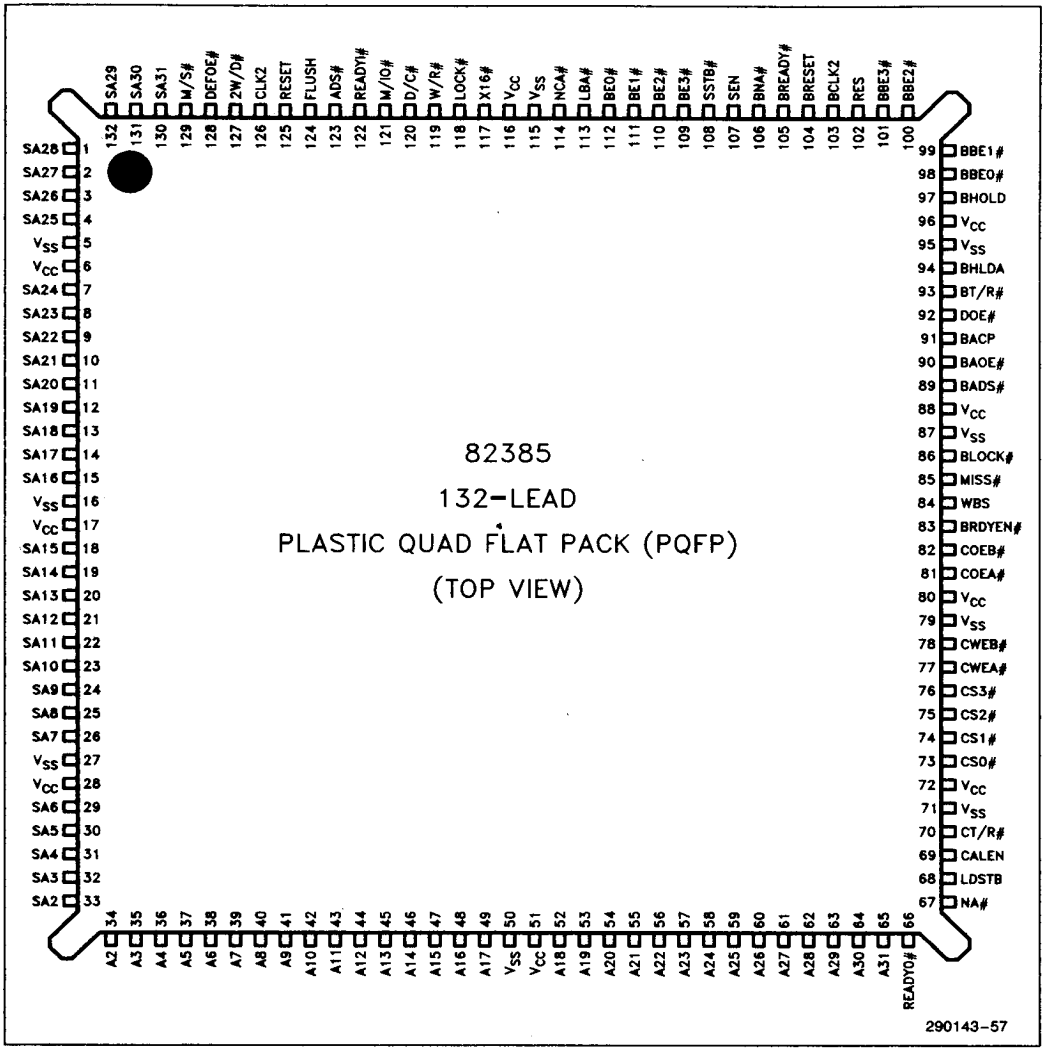


Figure 8-3. 82385 PQFP Pinout—View from TOP Side

Table 8-1. 82385 Pinout—Functional Grouping

PGA	PQFP	Signal
M2	65	A31
L3	64	A30
L2	63	A29
K3	62	A28
L1	61	A27
J3	60	A26
K2	59	A25
K1	58	A24
J2	57	A23
J1	56	A22
H2	55	A21
H3	54	A20
H1	53	A19
G1	52	A18
G2	49	A17
G3	48	A16
F1	47	A15
F2	46	A14
F3	45	A13
E1	44	A12
E2	43	A11
E3	42	A10
D1	41	A9
D2	40	A8
C2	39	A7
A1	38	A6
D3	37	A5
C3	36	A4
B2	35	A3
B3	34	A2
G14	112	BE0#
H13	111	BE1#
H12	110	BE2#
J14	109	BE3#
C13	126	CLK2
D13	125	RESET
K12	104	BRESET
L13	103	BCLK2

PGA	PQFP	Signal
C12	130	SA31
A13	131	SA30
B12	132	SA29
A12	1	SA28
C11	2	SA27
B11	3	SA26
C10	4	SA25
B10	7	SA24
A11	8	SA23
C9	9	SA22
A10	10	SA21
A9	11	SA20
B9	12	SA19
C8	13	SA18
A8	14	SA17
B8	15	SA16
B7	18	SA15
A7	19	SA14
A6	20	SA13
C7	21	SA12
B6	22	SA11
B5	23	SA10
A5	24	SA9
C6	25	SA8
A4	26	SA7
C5	29	SA6
B4	30	SA5
C4	31	SA4
A3	32	SA3
A2	33	SA2
J12	107	SEN
J13	108	SSTB#
L14	102	RESERVED

PGA	PQFP	Signal
—	116	V_{CC}
C1	6	V_{CC}
C14	17	V_{CC}
M1	28	V_{CC}
N13	51	V_{CC}
P1	72	V_{CC}
P3	80	V_{CC}
P11	88	V_{CC}
P13	96	V_{CC}
E13	123	ADS#
F14	119	W/R#
F12	120	D/C#
E14	121	M/IO#
F13	118	LOCK#
N3	67	NA#
G13	117	X16#
G12	114	NCA#
H14	113	LBA#
D14	122	READYI#
M3	66	READYO#
E12	124	FLUSH
M8	84	WBS
N8	85	MISS#
A14	128	DEFOE#
D12	127	2W/D#
B13	129	M/S#
M10	92	DOE#
M4	68	LDSTB
N11	97	BHOLD
M11	94	BHLDA

PGA	PQFP	Signal
B1	5	V_{SS}
B14	16	V_{SS}
M14	27	V_{SS}
N1	50	V_{SS}
N2	71	V_{SS}
N14	79	V_{SS}
P2	87	V_{SS}
P4	95	V_{SS}
P12	115	V_{SS}
P14	—	V_{SS}
N9	89	BADS#
M12	98	BBE0#
N12	99	BBE1#
L12	100	BBE2#
M13	101	BBE3#
P9	86	BLOCK#
K14	106	BNA#
N4	69	CALEN
P7	81	COEA#
M7	82	COEB#
N7	77	CWEA#
P6	78	CWEB#
M5	73	CS0#
M6	74	CS1#
N6	75	CS2#
P5	76	CS3#
N5	70	CT/R#
P8	83	BRDYEN#
K13	105	BREADY#
P10	91	BACP
M9	90	BAOE#
N10	93	BT/R#

8.3 PACKAGE DIMENSIONS AND MOUNTING

The 82385 package is a 132-pin ceramic Pin Grid Array (PGA). The pins are arranged 0.100 inch (2.5 mm) center-to-center, in a 14 x 14 matrix, three rows around (Figure 8-3).

A wide variety of available sockets allow low insertion force or zero insertion force mounting. These come in a choice of terminals such as soldertail, surface mount, or wire wrap.

8.4 PACKAGE THERMAL SPECIFICATION

The PGA case temperature should be measured at the center of the top surface opposite the pins, as in Figure 8-4. The case temperature may be measured in any environment to determine whether or not the 82385 is within the specified operating range.

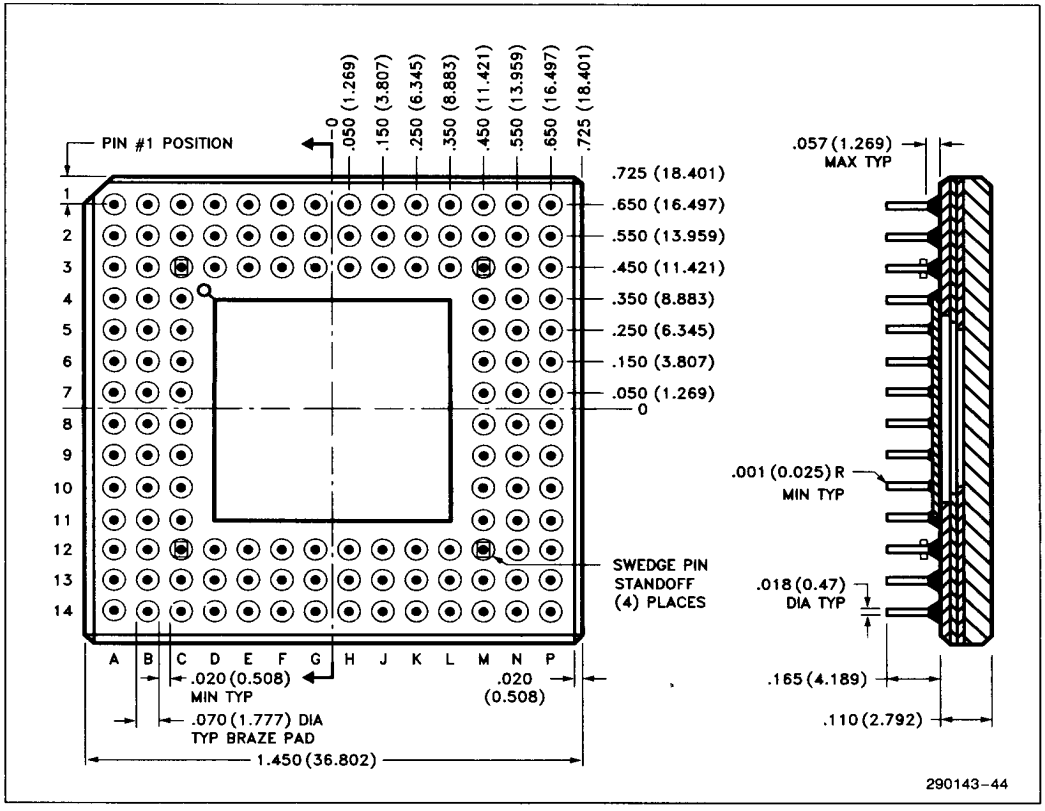


Figure 8-3.1. 132-Pin PGA Package Dimensions

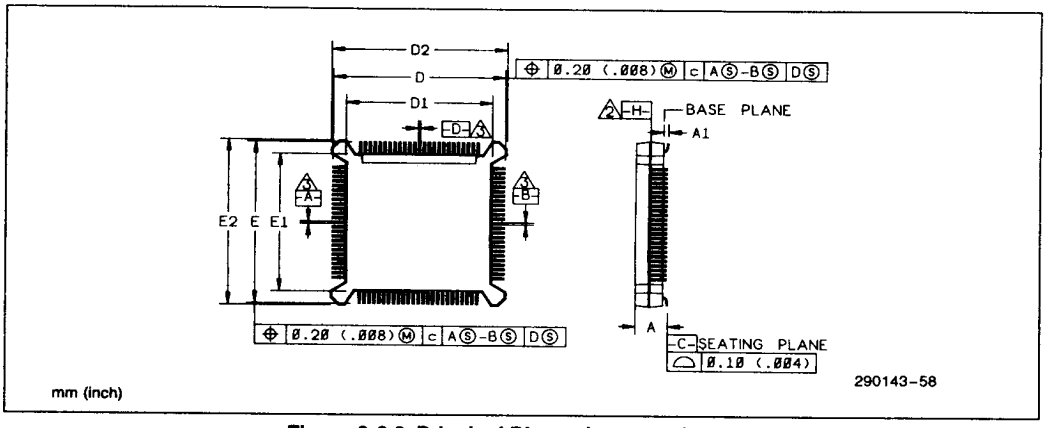


Figure 8-3.2. Principal Dimensions and Datums

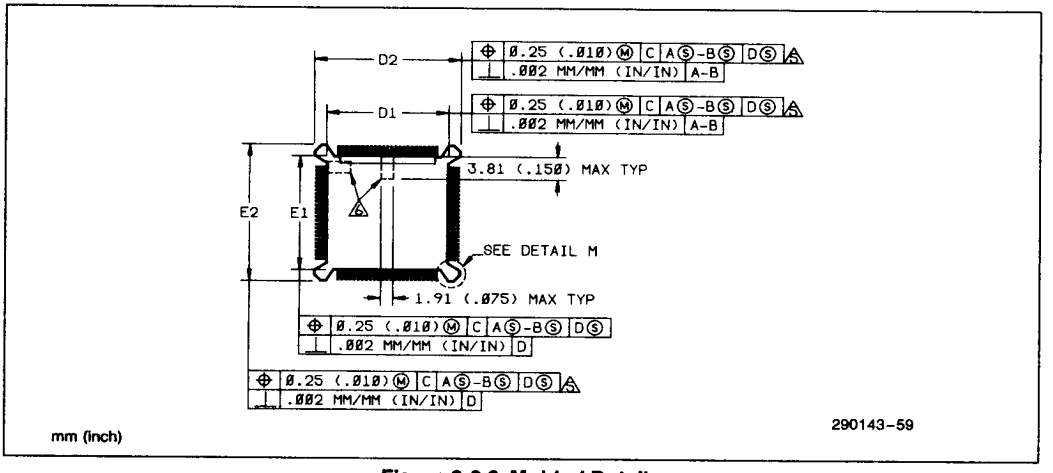


Figure 8-3.3. Molded Details

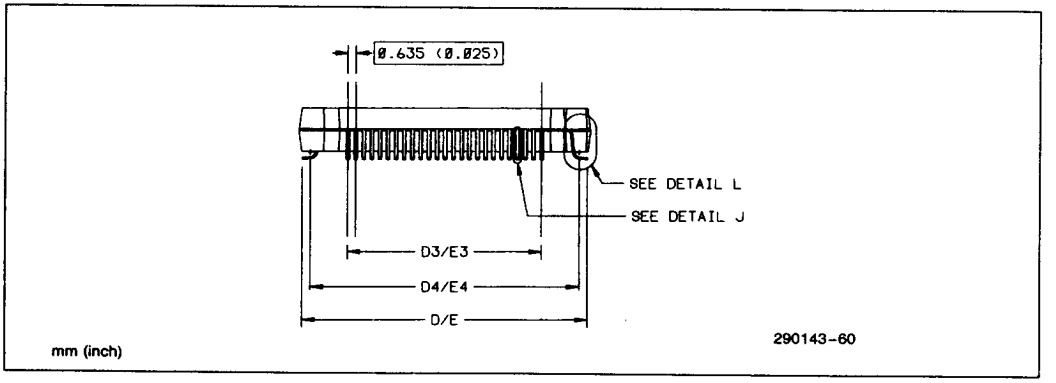


Figure 8-3.4. Terminal Details

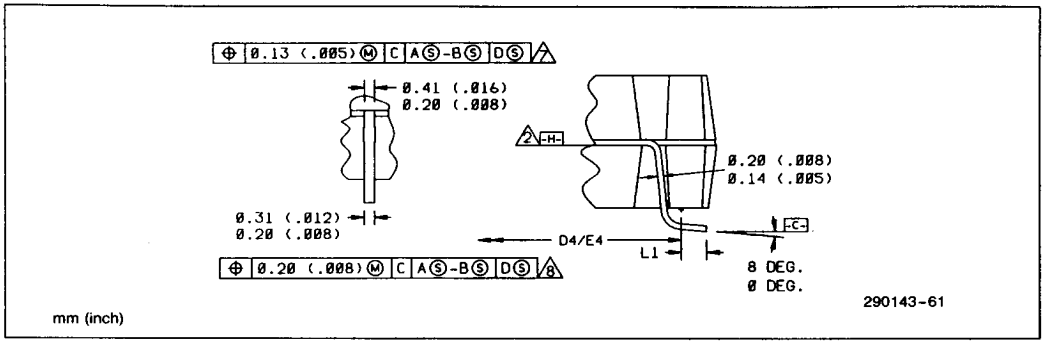


Figure 8-3.5. Typical Lead

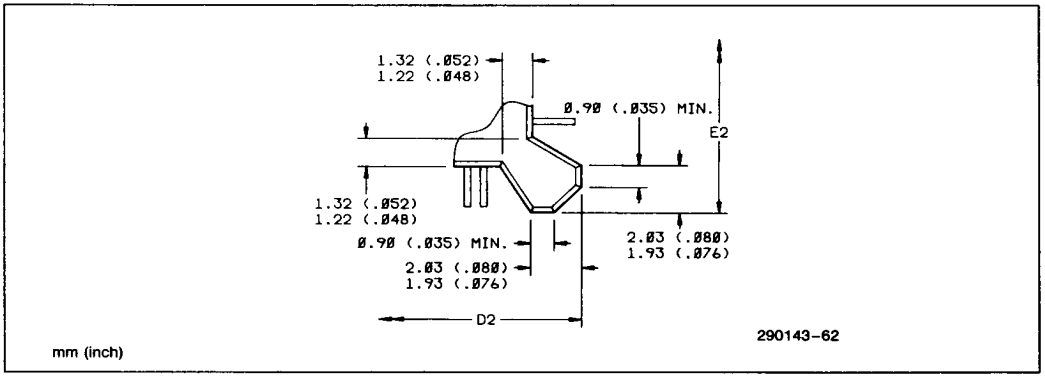


Figure 8-3.6. Detail M

PLASTIC QUAD FLAT PACK

Table 8-2. Symbol List for Plastic Quad Flat Pack

Letter or Symbol	Description of Dimensions
A	Package height: distance from seating plane to highest point of body
A1	Standoff: Distance from seating plane to base plane
D/E	Overall package dimension: lead tip to lead tip
D1/E1	Plastic body dimension
D2/E2	Bumper Distance
D3/E3	Footprint
L1	Foot length
N	Total number of leads

NOTES:
1. All dimensions and tolerances conform to ANSI Y14.5M-1982.
2. Datum plane -H- located at the mold parting line and coincident with the bottom of the lead where lead exits plastic body.
3. Datums A–B and -D- to be determined where center leads exit plastic body at datum plane -H-.
4. Controlling Dimension, Inch.
5. Dimensions D1, D2, E1 and E2 are measured at the mold parting line and do not include mode protrusion. Allowable mold protrusion of 0.18mm (0.007 in.) per side.
6. Pin 1 identifier is located within one of the two zones indicated.
7. Measured at datum plane -H-.
8. Measured at seating plane datum -C-.

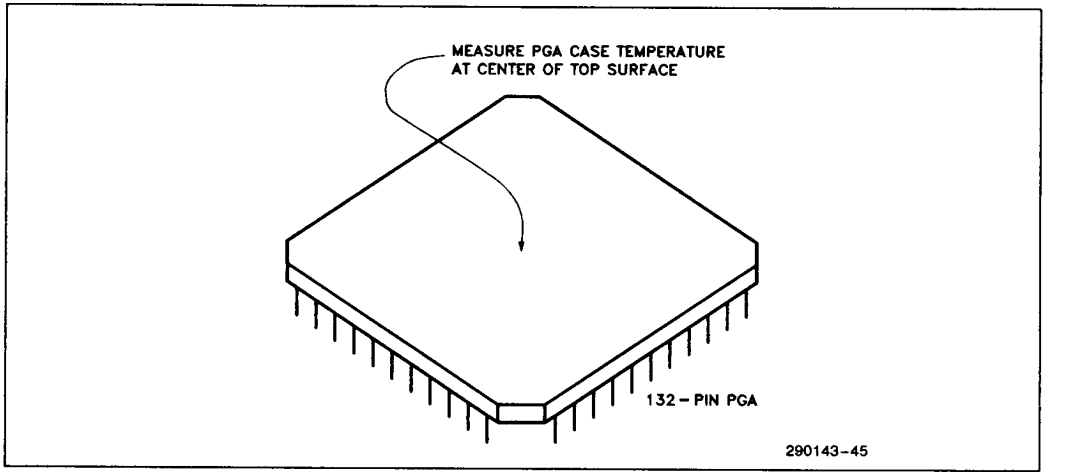


Figure 8-4. Measuring 82385 PGA Case Temperature

Table 8-3. 82385 PGA Package Typical Thermal Characteristics.

Parameter	Thermal Resistance—°C/Watt						
	Airflow—ft³/min (m³/sec)						
	0 (0)	50 (0.25)	100 (0.50)	200 (1.01)	400 (2.03)	600 (3.04)	800 (4.06)
θ Junction-to-Case (Case Measured as Figure 8.4)	2	2	2	2	2	2	2
θ Case-to-Ambient (No Heatsink)	19	18	17	15	12	10	9
θ Case-to-Ambient (with Omnidirectional Heatsink)	16	15	14	12	9	7	6
θ Case-to-Ambient (with Unidirectional Heatsink)	15	14	13	11	8	6	5

NOTES:
1. Table 8-3 applies to 82385 PGA plugged into socket or soldered directly onto board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$.
3. $\theta_{J\text{-}CAP} = 4°C/W$ (approx.)
   $\theta_{J\text{-}PIN} = 4°C/W$ (inner pins) (approx.)
   $\theta_{J\text{-}PIN} = 8°C/W$ (outer pins) (approx.)

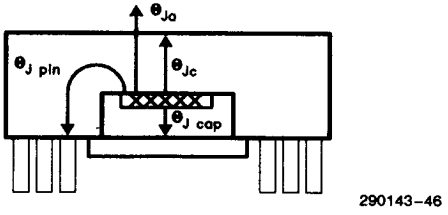


Table 8-4. 82385 132-Lead PQFP Package Typical Thermal Characteristics

Parameter	Thermal Resistance—°C/Watt						
	Airflow—lfm						
	0	50	100	200	400	600	800
θ Junction-to-Case (Case Measured as Figure 8.4)	5	5	5	5	5	5	5
θ Case-to-Ambient (No Heatsink)	23.5	22.0	20.5	17.5	14.0	11.5	9.5
θ Case-to-Ambient (with Omnidirectional Heatsink)	TO BE DEFINED						
θ Case-to-Ambient (with Unidirectional Heatsink)							

NOTES:
1. Table 8-4 applies to 82385 PQFP plugged into socket or soldered directly onto board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$.
3. $\theta_{J\text{-}CAP} = 4°C/W$ (approx.)
   $\theta_{J\text{-}PIN} = 4°C/W$ (inner pins) (approx.)
   $\theta_{J\text{-}PIN} = 8°C/W$ (outer pins) (approx.)

5

9.0 ELECTRICAL DATA

9.1 INTRODUCTION

This chapter presents the A.C. and D.C. specifications for the 82385.

9.2 MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Case Temperature under Bias ... −65°C to +110°C
Supply Voltage with Respect to V_{SS} ..... −0.5V to +6.5V
Voltage on Any Other Pin .... −0.5V to V_{CC} + 0.5V

NOTE:
Stress above those listed may cause permanent damage to the device. This is a stress rating only and functional operation at these or any other conditions above those listed in the operational sections of this specification is not implied.

Exposure to absolute maximum rating conditions for extended periods may affect device reliability. Although the 82385 contains protective circuitry to resist damage from static electrical discharges, always take precautions against high static voltages or electric fields.

9.3 D.C. SPECIFICATIONS V_{CC} = 5V ±5%; V_{SS} = 0V

Table 9-1. D.C. Specifications

Symbol	Parameter	Min	Max	Unit	Test Condition
V_{IL}	Input Low Voltage	−0.3	0.8	V	(Note 1)
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.3	V	
V_{CL}	CLK2, BCLK2 Input Low	−0.3	0.8	V	(Note 1)
V_{CH}	CLK2, BCLK2 Input High	3.7	V_{CC} + 0.3	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 4 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −1 mA
I_{CC}	Supply Current		300	mA	(Note 2) (Note 4)
I_{LI}	Input Leakage Current		±15	μA	$0V < V_{IN} \leq V_{CC}$
I_{LO}	Output Leakage Current		±15	μA	$0.45 < V_{OUT} < V_{CC}$
C_{IN}	Input Capacitance		10	pF	(Note 3)
C_{OUT}	Output Capacitance		10	pF	(Note 3)
C_{CLK}	CLK2 Input Capacitance		15	pF	(Note 3)

NOTES:
1. Minimum value is not 100% tested.
2. I_{CC} is specified with inputs driven to CMOS levels. I_{CC} may be higher if driven to TTL levels.
3. Not 100% tested. Test conditions f_C = 1 MHz, Inputs = 0V, T_{CASE} = Room.
4. 300 mA is the maximum I_{CC} at 33 MHz.
   275 mA is the maximum I_{CC} at 25 MHz.
   250 mA is the maximum I_{CC} at 20 MHz.

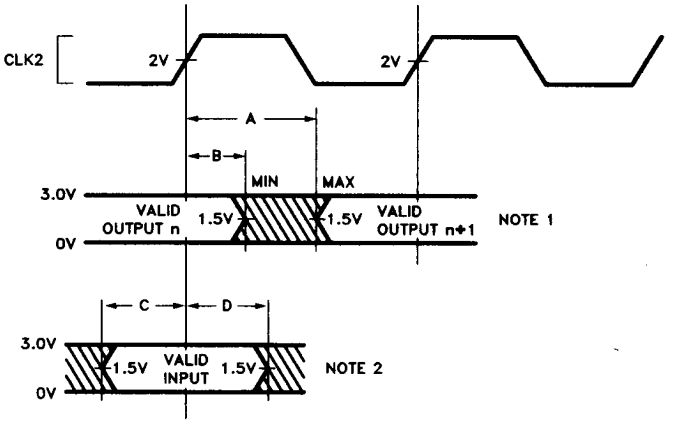
9.4 A.C. SPECIFICATIONS

The A.C. specifications given in the following tables consist of output delays and input setup requirements. The A.C. diagram's purpose is to illustrate the clock edges from which the timing parameters are measured. The reader should not infer any other timing relationships from them. For specific information on timing relationships between signals, refer to the appropriate functional section.

A.C. spec measurement is defined in Figure 9-1. Inputs must be driven to the levels shown when A.C. specifications are measured. 82385 output delays are specified with minimum and maximum limits, which are measured as shown. 82385 input setup and hold times are specified as minimums and define the smallest acceptable sampling window. Within the sampling window, a synchronous input signal must be stable for correct 82385 operation.

9.4.1 Frequency Dependent Signals

The 82385 has signals whose output valid delays are dependent on the clock frequency. These signals are marked in the A.C. Specification Tables with a Note 1.



LEGEND:
A—Maximum output delay specification
B—Minimum output delay specification
C—Minimum input setup specification
B—Minimum input hold specification

NOTES:
1. Under rated loading 82385 output (t_r and t_f) is typically ≤ 4.0 ns from 0.8V to 2.0V.
2. Input waveforms have t_r ≤ 2.0 ns from 0.8V to 2.0V.

Figure 9-1. Drive Levels and Measurement Points for A.C. Specification

5

A.C. SPECIFICATION TABLES

Many of the A.C. Timing parameters are frequency dependent. The frequency dependent A.C. Timing parameters are guaranteed only at the maximum specified operating frequency.

Table 9-2. 82385 A.C. Timing Specifications

V_{CC} = 5.0 ±5%

Symbol	Parameter	20 MHz		25 MHz		33 MHz		Units	Notes
		Min	Max	Min	Max	Min	Max		
T_{CASE}	Case Temperature	0	85	0	75	0	75	°C	
t1	Operating Frequency	15.40	20.00	15.40	25.00	15.40	33.33	MHz	
t2	CLK2, BCLK2 Clock Period	25.00	32.50	20.00	32.50	15.00	32.50	ns	
t3a	CLK2, BCLK2 High Time @ 2.0V	10		8		6.25		ns	
t3b	CLK2, BCLK2 High Time @ 3.7V	7		5		4.5		ns	(Note 8)
t4a	CLK2, BCLK2 Low Time @ 2.0V	10		8		6.25		ns	
t4b	CLK2, BCLK2 Low Time @ 0.8V	8		6		4.5		ns	(Note 8)
t5	CLK2, BCLK2 Fall Time		8		7		4	ns	(Notes 8, 9)
t6	CLK2, BCLK2 Rise Time		8		7		4	ns	(Notes 8, 9)
t7a	A2–A19, A21–A31 Setup Time	19		18		13		ns	(Note 1)
t7b	LOCK# Setup Time	16		14		9.5		ns	(Note 1)
t7c	BE(0–3)# Setup Time	19		14		10		ns	(Note 1)
t7d	A20 Setup Time	13		13		9		ns	(Note 1)
t8	A2–A31, BE(0–3)# LOCK# Hold Time	3		3		3		ns	
t9a	M/IO#, D/C# Setup Time	22		17		13		ns	(Note 1)
t9b	W/R# Setup Time	22		18		13		ns	(Note 1)
t9c	ADS# Setup Time	22		18		13.5		ns	(Note 1)
t10	ADS#, D/C#, M/IO#, W/R# Hold Time	5		3		3		ns	
t11	READYI# Setup Time	12		8		7		ns	(Note 1)
t12	READYI# Hold Time	4		4		3		ns	
t13a1	NCA# Setup Time (See t55b2)	21		18		13		ns	(Note 6)
t13a2	NCA# Setup Time (See t55b3)	16		13		9		ns	(Note 6)
t13b	LBA# Setup Time	10		8		5.75		ns	
t13c	X16# Setup Time	10		7		5.5		ns	
t14a	NCA# Hold Time	4		3		3		ns	
t14b	LBA#, X16# Hold Time	4		3		3		ns	
t15	RESET, BRESET Setup Time	12		10		8		ns	
t16	RESET, BRESET Hold Time	4		3		2		ns	
t17	NA# Valid Delay	15	34	4	27	4	19.2	ns	(25 pF Load) (Note 1)
t18	READYO# Valid Delay	4	28	4	22	3	15	ns	(25 pF Load) (Note 1)
t19	BRDYEN# Valid Delay	4	28	4	21	3	13	ns	

Table 9-2. 82385 A.C. Timing Specifications (Continued)
V_{CC} = 5.0 ±5%

Symbol	Parameter	20 MHz		25 MHz		33 MHz		Units	Notes
		Min	Max	Min	Max	Min	Max		
t21a1	CALEN Rising, PHI1	3	24	4	21	3	15	ns	
t21a2	CALEN Falling, PHI1	3	24	4	21	3	15	ns	
t21a3	CALEN Falling in T1P, PHI2	3	24	4	21	3	15	ns	
t21b	CALEN Rising Following CWTH Cycle	3	34	4	27	3	20	ns	(Note 1)
t21c	CALEN Pulse Width	10		10		10		ns	
t21d	CALEN Rising to CS# Falling	13		13		13		ns	
t22a1	CWEx# Falling, PHI1 (CWTH)	4	25	4	23	3	18	ns	(Note 1)
t22a2	CWEx# Falling, PHI2 (CRDM)	4	25	4	23	3	18	ns	(Note 1)
t22b	CWEx# Pulse Width	30		25		20		ns	(Notes 1, 2)
t22c1	CWEx# Rising, PHI1 (CWTH)	4	25	4	21	3	16	ns	(Note 1)
t22c2	CWEx# Rising, PHI2 (CRDM)	12	25	8	21	6	16	ns	(Note 1)
t23a	CS(0-3)# Rising	12	37	9	29	3	25	ns	(Note 1)
t23b	COEx# Falling to CS(0-3)# Falling	0		0		0		ns	(Note 1)
t24	CT/R# Valid Delay	12	38	9	30	3	22	ns	(Note 1)
t25a	COEx# Falling (Direct)	1	22	4	19.5	3	15	ns	(25 pF Load)
t25b	COEx# Falling (2-Way)	1	24.5	4	19.5	3	15	ns	(25 pF Load) (Note 1)
t25c1	COEx# Rising Delay @ T_{CASE} = Min	5	17	4	17.5	3	12	ns	(25 pF Load)
t25c2	COEx# Rising Delay @ T_{CASE} = Max	5	19	4	19.5	3	12	ns	(25 pF Load)
t25d	CWEx# Falling to COEx# Falling or CWEx# Rising to COEX# Rising when DEFOE# = V_{CC}	0	5	0	5	0	5	ns	(25 pF Load)
t26	CS(0-3)# Falling to CWEx# Rising	30		25		20		ns	(Notes 1, 2)
t27	CWEx# Falling to CS(0-3)# Falling	0		0		0		ns	
t28a	CWEx# Rising to CALEN Rising	0		0		2		ns	
t28b	CWEx# Rising to CS(0-3)# Falling	0		0		2		ns	
t31	SA(2-31) Setup Time	19		10		8		ns	
t32	SA(2-31) Hold Time	3		3		3		ns	
t33	BADS# Valid Delay	6	28	4	21	3	16	ns	(Note 1)
t34	BADS# Float Delay	6	30	4	30	4	25	ns	(Note 3)
t35	BNA# Setup Time	9		7		7		ns	
t36	BNA# Hold Time	15		4		2		ns	
t37	BREADY# Setup Time	26		18		13		ns	(Note 1)
t38	BREADY# Hold Time	4		3		2		ns	
t40a	BACP Rising Delay	4	20	4	16	2	12	ns	
t40b	BACP Falling Delay	4	22	4	20	2	18	ns	

5

Table 9-2. 82385 A.C. Timing Specifications (Continued)
V_{CC} = 5.0 ±5%

Symbol	Parameter	20 MHz		25 MHz		33 MHz		Units	Notes
		Min	Max	Min	Max	Min	Max		
t41	BAOE# Valid Delay	4	18	4	15	2	12	ns	
t43a	BT/R# Valid Delay	2	19	4	16	2	14	ns	
t43b1	DOE# Falling Delay	2	23	4	20	2	16	ns	
t43b2	DOE# Rising Delay @ T_{CASE} = Min	4	17	4	17	2	12	ns	
t43b3	DOE# Rising Delay @ T_{CASE} = Max	4	19	4	19	2	14	ns	
t43c	LDSTB Valid Delay	2	26	2	21	2	16	ns	
t44a	SEN Setup Time	11		9		7		ns	
t44b	SSTB# Setup Time	11		5		5		ns	
t45	SEN, SSTB# Hold Time	5		5		2		ns	
M/S# = V_{CC} (Master Mode)									
t_{46}	BHOLD Setup Time	17		15		11		ns	
t47	BHOLD Hold Time	5		3		2		ns	
t48	BHLDA Valid Delay	5	28	4	23	3	16	ns	
M/S# = V_{SS} (Slave Mode)									
t49	BHLDA Setup Time	17		15		11		ns	
t50	BHLDA Hold Delay	5		3		2		ns	
t51	BHOLD Valid Delay	5	28	4	23	3	18	ns	
t55a	BLOCK# Valid Delay	4	30	4	26	3	20	ns	(Notes 1,5)
t55b1	BBE(0–3)# Valid Delay	4	30	4	26	3	20	ns	(Notes 1, 7)
t55b2	BBE(0–3#) Valid Delay	4	30	4	26	3	20	ns	(Notes 1, 7)
t55b3	BBE(0–3)# Valid Delay	4	36	4	32	3	23	ns	(Notes 1, 7)
t55c	LOCK# Valid to BLOCK# Valid	0	30	0	26	0	20	ns	(Notes 1, 5)
t56	MISS# Valid Delay	4	35	4	30	3	22	ns	(Note 1)
t57	MISS#, BBE(0–3)#, BLOCK# Float Delay	4	32	4	30	4	25	ns	(Note 3)
t58	WBS Valid Delay	4	37	4	25	3	16	ns	(Note 1)
t59	FLUSH Setup Time	16		12		10		ns	
t60	FLUSH Hold Time	5		5		3		ns	
t61	FLUSH Setup to RESET Falling	26		21		16		ns	(Note 4)
t62	FLUSH Hold to RESET Falling	26		21		16		ns	(Note 4)

NOTES:
1. Frequency dependent specification.
2. Used for cache data memory (SRAM) specifications.
3. Float times not 100% tested.
4. This feature is tested only at 16 MHz.
5. BLOCK# delay is either from BPHI1 or from 386 LOCK#. Refer to Figure 5-3K and 5-3L in the 82385 data sheet.
6. NCA# setup time is now specified to the rising edge of PHI2 in the state after 386 DX addresses become valid (either the first T2 or the state after the first T2P).
7. BBE# Valid delay is a function of NCA# setup.
8. Not 100% tested.
9. t5 is measured from 0.8V to 3.7V.
   t6 is measured from 3.7V to 0.8V
   This parameter is not 100% tested and is guaranteed by Intel's test methodology.

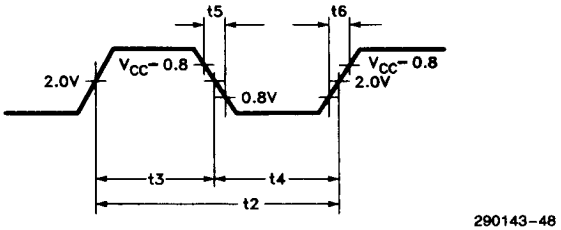
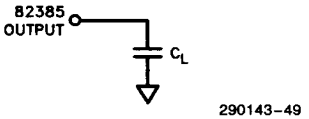


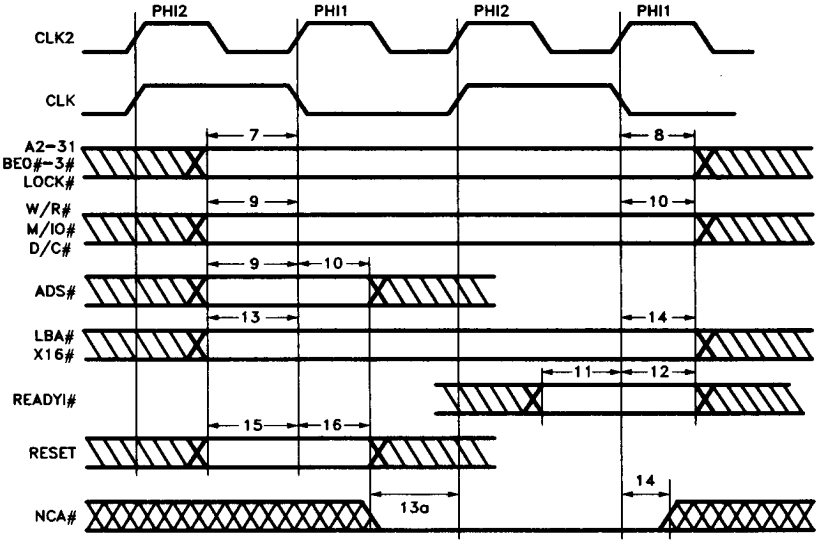
Figure 9-2. CLK2, BCLK2 Timing



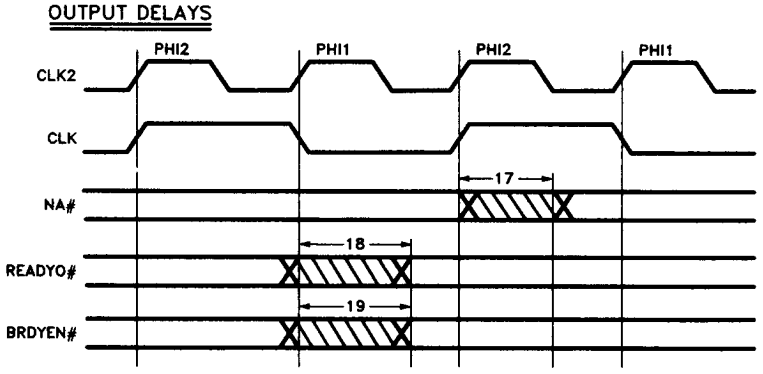
C_L indicates all parasitic capacitances.

Figure 9-3. A.C. Test Load

386 DX Interface Parameters

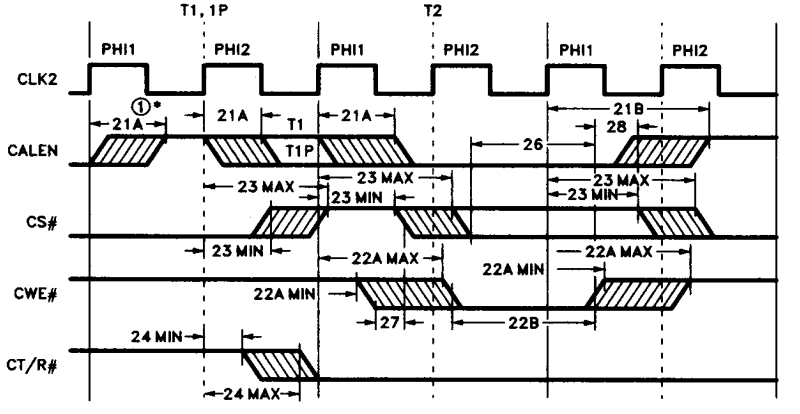


5



290143-51

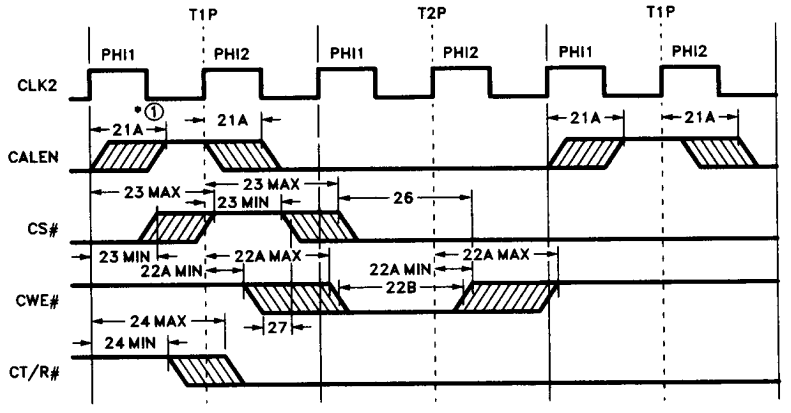
Cache Write Hit Cycle



290143-52

①*. This would be 21B if previous bus cycle was Cache Write Hit cycle.

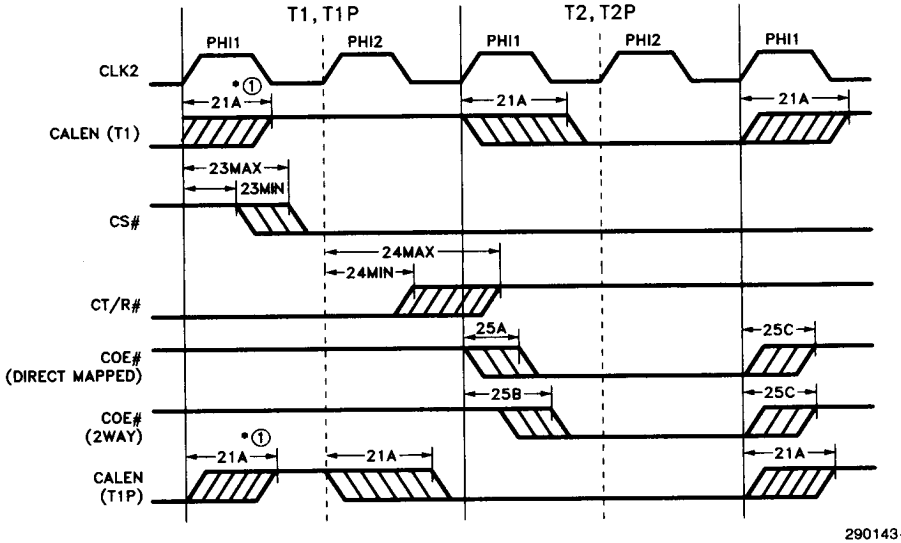
Cache Read Miss (Cache Update Cycle)



290143-53

*①. This would be 21B if previous bus cycle was Cache Write Hit cycle.

Cache Read Cycle

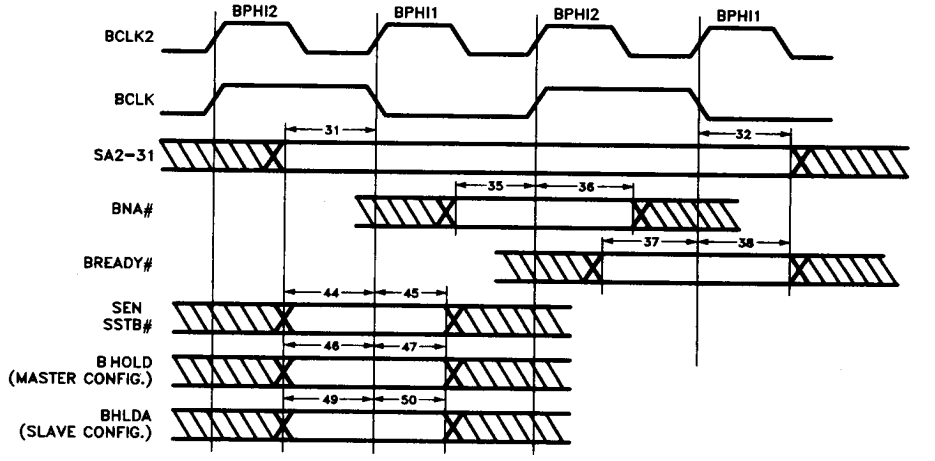


290143-54

*①. This would be 21B if previous bus cycle was Cache Write Hit cycle.

5

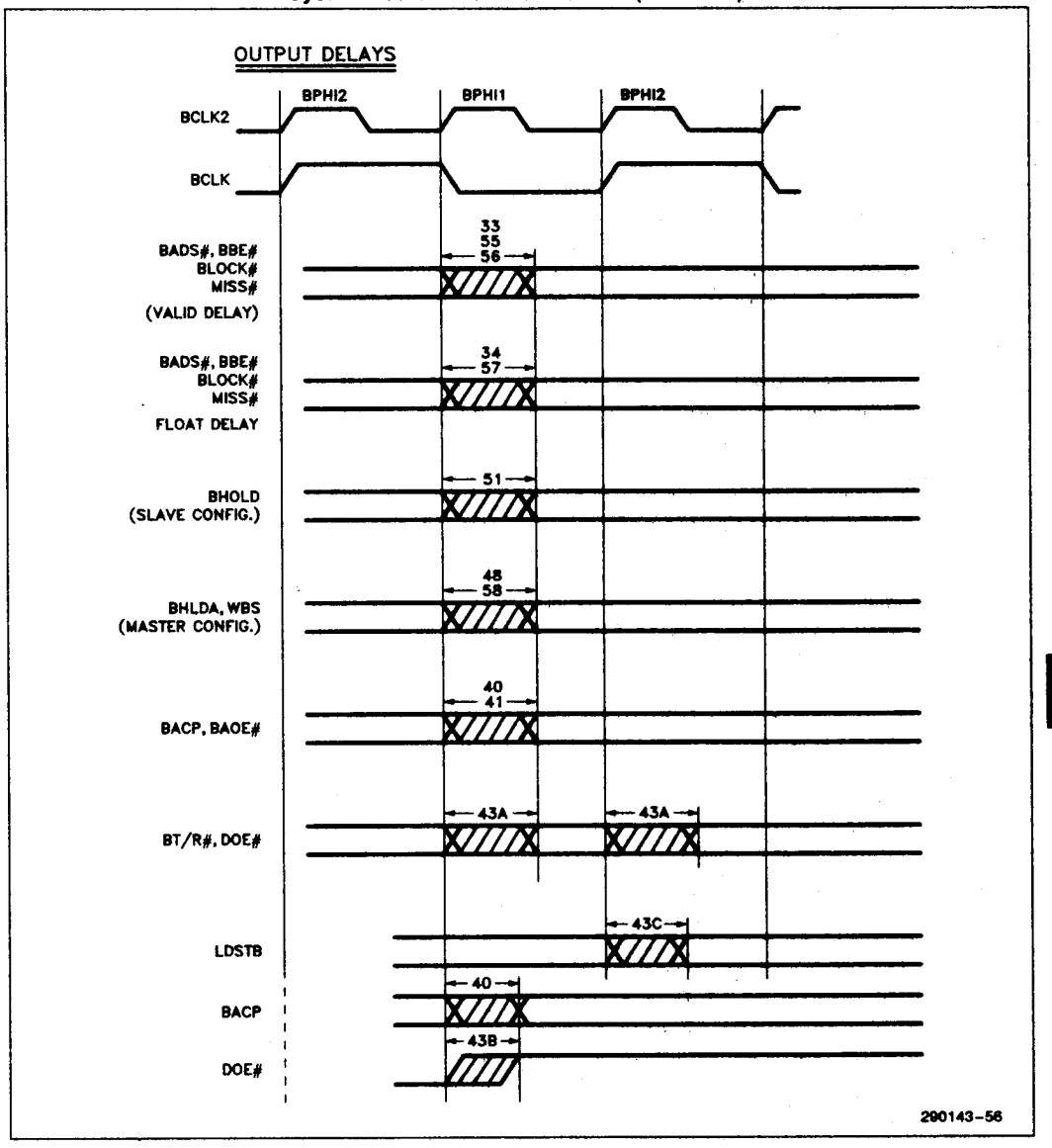
System Bus Interface Parameters



290143-55

*This would be 21B if previous cycle was Cache Write Hit.

System Bus Interface Parameters (Continued)



APPENDIX A

82385 Signal Summary

Signal Group/Name	Signal Function	Active State	Input/ Output	Tri-State Output?
386 DX INTERFACE				
RESET	386 DX Reset	High	I	—
A2–A31	386 DX Address Bus	High	I	—
BE0#–BE3#	386 DX Byte Enables	Low	I	—
CLK2	386 DX Clock	—	I	—
READYO#	Ready Output	Low	O	No
BRDYEN#	Bus Ready Enable	Low	O	No
READYI#	386 DX Ready Input	Low	I	—
ADS#	386 DX Address Status	Low	I	—
M/IO#	386 DX Memory / I/O Indication	—	I	—
W/R#	386 DX Write/Read Indication	—	I	—
D/C#	386 DX Data/Control Indication	—	I	—
LOCK#	386 DX Lock Indication	Low	I	—
NA#	386 DX Next Address Request	Low	O	No
CACHE CONTROL				
CALEN	Cache Address Latch Enable	High	O	No
CT/R#	Cache Transmit/Receive	—	O	No
CS0#–CS3#	Cache Chip Selects	Low	O	No
COEA#, COEB#	Cache Output Enables	Low	O	No
CWEA#, CWEB#	Cache Write Enables	Low	O	No
LOCAL DECODE				
LBA#	386 DX Local Bus Access	Low	I	—
NCA#	Non-Cacheable Access	Low	I	—
X16#	16-Bit Access	Low	I	—
STATUS AND CONTROL				
MISS#	Cache Miss Indication	Low	O	Yes
WBS	Write Buffer Status	High	O	No
FLUSH	Cache Flush	High	I	—
82385 INTERFACE				
BREADY#	385 Ready Input	Low	I	—
BNA#	385 Next Address Request	Low	I	—
BLOCK#	385 Lock Indication	Low	O	Yes
BADS#	385 Address Status	Low	O	Yes
BBE0#–BBE3#	385 Byte Enables	Low	O	Yes

82385 Signal Summary (Continued)

Signal Group/Name	Signal Function	Active State	Input/ Output	Tri-State Output?
DATA/ADDR CONTROL				
LDSTB	Local Data Strobe	Pos. Edge	O	No
DOE#	Data Output Enable	Low	O	No
BT/R#	Bus Transmit/Receive	—	O	No
BACP	Bus Address Clock Pulse	Pos. Edge	O	No
BAOE#	Bus Address Output Enable	Low	O	No
CONFIGURATION				
2W/D#	2-Way/Direct Map Select	—	I	—
M/S#	Master/Slave Select	—	I	—
DEFOE#	Define Cache Output Enable	—	1	—
COHERENCY				
SA2–SA31	Snoop Address Bus	High	I	—
SSTB#	Snoop Strobe	Low	I	—
SEN	Snoop Enable	High	I	—
ARBITRATION				
BHOLD	Hold	High	I/O	No
BHLDA	Hold Acknowledge	High	I/O	No

5

10.0 REVISION HISTORY

DOCUMENT: ADVANCE INFORMATION DATA SHEET
PRIOR REV: 290143-003 September 1988
NEW REV: 290143-004 September 1989

Change #	Page #	Para. #	Change
1.	Throughout	Fig. 8-3	PQFP Package added
2.	Throughout	Tables 8-2, 8-3	PQFP Info
3.	Throughout	Table 8-4	PQFP Thermal Resistance
4.	Throughout		A.C. Specifications Unified (20 MHz, 25 MHz, 33 MHz)
5.	Throughout		DEFOE# Specifications added to device