485TURBOCACHE MODULE
i486™ MICROPROCESSOR CACHE UPGRADE

82485MA (64k Module)
82485MB (128k Module)

- **High Performance**
  - **— Zero Waitstate Access**
  - **— One Clock Bursting**
  - **— Two-Way Set Associative**
  - **— BIOS ROM Cacheing**
  - **— 25/33 MHz Operation**
- **Range Of Price/Performance**
  - **— 0, 64k, 128k Cache With Single Socket**
  - **— Cascadable With Multiple Sockets**
- **High Integration**
  - **— Seven Square Inch Area**
  - **— Includes Tag, Data, Parity, and Controller**
- **Easy To Use**
  - **— Software Transparent**
  - **— End User/Dealer Installation**
  - **— Write-Through Memory Update**
  - **— Same Timing as i486™ CPU**
  - **— Same Invalidation Mechanism as i486 CPU**

The 485Turbocache Module is a performance upgrade for 25 MHz or 33 MHz i486™ Microprocessor systems. It provides up to 128k bytes of external cache memory in a single, end-user installable module. Support for the cache module upgrade is provided by a 113 pin socket in the i486 CPU system. A single socket allows three price/performance configurations: no cache, a 64k byte cache, or 128k byte cache. Additional modules may be cascaded for larger cache sizes. No jumpers, configuration software, or BIOS/applications/operating system support is required to get 5-30% (15% average) performance boost after installing the cache. Cache data integrity is monitored by a parity bit per byte.

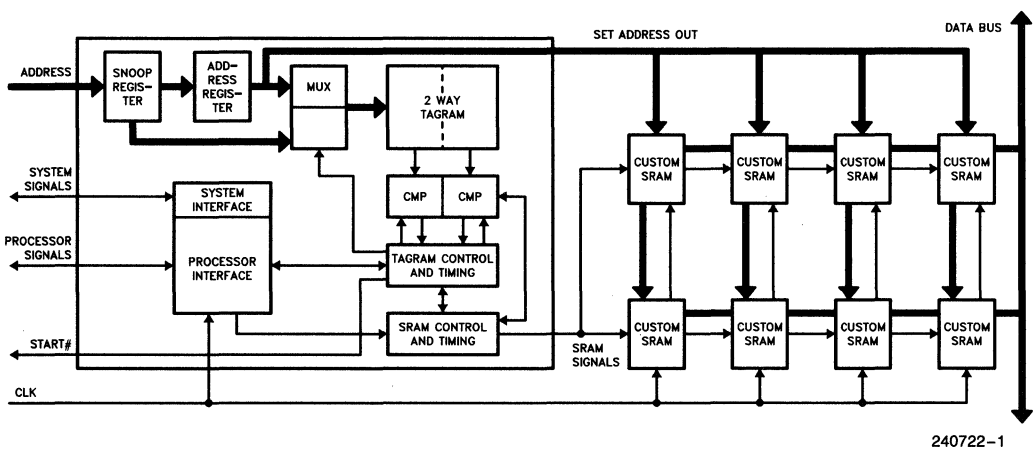


Figure 0.1. 485Turbocache Module Internal Block Diagram

5

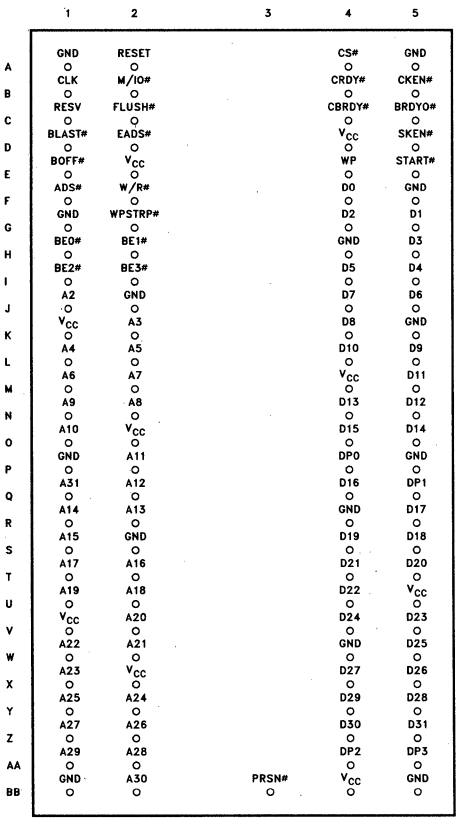
i486 and 486 are trademarks of Intel Corporation.

November 1990
Order Number: 240722-002

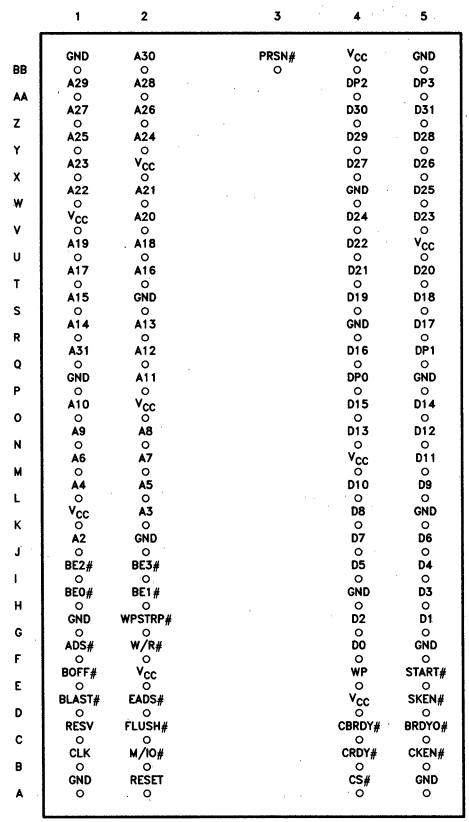
CONTENTS

CONTENTS

0.1 PINOUT



Top Side View



Pin Side View

Pin Cross Reference

Pin Name	Location	Pin Name	Location	Pin Name	Location	Pin Name	Location	Pin Name	Location
ADS#	F1	A20	V2	CBRDY#	C4	D10	L4	D29	YA
A2	J1	A21	W2	CKEN#	B5	D11	M5	D30	Z4
A3	K2	A22	W1	CLK	B1	D12	N5	D31	Z5
A4	L1	A23	X1	CRDY#	B4	D13	N4	EADS#	D2
A5	L2	A24	Y2	CS#	A4	D14	O5	FLUSH#	C2
A6	M1	A25	Y1	DP0	P4	D15	O4	M/IO#	B2
A7	M2	A26	Z2	DP1	Q5	D16	Q4	PRSN#	BB3
A8	N2	A27	Z1	DP2	AA4	D17	R5	RESET	A2
A9	N1	A28	AA2	DP3	AA5	D18	S5	RESV	C1
A10	O1	A29	AA1	D0	F4	D19	S4	SKEN#	D5
A11	P2	A30	BB2	D1	G5	D20	T5	START#	E5
A12	Q2	A31	Q1	D2	G4	D21	T4	WP	E4
A13	R2	BE0#	H1	D3	H5	D22	U4	WPSTRP#	G2
A14	R1	BE1#	H2	D4	I5	D23	V5	W/R#	F2
A15	S1	BE2#	I1	D5	I4	D24	V4	GND	A1, G1, P1, BB1,
A16	T2	BE3#	I2	D6	J5	D25	W5		J2, S2, H4, R4, BB5,
A17	T1	BLAST#	D1	D7	J4	D26	X5		W4, A5, F5, K5, P5,
A18	U2	BOFF#	E1	D8	K4	D27	X4	V_{CC}	K1, V1, E2, O2, X2,
A19	U1	BRDYO#	C5	D9	L5	D28	Y5		D4, M4, BB4, U5

Figure 0.2. 485Turbocache Module 64k/128k Pin Configuration

0.2 PIN DESCRIPTION OVERVIEW

Pin Name	Type	Active	Description
CONTROL SIGNALS			
CLK	I	–	**CLOCK** is the timing reference from which the 485Turbocache Module monitors and generates events. CLK must be connected to the i486 CPU CLK pin.
RESET	I	High	**RESET CACHE** forces the 485Turbocache Module to begin execution in a known state and must be connected to the i486 CPU RESET pin. It also causes all cache lines to be invalidated. Setup and hold times t_{23} and t_{24} must be met for recognition in any specific clock.
ADS#	I	Low	**ADDRESS STROBE** is generated by the i486 Microprocessor. It is used to determine that a new cycle has been started. Setup time t_7 must be met for proper operation.
M/IO#	I	–	**MEMORY/IO** is an i486 CPU generated cycle definition signal that indicates a Memory (M/IO# high) or I/O (M/IO# low) access. Setup time t_7 must be met for proper operation.
W/R#	I	–	**WRITE/READ** is an i486 CPU generated cycle definition signal used to indicate a Write (W/R# high) or Read (W/R# low) access. Setup time t_7 must be met for proper operation.
START#	O	Low	**MEMORY START** indicates that a cache read miss or a write has occurred and that the current access must be serviced by the memory system. START# is not activated for I/O cycles, and is not asserted if CS# is inactive.
BRDYO#	O	Low	**BURST READY OUT** is a burst ready signal driven by the 485Turbocache Module to the i486 CPU. It is activated when a read hit occurs to the 485Turbocache Module and should be a term in the BRDY# input to the i486 CPU.
CBRDY#	I	Low	**CACHE BURST READY IN** is the burst ready input from the memory system. It is applied to both the 485Turbocache Module and the i486 CPU BRDY# pin in parallel. CBRDY# is ignored during T1 and idle cycles. BLAST# determines the length of the transfer. All cacheable read cycles are 4 dword transfers. Setup and hold times t_9 and t_{10} must be met for proper operation.
CRDY#	I	Low	**CACHE READY IN** is the non-burst ready input from the system. Like CBRDY#, it is applied to both the cache and i486 CPU RDY# pin in parallel. CRDY# is ignored during T1 and idle cycles. Setup and hold times t_9 and t_{10} must be met for proper operation.
BLAST#	I	Low	**BURST LAST** is output by the i486 CPU and is sampled by the 485Turbocache Module to determine when the end of a cycle occurs. Setup and hold times t_8 and t_{8a} must be met for proper operation.
BOFF#	I	Low	**BACKOFF** is an i486 CPU input sampled by the 485Turbocache Module to indicate that a cycle be immediately terminated. If BOFF# is sampled active, the 485Turbocache Module will float its data bus. The 485Turbocache Module will ignore all cycles, except invalidation cycles, until BOFF# is deactivated. Setup and hold times t_{17} and t_{18} must be met for proper operation.
PRSN#	O	Low	**PRESENCE** is an active low output always asserted by the 485Turbocache Module. It may be used as a 485Turbocache Module presence indicator and should be connected via a 10K pullup resistor.

0.2 PIN DESCRIPTION OVERVIEW (Continued)

Pin Name	Type	Active	Description
ADDRESS SIGNALS			
A2-A31	I	–	**PROCESSOR ADDRESS LINES A2-A31** are the i486 CPU address lines used by the 485Turbocache Module. Address lines A2 and A3 are used as burst address bits. In the 64k 485Turbocache Module, A4-A14 comprise the set address inputs to the 485Turbocache Module and A15-A31 are used as the tag address. In the 128k 485Turbocache Module, A4 becomes a line select input, A5-A15 is the set address input and A16-A31 is used as the tag address. Setup time t_6 must be met for proper operation.
BE0#-BE3#	I	Low	**BYTE ENABLE** inputs are connected to the i486 CPU byte enable outputs. They are specifically used for completing partial reads from and writes to the 485Turbocache Module during hit cycles. During miss cycles, transfers are ignored if all the byte enables are not asserted since the 485Turbocache Module only caches 32-bit transfers. Setup time t_6 must be met for proper operation.
CS#	I	Low	**CHIP SELECT** is used to cascade 485Turbocache Module modules. Address bits may be decoded in order to cascade multiple devices or be decoded to selectively cache portions of memory. Setup and hold times t_{30} and t_{31} must be met for proper operation.
DATA SIGNALS			
D0-D31	I/O	–	**PROCESSOR DATA LINES D0-D31** are connected to the i486 CPU data bus. D0-D7 define the least significant byte while D24-D31 define the most significant byte. Setup and hold times t_{13} and t_{14} must be met for proper operation.
DP0-DP3	I/O	–	**DATA PARITY** are the parity bits associated with the data on the data bus. They are connected to the i486 CPU pins with the same name. Parity is treated by the 485Turbocache Module as additional data bits to be stored. Setup and hold times t_{13} and t_{14} must be met for proper operation.
CACHEABILITY SIGNALS			
CKEN#	O	Low	**CACHE ENABLE TO CPU** is the KEN# term generated by the 485Turbocache Module to the i486 Microprocessor. CKEN# is activated twice; First during T1 to enable a cache line fill, and second on the clock before the last BRDY# or RDY# to validate the line fill. CKEN# is ALWAYS active in T1, but will not validate a line fill if the line fill is a write protected line and WPSTRP# is low, or if the cycle is a read miss.
SKEN#	I	Low	**SYSTEM CACHE ENABLE** is an input from the main memory system to indicate whether the current line fill is cacheable in the 485Turbocache Module. It is sampled by the 485Turbocache Module exactly like KEN# is sampled by the i486 Microprocessor. Setup and hold times t_{11} and t_{12} must be met for proper operation.
FLUSH#	I	Low	**FLUSH CACHE** causes the 485Turbocache Module to invalidate its entire cache contents regardless of CS#. Any line fill in progress will continue, but will be invalidated immediately. The i486 CPU flush instruction does not affect the 485Turbocache Module. Setup and hold times t_{23} and t_{24} must be met for recognition in any specific clock.

0.2 PIN DESCRIPTION OVERVIEW (Continued)

Pin Name	Type	Active	Description
CACHEABILITY SIGNALS (Continued)			
WP	I	High	**WRITE PROTECT** defines a line as write protected. WP is sampled during the third transfer of a line fill and is maintained internally as a state bit. Any subsequent writes to a write protected line will have no effect. Setup and hold times t_{15} and t_{16} must be met for proper operation.
WPSTRP#	I	Low	**WRITE PROTECT STRAPPING OPTION** changes the behavior of CKEN#. CKEN# is always asserted in T1 to indicate a cacheable line transfer but is deasserted on the next clock. During read hit cycles, CKEN# is asserted again for the duration of the transfer to indicate a cacheable line fill. If WPSTRP# is strapped low, and a write protected line is being transferred, CKEN# is not activated again for the transfer. This prevents the i486 CPU from cacheing write protected lines during read hit cycles. WPSTRP# must be valid and not change two clocks before and after the falling edge of RESET.
INVALIDATE SIGNALS			
EADS#	I	Low	**VALID EXTERNAL ADDRESS STROBE** indicates that an invalidation address is present on the i486 CPU address bus. The 485Turbocache Module will invalidate this address, if present, but will only do so if CS# is active. The 485Turbocache Module is capable of accepting an EADS# every other clock. The 485Turbocache Module EADS# should be connected to the i486 CPU EADS# pin. Setup and hold times t_{19} and t_{20} must be met for proper operation.

1.0 FUNCTIONAL DESCRIPTION

1.1 Introduction

The 485Turbocache Module is a complete 2-way set-associative 64k or 128k cache housed in a 113-pin module. It contains 4 or 8 custom data SRAMs and the Intel 82485 cache controller. The cache module was designed to be cascadable to a maximum of 512k with the addition of more modules. The module was also designed so the system may easily detect a cache's presence and reconfigure itself accordingly. The 485Turbocache Module is a plug-in option that is an ideal i486™ Microprocessor cache solution.

The cache module interfaces directly to the i486 Microprocessor. Designing with the cache module is easy because it directly supports the timing of 25 MHz and 33 MHz systems. It is capable of reading and writing data in 0 waitstates, and performing 1 clock bursting. Because the 485Turbocache Module was designed exclusively for the i486 Microprocessor, it recognizes i486 CPU invalidations, use of BOFF#, and prematurely terminated cycles. The cache module is write-through so it supports the same i486 CPU consistency mechanisms, stores data parity, can cache BIOS in modes where the i486 CPU cannot, is software transparent, and may be an end-user installable upgrade.

Below are the order codes for the 485Turbocache Module:

Size	25 MHz	33 MHz
64k	82485MA-25	82485MA-33
128k	82485MB-25	82485MB-33

The following Functional Description describes the cache module's base architecture, its operation, features, and deviations from the i486 CPU specification.

5

1.2 Base Architecture

The 485Turbocache Module contains an 82485 cache controller and 4 (82485MA) or 8 (82485MB) SRAMs for a complete 64k or 128k cache. In either configuration, the 485Turbocache Module is 2-way set-associative with a 16 byte line size.

Figure 1.1 outlines the 82485 cache controller which is the heart of the 485Turbocache Module. Each WAY contains 2k tags with 17 bits per tag so it may store the complete 4G real address space. The tags also reference 2 valid bits and a write-protect bit. When the 82485 is configured as a 64k cache, as in the 64k 485Turbocache Module, each tag references a single, 16 byte line. When the 82485 is configured as a 128k cache, as in the 128k 485Turbocache Module, each tag is forced to reference two consecutive 16 byte lines; This is called sectoring. A 128k 485Turbocache Module contains 2 sectors per tag. The LS input (address bit A4) determines which sector of each tag is being selected.

The control units of the 82485 are responsible for three main functions: controlling the data SRAMs, controlling the tagram structure, and interfacing to the i486 CPU. Since these are independent units, the 82485 is capable of updating its tagram while data is being bursted into SRAM, or invalidating during a line fill to a different address. Special address registers in the 485Turbocache Module allow the i486 Microprocessor to drop its address in the first T2 (in response to AHOLD) and the system to issue a invalidate address with an i486 CPU hold time.

The 82485 uses the "Least Recently Used" algorithm to determine which tag should be invalidated on cache misses. A single LRU bit per tag is used to point to the tag that will be replaced should a replacement be required.

The data memory portion of the 485Turbocache Module is composed of a set of SRAMs that operate at fast 33 MHz speeds. They are capable of 0 wait-state reads and writes, and single clock bursting, and have minimized capacitive loading on the i486 CPU clock and data lines.

1.3 Cache Operation

To operate at high speeds, the 485Turbocache Module must begin its tag lookup to determine a cache hit or miss as soon as possible. During normal operation, this is done as soon as the i486 CPU generates an address. SRAM reads, SRAM writes, and system signals cannot be generated until a hit or miss has been determined. The following sections will discuss read miss, read hit, write, invalidate, and BOFF# cycles.

1.3.1 READ MISS

Figure 1.2 shows 485Turbocache Module activity during a normal read miss cycles. In T1, the 485Turbocache Module begins its tag lookup to see if the read cycle is a hit. Once it has been determined that the address is not present in the cache (a miss), START# is issued to indicate to the memory system that it must service the current cycle. The cache is then idle until SKEN#, the cache's KEN# input, is seen active. Should SKEN# be inactive and the burst line transfer from memory begin, the line is non-cacheable and is ignored.

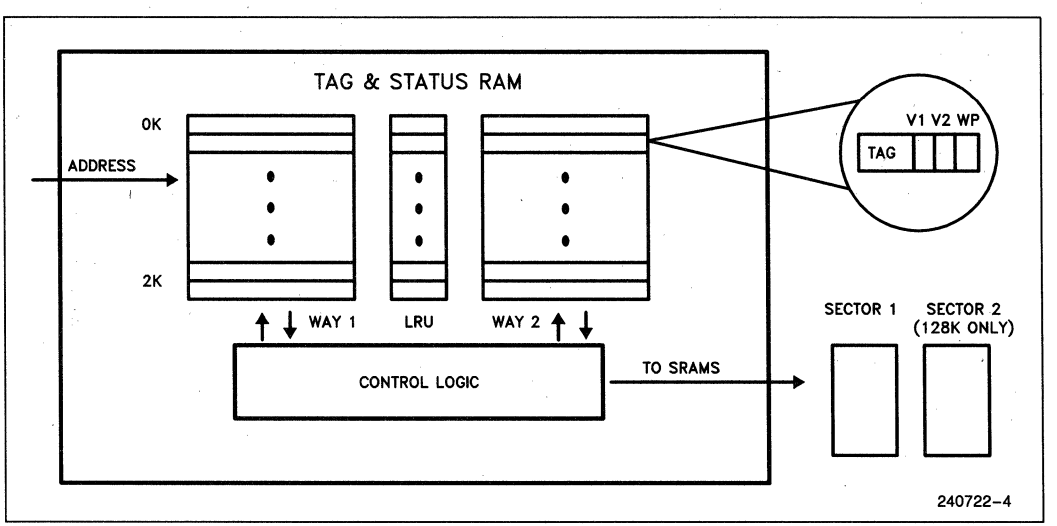


Figure 1.1. 82485 Cache Controller

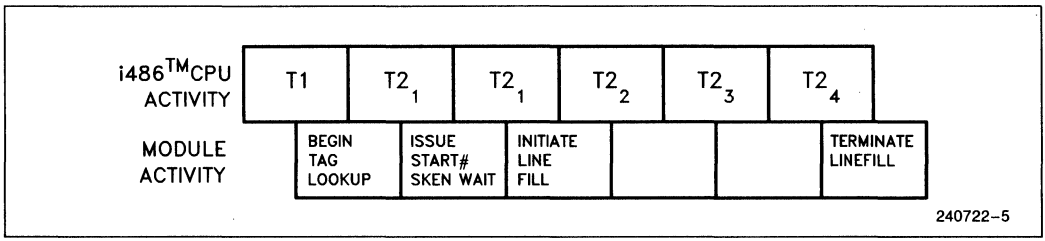


Figure 1.2. Normal Read Miss Cycle

Once SKEN# has been asserted, the 485Turbocache Module invalidates a line in the cache (or chooses a free line) in preparation for the bursted data (see section 4.3.1). The data is bursted into the cache and back to the i486 Microprocessor simultaneously. If an SKEN# preceded the last bursted item, then the line was cacheable, and the 485Turbocache Module updates its valid bit to indicate so. If the line is invalid, or aborted for any reason (BLAST#, BOFF#) the line is left invalid.

During a read miss cycle, the 485Turbocache Module cannot accept the data from memory in zero waitstates. The earliest data may be returned is the clock after START# is sampled active. START# is the signal that indicates that the memory system must complete the current cycle.

The 485Turbocache Module is also capable of handling non-burst and interrupted burst line fills. Refer to the section "4.0. Performance Considerations" for improving 485Turbocache Module performance during line fills. Note that the 485Turbocache Module only caches 32-bit transfers. The 485Turbocache Module does not input the i486 CPU inputs BS#8 or BS#16. All transfers are assumed to be 32-bit transfers with valid data on all 32 data lines.

1.3.2 READ HIT

During Read Hit cycles, the 485Turbocache Module responds directly to the i486 Microprocessor with a line of data in 5 clocks. The 485Turbocache Module asserts CKEN# (its KEN# output to the i486 CPU) in both T1 and the third T2 to indicate this as a cacheable transfer. Should the bursted line be write-protected, AND WPSTRP# is strapped low, CKEN# is high for the third T2 and the line is not cached by the i486 CPU. The only updating the 485Turbocache Module needs to perform during read hit cycles is to update the LRU bit to point to the WAY that was not transferred.

1.3.3 WRITE CYCLES

Since the 485Turbocache Module is a write-through cache, all write cycles are written by the i486 CPU to main memory. Figure 1.3 shows a write hit where the tag lookup in T1 is found to be a hit so the data is updated by the cache in T2. Write misses do not affect cache contents, nor do writes to write protected lines. Write hits will alter the LRU bit in the same way as a read hit.

5

1.3.4 INVALIDATION CYCLES

The 485Turbocache Module allows invalidation cycles to occur at any time by asserting AHOLD and EADS#. Self-invalidations, where AHOLD is not asserted, are allowed at any time except on the clock edge of the last transfer of a line fill. EADS# assertion allows both the CPU cache and 485Turbocache Module to be invalidated at the same time. Regardless of what the 485Turbocache Module is doing,

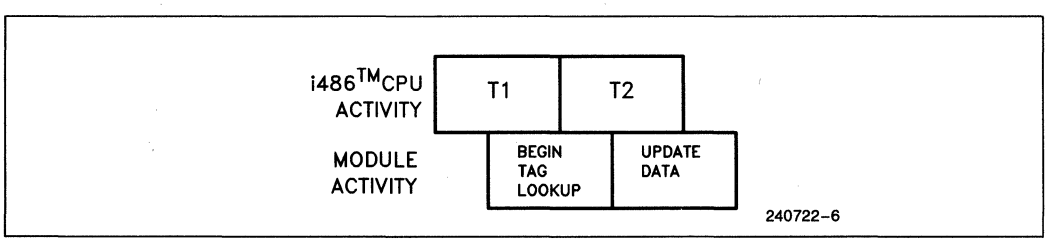


Figure 1.3. Write Hit Cycle

EADS# causes the address present on the address inputs of the 485Turbocache Module to be invalidated. This includes read hit, read miss, write, and BOFF# cycles.

There may be a performance penalty, however, if EADS# is asserted at a time when the tag memory of the 485Turbocache Module is in use. Since the 485Turbocache Module tags are single-ported, only one tag access per clock is allowed.

Figure 1.4 shows a read miss cycle with an invalidation lookup occurring in the third transfer of a line fill. Under normal conditions, the 485Turbocache Module would, on the next clock, validate the current line that is being filled. Since the EADS# occurred, the tagram is occupied on the next clock with a tag lookup to see if the invalidate is a hit, and the current line is not yet validated. If it is a hit, the next cycle is used to perform the actual invalidation. The following clock is spent validating the current line fill. Should the i486 Microprocessor begin a cycle immediately, the 485Turbocache Module is not able to perform its tag lookup until one clock cycle later when the tag memory is free. This causes START# to be delayed, and ultimately a memory read cycle from beginning.

For greatest performance, EADS# should not be issued in the second, third or fourth transfer of a cache line fill.

Self-Invalidations, EADS# asserted without AHOLD, are not allowed at the clock edge of the last T2 of a cycle (the first T1 clock edge of the next cycle). If a self-invalidation occurs in T1, ADS# and EADS# are sampled at the same time, the 485Turbocache Module will invalidate the line and assert START# as in a normal read miss cycle. If EADS# is asserted at any other time, START# is not asserted.

1.3.5 BOFF# CYCLES

When BOFF# is asserted, the 485Turbocache Module, like the i486 Microprocessor, will relinquish the bus in the next clock cycle. While BOFF# is asserted, as any other time, the 485Turbocache Module monitors EADS# to perform any invalidate cycles.

If BOFF# is asserted during a cache read hit (data is being transferred from cache to CPU), the 485Turbocache Module invalidates the line being transferred. Once BOFF# has been released and the cycle resumes, the 485Turbocache Module sees this as a cache miss and the memory system must supply the remaining data. If BOFF# is asserted during a cache read miss (memory is transferring to cache and CPU), the 485Turbocache Module will treat the line fill like an aborted fill, and the line will remain invalid. Once BOFF# is released and the cycle is restarted, the remainder of the line fill is treated like another aborted fill, and remains invalid.

Figure 1.5 is an example of an aborted line fill. Since the line transfer is interrupted before the transfer completes, it stays invalidated. Once the transfer resumes, the 485Turbocache Module sees a new cycle begin with ADS#, but it completes with BLAST# after three transfers. It treats this as an aborted line fill cycle, and the cycle is never validated.

Asserting BOFF# in the same clock as ADS# will cause the i486 CPU to float its bus in the next clock and leave ADS# floating low. Since ADS# is floating low, a peripheral device may think that a new bus cycle has begun even though the cycle was aborted. The 82485 handles this circumstance in most cases since an active ADS# in the clock BOFF# is deasserted is ignored. The only circumstance that must be handled by the system is as follows:

BOFF# is asserted in T1, and before BOFF# is deasserted, HOLD is asserted and remains asserted after BOFF# is deasserted (see Figure 1.6). In this circumstance it is necessary for the system to assure that ADS# is either driven to a valid level or pulled high in the clock after BOFF# is deasserted (meeting the 82485 ADS# setup time).

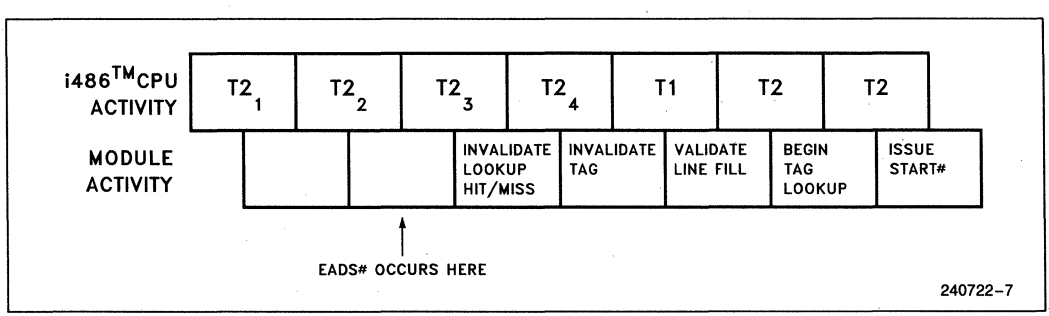


Figure 1.4. Invalidation During Read Miss

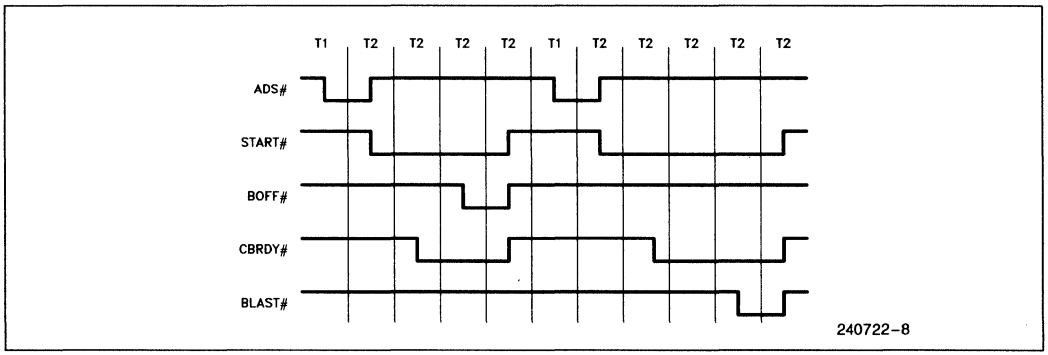


Figure 1.5. Aborted Line Fill

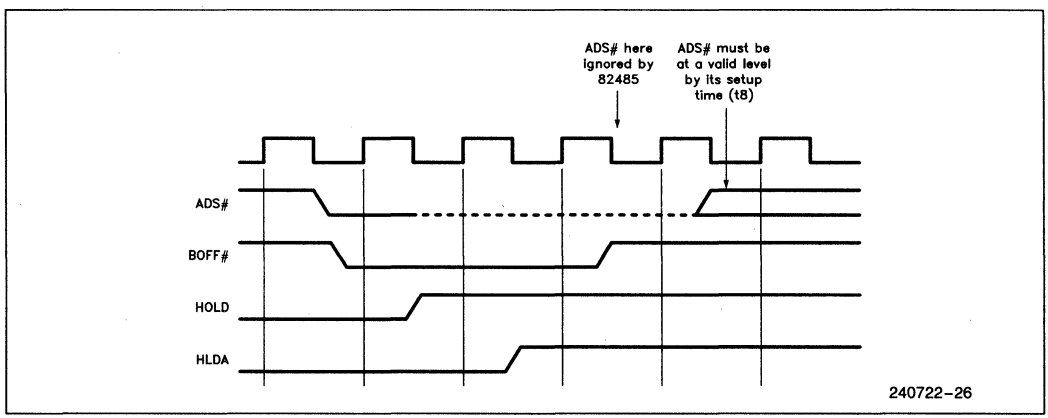


Figure 1.6. BOFF# Asserted in T1

There are several ways to avoid this system restriction:

1. Do not assert BOFF# in T1.
2. Use a "two clock" backoff: in the first clock AHOLD is asserted and in the second clock BOFF# is asserted. This guarantees that ADS# will not be floating low.
3. Do not assert HOLD when BOFF# is asserted.

1.4 Incompatibilities

Below are a list of some special design considerations that the 485Turbocache Module requires to be designed into an i486 CPU system. They have been summarized to point out any possible inconsistencies between the i486 CPU specification and the 485Turbocache Module specification:

1. Invalidation cycles may only be performed every two clocks. Unlike the i486 CPU, the 485Turbocache Module only allows EADS# assertion every other clock at most.
2. The minimum clock high voltage is slightly higher than the i486 CPU specification. It is still within TTL levels, however.
3. The i486 CPU will recognize HOLD during non-burst, non-cacheable, code prefetches. These prefetches are cacheable by the 485Turbocache Module. Since the module does not see the HLDA signal, another bus master could hold the CPU in mid-cycle, begin its own transfer, and coincidentally complete the cacheable transfer. This is only possible in systems that have another bus master that can drive the module's ADS pin. In these systems, the CPU's HLDA pin should be inverted and connected to the module's BOFF# input. This guarantees that cycles interrupted by HLDA will be aborted, and not cached, by the 485Turbocache Module.

2.0 SYSTEM INTERFACE

The following section describes the basic connection of the 485Turbocache Module in an i486 CPU

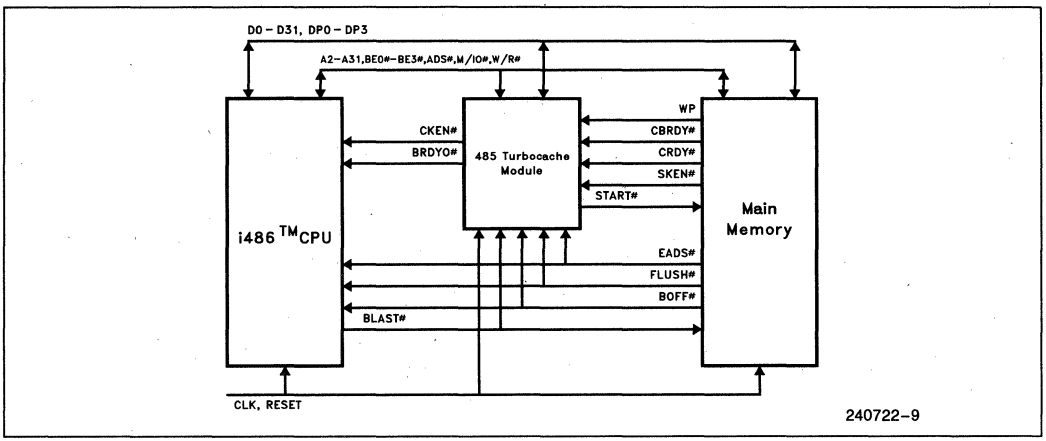


Figure 2.1. 485Turbocache Module Typical Configuration

system. The section highlights the CPU bus connections, memory bus connections, and gives specifics about their related signals.

A typical 485Turbocache Module connection to an i486 Microprocessor and memory subsystem is shown in Figure 2.1. All of the signals that the i486 CPU generate "feed-around" the 485Turbocache Module; That is, they go to both the 485Turbocache Module and the memory controller. In turn, most memory generated signals feed-around the 485Turbocache Module back to the CPU. This is what makes the 485Turbocache Module an optional cache. The following describes all the signals the 485Turbocache Module encounters.

2.1 i486™ Microprocessor Signals

The following 485Turbocache Module signals connect directly to the corresponding i486 CPU signals. These pins have the same name and functionality as the i486 Microprocessor pins.

2.1.1 ADDRESS LINES A2-A31

A2-A31 are the address lines generated by the i486 CPU and used by the cache as set and tag addresses. A 64k 485Turbocache Module cache will use A4-A14 as set address inputs and the remaining address bits as tag address. A 128k 485Turbocache Module uses A5-A15 as set address inputs, A4 as the line select bit for sectoring, and the remaining bits as tag address. Address lines A2 and A3 are used as burst address inputs.

The address lines are also used as invalidate inputs. At any time, if EADS# is asserted, the address that is present at the address inputs will be invalidated. The 485Turbocache Module will not invalidate unless CS# is sampled active. Note that the address is latched internally so that AHOLD assertion in T1 is permitted.

2.1.2 DATA LINES D0-D31 AND PARITY DP0–DP3

This is the processor data bus common to the i486 CPU, the 485Turbocache Module, and memory bus. The 485Turbocache Module transfers information to the CPU on read hits, and stores data from memory on read misses. The four parity bits, DP0-DP3 are treated just like extra data bits.

2.1.3 ADS#, W/R#, M/IO#

The processor control signals ADS#, W/R#, and M/IO# are used by the 485Turbocache Module to indicate the start of a new cycle, and identify the type of cycle. ADS# assertion indicates a T1 cycle and initiates the tag lookup process in the 485Turbocache Module. I/O cycles are ignored.

ADS# is the primary signal that activates the 485Turbocache Module. When ADS# goes low, the module begins the hit/miss tag lookup regardless of the state of Chip Select (CS#). For this reason, any bus master that controls the ADS# input to the module must meet the module address bus setup and hold times, regardless of the state of CS#. Chip Select, when inactive, disables the module outputs only. (Note that CS# must be asserted for invalidation cycles.)

2.1.4 BYTE ENABLES BE0#-BE3#

Byte enable inputs are used to complete partial byte or word writes to the 485Turbocache Module on cache write hit cycles. All other partial transfers are ignored by the 485Turbocache Module.

2.1.5 BLAST#

BLAST# is used by the 485Turbocache Module to indicate the end of a cycle. If BLAST# is asserted early during a cache line fill from a read miss, that transfer is left invalid by the 485Turbocache Module.

2.1.6 BOFF#

Once BOFF# is sampled by the 485Turbocache Module, it relinquishes control of the data bus in the next clock. If a read hit line transfer was in progress, that transfer will not continue once BOFF# is released. If a read miss transfer was interrupted by BOFF#, the 485Turbocache Module would mark the line as invalid even if the transfer continues once BOFF# has been released. The 485Turbocache Module will recognize invalidations during BOFF#, but will only do so if CS# is active.

2.1.7 FLUSH#

The 485Turbocache Module FLUSH# input behaves exactly like the i486 Microprocessor input. Once asserted, FLUSH# will invalidate the entire contents of its cache memory regardless of the state of CS#. While FLUSH# is asserted, the 485Turbocache Module continues to track CPU bus cycles and treats all accesses as cache misses, activating START# appropriately.

FLUSH# may be used asynchronously with both the i486 CPU and the 485Turbocache Module. If the proper pulsewidths are given, FLUSH# will be recognized, but, it is possible that the FLUSH# will be recognized on different clock edges for each device. This may happen if FLUSH# assertion or deassertion is near its setup and hold times when one device may recognize it and the other may not.

2.1.8 EADS#, AHOLD

EADS# assertion causes the 485Turbocache Module to invalidate the address present on the address bus if CS# is seen active. AHOLD need not be asserted, nor is it even used as an input to the 485Turbocache Module. EADS# may be asserted at most once every other clock as that is the fastest 485Turbocache Module invalidation rate. The section titled "invalidation cycles" describes where EADS# may be asserted for maximum performance.

EADS# may not be asserted on the clock edge of the last T2 of a cycle (the first T1 of the next cycle) if AHOLD is not asserted.

2.1.9 RESET

RESET is an asynchronous input that causes the 485Turbocache Module to reset its internal machines to a known state: its entire cache contents invalidated, and expecting the start of a new bus cycle. RESET must be high for at least 10 clocks for the 485Turbocache Module to reset properly from a warm boot. For a cold boot, RESET must remain active for 3000 ns (100 clocks at 33 MHz, 75 clocks at 25 MHz). There must be no bus activity for at least 4 clocks after the falling edge of RESET so the 485Turbocache Module can reset internally. The falling edge of RESET causes the 485Turbocache Module to sample its WPSTRP# strapping option.

2.2 CPU Bus Interface Signals

These are signals generated by the 485Turbocache Module, or decoded from the i486 CPU that correspond to the CPU bus.

2.2.1 CHIP SELECT CS#

Chip Select is used to select the proper 485Turbocache Module cache module if multiple modules are used, otherwise, with one 485Turbocache Module, CS# may be grounded. CS# is generated by decoding the lowest order tag addresses coming into the module. For example, two 128k cache modules would decode A16 for their chip selects. A16 high would select module 1, while A16 low would select module 2. The following table summarizes the addresses used for decoding:

Size	Modules	Address Bit(s) to Decode
64k	2	A15
64k	4	A15, A16
128k	2	A16
128k	4	A16, A17

For compatibility, A16 and A17 may be decoded for 64k modules. Performance may be increased because of increased granularity, however, if A15 and A16 are used.

With CS# inactive, invalidation cycles are ignored, START# is inactive, and CKEN# is inactive. CKEN# does, however, always activate in T1 as it is not possible for the 485Turbocache Module to recognize CS# before then.

If required, the LOCK# signal may be used as a term in the creation of CS#. If locked cycles do not generate CS#, START# must be generated externally so memory may handle the cycle.

2.2.2 CPU CACHE ENABLE CKEN#

CKEN# is generated by the 485Turbocache Module to indicate that its current transfer, during a read hit cycle, is cacheable. It is always driven (not an open-collector output) and must be used as one of the terms that generates KEN# to the i486 Microprocessor. CKEN# is always active in T1, but then goes inactive and remains inactive unless the cycle is a read hit cycle.

For read miss and write cycles, CKEN# goes inactive in T2 and remains inactive until the next T1. It is the responsibility of the system to generate the KEN# signal to the i486 CPU in these cases.

In a read hit cycle, CKEN# goes active again in the second T2 and remains active throughout the cycle. This forces external KEN# logic to activate KEN# and make the cycle cacheable to the i486 CPU. However, if the line being transferred is write-protected, AND the WPSTRP# pin is strapped low, CKEN# stays inactive in T2 and remains inactive throughout the cycle. This allows write protected lines in the 485Turbocache Module to be cacheable only to the 485Turbocache Module.

2.2.3 BURST READY OUT BRDYO#

The 485Turbocache Module generates BRDYO# when it is bursting data back to the i486 CPU during read hit cycles. BRDYO# is always driven (not an open collector output) and should be used by external logic to create the BRDY# input signal to the i486 CPU. Since the 485Turbocache Module is a zero waitstate, single clock burst cache, BRDYO# is activated in the first T2 until the fourth T2 unless the cycle is interrupted.

2.3 Memory Interface Signals

Memory Interface Signals are signals coming to or from the main memory subsystem. The only signal the 485Turbocache Module generates to the memory system is START#, which is the only signal that must be handled should the 485Turbocache Module be designed as an option.

2.3.1 PRSN#

This signal is tied low inside the 485Turbocache Module. If the system pulls this signal high with a 10K pullup resistor, cache presence will be indicated by that line being pulled low. PRSN# signal is used to indicate that external logic should only start memory cycles when START# goes active rather than from ADS# active.

2.3.2 START#

START# is a signal asserted by the 485Turbocache Module to indicate that the memory subsystem must process the current cycle. START# is always driven and valid and is asserted for all read miss cycles and memory write cycles. START# is not activated for I/O cycles, or if CS# is sampled inactive. START# is normally active in the first T2, but may be delayed if an invalidation cycle forced the previous cycle to be elongated (see 1.3.4 Invalidation cycles).

2.3.3 WRITE PROTECT WP

The Write Protect input is an active high input that indicates to the 485Turbocache Module that the current line transfer is write-protected. It is sampled on the clock edge of the third BRDY# of a line transfer of a read-miss cycle. The 485Turbocache Module saves this information as a single bit in each tag location. In 128k configurations where there is a single tag for 2 consecutive lines, the write protect bit is valid for both lines. If a location has been write-protected, and writes to that location will be ignored.

WP is a synchronous input and must meet the 485Turbocache Module setup and hold times regardless of whether it is being sampled or not.

2.3.4 WRITE PROTECT STRAPPING OPTION WPSTRP#

WPSTRP# is a strapping option that is sampled during RESET. It indicates whether write protected items in the 485Turbocache Module should be cacheable in the i486 CPU cache. If WPSTRP# is high , CKEN# will go active in T2 during all read hit cycles to indicate that they are cacheable. If WPSTRP# is low, CKEN# will be inactive in T2 for read hit cycles to locations that are write-protected. This allows write protected items to be cached by the 485Turbocache Module and not by the i486 CPU.

2.3.5 SYSTEM CACHE ENABLE SKEN#

The SKEN# input to the 485Turbocache Module is like the KEN# input to the i486 Microprocessor. It is sampled just like KEN#, the clock before the first and last transfers of a line fill, to indicate whether the line is cacheable. If the KEN# input to the i486 CPU is connected to the SKEN# input of the 485Turbocache Module, the i486 CPU internal cache and the 485Turbocache Module will cache the same items. It is possible to control KEN# and SKEN# separately so the 485Turbocache Module and i486 CPU cache different areas of memory.

SKEN# is a synchronous input and must meet the 485Turbocache Module setup and hold times regardless of whether it is being sampled or not.

2.3.6 CACHE READY AND BURST READY CRDY#, CBRDY#

CRDY# and CBRDY# are the ready and burst ready inputs to the 485Turbocache Module. They should behave exactly like the i486 CPU RDY# and BRDY# inputs. CBRDY# should be used in conjunction with BRDYO# to generate the i486 CPU BRDY# input. Likewise, CRDY# should be used to form the i486 CPU RDY# input.

The 485Turbocache Module does not sample the CBRDY# or CRDY# inputs during read hits, so it is not possible to artificially add waitstates to the 485Turbocache Module's burst transfer. The CBRDY# and CRDY# inputs must, follow 485Turbocache Module setup and hold times even outside the sampling window.

3.0 SYSTEM CONFIGURATIONS

Two of the most important features of the 485Turbocache Module are its cascadability and its optionality. Below, it is explained how to design a system with a single 485Turbocache Module, multiple 485Turbocache Modules and a socket for an optional 485Turbocache Module.

3.1 Single Cache

In a single cache configuration, the addition of a 485Turbocache Module requires no or little extra logic. Most of the signals are common to the i486 CPU, the memory bus controller, and the 485Turbocache Module. The others, such as KEN#, SKEN#, and START# will be discussed individually.

3.1.1 i486™ MICROPROCESSOR BUS INTERFACE

As seen in Figure 2.1, the i486 CPU-related signals are connected to both the 485Turbocache Module and the memory controller. These are the address bus, data and parity bus, ADS#, W/R#, M/IO#, BE0#-BE3#, BLAST#, RESET, and CLK.

Since a single 485Turbocache Module resides on the address bus, CS# may be tied low so the part is always chip selected.

3.1.2 MEMORY BUS INTERFACE

On the memory bus side, BOFF#, FLUSH#, and EADS# are connected to the i486 CPU and the 485Turbocache Module in parallel. The memory ready signals, CRDY# and CBRDY#, are connected directly to the 485Turbocache Module, but are combined with other system ready signals to form the i486 CPU RDY# and BRDY# inputs. One of the system ready signals is the 485Turbocache Module BRDYO# which must be ANDed with CBRDY# and other burst ready signals to form BRDY# into the CPU.

The memory system must also generate the WP input. If write-protection is not needed, WP may be tied to V_{SS}. If the system would like to prevent write-protected lines in the 485Turbocache Module from being cached by the i486, WPSTRP# should be tied to V_{SS}.

3.1.3 KEN# AND SKEN# GENERATION

The KEN# input to the i486 Microprocessor is a result of all the cache enable signals in the system. Since the 485Turbocache Module activates CKEN# only during a read hit cycle, the CKEN# output may be ANDed with the system cache enable signal to form KEN# to the i486 CPU.

If the 485Turbocache Module and i486 CPU internal cache will cache the same areas of memory, the KEN# input to the i486 CPU may be tied to the SKEN# input of the 485Turbocache Module. Otherwise, the memory system can generate 2 cache enable signals: One that is ANDed with CKEN# to produce KEN#, and another for the SKEN# input.

3.1.4 START# GENERATION

START# goes low to indicate that the memory system must complete the current cycle. This is true for all memory writes and read misses. It is the memory subsystem's responsibility to recognize I/O cycles and begin an I/O access without waiting for START#.

START# is asserted in T2, but may be delayed if there was an invalidation in the previous cycle (see 1.3.4 Invalidation cycles). Because the assertion of START# may be somewhat unpredictable, it is recommended that START# be used to either begin a DRAM RAS cycle, or enable DRAM output buffers.

Figure 3.1a shows that START# may be the indication to DRAM control to begin a cycle. Once START# is sampled active, a RAS and CAS cycle begin. This will incur an extra waitstate to cache read misses since the earliest a memory cycle will begin is the first T2.

Figure 3.1b shows that START# may enable DRAM data buffers. The actual DRAM cycle begins once ADS# and M/IO# are sampled low, but will not complete until the buffers have been gated allowing data to be written to the i486 CPU data bus. Should the cycle be a 485Turbocache Module read hit, the buffers are never enabled. Since the 485Turbocache Module takes 5 clock cycles to complete the burst transfer, RAS precharge time can easily be absorbed.

See 4.3.4 START# Predictability for detailed information how START# may be asserted in a predictable manner.

3.2 Multiple Cache

A multiple cache scheme is similar to the single cache scheme because all of the i486 Microprocessor bus interface signal connection remain the same. Like the single cache example, only KEN#, SKEN#, START#, and now CS#, need special handling. Figure 3.2 is an example of a 512k multiple cache configuration.

3.2.1 MEMORY BUS INTERFACE

Like the i486 Microprocessor bus interface signals, BOFF#, FLUSH#, and EADS# are connected to the CPU, memory system, and all caches in parallel. The ready and burst ready outputs from the memory system connect to the CRDY# and CBRDY# inputs to all 485Turbocache Module caches. The CBRDY# signal is then ANDed with the BRDYO# outputs from all 485Turbocache Modules to form BRDY# to the i486 CPU.

3.2.2 START#

START# is activated by a single 485Turbocache Module at a time because CS# is active for a single 485Turbocache Module at a time. START#, therefore, may be ANDed with all other START# signals to form a system start indication. See section 3.1 Single Cache for details how START# may be used.

3.2.3 KEN#

Like START#, CKEN# is only activated for chip selected modules. Therefore, all CKEN# outputs may be ANDed together to form the i486 CPU KEN# signal. A system cache enable signal must also be included in the AND terms since it is the system's responsibility to generate KEN# during read miss cycles.

3.2.4 SKEN#

Since SKEN# is used during read miss cycles and ignored otherwise, the system cache enable signal can be connected to all 485Turbocache Modules' SKEN# inputs. If multiple sources can create the

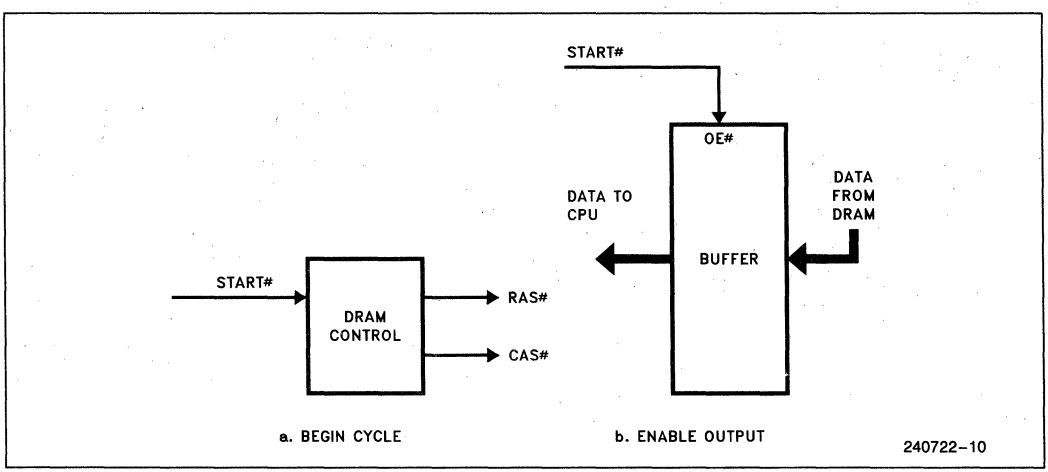


Figure 3.1. Using START# in DRAM Control

KEN# signal to the i486 CPU, KEN# may be fed into all 485Turbocache Modules. If the i486 CPU caches different memory locations than the second-level cache, SKEN# must be generated separately and then connected to all 485Turbocache Module inputs.

3.2.5 CS#

Chip select is used to identify which 485Turbocache Module is being addressed. It is the result of decoding the lowest order tag address bits. Figure 3.2 shows how a PLD chooses one of four 485Turbocache Modules. Anytime an address is present on the address bus, including invalidation cycles, one of the 485Turbocache Modules is selected.

3.3 Optional Cache

The 485Turbocache Module is an optional cache. However, its most powerful feature is allowing a system to reconfigure itself easily once a 485Turbocache Module has been installed. To accomplish this, the 485Turbocache Module is designed as a write-through cache with all signals feeding around to the memory subsystem whether the 485Turbocache Module is present or not. There are only a few considerations that need to be made to allow the 485Turbocache Module to be fully optional.

3.3.1 SIGNAL CONSIDERATIONS: START#, CKEN#, BRDYO#

If the 485Turbocache Module is not present in a system that expects it to be, the START# signal will never be asserted and memory will never begin a cycle. A solution to this problem is to connect the PRSN# presence pin into the memory controller that accepts START#. If PRSN# is high, the 485Turbocache Module is not present, and all memory cycles should begin with the assertion of ADS#. Note that START# should have a pullup resistor to ensure it is not left floating.

When the 485Turbocache Module is removed from a system the CKEN# and BRDYO# signals, which are combined with external logic to form KEN# and BRDY#, will be left floating. All CKEN#, BRDYO#, START#, and PRSN# pins should have pullup resistors tied to them. This assures an inactive state when no 485Turbocache Module is present.

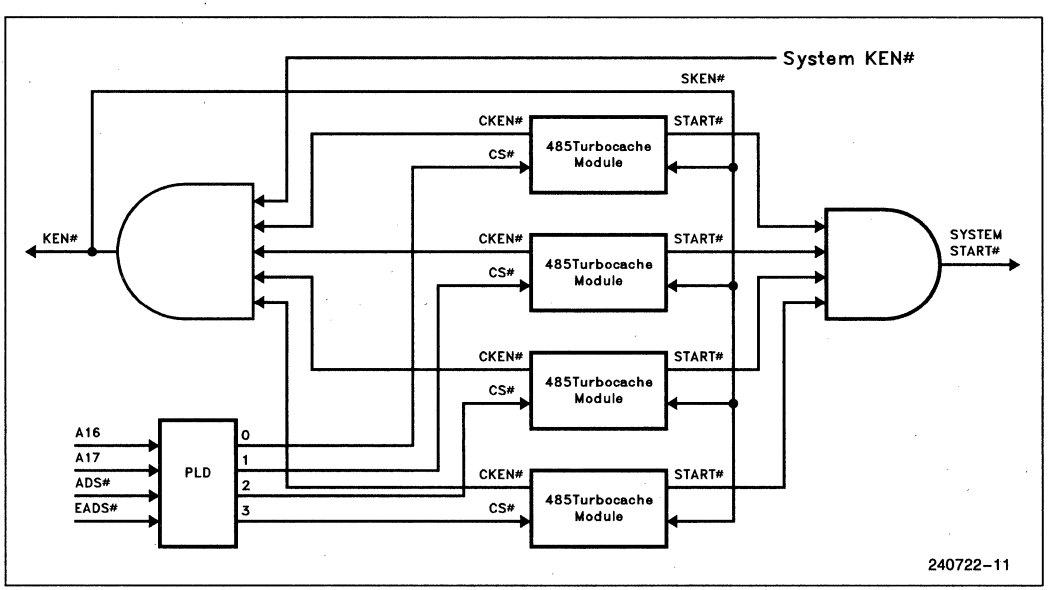


Figure 3.2 Multiple Cache Configuration

3.3.2 CONSIDERATIONS WITH MULTIPLE CACHES

As long as all the START#, CKEN#, BRDYO#, and PRSN# signals have pullup resistors tied to them, all empty cache sockets will respond like inactive caches. There is, however, a chip selecting problem since CS# decoding varies with the number of caches that are present.

Chip select decoding logic, like Figure 3.3 shows, should have all PRSN# pins as input. From this information, the correct chip select decoding can be generated. The logic in Figure 3.3 is able to keep CS1 asserted if one cache is detected, decode A16 if 2 caches are detected, or decode A16 and A17 if all 4 caches are present.

The most difficult problem to overcome when allowing an optional number of multiple caches is to account for capacitive load changes. Since each cache has a capacitive load on the data bus and clock lines, some amount of design effort must be spent resolving capacitive loading. When designing with 4 caches, each cache will probably have to receive a dedicated clock line. As well, the data bus will have to be buffered outside of the CPU and cache core.

4.0 OPERATIONAL/PERFORMANCE CONSIDERATIONS

The following sections provide more detailed information about operating and designing-in the 485Turbocache Module. This includes testing the cache, understanding sectoring, and making small performance adjustments.

4.1 Testing and Data Integrity

The 485Turbocache Module can monitor data integrity using parity bits. The i486 Microprocessor has the capability of outputting and checking data parity. The memory subsystem must also support parity to use the parity support on the 485Turbocache Module. This data parity information is stored with every byte inside the 485Turbocache Module, and is checked by the i486 CPU during data reads. To be able to identify data errors from memory or cache, the parity error check output (PCHK#) of the i486 CPU can be sampled.

Power up self test programs test main memory functionality on a cell by cell basis since parity logic is not capable of detecting all memory failures. It is also important to test cache memory. The following algorithm will test any number of 64k 485Turbocache Modules or 128k 485Turbocache Modules up to 512k of cache memory:

1. Flush or Reset the cache.
2. Write "1" to every bit of a 512k block of memory.
3. Read the 512k block twice; this fills the cache.
4. Disable CS# and write "0" to the 512k block; this fills memory.
5. Read the 512k block:
   - Repetitive assertions of START# indicate the cache boundary (size of cache)
   - Data $\neq$ 1 indicates bad tag or SRAM
6. Repeat with "0" in the cache and "1" in memory.

4.2 Sectored vs Non-Sectored Cache

The 64k 485Turbocache Module was designed as a 64k non-sectored cache; this means each tag of the cache points to 1 line of data in the cache memory. A 128k cache requires twice the number of tags to be non-sectored. This increases tag size, complexity, and reduces tag lookup speed. For this reason, the 128k 485Turbocache Module is a sectored cache. Each tag in the 128k 485Turbocache Module points to 2 consecutive lines in the cache. A Line Select bit, address bit A4, determines which line is being referenced.

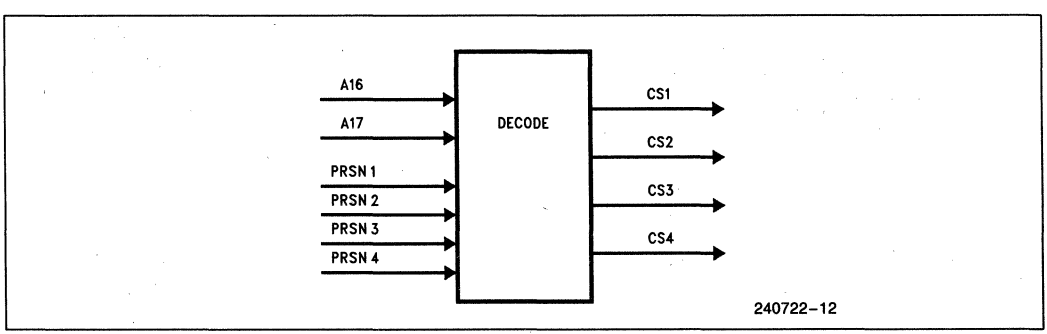


Figure 3.3. Chip Select Decoding

Figure 4.1 is an example of one tag in a sectored cache. If this tag points to address 2500h, then the adjacent line is reserved for address 2510h (A4 high). If, for example, address 2510 had been written first, the tag would still contain 25 and only address 2500 could be placed in the first line.

Since the Line Select Bit is used for a sectored architecture, all set and tag address bits are shifted higher in the address space. The 128k 485Turbocache Module internally compensates for this shift so pin-compatibility with the 64k 485Turbocache Module is maintained. This allows either cache configuration, 64k 485Turbocache Module or 128k 485Turbocache Module, to be hardware-transparent.

Because a sectored cache references 2 consecutive lines, the odds of filling both lines is reduced, and thus the hit rate of the cache. A sectored cache will have a slightly reduced hit rate compared to an equivalent non-sectored cache, but simulations have shown the performance penalties to be minimal (1 to 2 percent). Simulations have also shown that a two-way set associative sectored 128k cache offers significantly better performance than a direct mapped 128k non-sectored cache.

4.3 Performance Considerations

The following section offers a few special considerations that will increase cache performance or ease hardware design. These considerations are simply design notes and are not deviations from the i486 Microprocessor specification.

4.3.1 SKEN# ASSERTION

SKEN# is an input to the 485Turbocache Module to indicate the cacheability of a line during a read miss cycle. It is sampled exactly like KEN# in the i486 CPU, one clock before the first dword transfer of a line fill, and one clock before the last dword.

During a line fill, the 485Turbocache Module loads the dwords of the line directly into the appropriate spot in cache memory. This means that once SKEN# has been sampled active by the 485Turbocache Module, it must "commit" a line and invalidate a location to prepare for the incoming line. Once a line fill completes with a proper SKEN#, the line can be validated.

A potential performance loss exists if a system designer chooses, during non-cacheable cycles, to keep SKEN# active, but inactivate SKEN# the clock before the first transfer (see Figure 4.2). Once the 485Turbocache Module sees SKEN# low in the first T2, it commits a line in the cache by invalidating an entry despite the fact that SKEN# was later deasserted. The performance loss can be avoided if SKEN# was held inactive until cacheability could be determined.

4.3.2 INVALIDATION WINDOW

When an invalidation is requested with the assertion of EADS#, the 485Turbocache Module must immediately invalidate the address present on the address bus. If the tag portion of the 485Turbocache Module is in use, the invalidation takes priority and will suspend the other action. This may decrease performance. To avoid this, EADS# should not be issued in the second, third or fourth transfer of a cache read miss cycle. Section 1.3.4 Invalidation Cycles under Functional Description explains this in detail.

4.3.3 BOFF# ASSERTION

If BOFF# is asserted and the 485Turbocache Module is in the middle of a cacheable read miss cycle, the 485Turbocache Module treats the current line fill as non-cacheable. Once BOFF# is released and the cycle continues, the 485Turbocache Module will treat the rest of the cycle as a non-cacheable cycle.

In most systems BOFF# is a rare occurrence, thus the performance loss is negligible. If, however, BOFF# is regular and predictable, system performance can be increased by timing BOFF# so that the four dword transfers of a line fill are never interrupted. Section 1.3.5 BOFF# Cycles under Functional Description explains aborted cycles in more detail.

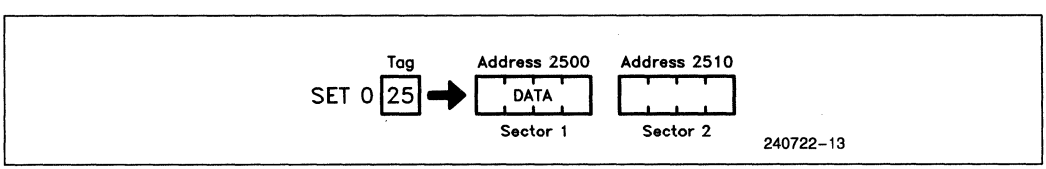


Figure 4.1 Sectored Example

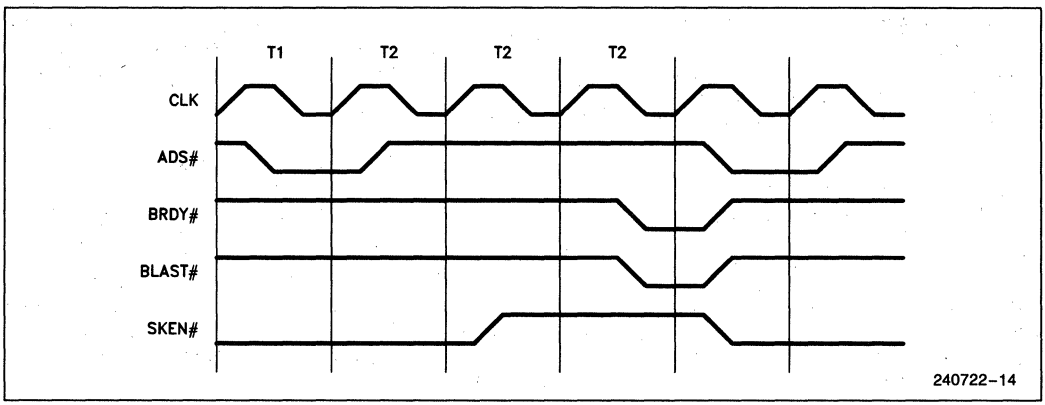


Figure 4.2 Method of SKEN# Generation Not Recommended

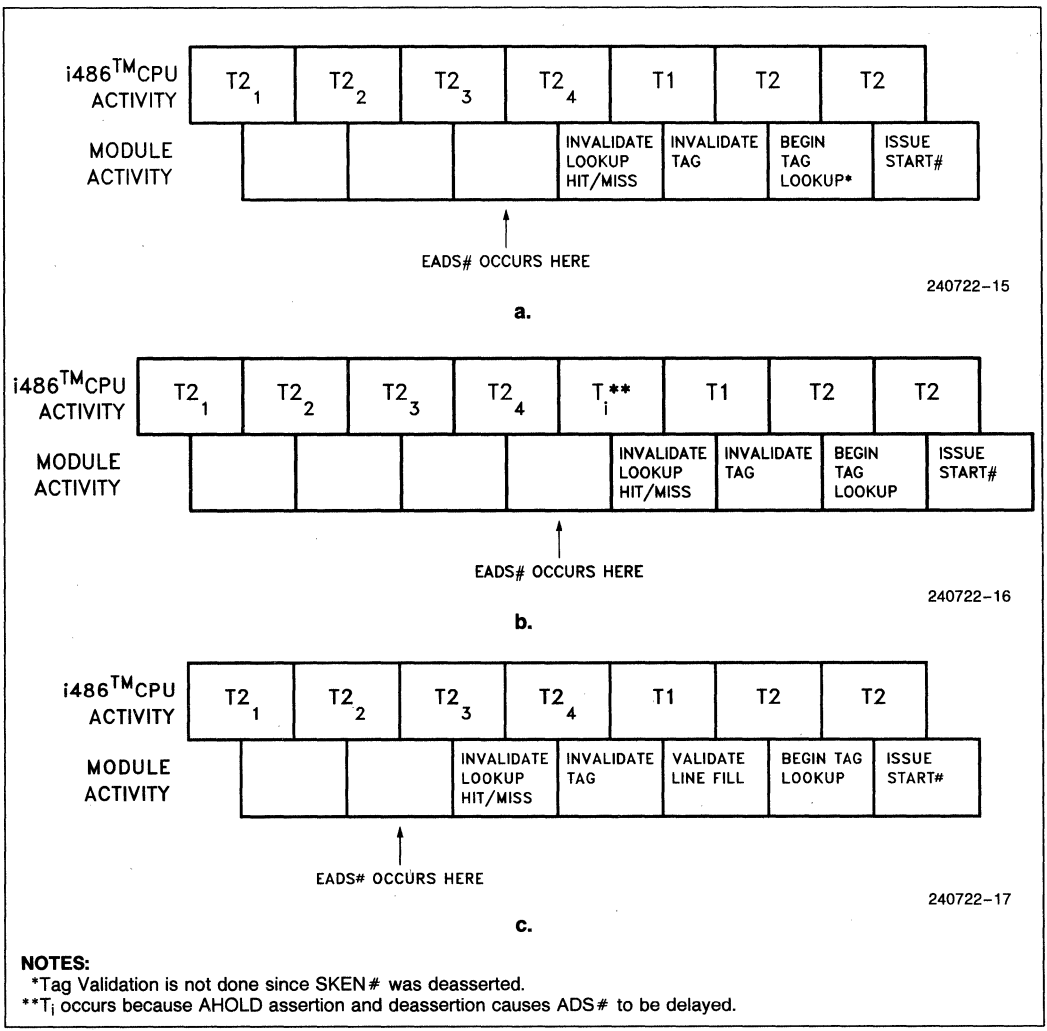
4.3.4 START# PREDICTABILITY

START# is asserted in the first T2 of a read miss cycle unless an invalidation occurred in the previous cycle. The section titled "Invalidation Cycles" explains why START# may be delayed. If START# must be a predictable signal to the system, and invalidation cycles cannot be timed to occur before the second transfer of a read miss cycle, there is a way to ensure the predictability of START#.

When EADS# is asserted towards the end of a read miss cycle, there are 3 tag accesses that need to be made before T1 of the next cycle: invalidate lookup, the actual invalidation (if a hit), and validation of the current line fill (if cacheable). Since there is no way to predict the hit/miss possibility of an invalidation request, it is assumed that 2 tag accesses will be required to service it. One tag access can be saved, then, by making the current line fill non-cacheable.

To do this, SKEN# to the 485Turbocache Module may be deasserted if AHOLD is detected. If SKEN# is deasserted the clock before the last CBRDY#, the line is non-cacheable. Figure 4.3a shows how assertion of EADS# during the third transfer of a burst cycle incurs a 1 clock delay in START#. Figure 4.3b shows EADS# assertion in the fourth clock, but since AHOLD will cause the CPU to delay ADS# at least one extra clock, START# is delayed only 1 clock as well. Assertion of EADS# in the second transfer of a burst causes a 1 clock delay in START# without deasserting SKEN# (see Figure 4.3c), so there is no advantage in dropping SKEN# for EADS# assertion then.

In summary, if SKEN# is deasserted in response to AHOLD during the third of fourth transfer of a line fill, START# will be delayed at most 1 clock. This makes START# predictable: It will always be valid in the second T2 of a read miss cycle. Note that if START# was not delayed, its value is retained in the second T2.

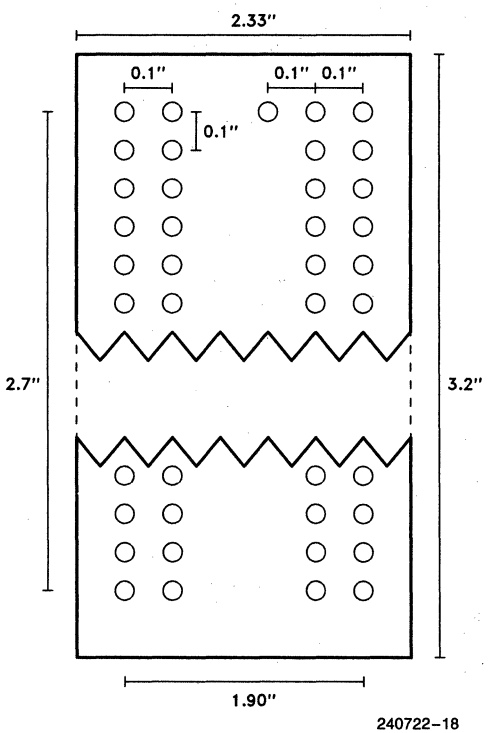


NOTES:
*Tag Validation is not done since SKEN# was deasserted.
**T_i occurs because AHOLD assertion and deassertion causes ADS# to be delayed.

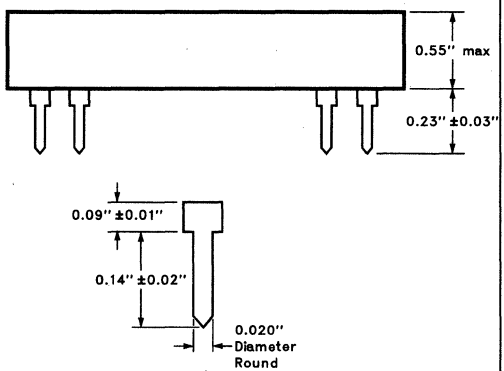
Figure 4.3 Predictable START# Delay

5

5.0 MECHANICAL SPECIFICATIONS



Pin Side View



Figures Not Drawn To Scale

6.0 ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature under Bias ....0°C to +70°C

Storage Temperature .....−55°C to +150°C

Voltage on Any Pin
with Respect to Ground ...−0.5V to V_{CC} + 0.5V

Power Dissipation:
64k 485Turbocache Module 4W
128k 485Turbocache Module 6W

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

7.0 D.C. CHARACTERISTICS (V_{CC} = 5V ±5%)

Symbol	Parameter	Min	Max	Unit	Notes
V_{IL}	Input Low Voltage	−0.3	+0.8	V	
V_{IH}	Input High Voltage	2.2	V_{CC} + 0.3	V	
V_{OL}	Output Low Voltage		0.4	V	1
V_{OH}	Output High Voltage	2.4		V	2
V_{CIL}	Clock Input Low Voltage	−0.3	+0.8	V	
V_{CIH}	Clock Input High Voltage	2.2	V_{CC} + 0.3	V	
I_{CC}	Supply Current Supply Current		800 1200	mA mA	82485MA 82485MB
I_{LI}	Input Leakage Current: D0–D31 and Parity D0–D31 and Parity CLK CLK TAI15, TAI16, WPSTRR# All Other Inputs		 ±20 ±40 ±55 ±75 −400 ±15	 µA µA µA µA µA µA	 82485MA 82485MB 82485MA 82485MB
I_{LO}	Output Leakage Current: D0–D31 and Parity D0–D31 and Parity		 ±20 ±40	 µA µA	 82485MA 82485MB
C_{IN}	Input Capacitance: —A2 Through A31 —BE0 Through BE3 —ADS#, M/IO#, W/R# —CRDY#, CBRDY# —D0 Through D31 —D0 Through D31		 20 20 20 20 25 45	 pF pF pF pF pF pF	 82485MA 82485MB
C_{CLK}	Clock Input Capacitance Clock Input Capacitance		45 75	pF pF	82485MA 82485MB
C_O	I/O Capacitance		25	pF	

NOTES:
1. Measured at 4.5 mA.
2. Measured at 1.0 mA.

5

8.0 A.C. CHARACTERISTICS (V_{CC} = 5V ±5%)

All A.C timings are tested with a capacitive load of 50 pF unless otherwise specified.

Symbol	Parameter	25 MHz		33 MHz		Fig.	Notes
		Min(ns)	Max(ns)	Min(ns)	Max(ns)		
t_1	CLK Period	40	42	30	42	9.1	
t_2	CLK High Time	14		11		9.1	1
t_3	CLK Low Time	14		11		9.1	2
t_4	CLK Fall Time		4		3	9.1	
t_5	CLK Rise Time		4		3	9.1	
t_6	A2–A31, BE0#–BE3# Setup Non-Snoop	17		13 *		9.3	
t_{6a}	A2–A31, BE0#–BE3# Hold Non-Snoop	3		3		9.3	
t_7	ADS#, M/IO#, W/R# Setup	17		13		9.2	
t_{7a}	ADS#, M/IO#, W/R# Hold	3		3		9.2	
t_8	BLAST# Setup	10		9		9.2	
t_{8a}	BLAST# Hold	3		3		9.2	
t_9	CRDY#, CBRDY# Setup	8		5		9.2	
t_{10}	CRDY#, CBRDY# Hold	3		3		9.2	
t_{11}	SKEN# Setup	8		5		9.2	
t_{12}	SKEN# Hold	3		3		9.2	
t_{13}	D0–D31, DP0–DP3 Setup	5		5		9.2	
t_{14}	D0–D31, DP0–DP3 Hold	3		3		9.2	
t_{15}	WP Setup	8		8		9.2	3
t_{16}	WP Hold *	3		3		9.2	
t_{17}	BOFF# Setup	10		8			
t_{18}	BOFF# Hold	3		3			
t_{19}	EADS# Setup	8		5		9.3	4
t_{20}	EADS# Hold	3		3		9.3	
t_{21}	A4–A31 Setup (Snoop)	5		5		9.3	
t_{22}	A4–A31 Hold (Snoop)	3		3		9.3	
t_{23}	RESET, FLUSH# Setup	10		5		9.4	
t_{24}	RESET, FLUSH# Hold	3		3		9.4	
t_{25}	RESET, FLUSH# Pulse Width (Asynchronous Use)	60		45			

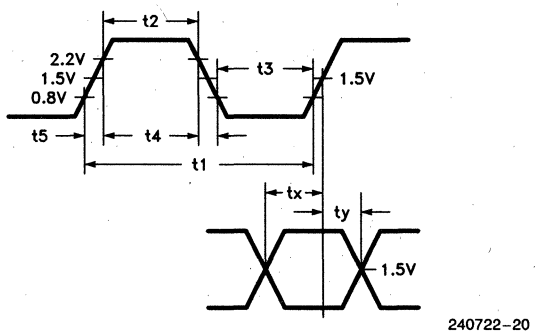
8.0 A.C. CHARACTERISTICS (V_{CC} = 5V ±5%) (Continued)

All A.C timings are tested with a capacitive load of 50 pF unless otherwise specified.

Symbol	Parameter	25 MHz		33 MHz		Fig.	Notes
		Min(ns)	Max(ns)	Min(ns)	Max(ns)		
t_{26}	BRDYO# Valid		22		16	9.3	
t_{27a}	CKEN# Valid	3	18	3	15	9.3	
t_{27b}	CKEN# Hold		12		12	9.3	5
t_{28}	START# Valid	5	22	5	16	9.2	
t_{29}	D0–D31 Valid (Read Hit)		30		24	9.3	6
t_{30}	CS# Setup	6		6		9.2	7
t_{31}	CS# Hold	3		3		9.2	

NOTES:
1. At 2.2V.
2. At 0.8V.
3. Setup to CLK edge of third BRDY# in line fill.
4. Setup to CLK edge where EADS# is valid.
5. Hold time from CLK edge in which CKEN# will be sampled.
6. Valid up to C_L = 100 pF.
7. At the clock edge in which ADS# or EADS# is sampled.

9.0 WAVEFORMS



tx = input setup times
ty = input hold times, output float, valid and hold times

Figure 9.1. CLK Waveforms

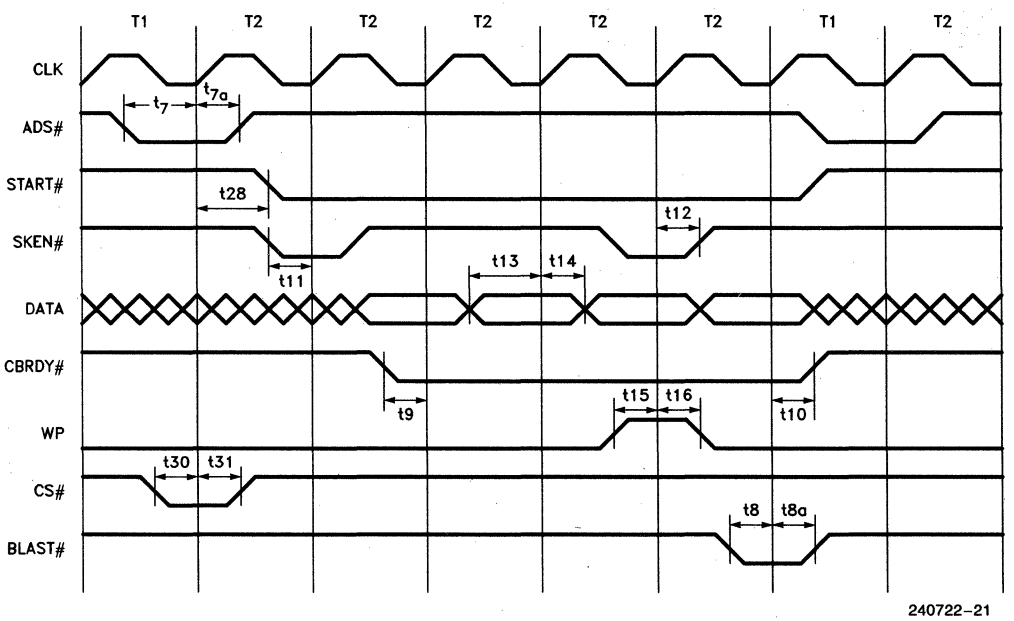
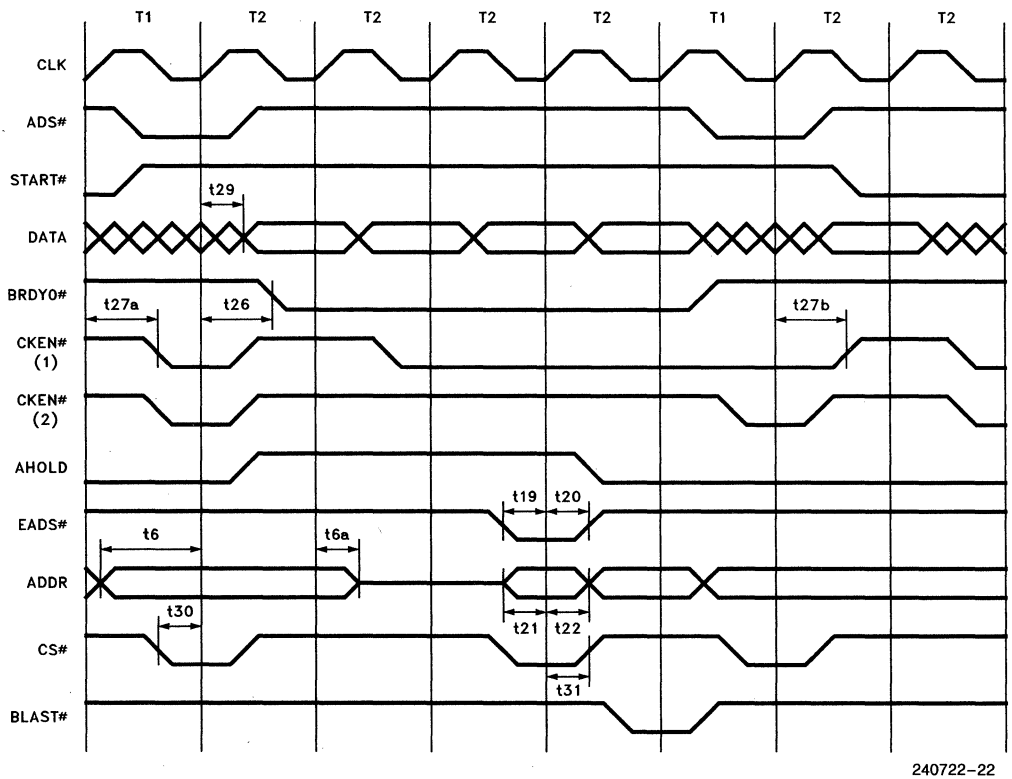


Figure 9.2. Write Protected Read Miss



1. Normal CKEN# behavior.
2. CKEN# behavior if line is Write Protected and WPSTRP# is low.

Figure 9.3. Read Hit Cycle and Write Cycle with Invalidation

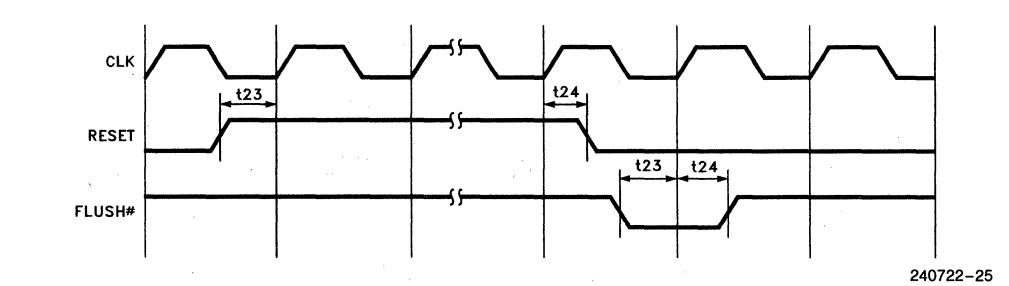


Figure 9.4. RESET and FLUSH #

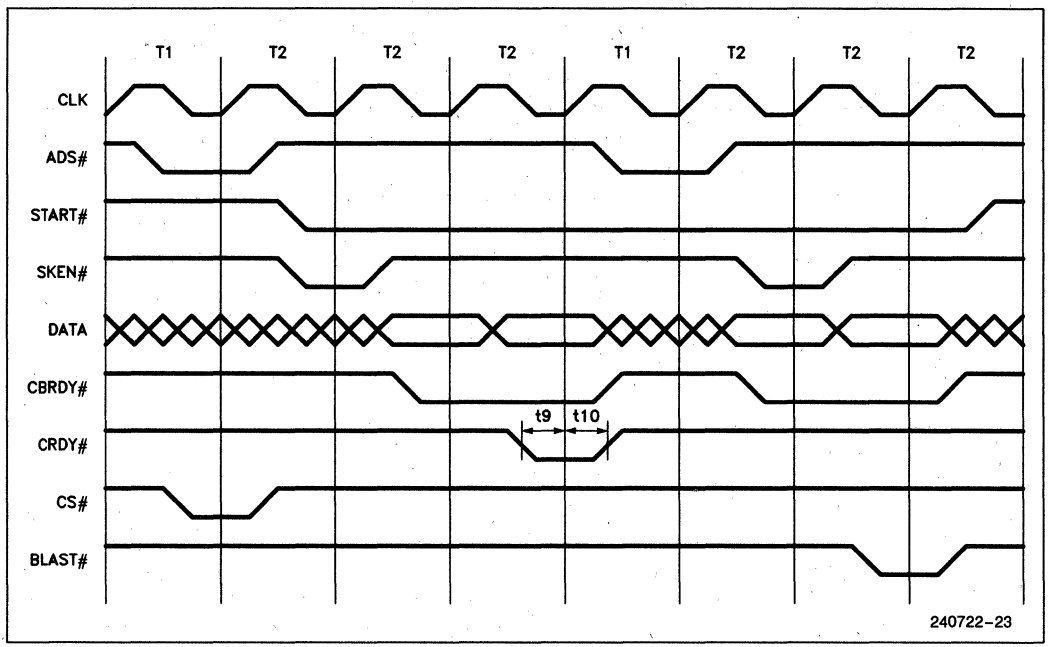


Figure 9.5. Multiple Cycle Line Fill

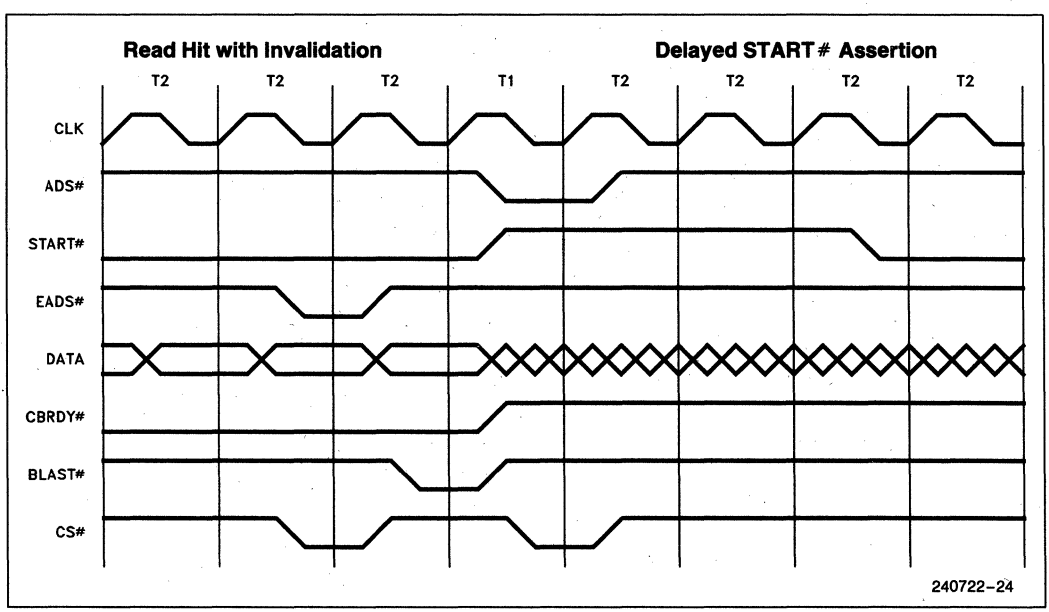


Figure 9.6. Invalidation Causing Delayed START#

10.0 REVISION HISTORY

Revision -002 of the 485Turbocache Module Data Sheet contains several updates and corrections to the original version. A revision summary of major changes is listed below:

Throughout Document — The name of the cache module has been changed from Turbocache 486 Module to 485Turbocache Module.

Section 1.3.1 — Clarified that all transfers seen by the 485Turbocache Module are assumed to be 32-bit transfers.

Section 1.4 — Removed one incompatibility between the 485Turbocache and the i486 CPU.

Section 2.1.9 — Corrected RESET specifications.

Section 5.0 — Made mechanical specifications more precise.

Section 6.0 — Modified absolute maximum ratings.

Section 7.0 — Modified V_{IH} and V_{OL} specifications. Added input and output leakage current specifications.

Section 8.0 — Corrected AC specifications t_7 and t_{25}. Added AC specifications t_{6a} and t_{7a}.

82485
SECOND LEVEL CACHE CONTROLLER FOR THE i486™ MICROPROCESSOR

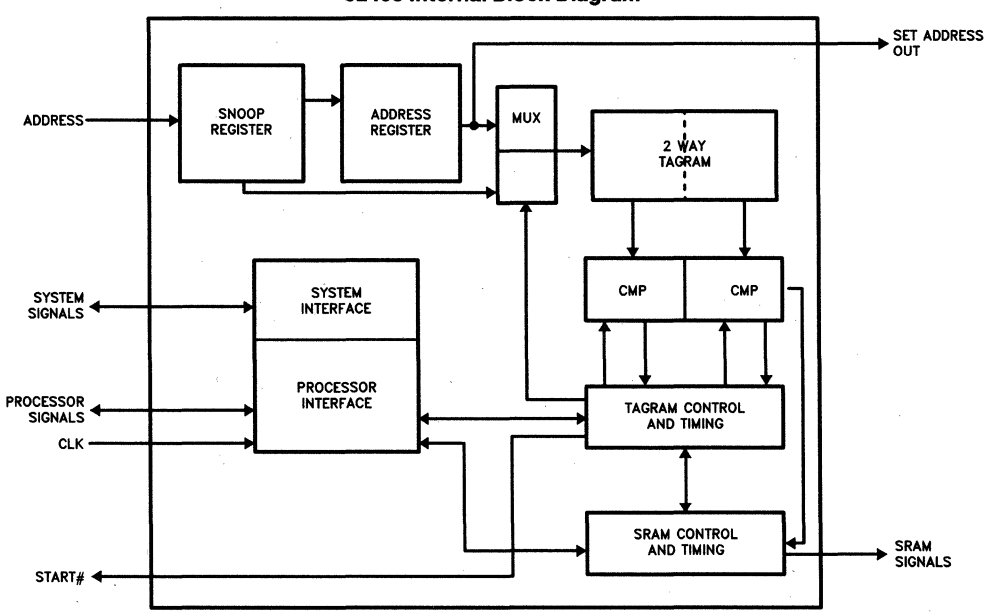
- **High Performance**
  - **— Zero Wait State Access on Cache Hit**
  - **— One Clock Bursting**
  - **— Two-Way Set Associative**
  - **— Write Protect Attribute Per Tag**
  - **— Start Memory Cycles in Parallel**
- **Easy to Use**
  - **— Matches i486™ Microprocessor Bus Timing**
  - **— Supports Invalidation Cycles**
  - **— Maintains Memory on Writes**
- **High Integration**
  - **— Single Chip Tag RAM and Controller**
  - **— No Logic Needed for CPU and Cache Connection**
  - **— Maps Full 4 Gigabyte Address Space**
- **Flexible System Configurations**
  - **— Supports 64K or 128K Cache Memory Per Controller**
  - **— Allows Multiple Controllers for Larger Cache Size**
  - **— Supports Non-Cacheable Memory Areas**

The 82485 is a second-level cache controller designed to improve the performance of i486™ Microprocessor systems. One 82485 cache controller supports 64K or 128K bytes of second level cache memory that maps to the entire 4 Gigabytes of the i486 microprocessor address space. The controller is completely software transparent. Several controllers may be cascaded to provide larger cache sizes. One controller plus SRAMs provides a 64K or a 128K cache. External EPROM can be cached yet remain write protected. The 82485 is fully compatible with the i486 microprocessor. All i486 CPU bus cycles and timings are supported.

A complete, optional second level cache controller using the 82485 is available as the 485Turbocache Module from Intel (data sheet order number 240722).

i486 is a trademark of Intel Corporation.

82485 Internal Block Diagram



For the complete data sheet on this device, contact Intel's Literature Distribution Dept., (800) 548-4725.

240831-1

November 1990
Order Number: 240831-001