486 Green PC VESA/ISA/PCI Chipset

SiS 85C496/497

Preliminary

Rev 3.0

July 1995


REDUCE

Part I

1. SiS85C496/497 Overview

SiS85C496 ***PCI & CPU Memory Controller (PCM)***
SiS85C497 ***AT Bus Controller & Megacell (ATM)***

The SiS 486-VIP (VESA/ISA/PCI) chips are two-chip solution ideally for Intel's 80486, SL Enhanced 486, P24D/P24T/DX4 CPU, AMD's 486, Enhanced Am486 and Cyrix's Cx486 (M7)/Cx 5x86 CPU based on green AT system. By supporting the most popular industrial standard system interfaces, it provides flexible configurations for system design and applications.

The SiS85C496 PCI & CPU Memory Controller (PCM) integrates the Host Bridge (Host Interface), the cache and main memory DRAM Controller, the PCI Bridge, the built-in IDE Controller, and the FS-Link Bus (Fast Slow Link Bus). It provides the address paths and bus control for transfers among the Host (CPU/L1 cache), main memory (L2 cache and DRAM), the Peripheral Component Interconnect (PCI) Bus, and the FS-Link Bus. The L2 cache controller supports both write-through and write-back cache policies and cache sizes up to 1 MBytes. The cache memory can be built using standard asynchronous SRAMs. The main memory DRAM controller interfaces DRAM to the Host Bus, PCI Bus, and FS-Link Bus. Up to eight single sided SIMMs or four double sided SIMMs provide a maximum of 255 MBytes of main memory. The installation of DRAM SIMMs is "Table-Free", which allows the SIMMs be installed into any slot location and any combinations. The built-in IDE hard disk controller allows CPU accessing hard disk and also provides higher system integration with lower system cost. The 85C496 is intended to be used with the SiS85C497 which is a AT Bus Controller with built-in 206 controller.

The SiS85C497 AT Bus Controller and Megacells (ATM) provides the interface between PCI/CPU/Memory Bus (fast machine) and the ISA Bus (slow machine). It also integrates many of the common I/O functions in today's ISA based PC systems. The 85C497 comprises the FS-Link interface (Fast-Slow Link interface), ISA bus controller, DMA controller and data buffers to isolate the FS-Link Bus from the ISA Bus and to enhance performance. It also integrates a 14 channel edge/level interrupt controller, refresh controller, a 8-bit BIOS timer, three programmable timer/counters, non-maskable-interrupt (NMI) control logic, Power Management Unit, and RTC. Figure 1.1 shows the system block diagram.

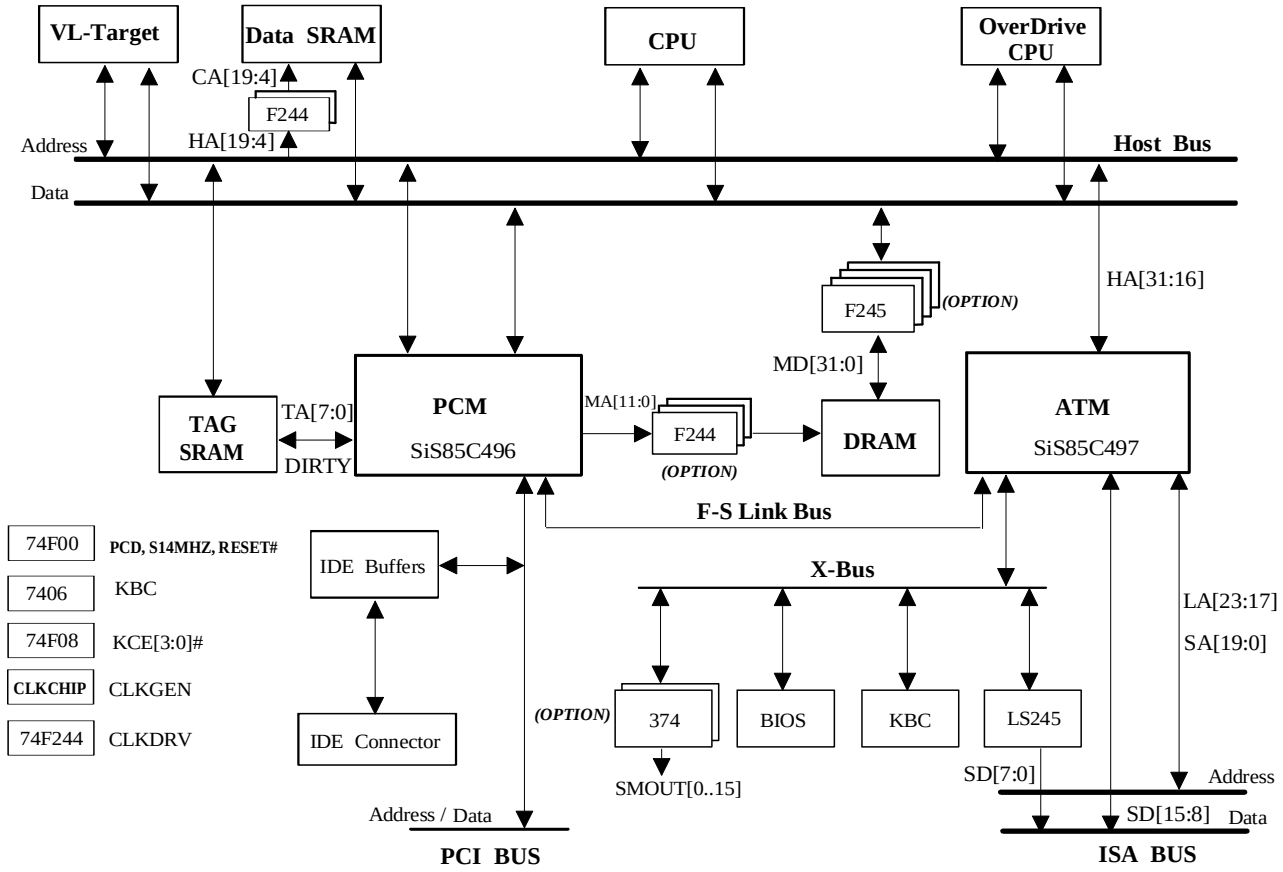


Figure 1.1 SiS85C496/497 System Block Diagram

Part II

1. SiS85C496 PCI & CPU Memory Controller

1.1 Features

- **Host Bus**
  - Supports Intel 486, P24D, P24T, DX4, SL-Enhanced 486, AMD 486, Enhanced Am486, and Cyrix M7/Cx 5x86 in 25/33/40/50 Mhz, 5V CPU.
- **VESA Bus Slave**
  - Supports VESA Bus Specification Rev. 2.0p with Local Device Target only.
- **PCI Local Bus**
  - Supports PCI Bus Specification Rev. 2.0 with up to 4 PCI Masters.
  - Implements 3 Level Post Write Buffer for CPU write PCI Target Memory Cycle.
  - Supports Back to Back Single Memory Write to PCI Burst Write.
  - Supports PCI Interrupt Steering with Four PIRQ Inputs.
  - Supports PCI Master Burst Accesses On-Board Memory Up to 64 Double Word Long.
  - Supports Concurrency PCI Bus.
  - Snoop Filter and Advanced Snooping for Reducing CPU Snoops During Sequential PCI Master Accesses On-Board Memory Cycles.
  - Supports PCI Bus PCI to PCI Bridge.
- **Supports L1 Cache Write Back CPU (P24T/P24D/M7/Enhanced Am486) systems**
- **Supports Cx 5x86 Linear Burst Order Mode.**
- **L2 Cache Controller**
  - Write-Back or Write-Through Schemes
  - Bank Interleave/Non-Interleave Cache Access
  - Cache Size: 64K/128K/256K/512K/1MB
  - 8 bit or 7 bit Tag (Combined Tag and Dirty SRAM) with Direct-Mapped cache organization.
  - Optional Separate Dirty SRAM.
- **DRAM Controller**
  - Supports 8 Banks Non-Interleaved Access for Single and Double Sided SIMMs up to 255 MBytes.
  - Supports DRAM CAS Before RAS Refresh.
  - Supports "Table-Free" DRAM configuration.
  - Programmable driving current for the DRAM signals.
  - Supports Symmetrical and Asymmetrical DRAMs.
  - Supports 256K/512K/1M/2M/4M/8M/16M/32M xN Fast Page Mode and EDO DRAM.
- **Built-In Local Bus IDE Interface**
  - Supports Data Conversion for the Double Word Accessing
  - Supports Symmetry Configuration for Channel 1 and Channel 0, PIO Mode IDE Hard Disks.
  - Supports Mode 3 and above Timing.
  - Supports Individual Drive Timing Setting for Optimal Performance.
  - Supports Posted Write Buffers and Pre-fetch Buffer.
  - Supports Primary IDE or Secondary IDE Addressing (1Fx/17x)

- **Fast-Slow Link Interface**
  - Linkage to ISA Bridge by FS-Link Interface.
  - Fast Access to BIOS, ISA Memory Holes, and Interrupt Acknowledge Cycle by FS-Link.
  - Two Programmable Non-Cacheable Regions
  - Two Programmable PCI Memory Holes and One Programmable ISA Memory Holes.
- **208-Pin PQFP**
- **0.6μm Low Power CMOS Technology**

1.2 Architectural Overview

The 85C496 provides six basic functions: Host Interface, Cache controller, DRAM controller, PCI Bridge, IDE interface and FS-Link interface. This section describes an overview of these functions.

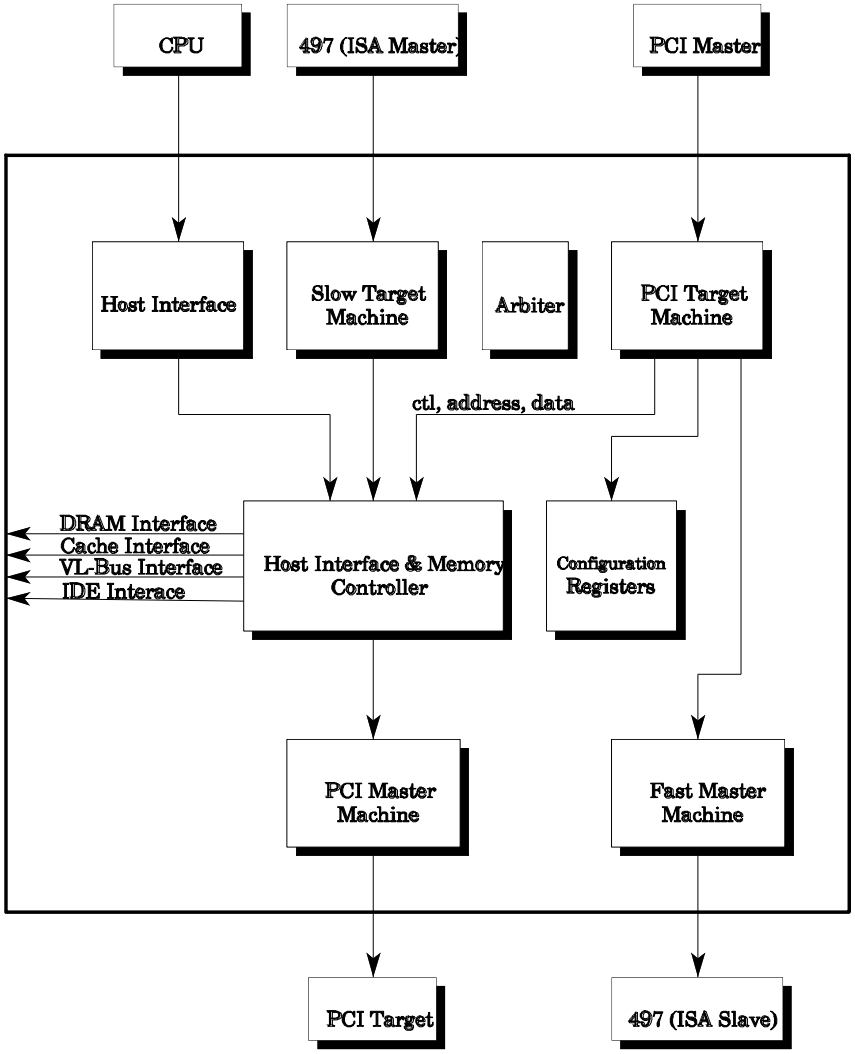


Figure 1.2 Architectural Overview Diagram

1.2.1 Host Interface Operations

The Host Interface provides an interface among the Bus masters (CPU, PCI master, and ISA master), the main memory (L2 cache and DRAM), and the I/O devices (VL-Slave, PCI target, and ISA slave).

1.2.2 Cache Operations

The secondary cache controller in 85C496 supports direct mapped cache organization. The implementation of the level two (L2) cache is done by using the standard SRAMs. Both cache write-back and write-through policy are supported. The SRAMs is external to the 85C496 and located on the Host address and data bus.

The structure of the level two cache uses same cache line size (four double words or 16 bytes) as the level one cache inside Intel 486 compatible CPUs. With direct mapped cache organization, 85C496 supports 4K, 8K, 16K, 32K, or 64K cache lines to implement 64K, 128K, 256K, 512K or 1M Bytes of L2 cache. Associated with each line of cache data, there are address tag field and a dirty bit to control the access of L2 cache. Cache tag field is used for address comparison to determine whether L2 cache contains the data for the access. While the dirty bit is used in conjunction with write-back policy to determine the status of the data. The 85C496 supports both 7-bit and 8-bit wide address tag field. Together with one dirty bit, The 85C496 provides three different combinations of cache implementation to meet different system integration requirements.

During a memory read or write operation, the 85C496 searches the L2 cache first before forwarding the access to the main memory. For the I/O read or write operations, L2 cache is not involved since I/O operations are not cacheable.

The operation of the secondary cache is dependent on the cache policy and can be programmed by Register 42h~43h bit 8.

Write-Through

The basic concept of cache write-through policy is that the contents of main memory is always consistant with those in the L2 cache. During memory read operations, the 85C496 searches the L2 cache first. If the addressed data is located in the L2 cache (cache hit) then data is return directly from the L2 cache. If the addressed data is not located in the L2 cache (cache miss) then the access is forwarded to main memory. When data is read from the main memory, L2 cache is also being updated (cache line fill). For memory write operations, the 85C496 searches the L2 cache first, if the addressed memory location is in the L2 cache then data is written both to the L2 cache and the main memory to maintain data consistancy. If the addressed data is not located in L2 cache then only main memory is written with that data, and no operation is performed on L2 cache (no write allocate).

Write-Back

The basic concept of cache write-back policy is that the data in the L2 cache and the main memory is updated only whenever is necessary. During a memory write operation, the 85C496 searches the L2 cache first. If the addressed memory location is in L2 cache then data is written to L2 cache only but not to the main memory. If the addressed memory location is not in L2 cache then only main memory is written with that data, no operation is done to L2 cache (no write allocate). For a memory read operation, the 85C496 searches the L2 cache first. If the addressed data locates in the L2 cache (cache hit) then data is return directly from L2 cache. If the addressed data does not locate in L2 cache (cache miss) then the access is forwarded to main memory. When data is return from main memory, L2 cache will also be updated (cache line fill). However, if the cache line fill operation is to a dirty line (the Dirty bit associated with cache line is set), the dirty line is first written back to main memory before the new line is brought into the L2 cache. For a dirty line write-back operation, the 85C496 first performs a read from the dirty cache line and then writes the data to main memory.

In addition to supporting burst reads for the cache line fills of the CPU, the 85C496 is capable of accepting burst write data of the CPU's internal cache dirty line(s) during CPU write-back cycles. The support of the CPU burst write cycle is optional through the control of the Configuration Registers in the 85C496.

1.2.3 DRAM Memory Operations

The 85C496 can support 8 rows of DRAM, and memory size from 1 MBytes up to 255 MBytes. Each bank could be single or double sided 32 bits Fast Page Mode or Extended Data-Out DRAM. The installed DRAM type can be 256K/512K x 36, 1M/2M x 36 or 4M/8M/16M/32M x 36 SIMMs.

1.2.4 PCI Bridge Operations

PCI bridge communicates to the host interface and FS-link interface for cycle translation and redirection.

- **PCI bridge to HOST interface communication**

The PCI bridge provides an interface between bus masters (CPU and ISA master) and the PCI bus. A PCI cycle is generated by the PCI bridge if a CPU or ISA master- initiated cycle is not targeted to VL-slave, cache, or on-board memory. The PCI cycle is completed if PCI target claims the cycle and returns the TRDY#. One exception is master abort that occurs when none of the PCI targets claim the PCI cycle in a certain amount of time. In this case the PCI bridge returns all ones on the data bus to the host bridge then completes the cycle.

For PCI master accessing the cache or on-board memory, PCI bridge requests Host bridge for the memory cycle. The memory access can be single or burst transfers.

If the address of the undergoing cycle is within ISA memory area, the PCI bridge redirects the cycle to ISA bus via FS-link interface. All these information are passed from the PCI bridge target machine via host interface to the PCI bridge master machine then to the FS-link interface.

- **PCI bridge to FS-link communication**

When a PCI cycle is targeted at the ISA slave, the PCI bridge redirects the cycle to ISA bus via FS-link machine. The FS-link cycle can be initiated only through the PCI bridge. The PCI bridge redirects all the unclaimed PCI I/O cycle with address under 64k to ISA bus. It also redirects all the unclaimed memory cycles with address within ISA memory hole or ROM area to ISA bus.

- **Inside PCI bridge**

PCI bridge incorporates a 3-level post write buffer for PCI memory write cycle. When CPU writes PCI memory, host interface will receive ready from bridge before the data is actually written to the target. In this way, CPU access cache or on-board memory cycle can be executed concurrently with PCI memory write cycle. Note that only PCI memory write cycle can be posted in to the write buffer. I/O cycles, configure cycles and ISA memory cycles are not posted.

1.2.5 FS-Link Operations

85C496 implements two link machines, fast master machine and slow target machine. They provide the same purposes as fast target machine and slow master machine in 85C497. Resources with higher bandwidth such as CPU, cache, main memory and PCI bus are considered as fast interface. While ISA resources which has lower bandwidth are considered as slow interface. The link machines in both chips provide the necessary interface control between fast interface and slow interface. When CPU master or PCI master wants to access ISA slave, the fast master machine is requested to transfer bus cycle information to the 85C497 fast link target machine. The 85C497 fast link target machine then converts cycle to AT bus controller for the ISA bus access. For ISA master, on the other hand, when it wants to access resource such as main memory, the AT bus controller requests the slow link master machine for the FS-link bus, then the 85C497 slow link master machine converts the ISA cycle to the 85C496 slow link target machine for that access.

2. Detailed Functional Description

There is the functional description of the 85C496 interfaces and functional blocks in this section which includes the following topics: CPU/Cache Interface, DRAM Control Interface, PCI Interface, FS-Link Interface, IDE Controller, Clocks and Reset Logic.

2.1 CPU/Cache Interface

There are some essential topics about the interface between CPU and Cache discussed in this section.

- L2 cache Controller
- The cache policy configuration between the L1 and L2 caches.
- Snoop protocol

2.1.1 L2 Cache Controller

85C496 provides control for level 2 cache ranging in size from 64K, 128K, 256K, 512K up to 1M byte, which uses asynchronous data SRAM of one bank or two, tag width can be either 7 or 8 bits, and dirty bit is optional too. 85C496's secondary-level cache protocol do not need an extra SRAM for the valid bit.

Second level cache supported by 85C496 has line size of 16 bytes or 4 double words with direct map organization, cache lines are updated at reads, but not on writes. Caching protocol can be configured to be either write through or write back.

During CPU, PCI, ISA master or DMA accesses, data is read from the level 2 cache on a cache hit. Data is written to cache only on a cache hit when the level 2 cache is configured in write back mode. When the level 2 cache is configured in write through mode, data will be written to both cache and DRAM on cache hits.

Only the data in the main memory can be allocated in the level 1 and level 2 cache, memory on either VESA bus, PCI bus or ISA bus will not be cached.

The portion of on-board DRAM that's above the cacheable DRAM size will not be cached in the level 2 cache, however, software resides in this area will not be suffered because 85C496 still caches data in this location to the CPU level 1 cache.

For example; if 10M bytes of DRAM, 64K bytes of cache with 7 bit tags is installed. Then the lower 8M DRAM will be cached in both level 1 and level 2 cache, and the upper 2M DRAM which is not cacheable in the level 2 cache will still be cached in the level 1 cache.

Shadow cacheable

UMB (Upper Memory Block) which is not in main memory area is by default non-cacheable. However, once shadowed, the RAM segments 0C0000-0FFFFFh can be further cached in Level 2 and Level 1 cache. By setting Register 46h, each 32K memory segments can be individually enabled to be cacheable by both 85C496's secondary level cache and CPU's internal cache.

System designer should be aware that for Intel P24D and P24T CPU operating in write back mode, shadow cacheable areas are not write protected in CPU's internal cache because once the data line being filled in CPU's level 1 cache, there will be no observable bus cycle outside the CPU while CPU write on these cache lines. The line by line write-back / write-through mode (WB/WT# pin) is not supported by the 85C496. Therefore programmers must make sure that software do not write on these locations unless they really meant to change it's contents. For Cyrix Cx486DX, Cx486DX2 these areas can still be write protected by setting the WT1 bit in CPU's CCR2 register.

Non-cacheable area

Three non-cacheable areas can be defined by setting register 50h through 55h. Only the main memory can be cached, memory on the extension bus is by default non-cacheable. The non-cacheable areas are programmed by specifying the starting address and the size of that area, each holes can be sized as either 64k, 128k, 256k, 512k, 1M, 2M or 4M.

Cache Sizing

To perform the level 2 cache auto sizing, the bit 1 of register 43-42 should be set to force level 2 cache always hit (bit1 = 1). After the cache size is detected, the bit 1 of register 43-42 should then set to 0 for the normal operation.

Cache initialization

85C496 do not need a valid bit SRAM for the level 2 cache. To do so it assumes all the cache line in the level 2 cache are valid. Therefore level 2 cache must be initialized with accurate data from main board memory before it can be enabled. To initialize level 2 cache, the bit 1 of register 43-42 should be set to normal operation mode (bit1 = 0). Since the address tag field is at unknown state, the address generated for memory read (line fill) can be a cache hit, therefore it is necessary to read of a block of main memory data equal to twice the size of level 2 cache. Then cache is updated and cache initialization is completed.

Cache timing

The Level 2 cache data SRAM can be either 1 bank or 2 banks, in two bank configuration accesses are interleaved. The bit 4 of Register 43-42 controls whether cache accesses are interleaved.

Depending on the system clock rate, and the speed of the data/tag/dirty bit SRAM being used, CPU access level 2 cache can be completed in either zero or one wait state. Therefore single read or write takes either 2 clocks or 3 clocks, and burst read or write takes either 2-1-1-1, 2-2-2-2, 3-1-1-1, or 3-2-2-2 clocks.

The bit 9 of Register 43-42 selects the lead off cycle to be either zero or one wait state, the bit 10 of register 43-42 selects the subsequent three read cycles to be either zero or one wait state, and the bit 11 of register 43-42 selects the subsequent three write cycles to be either zero (with this option, the bits[1:0] of register 40: DRAM speed should be 'Faster' or 'Fastest') or one wait state. The bit 11 of register 43-42 also selects the single write cycles to be either zero or one wait state.

Note that the bit 9 of register 43-42 also defines the lead off cycle wait state for DRAM cycles. Moreover burst writes is only applicable with CPUs that supports burst write and the bits[6:5] of register 40 has being enabled for 85C496 to respond accordingly.

- **Cache Line Description**

The internal cache of the 80486 has a 16-byte line size. When a read miss occurs in the internal cache, the 80486 initiates off-chip memory read cycles to update the current cache line. The 80486 reads in 16-byte blocks (4 doublewords). To increase the bus throughput, the 80486 provides a burst mode transfer. Four doublewords can be read sequentially in 5 processor clocks (2-1-1-1).

The second level cache provided by the SiS85C496 also has a 16-byte line size. It supports the 80486 burst read cycles to do the fastest cache line fill. When the 80486 internal and external caches encounter a read miss, they are simultaneously updated with the data line read from DRAM.

In addition to supporting burst reads for the cache line fills of the CPU, the 85C496 is capable of handling burst writes from the CPU's internal cache dirty line write back cycles. The CPU burst write support is optional by programming the Configuration Registers. The 85C496 supports the cache size up to 1MB and the DRAM size up to 255MB.

The address bits used to access cache lines depends on the size of the cache as follows:

Table 2.1 Cache Line Selection

Cache size	Tag field (8 bit tag)	Tag field (7 bit tag)	Cache line selection
64K byte	A[23:16]	A[22:16]	A[15:4]
128K byte	A[24:17]	A[23:17]	A[16:4]
256K byte	A[25:18]	A[24:18]	A[17:4]
512K byte	A[26:19]	A[25:19]	A[18:4]
1M byte	A[27:20]	A[26:20]	A[19:4]

The 85C496 provides cache data and tag control signals for a two banks, interleaved cache implemented using standard asynchronous SRAMs. Figure 2.5 shows the block diagram of the CPU-L2 Cache connections.

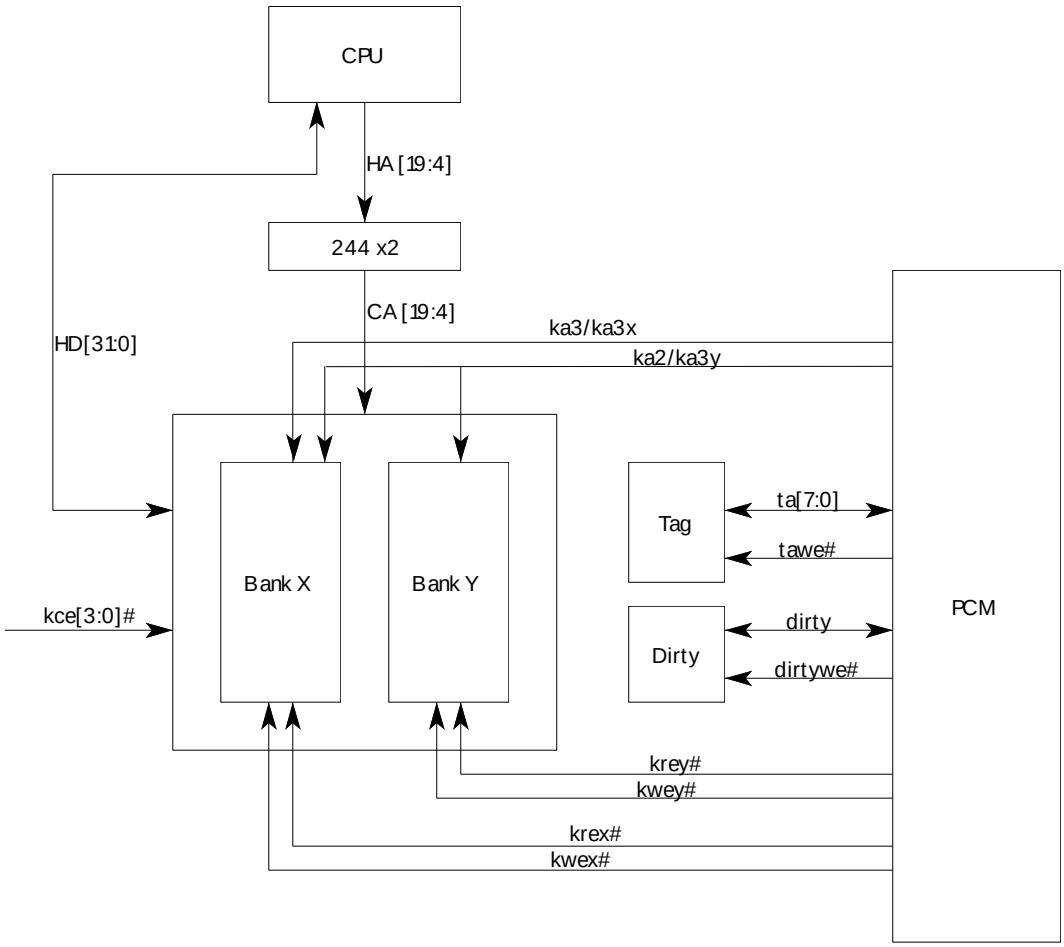


Figure 2.5 CPU-L2 Cache Connections

L2 Cache Size Options

Table 2.2 gives the information about the size option of L2 Cache and SRAM components. This table also shows Tag and Dirty SRAM requirements. The Tag and Dirty bits may be combined in a proper speeded 8-bit SRAM.

Table 2.2 L2 Cache Size Options

Cache size	Data SRAM	Tag SRAM	Dirty SRAM
64K byte	8 (8K x 8)	1 (4K x 8/7)	1 (4K x 1/0)
128K byte	4 (32K x 8)	1 (8K x 8/7)	1 (8K x 1/0)
256K byte	8 (32K x 8)	1 (16K x 8/7)	1 (16K x 1/0)
512K byte	4 (128K x 8)	1 (32K x 8/7)	1 (32K x 1/0)
1M byte	8 (128K x 8)	1 (64K x 8/7)	1 (64K x 1/0)

Table 2.3 shows the cacheable DRAM size with different L2 cache settings. The cacheable DRAM size is determined by cache size as well as the tag address field width. The on-board memory beyond the cacheable size is not cacheable by the L2 cache, however it is still cacheable by the 80486 internal cache.

Table 2.3 Cacheable Size Options

Cache size	Cacheable DRAM size (8 bit tag)	Cacheable DRAM size (7 bit tag)
64K byte	16M byte	8M byte
128K byte	32M byte	16M byte
256K byte	64M byte,	32M byte
512K byte	128M byte	64M byte
1M byte	255M byte	128M byte

L2 Cache Performance Options

Table 2.4 gives the information about the performance options of the L2 cache and DRAM.

Table 2.4 L2 Cache / DRAM Performance

Fast Page Mode DRAM

Cycle Definition	Total Clocks at Host Interface			
Host-to-Memory	**25 MHz**	**33 MHz**	**40 MHz**	**50 MHz**
L2 read hit, non-burst	2	2	3	3
L2 read hit, burst	2-1-1-1	2-1-1-1	3-2-2-2	3-2-2-2
L2 write hit, non-burst	2	2	3	3
L2 write hit, burst	2-1-1-1	2-1-1-1	3-2-2-2	N/A
DRAM read, non-burst, page-hit	3	4	5	6
DRAM read, non-burst, row-miss	5	6	7	8
DRAM read, non-burst, page-miss	7	8	10	12
DRAM read, burst, page-hit	3-2-2-2	4-3-3-3	5-3-3-3	6-4-4-4
DRAM read, burst, row-miss	5-2-2-2	6-3-3-3	7-3-3-3	8-4-4-4
DRAM read, burst, page-miss	7-2-2-2	8-3-3-3	10-3-3-3	12-4-4-4
DRAM write, non-burst, page-hit	3	4	5	5
DRAM write, non-burst, page-start	5	6	7	7
DRAM write, non-burst, page-miss	7	8	10	11
DRAM write, burst, page-hit	3-2-2-2	4-3-3-3	5-3-3-3	N/A
DRAM write, burst, page-start	5-2-2-2	6-3-3-3	7-3-3-3	N/A
DRAM write, burst, page-miss	7-2-2-2	8-3-3-3	10-3-3-3	N/A

EDO DRAM

Cycle Definition	Total Clocks at Host Interface			
Host-to-Memory	**25 MHz**	**33 MHz**	**40 MHz**	**50 MHz**
L2 read hit, non-burst	2	2	3	3
L2 read hit, burst	2-1-1-1	2-1-1-1	3-2-2-2	3-2-2-2
L2 write hit, non-burst	2	2	3	3
L2 write hit, burst	2-1-1-1	2-1-1-1	3-2-2-2	N/A
DRAM read, non-burst, page-hit	3	4	5	5
DRAM read, non-burst, row-miss	5	6	7	7
DRAM read, non-burst, page-miss	7	8	10	11
DRAM read, burst, page-hit	3-2-2-2	4-2-2-2	5-2-2-2	5-2-2-2
DRAM read, burst, row-miss	5-2-2-2	6-2-2-2	7-2-2-2	7-2-2-2
DRAM read, burst, page-miss	7-2-2-2	8-2-2-2	10-2-2-2	11-2-2-2
DRAM write, non-burst, page-hit	3	4	5	5
DRAM write, non-burst, page-start	5	6	7	7
DRAM write, non-burst, page-miss	7	8	10	11
DRAM write, burst, page-hit	3-2-2-2	4-3-3-3	5-3-3-3	N/A
DRAM write, burst, page-start	5-2-2-2	6-3-3-3	7-3-3-3	N/A
DRAM write, burst, page-miss	7-2-2-2	8-3-3-3	10-3-3-3	N/A

NOTE*:* Burst write function is applicable only with L1 W/B CPU.

Cycle Definition	Total Clocks at Host Interface			
PCI-to-Memory	**25 MHz**	**33 MHz**	**40 MHz**	**50 MHz**
L2 read hit, non-burst	6 T 9 B	6 T 9 B	6 T 7 B	6 T 7 B
L2 read hit, burst	6-4-4-4... T 9-6-4-4... B	6-4-4-4... T 9-6-4-4... B	6-4-4-4... T 7-5-4-4... B	6-4-4-4... T 7-5-4-4... B
L2 read miss, non-burst	7 T 10 B	8 T 11 B	7 T 8 B	7 T 9 B
L2 read miss, burst	7-5-5-5... T 10-7-5-5... B	8-6-6-6... T 11-8-6-6... B	7-5-5-5... T 8-6-5-5... B	7-5-5-5... T 9-7-5-5... B
L2 write hit, non-burst	6 T 9 B	6 T 9 B	6 T 7	6 T 7
L2 write hit, burst	6-4-4-4... T 9-6-4-4... B	6-4-4-4... T 9-6-4-4... B	6-4-4-4... T 7-5-4-4... B	6-4-4-4... T 7-5-4-4... B
L2 read miss, non-burst	7 T 10 B	8 T 11 B	7 T 8 B	7 T 9 B
L2 write miss, burst	7-5-5-5... T 10-7-5-5... B	8-6-6-6... T 11-8-6-6... B	7-5-5-5... T 8-6-5-5... B	7-5-5-5... T 9-6-5-5... B

NOTE : The above timing is with reference to the PCI clock.

T : CPU L1 cache in Write Through Mode.

B : CPU L1 cache in Write Back Mode.

Those following figures show the Cache read cycle and Cache write cycle.

L2 Cache Fill Without Dirty Line Replacement

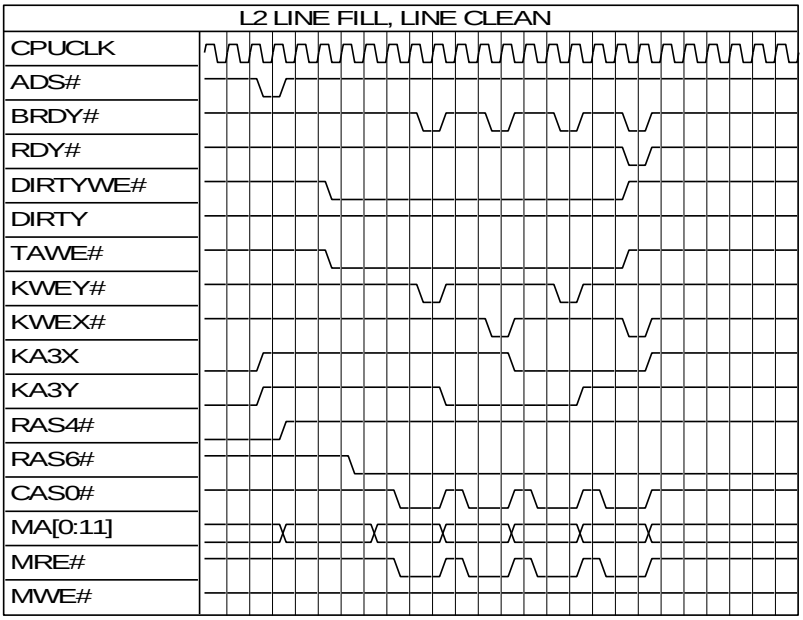


Figure 2.6 Level 2 Cache Line Fill with no Dirty

L2 Cache Line Fill with Dirty Line Replacement

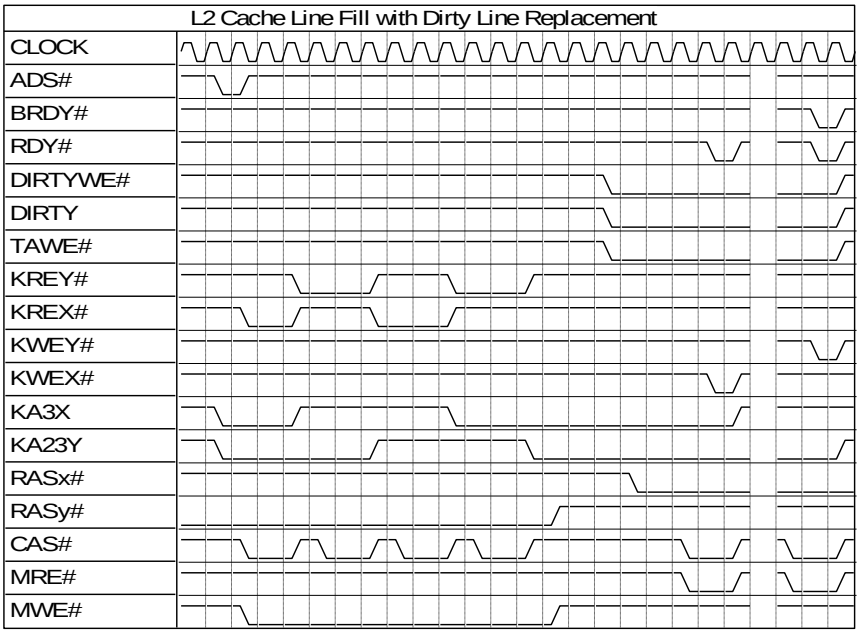


Figure 2.7 Level 2 Cache Line Fill with Dirty Line Replacement

L2 Cache Read Hit, Burst Transfer in Write Through or Write Back Mode

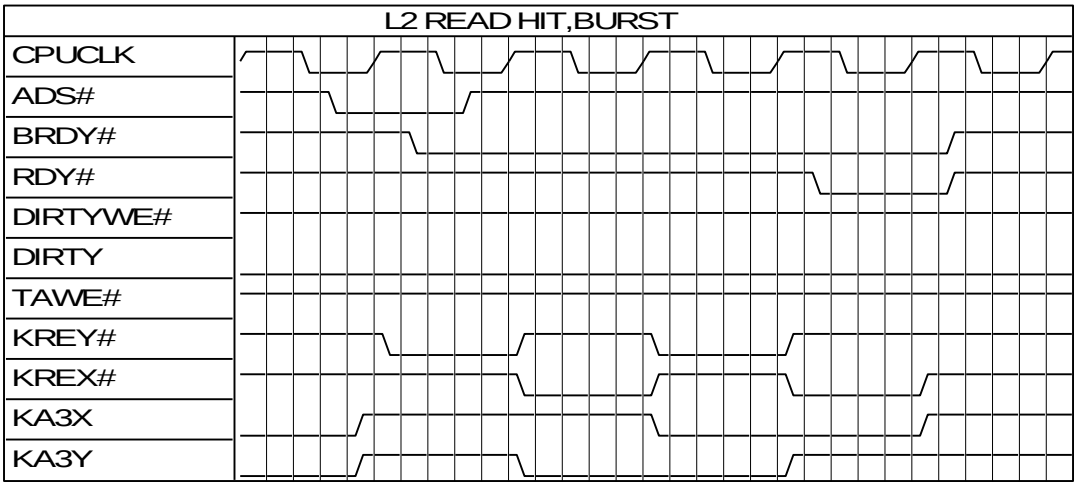


Figure 2.8 L2 Cache Read Hit, Burst Transfer in Write Through or Write Back Mode

L2 Cache Write Hit, Non-Burst Transfer in Write Back Mode

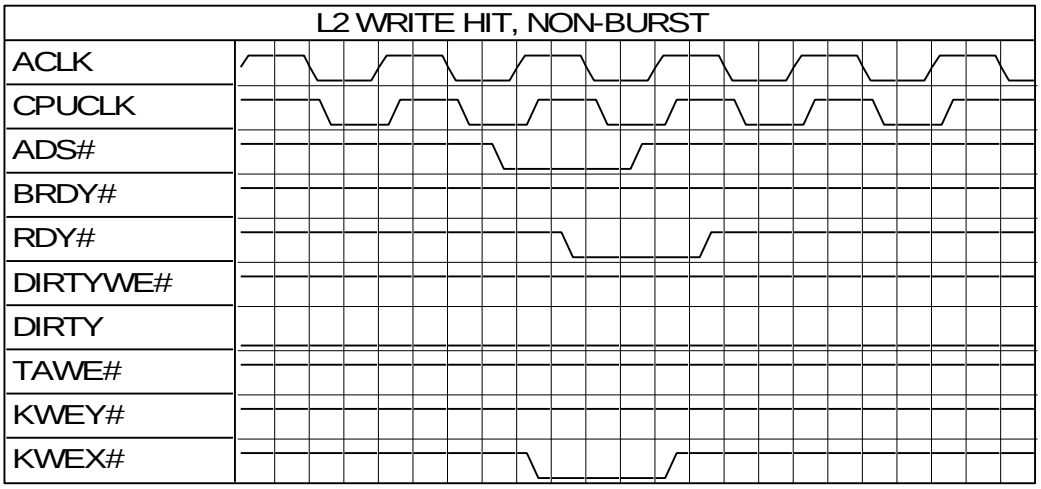


Figure 2.9 L2 Cache Write Hit, Non-Burst Transfer in Write Back Mode

L2 Cache Write Hit, Burst Transfer in Write Back Mode

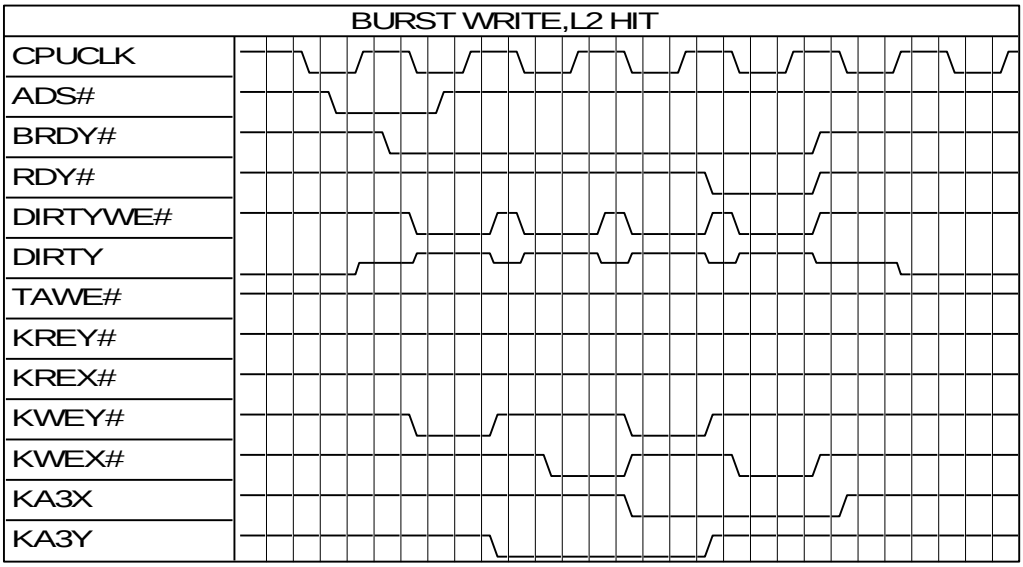


Figure 2.10 L2 Cache Write Hit, Burst Transfer in Write Back Mode

2.1.2 L1/L2 Cache Mode Configurations

The 85C496 supports the L1 and L2 caches with write-through (WT)/write-back (WB) mode. Both caches can be in the write-back mode in a given configuration. Table 2.5 shows the five configurations supported by the 85C496. The 85C496 can also enable/disable L1 (System BIOS disable L1 cache via CPU Control Register) or L2 cache (System BIOS disable L2 cache via Cache Configure Register 43-42 bit 0) individually.

Table 2.5 L2 Cache Configurations

Processor Type	L1 Cache Mode	L2 Cache Mode
486SX/DX/DX2/DX4/SL-Enhanced	WT	WT/WB
P24D/P24T	WT/WB	WT/WB
M7/Cx 5x86	WT/WB	WT/WB
Am486 DX4/DX2/DX/SX	WT	WT/WB
Enhanced Am486 DX4/DX2	WT/WB	WT/WB

2.1.3 Snoop Operations

The Snoop operation by 85C496 ensures data consistency between the L1 cache, L2 cache and main memory.Once the Snoop is enabled by 85C496, it monitors PCI/ISA Bus accesses to main memory. The type of PCI/ISA access (read or write) and the caching modes(write-through or write-back) of the L1 and L2 caches determines the snoop operation be taken. All the caching modes are shown below:

Table 2.6 Snoop Protocol

Type of Access	L1:WT, L2:WT	L1:WT, L2:WB	L1:WB, L2:WT	L1:WB, L2:WB
PCI/ISA Master Read of Main Memory	**No Snoop.** Data in L1 and L2 is consistent with data in main memory.	**Snoop L2.** L2 is Snooped for dirty data. If L2 line is dirty, then the 85C496 writes cache line back to main memory before permitting PCI/ISA Master to read memory. L2 line is marked as clean (Dirty = 0). No L1 action is taken.	**Snoop L1.** L1 is Snooped for dirty data. If L1 line is dirty, then the CPU writes cache line back to main memory before permitting PCI/ISA Master to read memory. During write-back, data is also written to L2.	**Snoop both L1 and L2.** Both L1 and L2 caches are Snooped for dirty data.
PCI/ISA Master Write to Main Memory	**Snoop both L1 and L2.** Invalidate L1 and overwrite L2 on Snoop hit.	**Snoop both L1 and L2.** Invalidate L1 and overwrite L2 on Snoop hit.	**Snoop both L1 and L2.** Invalidate L1 and overwrite L2 on Snoop hit. If L1 Snoop hits a dirty line, then the CPU writes cache line back to main memory before permitting PCI /ISA to write to main memory.	**Snoop both L1 and L2.** Invalidate L1 and overwrite L2 on Snoop hit. If L1 Snoop hits a dirty line, then the CPU writes cache line back to main me-mory before per-mitting PCI /ISA to write to main memory.

Advanced Snooping and Snoop Filter

The 85C496 integrates an Advanced Snooping mechanism that eliminates the main memory access delay due to Snoop cycles. During a PCI burst cycle, the 85C496 performs Snoop cycle in advance before the actual burst cycle is generated. The snoop filter is used so that the advanced snooping is performed in very 4 Double words.

2.2 DRAM Interface

The size of DRAM supported by 85C496 range from 1M to 255M bytes, UMB (upper memory block) is excluded from the main memory, and can either be used to shadow BIOS or extension card ROM, or be appended to the top of main memory or remapped for use as SMRAM.

Three memory holes can also be used to extract portions of DRAM from main memory. All non-main memory accesses will first be presented to VESA bus, then forwarded to PCI bus, and only accesses to memory hole 2 (ISA memory hole) will be further forward to ISA bus. Memory hole do not necessary have to be configured within the range of main memory. But it is illegal to install any VESA, PCI or ISA bus memory overlapped with main memory.

The 85C496 DRAM interface is designed to support DRAM with FP/EDO mode with CAS# before RAS# refresh. DRAM type can be 256K/512K by 32/36 bits, 1M/2M by 32/36 bits, 4M/8M/16M/32M by 32/36 bits or any combination of the above.

85C496 has multiple share function pins to meet individual system requirement. MA11 is shared with RAS7 at pin 157, and shared with DIRTY or fourth PCI master request at pin 126. If any 16M by 32/36 bits DRAM is installed, then either the DRAM on row 7 will not be accessible, or the cache interface should be implemented without the dirty bit (always dirty), or fourth PCI master is not supported. More detailed information is in section 2.8.

2.2.1 DRAM Organization

The 85C496 DRAM Interface is organized into eight rows (RAS#[7:0]), with each row consisting of one 36-bit bank (non-interleaved, 32 bits data and 4 bits data parity). The DRAM organization is shown in Figure 2.11. Each row is four bytes wide (CAS#[3:0]).

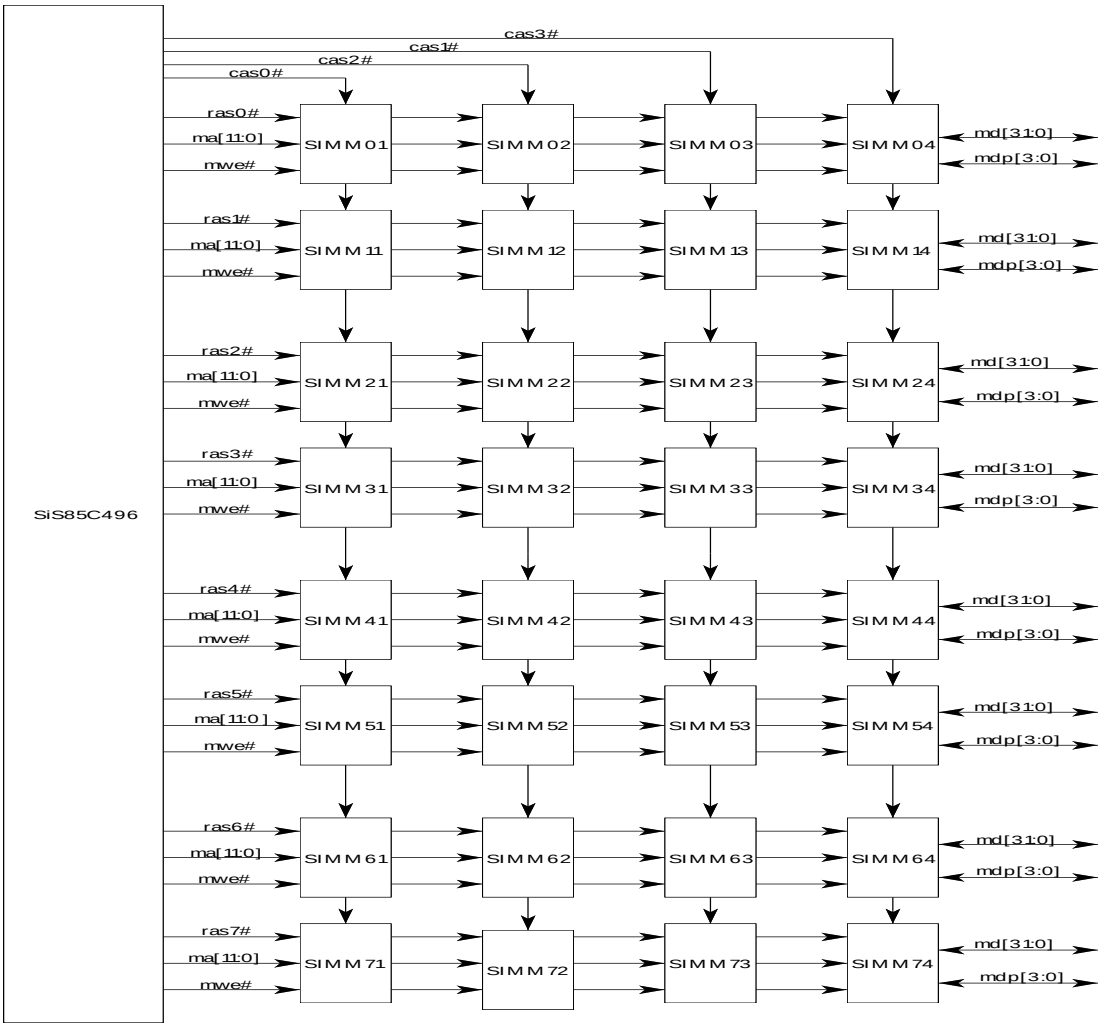


Figure 2.11 DRAM Organization Supporting Single & Double Bank SIMMs

2.2.2 DRAM Size Configuration

85C496 supports 8 rows of DRAMs each 32 or 36 bits wide. Register 4Fh to 48h defines the address boundary of these 8 rows of memory. The 8 rows of DRAMs may be implemented in 8 banks of single-sided SIMMs, 4 banks of double-sided SIMMs or any other combinations as required. Access to the rows are not interleaved and need not to be populated starting from row 0 or in consecutive sequence.

The DRAM may be installed with 256K/512K by 32/36 bits, 1M/2M by 32/36 bits, 4M/8M/16M/32M by 32/36 bits type. Since these register came up with initial value of 00h after power up, 85C496 do not recognize any DRAM at that time. To configure the DRAM size registers, software routine must first set each row to its maximum capacity and attempt to write, read a specific pattern in order to determine whether DRAM is actually installed in that slot, and if installed, what is the size and type of the DRAM. After going through each rows in the same manner, the 8 address boundaries to be written in the registers can be calculated by adding up the number of megas of DRAM between each bank and bank 0. For example, if after DRAM sizing routine, it's found that row 2, 3, 5 is populated with 16M, 1M, 4M of

DRAM respectively, then the row boundary registers should be programmed in the following manner:

Table 2.7 DRAM Auto Detection

	DRAM Size	Row Boundary Register
Row 7	0 M	21 (15h)
Row 6	0 M	21 (15h)
Row 5	4 M	21 (15h)
Row 4	0 M	17 (11h)
Row 3	1 M	17 (11h)
Row 2	16 M	16 (10h)
Row 1	0 M	0 (0h)
Row 0	0 M	0 (0h)

Shadow RAM

Memory location 0C0000h-0FFFFFh in standard PC system are reserved for use by system BIOS and extension cards, 85C496 provides a shadow mechanism that enables the contents of these ROMs in that area be duplicated to the same location in main memory that can be accessed by CPU in very little wait states. Therefore overall system performance will be greatly improved because low speed ROM accesses are eliminated.

85C496 implements read and write access attributes to control the access of the shadow RAM areas. To use shadow RAM, first register 44h - 45h have to be programmed to enable the areas to be shadowed, while setting shadow RAM as read disable and write enable, read then write the same address will copy the contents of the ROMs into their corresponding locations into main memory. After the whole memory segment being transport to DRAM, shadowing can be enabled by setting shadow RAM as read enable and write disable. Any further reads to those locations will be accessing main memory and writes will be directed to the ROMs.

PCI master, ISA master and DMA is not permitted to access 0C0000h-0FFFFFh unless the bit 10 of register 44-45h is enabled. Once enabled, the read enable, write protect settings in the bits[8:9] of register 44-45h still control the read/write attribute of that area.

Memory hole

Since 85C496 can handles on-board main memory up to 255M bytes, any memory resides on extension cards that locates on the location overlapped with main memory will not be accessed by CPU. To solve this problem 85C496 provides three memory holes that can be programmed by register 50h through 55h to prevent those areas from interpreted as main memory. The memory holes are programmed by specifying the starting address and the size of that hole, each holes can be sized as either 64K, 128K, 256K, 512K, 1M, 2M or 4M.

Relocate

Again in standard PC system, UMB (upper memory block) in main memory is not used by the system. To recover the unused DRAM, 85C496 relocates memory segment 0A0000-0BFFFFh and 0D0000-0EFFFFh to the top of DRAM installed. To facilitate the relocation feature, if total DRAM installed is 8M or less, memory segment D and E is not shadowed, and SMRAM

relocation is not enabled, 256K byte of DRAM will be appended to the top of DRAM after enabling relocation by setting the bit 0 of register 47.

SMRAM

SMRAM may be located at either 060000-06FFFFh or 0E0000-0EFFFFh, and can be physically mapped to 0A0000-0AFFFFh or 0B0000-0BFFFFh. The DRAM relocation feature is not applicable if SMRAM is utilized.

The bits[4:3] of register 5A selects the SMRAM base address and to which location it relocates. The bit 2 of register 5A forces remapping even when CPU is not in SMI mode so that SMI routines can be initialize at that area. Contrarily, the bit 1 of register 5A forces relocation ineffective during SMI mode to enable the un-remapped data at that address be accessed. SMRAM area should not be cacheable. To use logical address of 6XXXX segment or Exxxx segment, it is BIOS engineer's responsibility to make sure that these addresses are set to non-cacheable. While accessing SMRAM, the contents will then not be cached by either level 1 or level 2 cache, therefore it is not necessary to flush the contents of the cache upon existing SMI mode.

Another possible way of implementing SMRAM is by shadowing segment 0E0000-0EFFFFh and use it directly as SMRAM, therefore it is no need for relocation, and the contents will be accessible regardless whether it is in SMI mode or not.

DRAM timing

85C496 is designed with very flexible DRAM control timing, basic timings are selected by the bits[1:0] of register 40, while more detail timings are individually adjustable (all 'clocks' listed below refers to CPUCLK).

RAS# precharge time:

The bits[1:0] of register 40 selects the RAS# precharge time to be 3 clocks at 'Fastest' and 'Faster', 4 clocks at 'Slower' and 5 clocks at 'Slowest'.

RAS# to CAS# delay:

The bit 3 of register 41 selects RAS# to CAS# delay to be either 2 clocks or 3 clocks.

CAS# precharge time:

The bit 0 of register 41 selects CAS# precharge time to be either 1 clock or 2 clocks.

These setting should be selected to meet DRAM minimum timing requirements for proper operation.

FP/EDO DRAM Write with Non-Burst Transfer and Page start

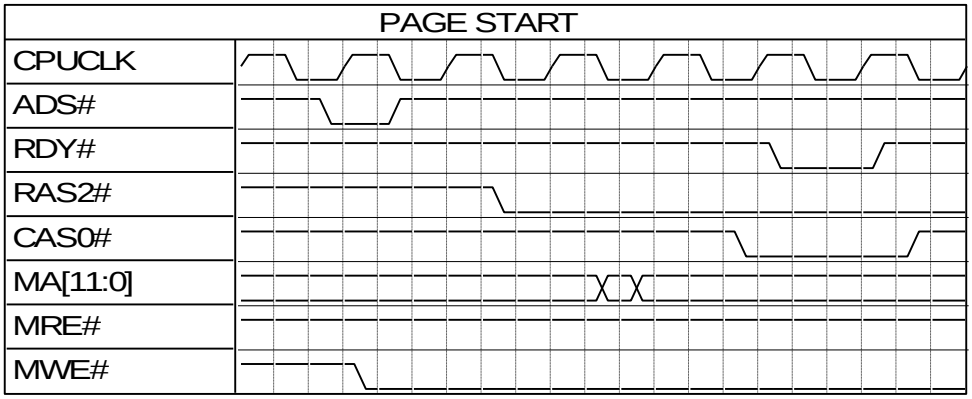


Figure 2.12 FP DRAM Write with Non-Burst Transfer and Page start

FP/EDO DRAM Read with Non-Burst Transfer and Page Miss

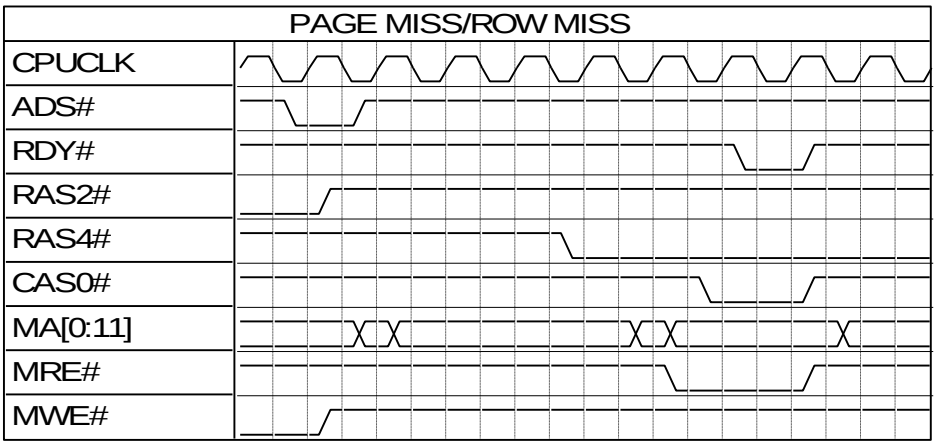


Figure 2.13 FP DRAM Read with Non-Burst Transfer and Page Miss

FP DRAM Read with Burst Transfer and Row Miss

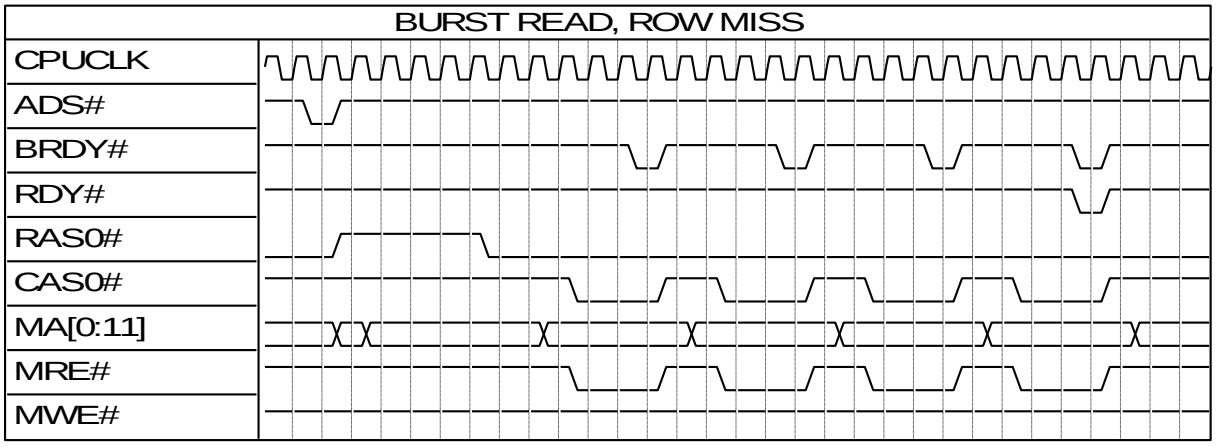


Figure 2.14 FP DRAM Read with Burst Transfer and Row Miss

EDO DRAM Read with Burst Transfer and Page Miss

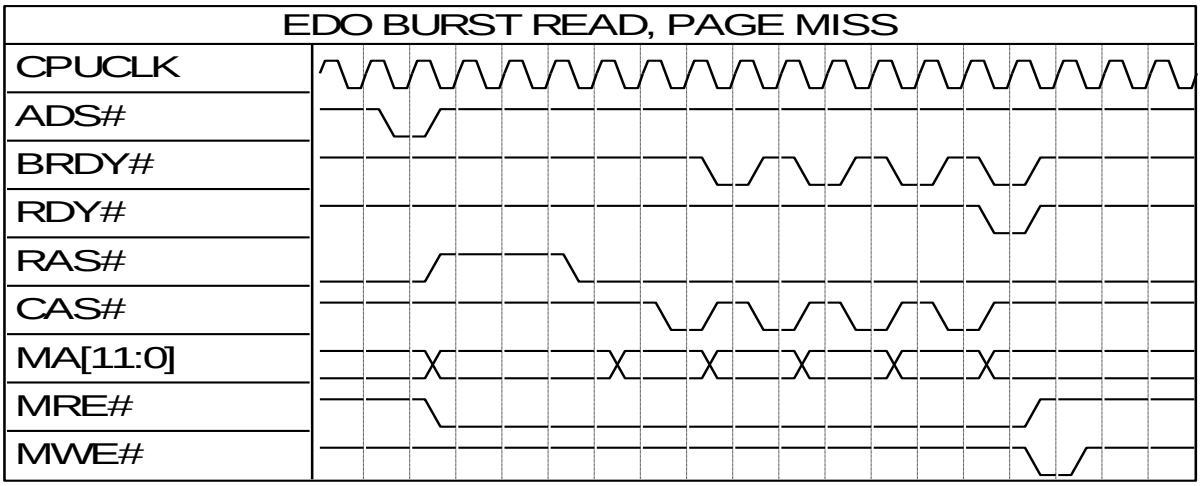


Figure 2.14-1 EDO DRAM Read with Burst Transfer and Page Miss

EDO DRAM Read with Burst Transfer and Page Hit

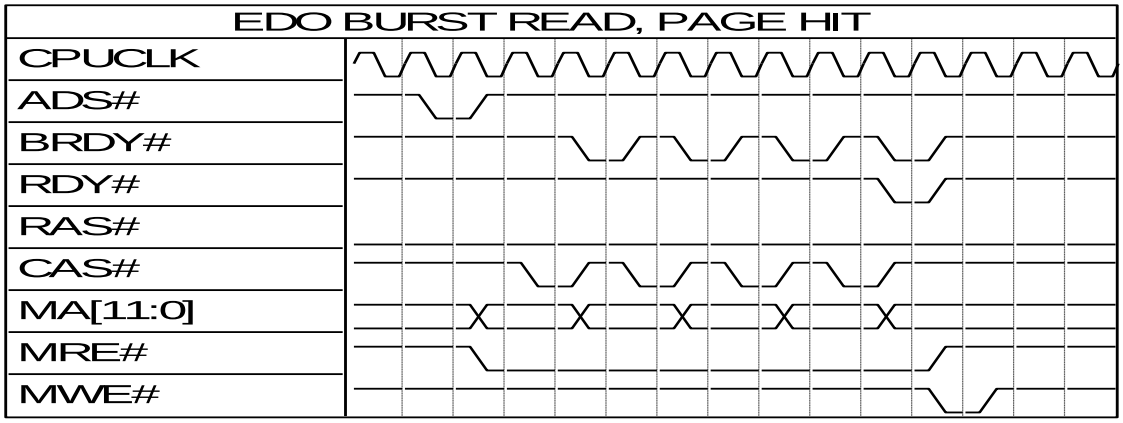


Figure 2.14-2 EDO DRAM Read with Burst Transfer and Page Hit

FP/EDO DRAM Write with Burst Transfer and Page Start

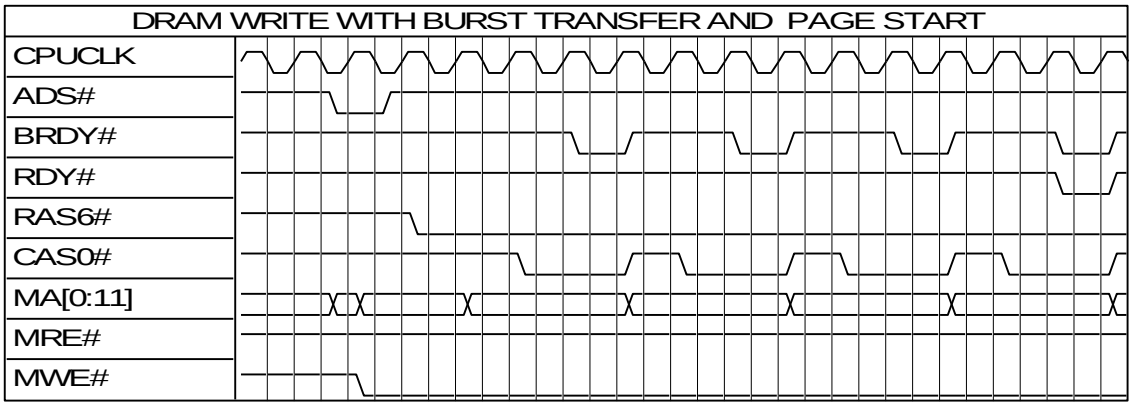


Figure 2.15 FP DRAM Write with Burst Transfer and Page Start

Parity

The bit 7 of register 41 enables checking for DRAM parity error during read cycles, and if register 05-04[8] is enabled for SERR# output, then parity errors will activate SERR# output to instruct 85C497 to assert NMI which further alerts CPU of this faulty situation. DRAM parity bits are generated by 85C496 during level 2 cache write back DRAM cycles to keep parity bits updated, also parity bits are generated during PCI master and ISA master to main board DRAM write cycles.

Refresh

85C496 enforces CAS# before RAS# refresh and supports RAS# staggering that reduces power surge induced by simultaneously switch of the RAS# lines. Refresh is triggered by the OUT1 input from 85C497. 85C496 do not hold CPU during refresh in progress, also the refresh cycle is concurrent with every other cycles in the system except DRAM cycles so that the system time consumed by refresh is very few. The bit 4 of register 41 enables power saving slow refresh by reducing the rate of refresh to one refresh per every 4 OUT1 refresh requests.

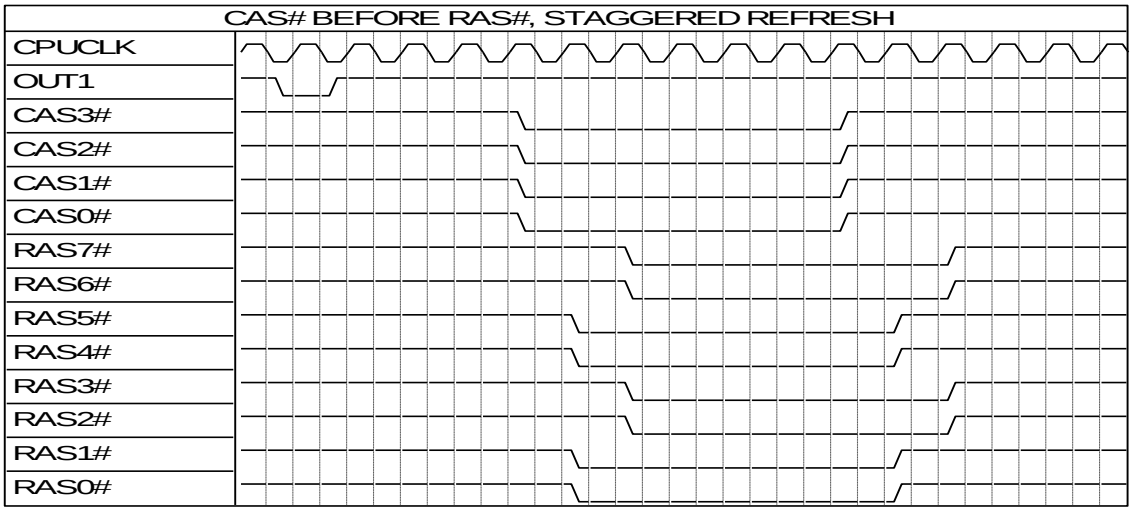


Figure 2.16 CAS# before RAS#, staggered refresh

High current drive support

Since 85C496 supports up to 8 rows of DRAMs, the capacitance loading of the DRAM interface in 85C496 varies fairly distantly, therefore 85C496 provides programmable I/O buffers which have driving capability of 12 or 24mA.

System designer may program register 57[6:4] to select CAS#[3:0], MA[10:0] and MWE# depending on the number and type of DRAM populated and the actual board wire layout. If the ultimate timing still do not meet the minimum requirements after setting this register and turning other corresponding DRAM timing settings, external TTL driver should be installed to provide sufficient current drive for the DRAMs.

2.2.3 DRAM Address Translation

Table 2.8.1 and 2.8.2 show the Host/PCI/ISA address multiplexing scheme for generation of row and column addresses at the DRAM interface (MA[11:0]).

Table 2.8.1 DRAM Address Translation

	256K/512K		1M/2M		4M/8M/16M	
MA#	CAS	RAS	CAS	RAS	CAS	RAS
MA 0	A3	A13	A3	A13	A3	A13
MA 1	A2	A12	A2	A12	A2	A22
MA 2	A4	A14	A4	A14	A4	A14
MA 3	A5	A15	A5	A15	A5	A15
MA 4	A6	A16	A6	A16	A6	A16
MA 5	A7	A17	A7	A17	A7	A17
MA 6	A8	A18	A8	A18	A8	A18
MA 7	A9	A19	A9	A19	A9	A19
MA 8	A10	A11	A10	A20	A10	A20
MA 9	A21	A20	A11	A21	A11	A21
MA 10	A23	A22	A23	A22	A12	A23
MA 11	A25	A24	A25	A24	A25	A24

Table 2.8.2 4K Refresh Asymmetric DRAM Address Translation

	1M (12x8)		2M (12x9)		4M (12x10)	
MA#	CAS	RAS	CAS	RAS	CAS	RAS
MA 0	A3	A13	A3	A13	A3	A13
MA 1	A2	A12	A2	A12	A2	A12
MA 2	A4	A14	A4	A14	A4	A14
MA 3	A5	A15	A5	A15	A5	A15
MA 4	A6	A16	A6	A16	A6	A16
MA 5	A7	A17	A7	A17	A7	A17
MA 6	A8	A18	A8	A18	A8	A18
MA 7	A9	A19	A9	A19	A9	A19
MA 8		A20	A10	A20	A10	A20
MA 9		A21		A21	A11	A21
MA 10		A10		A22		A22
MA 11		A11		A11		A23

2.3 PCI Interface

The PCI Interface of 85C496 is to provide the interface between the CPU/main memory /ISA Bus and the PCI Bus. When PCI-to-main memory transfer happens, the 85C496 is a target; when CPU-to-PCI or ISA-to-PCI transfer happens, the 85C496 is a master.

Both of the non-configuration and configuration address spaces can be read or written by the Host. The 85C496 is both a master and target for the transaction, when the CPU is accessing

the 85C496's configuration registers. It is essential to know that PCI-to-CPU accesses are not permitted.

This section describes PCI cycle termination, exclusive access cycles, parity support and system error generation on the PCI Bus.

2.3.1 Functional Overview

Command Set

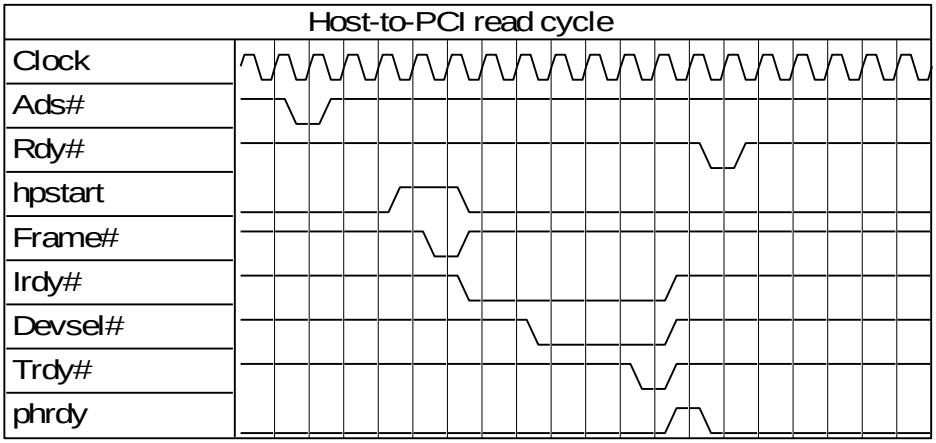
PCI Bus commands indicate to the target the type of transaction desired by the master. These commands are presented on the C/BE[3:0]# signals during the address phase of a transfer. Table 2.9 summaries the 85C496's support of the PCI Bus commands.

Table 2.9 PCI Bus Commands Supported

C/BE[3:0]#	Command Type	85C496 as a Master	85C496 as a Target
0000	Interrupt Acknowledge	Yes	No
0001	Special Cycle	No	No
0010	I/O Read	Yes	Yes (for ISA)
0011	I/O Write	Yes	Yes (for ISA)
0100	Reserved	No	No
0101	Reserved	No	No
0110	Memory Read	Yes	Yes
0111	Memory Write	Yes	Yes
1000	Reserved	No	No
1001	Reserved	No	No
1010	Configuration Read	Yes	Yes
1011	Configuration Write	Yes	Yes
1100	Memory Read Multiple	No	Yes[1]
1101	Dual Address Cycle	No	No
1110	Memory Read Line	No	Yes[2]
1111	Memory Write and Invalidate	No	Yes[3]

2.3.2 Host-to-PCI Cycles

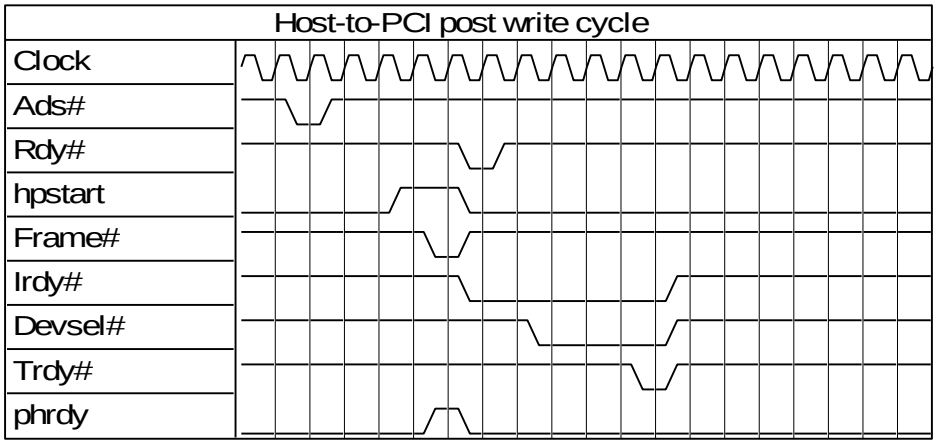
Read Cycle



NOTE: Hpstart, phrdy is 85C496's internal signal.

Figure 2.17 Host-to-PCI Read cycle

Single Write Cycle



NOTE: Hpstart, phrdy is 85C496's internal signal.

Figure 2.18 Single Host-to-PCI Posted Write Cycle

Multiple Write Cycle

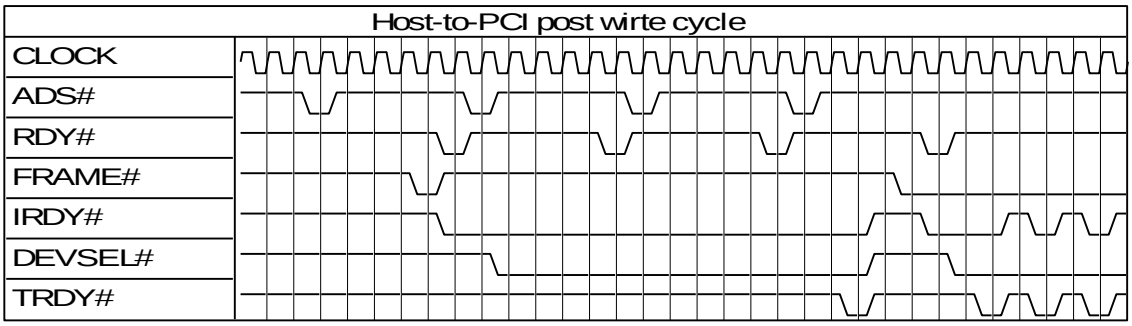


Figure 2.19 Host Write to Burst PCI Cycle

2.3.3 Host-to-PCI Configuration Cycles

The 85C496 supports PCI configuration read and write cycles for the CPU to access 85C496 configuration space register. When 85C496's configuration space is enabled via the CONFIG_ADDRESS register, any CPU I/O cycles to CONFIG_DATA register invokes PCI configuration cycle.

Note that the 85C496 hardwires AD16 as 496's IDSEL during configuration cycles. When 85C496 is the target of the configuration cycle, the 85C496 asserts DEVSEL# to claim the cycle on PCI bus. Therefore Host-to-PCM cycle, the 85C496 is the target as well as the initiator. The Host-to-PCM configuration cycle is completed when the 85C496 asserts both IRDY# and TRDY# acitve.

2.3.4 PCI Cycle Transaction Termination

Termination of a PCI cycle may be initiated by either the master or the target. All transactions are concluded when both FRAME# and IRDY# are deasserted, indicating an IDLE cycle state.

• Master-Initiated Termination

The 85C496's mechanism used in master-initiated termination is when FRAME# is deasserted and IRDY# is asserted. This signals the target that the final data phase is in progress. The 85C496 may initiate termination by one of the following conditions.

- Cycle completion
- Master abort

Cycle completion

Cycle completion refers to the termination when the 85C496 has concluded a transaction. 85C496 deasserts FRAME# and asserts IRDY# to indicate that this is a final data transfer. Both IRDY# and TRDY# must remain activated until the data phase is completed.

Master Abort

The 85C496 generates a master abort (Figure 2.20) when no target responds to an 85C496-initiated transaction. The 85C496 determines that there will be no response to a trasaction if it samples DEVSEL# deasserted at clock 6. Once the 85C496 has detected the missing DEVSEL# , FRAME# and IRDY# are deasserted. The 85C496 terminates the transaction with IRDY# on the Host Bus when the transaction is a CPU-to-PCI memory or I/O write. The 85C496 returns FFFFFFFFh to the Host CPU when the transaction is a CPU-to-PCI memory , I/O read, or Configure read.

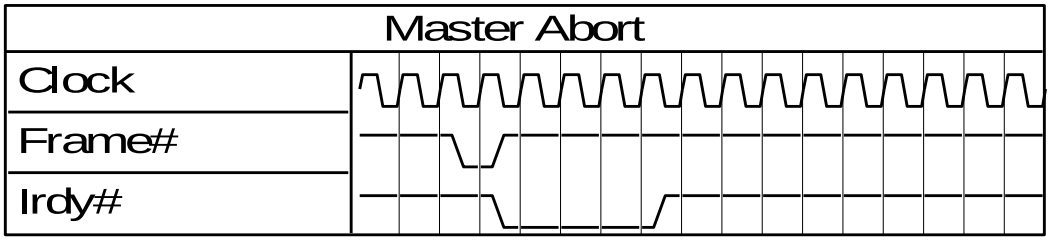


Figure 2.20 Master Abort Termination

• Target Initiated Termination

The mechanism uses the STOP# signal to initiate a target termination. A transaction can be terminated by the target in any of the following ways.

- Disconnect
- Retry
- Target abort

Disconnect

The 85C496 can initiate a disconnect when it is the target for any of the following reasons.

- PCI-to-85C496/85C497 Configure cycle, and PCI-to-85C497 cycles while the master intends to conduct a burst transaction.
- PCI master cycle to main memory across 256-byte boundary.

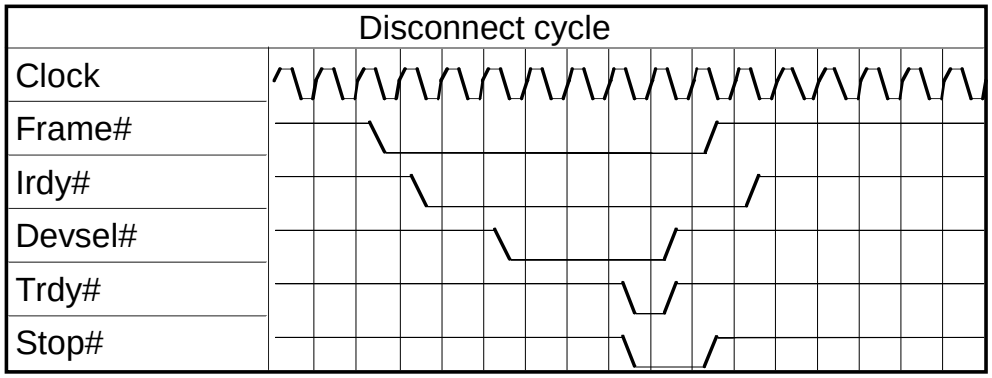


Figure 2.21 Target Initiated Disconnect

Retry

When a target is not able to process the transaction, it initiates a retry termination (Figure 2.22). While the 85C496 becomes the cycle inititor and detect a retry, it releases the bus for two clocks. The 85C496 will retry the cycle by arbitrating for the bus again and initiating the transaction when it is granted the ownership of the bus. The 85C496 keeps on retrying the cycle untill a successful transaction.

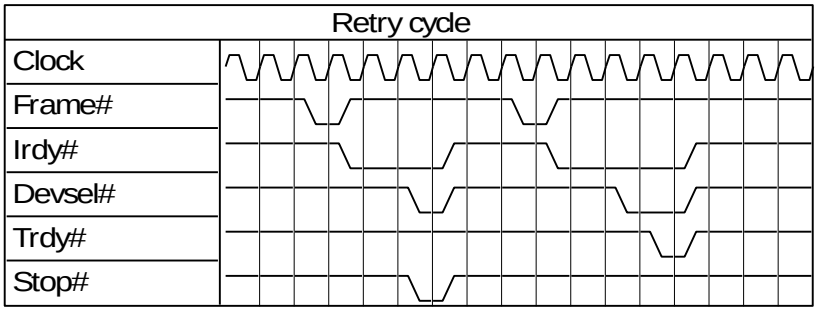


Figure 2.22 Target Initiated Retry

As a target, the 85C496 issues a retry if it is busy on write-back activity incurred by the previous PCI cycle. Additionally, the 85C496 issues a retry if it is locked by another PCI master.

Target Abort

When a target detects a fatal error which causes a abnormal termination, it can terminate a transaction with a target abort (Figure 2.23). After receiving a target abort, the 85C496 will assert the SERR# signal on PCI Bus to signal a fatal error condition. As a target, the 85C496 never generates a target abort.

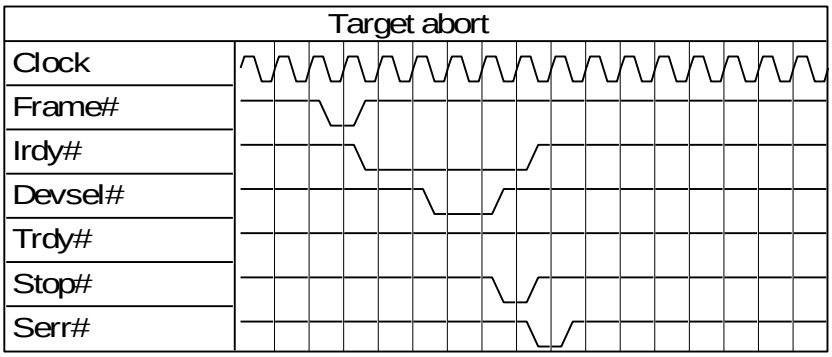


Figure 2.23 Target Initiated Target Abort

2.3.5 Exclusive Accesses

The PCI specification provides an exclusive access mechanism which allows non-exclusive accesses to proceed in the face of exclusive accesses.

- **85C496 as a Master Initiating Exclusive Access**

As a master. 85C496 never initiates a lock cycle. As a target, the 85C496 locks itself if it samples LOCK# negated during the address phase of the PCI cycle. When both FRAME# and LOCK# are negated, the 85C496 unlocks itself. If LOCK# is asserted during the address phase, the 85C496 responds by asserting STOP# and TRDY# (retry).

2.3.6 Parity Support

PCI Parity signal is an even parity generated across AD[31:0] and C/BE[3:0]. The 85C496 does parity checking during data phase for a read when it acts as a PCI master, and during address and data phase for a write cycles, when it acts as a PCI slave, if the PCI parity check enable bit is enabled.

2.3.7 Conditions for Generating SERR# Signal

The 85C496 asserts SERR# when either a main memory parity error or PCI Bus parity error has occured. The 85C496 also asserts SERR# for one clock when it detects a target abort on a 85C496 initiated per cycle. When the 85C496 asserts SERR#, it will sets bit 14 of Register 06h~07h to 1.

2.3.8 PCI Arbitration

The internal PCI arbiter supports four PCI masters and one ISA/DMA master. The five masters are arranged in a binary-tree group rotation priority scheme. The binary-tree group rotation scheme is implemented similar to binary tree architecture. The five leaves represent these masters. When one of the five masters holds the bus and initiates one cycle, all the ancestor nodes simultaneously change priority to make this master priority become the lowest. The another node doesn't change its priority because both masters in this node don't hold the bus. The PCI arbitration priority scheme can be programmed to be one of the following four modes, weakest, weaker, stronger, strongest through bit 7, 6 of PCI configuration register 56h.

In weakest mode, PCI bus arbiter grants the bus to a CPU request when no other master request presents. The worst case throughput of CPU request is 0%. In weaker mode, arbiter will grant the bus to CPU at least once in every eight arbitration cycles. The worst case throughput of CPU requests is 12.5%. In stronger mode, arbiter will grant the bus to CPU once in every four arbitration cycles. The worst case throughput of CPU request is 25%. In strongest mode, arbiter will grant the bus to CPU once in twice arbitration cycles. The worst case throughput is 50%.

Priority Change Scheme:

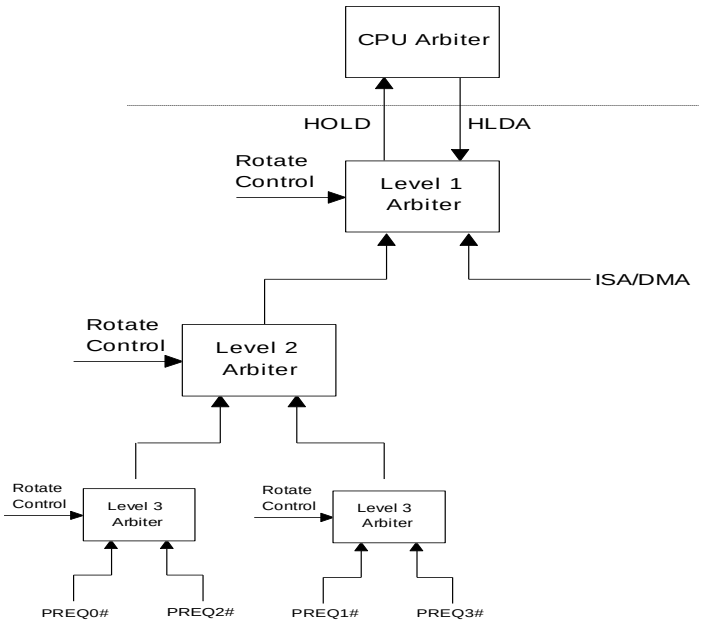


Figure 2.24 Arbiter Configuration Diagram

CPU Weakest Mode:

CPU → PCI0 → ISA→ PCI1 → ISA→ PCI2 → ISA→ PCI3 → ISA → PCI0 → .....→ CPU

CPU Weaker Mode:

CPU → PCI0 → ISA → PCI1 → ISA→ PCI2 → ISA → PCI3 → CPU → ISA→PCI0 → ISA → PCI1→.....→ CPU

CPU Stronger Mode:

CPU → PCI0 → ISA → PCI1 → CPU → ISA → PCI2 → ISA → CPU → PCI3→ ISA PCI0→ CPU → ISA → PCI1 → ISA →.....→ CPU

CPU Strongest Mode:

CPU → PCI0 → CPU →ISA →CPU →PCI1 → CPU →ISA →CPU → PCI2 → CPU → ISA → CPU →PCI3 →CPU →ISA →CPU →.....→ CPU

Arbitration Signal Protocol:

A PCI master requests the bus by asserting REQ#. Master must only use REQ# to signal a true need to use the bus. An agent must never use REQ# to "park" itself on the bus. The arbiter may deassert an mater's GNT# on any clock. An master must ensure its GNT# is asserted on the clock edge it wants to start a transaction. If GNT# is deasserted, the transaction must not proceed. Once asserted, GNT# may be deasserted according to the following.

a. If GNT# is deasserted and FRAME# is asserted, the bus transaction is valid and will continue.
b. One clock delay is required between the deassertion of a GNT# and the assertion of the next GNT#.
c. While FRAME# is deasserted, GNT# may be deasserted at any time in order to service a higher priority master, or in response to the associated REQ# being deasserted.

A DMA/ISA master requests the bus by asserting SHOLD. Arbiter then asserts FHLDA to indicate that host and PCI bus are held by DMA/ISA master. FHLDA can be asserted only when PCI bus is idle because once FHLDA is asserted, the cycle of DMA/ISA master will initiate cycle into host or PCI bus any time. If FHLDA is asserted, it can not be deasserted until SHOLD is deasserted because DMA/ISA master is not preemptable.

If a PCI master obtains bus grant over 16 PCI clocks and doesn't assert FRAME#, its request will be masked two PCI clocks. If PCI bus is locked, the request of DMA/ISA master will be masked. When post write buffer is busy, all acknowledges will be masked.
The arbitration logic supplies a mechanism for PCI bus parking. Parking is only allowed for the device which is tied to 496 PCI master request. Internal 496 PCI master grant will be asserted when no other master is currently using or requesting the bus. This achieves the minimum PCI arbitration latency possible. When PCI bus idle and 496 PCI master into Parking state, the 496 PCI master must ensure that AD[31:0], C/BE[3:0], and (one PCI clock later) PAR are driven.

The request and grant pins of the third PCI master are multiplexed by VL target local device selection and ready pins through bit3 of PCI configuration register 57h.

The request and grant pins of the fourth PCI master are multiplexed by pin 126 and 127 pins through bits[2:0] of PCI configuration register 57h.

2.3.9. CPU Back off support for PCI Bus PCI-to-PCI Bridge

PCI bus deadlock occurs when the upstream memory post-write buffers in the PCI-to-PCI bridges are turned on and both CPU and the PCI master behind the PCI-to-PCI bridge initiated cycle toward each other.

As described in the *PCI local bus specifications 2.1*, ‘Potential deadlock scenario when using PCI to PCI bridge’ of section 3.10. Deadlock occurs because the PCI-to-PCI Bridge cannot allow a read to transverse it while holding posted write data, or the agent that initiated the PCI access, cannot allow the PCI-to-PCI Bridge to flush data until it completes the read.

Therefore to resolve deadlock due to PCI-to-PCI Bridge unable to respond to a read while holding posted write data is to disable the post-write buffers in the PCI-to-PCI Bridge so that the PCI master behind the PCI-to-PCI Bridge never leaves data in its post-write buffers while is hasn't being granted the ownership for the root PCI bus.

PCI-to-PCI Bridge should be able to respond to a write regardless the state of the post-write buffer since the *PCI to PCI bridge architecture specification 1.0,* section 7.3 'Posting write data' requires that when a bridge has posted data it must accept write data in the opposite direction without flushing the buffer first.

To resolve the scenario that the agent cannot allow PCI-to-PCI Bridge to flush data until it completes the read, CPU read cycles will be backed off with the 85C496 at dead locks. Register 67 bit 3 enables this feature. However 85C496 will not back off CPU if the cycle is a CPU lock cycle.

Since data written to CPU to PCI post-write buffers by CPU can not be backed off, a non-postable area is accommodated by the 85C496 to allocate the memory space behind the PCI-to-PCI bridge as non-postable. This non-postable area can be setup by programming Register 64~65h.

2.4 FS-Link Interface

2.4.1 Fast Link Machine

For concurrence detecting PCI and ISA cycle, PCI target machine starts fast link machine cycle by asserting FADS#/FRDY# in the next PCI clock after FRAME# is asserted. If FADS#/FRDY# is asserted over three PCI clocks, the AT cycle will proceed normally. Others the AT cycle is abort in the following conditions.

1. PCI positive decode except accessing 497 configuration space.
2. ISA abort in the following cases.
   (1) I/O address over 64K.
   (2) memory address over 16M and not in ISA space.
   (3) access ISA and is retried by 496 PCI target machine.

In ISA abort condition, master abort will occur in PCI bus when no other device response the cycle. If DMA(ISA master) as master and post write buffer is busy, FADS#/FRDY# can not be asserted.

ISA space: M/IO#=0 or ISA memory hole or BIOS memory no shadow or C, D ROM segment in 1 M or A, B video RAM in 1 M and being set forward to ISA.

AHOLD is asserted after confirming that it is not onboard memory or VL bus cycle, and deasserted after PCI target ready being asserted in read cycle or fast link machine in turn around state in write cycle. When ISA space is accessed by CPU, it can not be post written to buffer. because if done, the concurrence of CPU cycle and fast link machine cycle will occur and it will make host address bus conflict.

After addresses, commands, byte enables and data transfer from 496 to 497 via HA[31:16], fast link machine into wait state until SADS#/SRDY# is asserted one PCI clock. If write cycle then fast link machine into idle state. In read one word cycle, it will latch data and go into idle state. In read double words, it will latch low word and then high latch word in the next PCI clock and go into idle state.

For pin counts consideration, the data communication between 496 and 497 via HA[31:16]. When FADS#/FRDY# is asserted, the first cycle is turn around. The high order addresses(A[31:16]) are transferred in the second cycle, the byte enables/commands are transferred in the third cycle and the low order addresses(A[15:0]) are transferred in the forth cycle. If write cycle and the ninth bit of command is zero, low word data and high word data are transferred in the following two PCI clocks. The format of command is the same as PCI and located in the second four bits of command/byte enable. The first four bits are four byte enables. The ninth bit is the delay IRDY# which can indicate the write data whether is valid or not.

For some VL add on cards, HA[23:2] are regarded as LA[23:17] and SA[16:2], so between fast link machine turn around state and SADS#/SRDY# is asserted, the values of HA will still keep the same values as the addresses of CPU or PCI master.

The following two figures are shown that CPU reads/writes ISA target through fast-link machine.

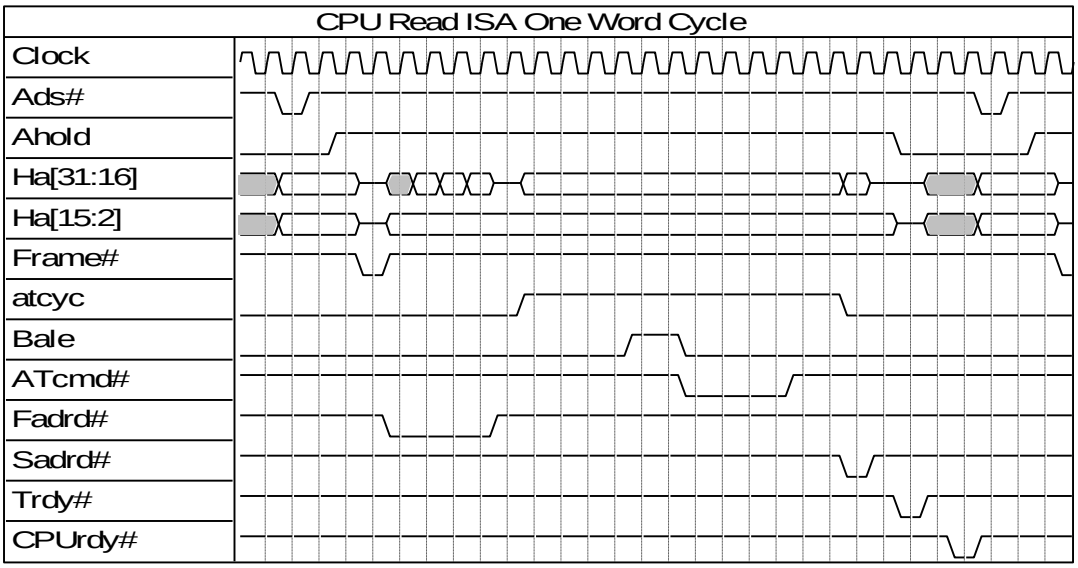


Figure 2.25 CPU Reads ISA Slave

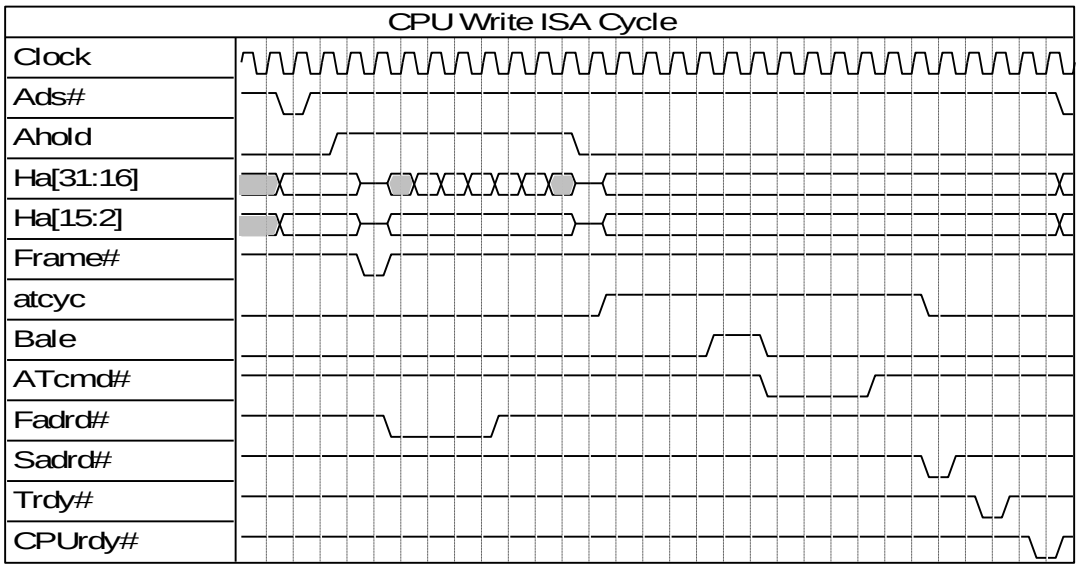


Figure 2.26 CPU Writes ISA Slave

The following two figures are shown that PCI master reads/writes ISA target through fast-link machine.

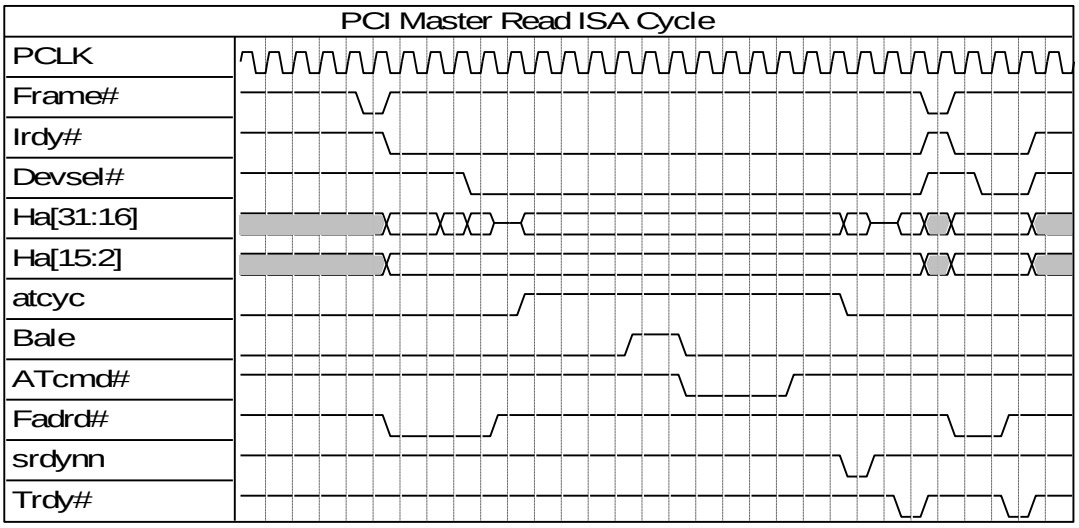


Figure 2.27 PCI Master Reads ISA Slave

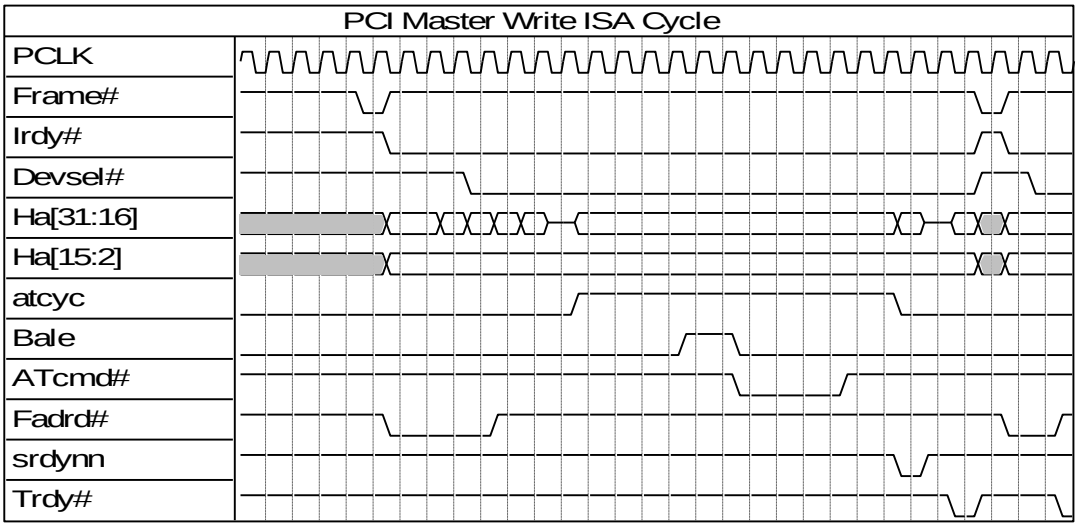


Figure 2.28 PCI Master Writes ISA Slave

The protocol of fast link machine is illustrated below :

Cycle		Description
Idle (the first cycle)	T0	High Impedence
High Address	T1	496 F.L.M.Addr[31:16] ⇒ HA[31:16] ⇒ 497 F.L.M.
Command/Byte Enable	T2	496 F.L.M. PCI IRDY# ⇒ HA24 ⇒ 497 F.L.M. 496 F.L.M.Command[3:0] ⇒ HA[23:20] ⇒ 497 F.L.M. 496 F.L.M. Byte Enable[3:0] ⇒ HA[19:16] ⇒ 497 F.L.M.
Low Address	T3	496 F.L.M.Addr[15:0] ⇒ HA[31:16] ⇒ 497 F.L.M.
Low Word data (write)	T4	496 F.L.M.data[15:0] ⇒ HA[31:16] ⇒ 497 F.L.M.
High Word data (write)	T5	496 F.L.M.data[31:16] ⇒ HA[31:16] ⇒ 497 F.L.M.
Turn Around	T4 or T6	High Impedence
Wait		496 F.L.M. Addr[15:2] ⇒ HA[15:2] 497 F.L.M. Addr[31:16] ⇒ HA[31:16]
Last Wait (read)		497 one word ⇒ HA[31:16] ⇒ 496 F.L.M. 497 low word of double word ⇒ HA[31:16] ⇒ 496 F.L.M.
Read Double Word data		497 high word of double word ⇒ HA[31:16] ⇒ 496 F.L.M.
Idle (the last cycle)		High Impedence

NOTE: S.L.M. means Slow Link Machine.

2.4.2 Slow-Link Machine

When ISA master or DMA want to access the upper memory(onboard, VL or PCI), 497 will assert SHOLD. After arbiter asserting FHLDA to indicate that host and PCI bus are held by DMA or ISA master, 497 will assert SADS#/SRDY# one PCI clock. The first two PCI clocks are turn around. Then the high order addresses are transferred in the third cycle and the command/byte enables are transferred in the forth cycle and the low order addresses are transferred in the fifth cycle. If write cycle then the write data are transferred in the following cycle. After the need data are transferred, the slow link machine goes into wait state until FADS#/FRDY# is asserted by 496 to indicate the cycle is ready and if read cycle the data is valid in HA[31:16]. Because ISA master or DMA can only access the upper memory, the format of command/byte enables is the first four bits are byte enables and the fifth bit is write/read bit.

The following two figures are shown that ISA master or DMA reads/writes onboard memory through slow-link machine.

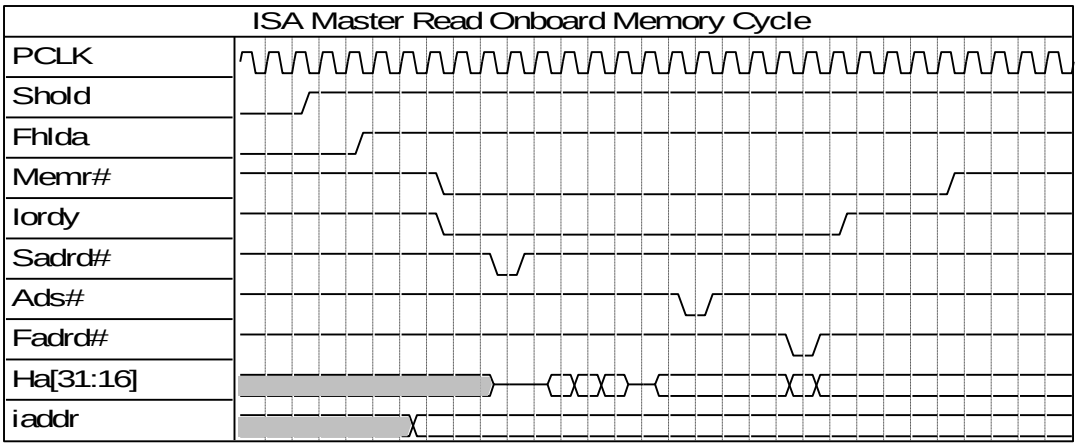


Figure 2.29 ISA Master Reads Onboard Memory

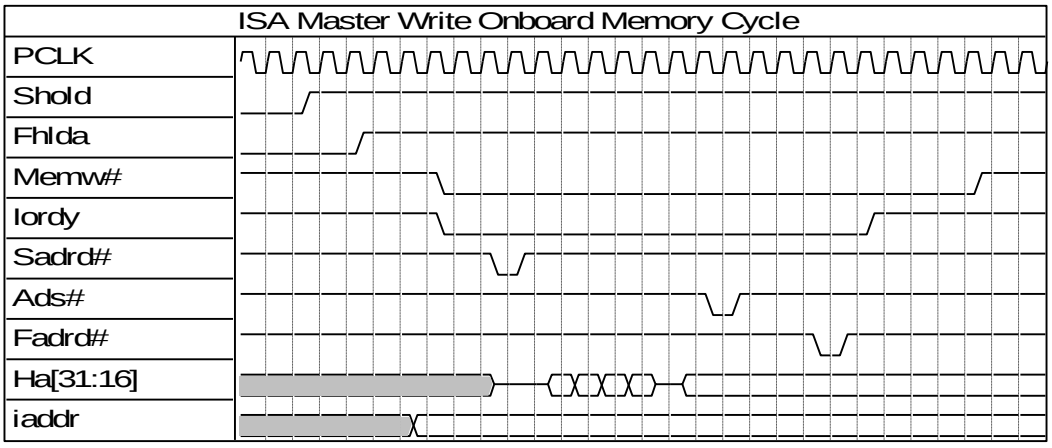


Figure 2.30 ISA Master Writes On-board Memory

The protocol of slow link machine is illustrated below:

Cycle		Description
Idle (the first cycle)	T0	497 S.L.M. to inform 496 tristate HA.
Idle (the second cycle)	T1	High Impedence
High Address	T2	497 S.L.M.Addr[31:16] ⇒ HA[31:16] ⇒ 496 S.L.M.
Command/Byte Enable	T3	497 S.L.M. MEMW#/MEMR# ⇒ HA20 ⇒ 496 S.L.M. 497 S.L.M. Byte Enable[3:0] ⇒ HA[19:16] ⇒ 496 S.L.M.
Low Address	T4	497 S.L.M. Addr[15:0] ⇒ HA[31:16] ⇒ 496 S.L.M.
One Word data (write)	T5	497 S.L.M. data[15:0] ⇒ HA[31:16] ⇒ 496 S.L.M.
Turn Around	T5 or T6	High Impedence
Wait		496 S.L.M. Addr[31:2] ⇒ HA[31:2]
Last Wait (read)		496 S.L.M. Data[15:0] ⇒ HA[31:16] ⇒ 497 S.L.M.
Idle (the last cycle)		High Impedence

NOTE: S.L.M. means Slow Link Machine.

2.5 IDE Controller

The built-in IDE controller of 85C496 acts as a VL-target, all control pins to IDE-drives are shared with PCI AD bus. Due to this restriction, IDE cycle and PCI cycle cannot be executed concurrently. Bit 31 of AD bus needs to be connected to a pull-high resistor because it is used as IORDY of IDE interface and some IDE drives have omitted this pin. Pin sharing of IDE interface and PCI interface are listed in the Pin Description Section.

The IDE controller supports 16-bit and 32-bit data transfer for IDE data register 1F0/170, other IDE registers R/W are 8-bit. When a 32-bit access to IDE data register is detected by IDE controller, it convert the access to 2 16-bit accesses.

The most extensive mode allowed user to assign address 1FX to channel 0 and 17X to channel 1 that enable the built-in IDE controller supporting up to 4 hard drives. (Each channel may connect to a master drive and a slave drive).

To speed up the data transfer between CPU and the hard disk, the IDE controller provides a prefeatch buffer and a post-write buffer which are both one-level deep and double-word wide for each IDE channels.

The operating speed may also be configured to accommodate hard drives with different access time. The cycle time (recovery time + command active time) is determined by CPU clock and the setting in configure registers. The shortest recovery time is 1 CPU clock and the command active time is 1 CPU clock, which meet the standard of AT Attachment Interface mode 3, 4 and above requirement timing.

Refer to 85C496 configure register 58h~59h, 60h~61h, 62h~63h for more details about configuring the built-in IDE controller.

2.6 VESA Device Interface

85C496 supports VESA VL-bus standard target interfaces. To enable VESA target cycles, register 57[3] must be set to enable the LBD#, LRDY# pin. Note that if VL bus is employed, then 85C496 supports only three PCI masters instead of four.

VL-bus local bus target can be accessed by CPU or ISA masters, PCI master accessing VL-bus local bus target is not supported because the fundamental differences between these two buses.

VL target responds to CPU or ISA master command by asserting LBD# to claim their cycle. Register 59-58[0] selects the timing at which 85C496 samples LBD#. According to VESA VL-bus specifications, in systems operating at 33MHz or less VL target should assert LBD# before the end of the first clock after ADS#, and for systems at 40MHz or higher, LBD# should be asserted before the end of the second clock after ADS#.

If the VL target asserts LBD# to claim the cycle, 85C496 releases control and monitors LRDY#, once LRDY# being sampled active, 85C496 returns RDY# to CPU or ISA bridge at the next clock and the cycle completes.

85C496 also supports shadowing of I/O port writes to palette lookup table as recommended by VESA VL-bus standard. If enabled by register 59-58[0], then the I/O write cycles to location 03C6-03C9h will be broadcast to PCI bus and further to ISA bus regardless whether a VL target claims the cycle or not. This feature allows every I/O devices at this location to be updated with identical data.

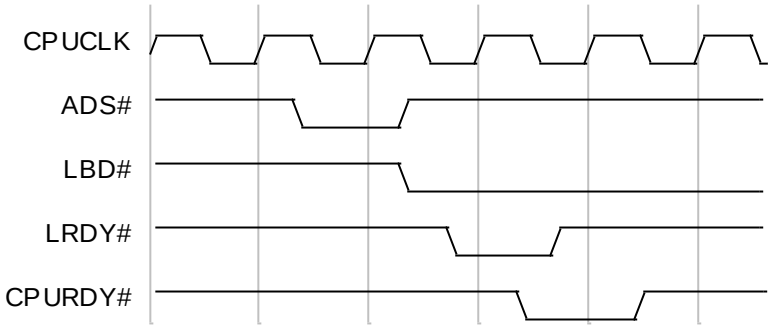


Figure 2.31 VESA VL-bus target cycle, 33MHz

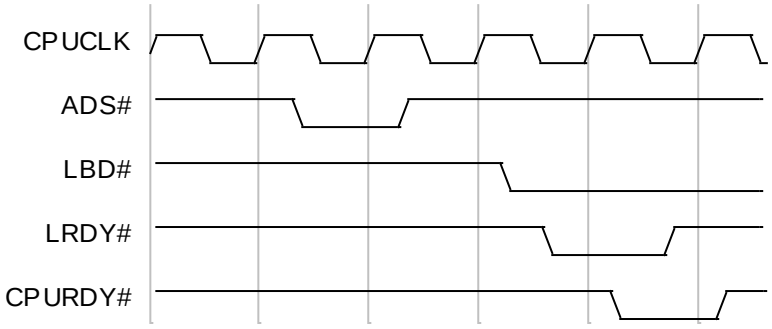


Figure 2.32 VESA VL-bus target cycle, 50MHz

2.7 Multiplexed pins

Several 85C496 I/O pins have multiple functions:

Pin 207, 204 can be used as either VESA VL-bus device control or the arbitration pins of the third PCI master.

If there is no 16M by 32/36 bit DRAM installed, then pin 157 may be configured as RAS7#, pin 126 as DIRTY or PREQ3# for the fourth PCI master and pin 127 as DIRTYWE# or PGNT3#. For more details about the function combinations, please refer to Register 57h for fourth PCI master. Otherwise, either pin 157 or 126 should be configured to be MA11 instead.

Pin 158, 159 can be used as two banks of DRAM, PCI to PCI bridge on PCI bus, or one bank DRAM and CPU reset command from keyboard controller. Refer to 85C496 Register 67h bit 3 for more detailed information about the definition of pin 158 and 159.

2.8 Clocks

Both CPU and PCI clock are supplied clock by external clock generator. CPU clock may varies from 50 MHz to 2 MHz. When the CPU clock frequency is lower than or equal to 33MHz, the PCI clock frequency is equal to CPU and when the CPU clock frequency is higher than 33 MHz., the PCI clock frequency is half of CPU clock.

2.9 Reset Logic

Both 85C496 and 85C497 receive power good reset input. 85C497 uses fast pwrgd (3us) to generate DRVRST, PCIRST# and internal reset and 85C496 uses slow pwrgd (5us) to generate INIT. 85C497 generates CPURST based on the states of INIT and cpurst_en bit. The detailed sequences are explained in the following section.

The only reset logic in 85C496 is to generate INIT signal to 85C497 and CPU as described in the following sequences:

a. When PWRGD is asserted, it triggers the pd5us circuit.
b. 85C496 configuration registers are initialized when PWRGD is asserted until pd5us is active (85C496 reset signal)

c. INIT signal is generated when PWRGD is asserted and extend 25 more CPU clocks after pd5us is asserted.
d. INIT is generated under three conditions: power on reset, fast keyboard reset, or shut down special cycle.
e. 85C496 internal state machine is initialized when rst486 signal is asserted. The rst486 is asserted when INIT is asserted and cpurst_en is enabled (cpurst_en is default enable, it is disabled by BIOS during POST in the normal operation unless the CPU type is 486, or CPU type is M7 and the WBAK bit in CCR2 is disabled, or if the init function is disable).

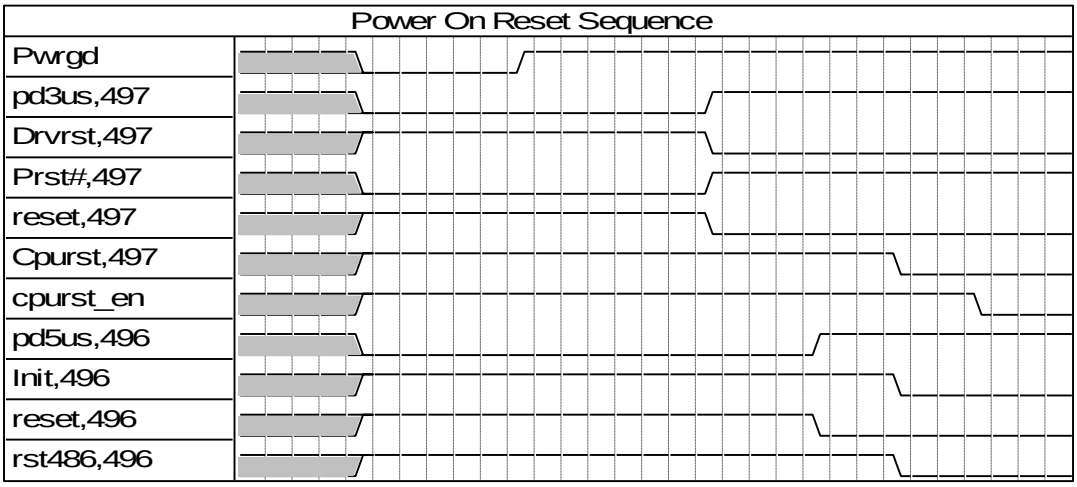


Figure 2.33 Power On Reset Sequence

3. Pin Assignment and Signal Description

This section contains a detailed description of each signal. The signals are arranged in functional groups according to the interface.

Note that the '#' symbol at the end of a signal name indicates that the active, or asserted state occurs when the signal is at a low voltage. When '#' is not present after the signal name, the signal is asserted when at the high voltage level.

The terms assertion and negation are used extensively. This is done to avoid confusion when working with a mixture of 'active-low' and 'active-high' signals. The term assert, or assertion indicates that a signal is active, independent of whether that level is represented by a high or low voltage. The term negate, or negation indicates that a signal is inactive.

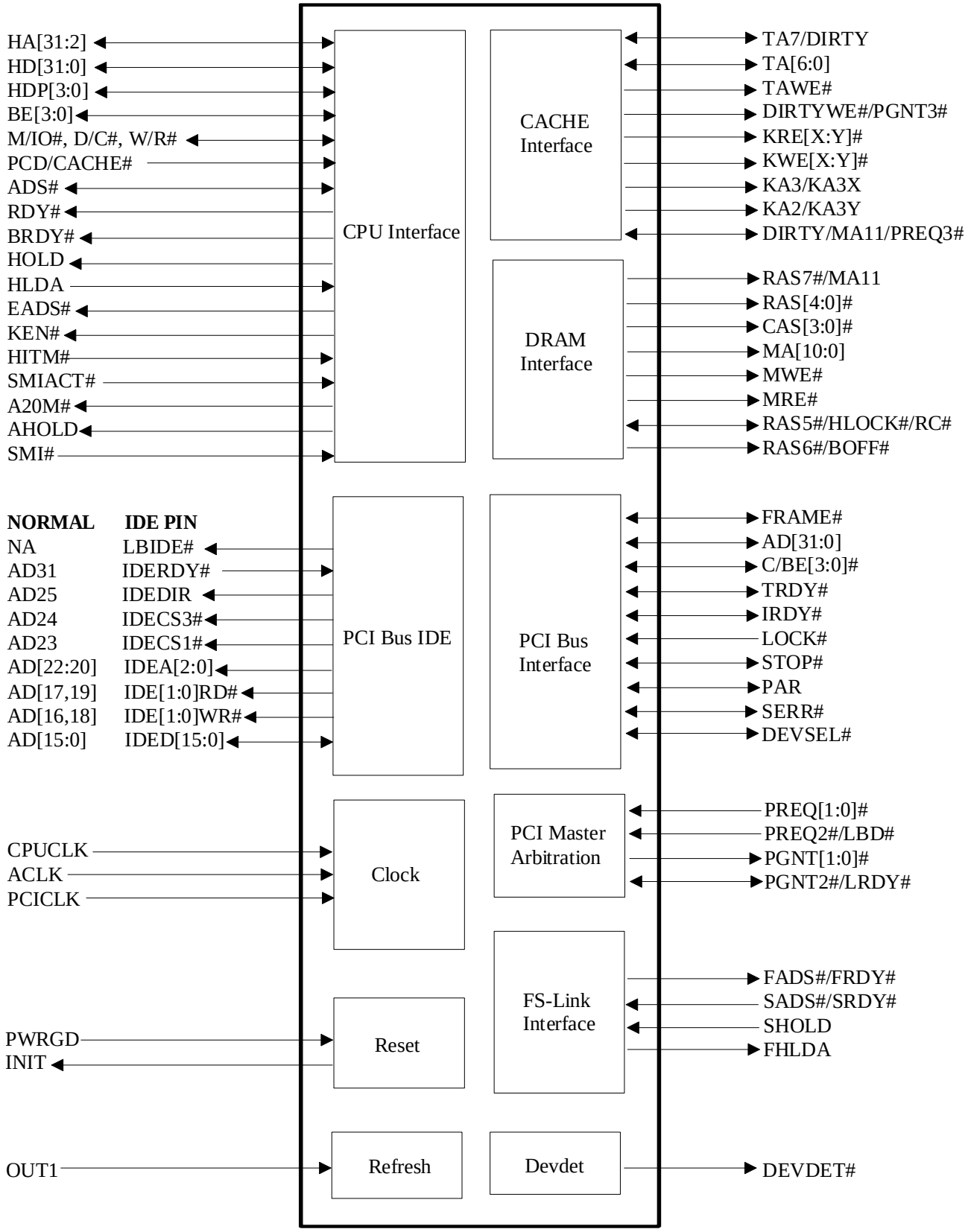


Figure 3.1 85C496 Component Block Diagram

3.1 Pin Assignment

85C496 Package Pinout

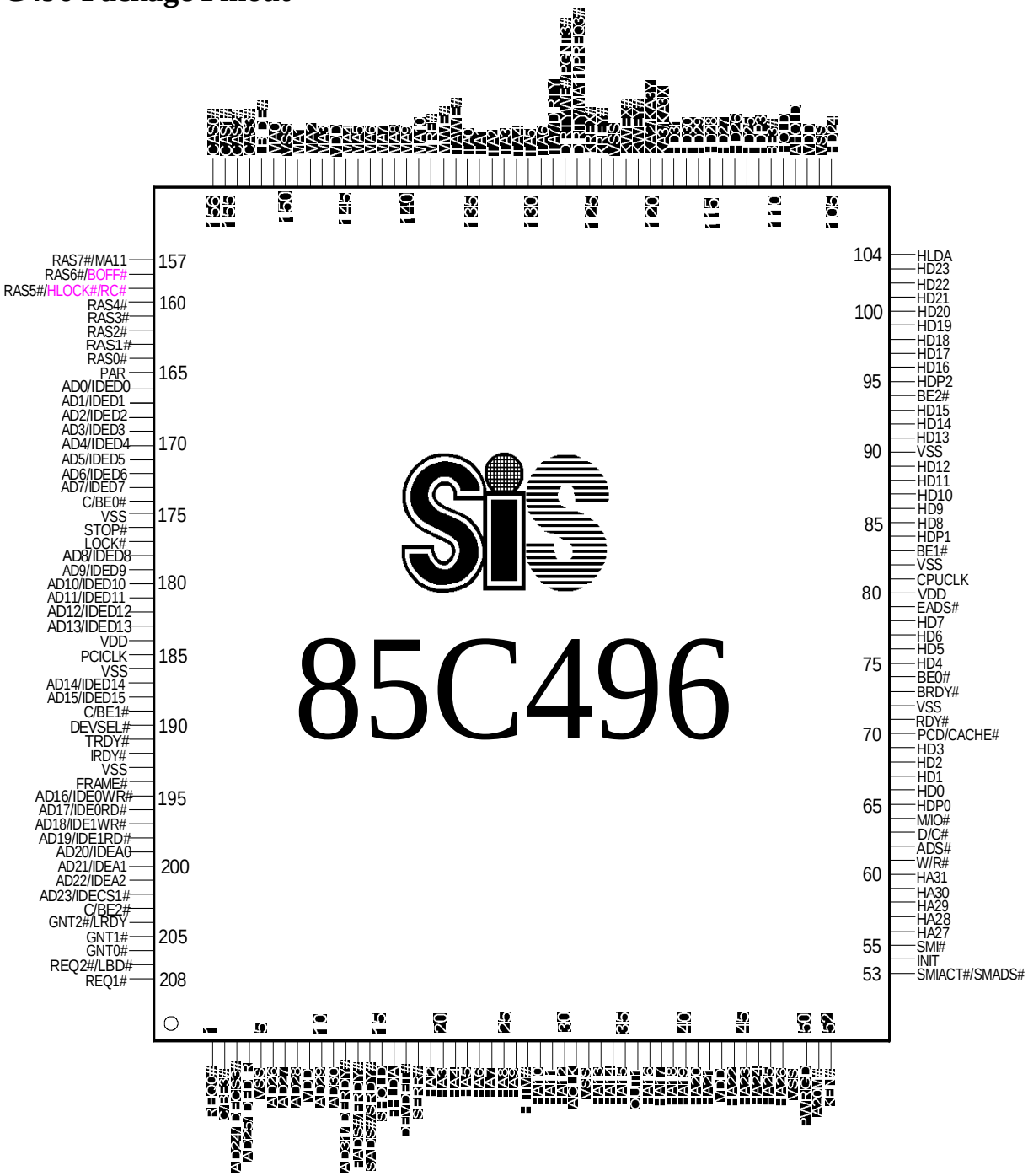


Figure 3.2 SiS85C496 Pin Assignment

85C496 Pin Assignment List(# means active low)

1 = REQ0#	53 = SMIACT#/SMADS#	105 = HD24	157 = RAS7#/MA11
2 = C/BE3#	54 = INIT	106 = VSS	158 = RAS6#/BOFF#
3 = AD24/IDECS3#	55 = SMI#	107 = VDD	159 = RAS5#/HLOCK#/RC#
4 = AD25/IDEDIR	56 = HA27	108 = AHOLD	160 = RAS4#
5 = VSS	57 = HA28	109 = HOLD	161 = RAS3#
6 = AD26	58 = HA29	110 = BE3#	162 = RAS2#
7 = AD27	59 = HA30	111 = HDP3	163 = RAS1#
8 = AD28	60 = HA31	112 = HD25	164 = RAS0#
9 = VDD	61 = W/R#	113 = HD26	165 = PAR
10 = AD29	62 = ADS#	114 = HD27	166 = AD0/IDED0
11 = AD30	63 = D/C#	115 = HD28	167 = AD1/IDED1
12 = AD31/IDERDY#	64 = M/IO#	116 = HD29	168 = AD2/IDED2
13 = FADS#/FRDY#	65 = HDP0	117 = HD30	169 = AD3/IDED3
14 = SADS#/SRDY#	66 = HD0	118 = HD31	170 = AD4/IDED4
15 = SHOLD	67 = HD1	119 = KA3/KA3X	171 = AD5/IDED5
16 = FHLDA	68 = HD2	120 = KA2/KA3Y	172 = AD6/IDED6
17 = DEVDET#	69 = HD3	121 = KWEY#	173 = AD7/IDED7
18 = SERR#	70 = PCD/CACHE#	122 = KWEX#	174 = C/BE0#
19 = HA2	71 = RDY#	123 = VSS	175 = VSS
20 = HA3	72 = VSS	124 = KREY#	176 = STOP#
21 = HA4	73 = BRDY#	125 = KREX#	177 = LOCK#
22 = HA5	74 = BE0#	126 = DIRTY/MA11/PREQ3#	178 = AD8/IDED8
23 = HA6	75 = HD4	127 = DIRTYWE#/PGNT3#	179 = AD9/IDED9
24 = HA7	76 = HD5	128 = TA7/DIRTY	180 = AD10/IDED10
25 = HA8	77 = HD6	129 = TA6	181 = AD11/IDED11
26 = HA9	78 = HD7	130 = TA5	182 = AD12/IDED12
27 = HITM#	79 = EADS#	131 = TA4	183 = AD13/IDED13
28 = HA10	80 = VDD	132 = TA3	184 = VDD
29 = HA11	81 = CPUCLK	133 = TA2	185 = PCICLK
30 = HA12	82 = VSS	134 = TA1	186 = VSS
31 = ACLK	83 = BE1#	135 = TA0	187 = AD14/IDED14
32 = VSS	84 = HDP1	136 = TAWE#	188 = AD15/IDED15
33 = HA13	85 = HD8	137 = MWE#	189 = C/BE1#
34 = HA14	86 = HD9	138 = MRE#	190 = DEVSEL#
35 = HA15	87 = HD10	139 = MA10	191 = TRDY#
36 = OUT1	88 = HD11	140 = MA9	192 = IRDY#
37 = HA16	89 = HD12	141 = MA8	193 = VSS
38 = HA17	90 = VSS	142 = MA7	194 = FRAME#
39 = HA18	91 = HD13	143 = MA6	195 = AD16/IDE0WR#
40 = HA19	92 = HD14	144 = MA5	196 = AD17/IDE0RD#
41 = HA20	93 = HD15	145 = MA4	197 = AD18/IDE1WR#
42 = HA21	94 = BE2#	146 = VDD	198 = AD19/IDE1RD#
43 = VDD	95 = HDP2	147 = MA3	199 = AD20/IDEA0
44 = HA22	96 = HD16	148 = MA2	200 = AD21/IDEA1
45 = HA23	97 = HD17	149 = MA1	201 = AD22/IDEA2
46 = HA24	98 = HD18	150 = VSS	202 = AD23/IDECS1#
47 = HA25	99 = HD19	151 = MA0	203 = C/BE2#
48 = HA26	100 = HD20	152 = LBIDE#	204 = GNT2#/LRDY#
49 = VSS	101 = HD21	153 = CAS3#	205 = GNT1#
50 = PWRGD	102 = HD22	154 = CAS2#	206 = GNT0#
51 = A20M#	103 = HD23	155 = CAS1#	207 = REQ2#/LBD#
52 = KEN#	104 = HLDA	156 = CAS0#	208 = REQ1#

3.2 Signal Description

3.2.1 Host CPU Interface

Signal Name	Type	Pin	Description
A20M#	O	51	**Address bit 20 Mask.** 496 forces A20M# to the CPU low for support of real mode compatible software. This signal is generated either from fast key board reset register or Port 92h. Writing a 0 to bit 1 of Port 92h Register forces A20M# low, if A20GATE (496 internal fast key board reset register bit) is also low. Writing a 1 to bit 1 of the Port 92h Register forces A20M# to the CPU high, regardless of the state of A20GATE. Upon reset, this signal is driven high.
ADS#	I/O	62	**Address Status.** ADS# is driven by the CPU to indicate that the address and byte enable signals and the bus cycle definition signals are valid.
AHOLD	O	108	**Address Hold.** It forces the CPU to release its address bus in the next clock cycle. The 496 asserts this signal in preparation to perform fast-slow link transfer. The 496 always drives the address on the Host Bus on the second clock after AHOLD is asserted and continues to drive the addresses until AHOLD is negated.
BE[3:0]#	I/O	110, 94, 83, 74	**Byte Enable.** The byte enable signals indicate which byte lanes on the CPU data bus carry valid data during the current bus cycle. BE3# indicates that the most significant byte of the data bus is valid while BE0# indicates that the least significant byte of the data bus is valid.
BRDY#	O	73	**Burst Ready.** BRDY# indicates that data presented are valid during a burst cycle.
EADS#	O	79	**External Address Status.** EADS# is a tri-state output pin. It indicates that a valid external address has been driven onto the CPU address pins. This address will be used for the CPU to perform an internal cache invalidation cycle.
HA[31:2]	I/O	60-56, 48-44, 42-37, 35-33, 30-28, 26-19	**Host Address.**

Signal Name	Type	Pin	Description
HD[31:0]	I/O	118-112, 105, 103-96, 93-91, 89-85, 78-75, 69-66	**Host Data Bus.**
HDP[3:0],	I/O	111, 95, 84, 65	**Host Data Parity.** Data parity bits generate even parity bits in memory write cycles and accept even parity bits in memory read cycles.
HLDA	I	104	**Hold Acknowledge.** HLDA comes from the CPU in response to a HOLD request. It is active high and remains driven during bus hold period. HLDA indicates that the CPU has given the bus to another bus master.
HITM#	I	27	**Hit Modified.** HITM# is used during Snoop operations to indicate that a hit to a modified cache line has occurred in the L1 cache. If HITM# is asserted, the 496 stalls the PCI-memory or ISA-memory cycles that caused the Snoop until the dirty line in the L1 cache is written back to memory.
HOLD	O	109	**CPU Host Request.** The Host bus request is used to request the control of the CPU bus. HLDA will be asserted by the CPU after completing the current bus cycle.
KEN#	O	52	**Cache Enable.** The CPU cache enable pin is used when the current cycle is cacheable to the internal cache of the CPU. It is an active low siginal asserted by the SiS85C496 during cacheable cycles. **NOTE:** KEN# is for 496 defined non-cacheable cycles and the PCD/CACHE# signal is for CPU defined non-cacheable cycles.
M/IO#, D/C#, W/R#	I/O	64, 63, 61	**Memory/Input-Output.** definition is an input to indicate an I/O cycle when low, or a memory cycle when high. **Data/Control.** is used to indicate whether the current cycle is a data or control access. **Write/Read.** from the CPU indicates whether the current cycle is write or read access. It is an output during the PCI master cycles.
RDY#	O	71	**Ready.** It indicates that the current non-burst cycle is complete when the CPU accesses on-board memory or ISA target.

Signal Name	Type	Pin	Description
PCD/CACHE#	I	70	**Page Cache Disable/Cacheable.** Page cache disable pin reflects the state of the page attribute bits of the CPU. The cache pin indicates an internally cacheable read cycle or a burst write-back cycle. If this pin is driven inactive during a read cycle, the CPU will not cache the returned data, regardless of the state of the KEN# pin.
SMIACT# /SMADS#	I	53	**System Management Interrupt Active.** For Intel processor, this signal is an active low input indicating that the processor is operating in SMM. For Cyrix processors, this signal is an active low input indicating the SMM address status. This pin is used for the memory controller to determine the memory space. For Intel CPU and AMD, when SMIACT# is active, the memory controller will map all the CPU generated memory cycle to SMRAM space. If the SMIACT# is inactive, on the other hand, the memory controller will map the cycle to normal DRAM space.
SMI#	I	55	**System Management Interrupt.** An active SMI# signal indicates 496 that the system is in the system management mode. 496 maps memory cycle to the SMRAM area when in system management mode.

3.2.2 497 Interface (FS Link, Refresh, Deturbo, PMU)

Signal Name	Type	Pin	Description
DEVDET#	O	17	**Device Detect.** This signal is generated to inform the 497 PMU the status of Device Detection information and the PCI master request.
FADS#/FRDY#	O	13	**Fast Address Status/Fast Ready.** For Host-to-ISA cycles, this signal is generated to the 497 indicates the request of the AT cycle. For ISA-to-memory cycles, on the other hand, this signal indicates that the 496 has presented valid data to the 497 in response to a read, or that the 496 has accepted data from the 497 in response to a write.
FHLDA	O	16	**Fast Hold Acknowledge.** When active, it indicates that the Host Bus is available for the 497.
OUT1	I	36	**Output 1.** This is an output signal from 8254 Counter/Timer located inside the 497 and used for the 496 DRAM Refresh Controller.
SADS#/SRDY#	I	14	**Slow Address Status/Slow Ready.** For ISA-to-memory cycles, this signal is generated from the 497 to the 496 indicates the request of the memory access. For Host-to-ISA cycles, on the other hand, this signal indicates that the 497 has presented valid data to the 496 in response to a read, or that the 497 has accepted data from the 496 in response to a write.
SHOLD	I	15	**Slow Hold Request.** The 497 generates this signal to the 496 to request the Host Bus.

3.2.3 PCI Interface

Signal Name	Type	Pin	Description
AD[31:0]	t/s	12-10, 8-6, 4-3, 202-195, 188-187, 183-178, 173-166	**PCI Address/Data.** During the first clock of a PCI transaction, AD[31:0] contains a physical address (32 bits). During subsequent clocks, AD[31:0] contains data.
C/BE[3:0]#	t/s	2, 203, 189, 174	**Command/Byte Enable.** PCI Bus Command and Byte Enables define the PCI command during the address phase of a PCI cycle, and the PCI byte enables during the data phases. C/BE[3:0]# are outputs when the 85C496 is a PCI bus master and inputs when it is a PCI slave.
DEVSEL#	s/t/s	190	**Device Select.** The 496 drives DEVSEL# based on the DRAM address range being accessed by a PCI master, or if it decodes the current configuration cycle targeted to the 496 or the 497, or if it decodes the current memory cycle targeted to the ISA memory hole, or if it decodes an I/O cycle targeted to ISA I/O space (less than 64K). As an input it indicates if any device has responded to current PCI bus cycle initiated by the 85C496..
FRAME#	s/t/s	194	**Frame#.** is an output when the 85C496 is a PCI bus master. The 85C496 drives FRAME# to indicate the beginning and duration of an access. When the 85C496 is a PCI slave, FRAME# is an input signal.
PGNT[1:0]#	t/s	205, 206	**PCI Bus Grant.** PGNT[1:0]# indicates that access to the PCI Bus has been granted to the PCI master request PREQ[1:0]#.
PGNT2#/LRDY#	t/s	204	**PCI Bus Grant/Local Ready.** PGNT2# indicates that access to the PCI Bus has been granted to the PCI master request (PREQ2#). For Host-to-VESA cycles, an assertion of LRDY# indicates that the target has presented valid data to the CPU in response to a read, or that the target has accepted data from the CPU in response to a write.

IRDY#	s/t/s	192	**Initiator Ready.** is an output when 85C496 is a PCI bus master. The assertion of IRDY# indicates the current PCI bus master's ability to complete the current data phase of the transaction. For a read cycle, IRDY# indicates that the PCI bus master is prepared to accept the read data on the following rising edge of the PCI clock. For a write cycle, IRDY# indicates that the bus master has driven valid data on the PCI bus. When the 85C496 is a PCI slave, IRDY# is an input.
LOCK#	s/t/s	177	**Lock.** indicates an exclusive bus operation that may require multiple transactions to complete. When LOCK# is sampled asserted at the beginning of a PCI cycle, the 85C496 considers itself a locked resource and remains in the locked state until LOCK# is sampled negated on a new PCI cycle.
PAR	t/s	165	**Parity.** Parity is an even parity generated across AD[31:0] and C/BE[3:0]. The 496 does parity checking during data phase for a read when it acts as a PCI master, and during address and data phase for a write cycles, when it acts as a PCI slave, if the PCI parity check enable bit is enabled.
PREQ[1:0]#	t/s	208, 1	**PCI Bus Request.** PREQ[1:0]# indicate to the PCI Bus arbiter that the PCI master 0 or PCI master 1 desire use of the PCI Bus.
PREQ2#/LBD#	t/s	207	**PCI Bus Request/Local Bus Device.** PREQ2# indicates to the PCI Bus arbiter that the PCI master 2 desires use of the PCI Bus. For Host-to-VESA cycles, this signal is used as the Local Bus Device and is driven by the VESA slave.
STOP#	s/t/s	176	**Stop.** STOP# indicates that the bus master must immediately terminate its current bus cycle at the next clock edge and release control of the PCI Bus. STOP# is used to disconnect, retry, and abort sequences on the PCI Bus.
SERR#	o/d	18	**System Error.** System error is an open drain output for reporting errors.

TRDY#	s/t/s	191	**Target Ready.** TRDY# is an output when the 496 is PCI slave. The assertion of TRDY# indicates the target agent's ability to complete the current data phase of the transaction. For a read cycle, TRDY# indicates that the target has driven valid data onto the PCI bus. For a write cycle, TRDY# indicates that the target is prepared to accept data from the PCI bus. When the 496 is a PCI master, it is an input.

3.2.4 IDE Interface

Signal Name	Type	Pin	Description
AD31/IDERDY#	t/s (I)	12	**IDE Ready.** If the hard disk is not fast enough to complete data transaction during command strobe, it drives IDERDY# low and the IDE controller will prolong the command strobe.
AD25/IDEDIR	t/s (O)	4	**IDE Direction Control.** This signal is not a standard ATA interface signal, it is used for the direction control of external data latch. During a read cycle, data flow from IDE drives into 496. In a write cycle, data flow from 496 to Hard disks.
AD24/IDECS3#	t/s (O)	3	**IDE Chip Select.** Part of register index. IDECS1# selects the command block register and IDECS3# select the control block register.
AD23/IDECS1#	t/s (O)	202	**IDE Chip Select.** Part of register index. IDECS1# selects the command block register and IDECS3# select the control block register.
AD[22:20]/IDEA [2:0]	t/s (O)	201-199	**IDE Address Bus.** Address bits indicating which register is to read from or write into. IDEA[2:0] and IDECS[3,1]# forms complete register index.
AD[17,19]/IDE[1 :0]RD#	t/s (O)	196, 198	**IDE Read strobe.** The falling edge of IDE[1:0]RD# enable data from a register onto data bus. IDE controller latches data at the rising edge of IDE[1:0]RD#.
AD[16,18]/IDE[1 :0]WR#	t/s (O)	195, 197	**IDE Write. Write strobe.** The rising edge of IDEWR[1:0]N clocks data into a register of the drives.
AD[15:0]/IDED[15:0]	t/s (I/O)	188-187, 183-178, 173-166	**IDE Data bus.** 8-bit or 16-bit bidirection data bus between HOST and IDE drives.
LBIDE#	O	152	**Local Bus IDE device.**

3.2.5 Cache Interface

Signal Name	Type	Pin	Description
DIRTY/MA11 /PREQ3#	I/O	126	**Dirty / Memory Address Bit 11 / PCI Bus Request.** This pin is used as dirty bit when the external dirty SRAM is used, otherwise it is used as memory address bit 11 when 16Mx36 type DRAM is used. This pin also indicate to the PCI bus arbiter that the PCI master 3 desires use of the PCI bus.
DIRTYWE#/ PGNT3#	O	127	**DIRTY bit Write Enable / PCI Bus Grant.** When an external DIRTY bit is used, this signal controls the write enable. PGNT3# indicates that access to the PCI bus has been granted to the PCI master request (PREQ3#).
KA3/KA3X	O	119	**Cache Address bit 3/Even bank indication.** This signal is used as the cache address bit 3 during cache non-interleaved mode; and it is used as the even bank indication during cache interleaved mode.
KA2/KA3Y	O	120	**Cache Address bit 2/Odd bank indication.** This signal is used as the cache address bit 2 during cache non-interleaved mode; and it is used as the odd bank indication during cache interleaved mode.
KRE[X:Y]#	O	125-124	**Cache Read Enable.** KREX# is used to indicate cache read enable for even bank and KREY# is used to indicate cache read enable for odd bank.
KWE[X:Y]#	O	122-121	**Cache Write Enable.** KWEX# is used to indicate cache write enable for even bank and KWEY# is used to indicate cache write enable for odd bank.
TA7/DIRTY	I/O	128	**Tag Address Bit 7 / Dirty bit.** This pin can be used as Tag Address bit 7 or the dirty bit.
TA[6:0]	I/O	129-135	**Tag Address.**
TAWE#	O	136	**Tag Address Write Enable.**

3.2.6 DRAM Interface

Signal Name	Type	Pin	Description
CAS[3:0]#	O	153-156	**Column Address Strobe.** They are the DRAM column address latch signals for bytes lane 3,2,1 and 0 respectively.
MA[10:0]	O	139-145, 147-149, 151	**Multiplexed DRAM Address.** The MA[10:0] provide the row and column address to the DRAM.
MRE#	O	138	**DRAM Read Enable.** This signal is used to control the direction of external buffer for DRAM data bus if the buffer is provided. This signal is asserted whenever CPU reads DRAMs data.
MWE#	O	137	**DRAM Write Enable.** This signal is connected to the DRAM write enable (WE#) pin. MWE# is asserted during DRAM write cycle.
RAS7#/MA11	O	157	**Row Address Strobe / Multiplexed DRAM Address.** This pin is used as Row Address Strobe 7 in default configuration. This pin becomes MA11 only when both the external Dirty bit and 16Mx36 DRAM type are used.
RAS6#/BOFF#	O	158	**Row Address Strobe 6/ Back Off.** This pin is used as Row Address Strobe 6 in default configuration, or as CPU back-off when supporting PCI-to-PCI bridge back off feature.
RAS5#/HLOCK#/RC#	I/O	159	**Row Address Strobe 5/ Host LOCK/ Reset CPU.** This pin is used as Row Address Strobe 5 in default configuration, or CPU bus lock input when supporting PCI-to-PCI bridge back off feature, or as Reset CPU input from external keyboard controller when disabling the 85C496 fast reset feature.
RAS[4:0]#	O	160-164	**Row Address Strobe.** DRAM row 4:0 address strobe output signal. These signals are used to latch the row addresses on the MA[11:0] bus into the DRAMs.

3.2.7 Clock

Signal Name	Type	Pin	Description
ACLK	I	31	**Advance Clock.** This is the advance clock and used by the 496. This clock shall be 3 to 5 ns earlier than the CPU clock. Frequencies supported by the external clock generator include 25, 33, 40, and 50 Mhz.
CPUCLK	I	81	**CPU Clock.** This clock is used by both CPU and the 496. Frequencies supported by the external clock generator include 25, 33, 40, and 50 Mhz.
PCICLK	I	185	**PCI Clock.** This clock is used by both PCI Bus and the 496. Frequencies supported by the external clock generator include 20, 25, and 33 Mhz.

3.2.8 Reset

Signal Name	**Type**	**Pin**	**Description**
INIT	O	54	**Initialize.** The 496 generates INIT to both CPU and 497 when fast keyboard reset, or CPU shutdown special cycle is occurred.
PWRGD	I	50	**Power Good.** When asserted, the power good input forces all the 496's internal registers and state machines to their default state.

3.2.9 Power Pins

Signal Name	**Type**	**Pin**	**Description**
VSS	I	5, 32, 49, 72, 82, 90, 106, 123, 150, 175, 186, 193	Ground.
VDD	I	9, 43, 80, 107, 146, 184	Power. +5V Volt supply.

4. Electrical Characteristics

4.1 Maximum Ratings

Case Temperature under bias ..... 0°C to 70°C
Storage Temperature ..... -40°C to 125°C
Voltage on any pin with respect to ground ..... -0.3 Volts to Vcc+0.3 Volts
Supply voltage with respect to Vss ..... -0.3 Volts to 7.0 Volts

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 reliability.

4.2 SiS85C496 D.C. Characteristics

Table 4.1 D.C. Specifications (V_{DD} = 5V ± 5%, T_{amb} = 0 to 70 °C)

Symbol	Parameter	Min.	Max.	Unit	Condition
V_{IL}	Input Low Voltage	-0.5	+0.8	V	
V_{IH}	Input High Voltage	+2.0	Vcc+0.5	V	
V_{T+}	TTL Schmitt Trigger, rising threshold	+1.5	+2.0	V	
V_{T-}	TTL Schmitt Trigger, falling threshold	+0.8	+1.1	V	
DT_T	TTL Schmitt Trigger, hysteresis (V_{T+} - V_{T-})	+0.4		V	
I_{IL}	Input Low Current TTL Input TTL w/pull-up	-10 -200	+10 -10	μA μA	0 < V_{IN} < Vcc
I_{IH}	Input High Current	-10	+10	μA	V_{IN} = V_{DD}
V_{OL}	Output Low Voltage		+0.4	V	@ I_{OL} max.
V_{OH}	Output High Voltage	+2.4		V	@ I_{OH} min.
I_{OL}	Output Low Current			mA	@ V_{OL} max.
I_{OH}	Output High Current			mA	@ V_{OH} min.
C_{IN}	Input Capacitance		10	pF	
C_{OUT}	Output Capacitance		10	pF	

4.3 A.C. Characteristics

The A.C. Specifications given in the following tables consists of output valid delays (float to active delay), output float delay (active to float delay), input setup and input hold requirements. These specifications are given for the functional operating range of the device. Timing specifications are given in nanoseconds (ns) unless otherwise specified. The test conditions are Vcc = 5V ± 5%, Tamb = 0°C to +70°C, CL = 35 pf unless otherwise specified.

Table 4.2 CPU Interface Timing Parameters (All the signals are based on ACLK rising edge)

Symbol	Parameter	Min	Max	Units	Notes
	CPU Interface				
t1	HA[31:2], HBE[3:0]#, HD[31:0], HDP[3:0] Input Setup Time	5		ns	
t2	HA[31:2], HBE[3:0]#, HD[31:0], HDP[3:0] Input Hold Time	3		ns	
t3	HA[31:2], HBE[3:0]#, HD[31:0], HDP[3:0] Output Valid Delay	6	17	ns	
	ADS#, M/IO#, D/C#, W/R# Input Setup Time	9		ns	
	ADS#, M/IO#, D/C#, W/R#, Input Hold Time	3		ns	
	ADS#, M/IO#, D/C#, W/R#, Output Valid Time	6	14	ns	
t5	PCD/CACHE#, HLDA, HITM#, SMI#, SMIACT# Input Setup Time	5		ns	
t5	PCD/CACHE#, HLDA, HITM#, SMI#, SMIACT# Input Hold Time	3		ns	
t7	RDY#, BRDY#, HOLD, AHOLD, EADS#, KEN#, A20M#, Output valid Delay	6	14	ns	

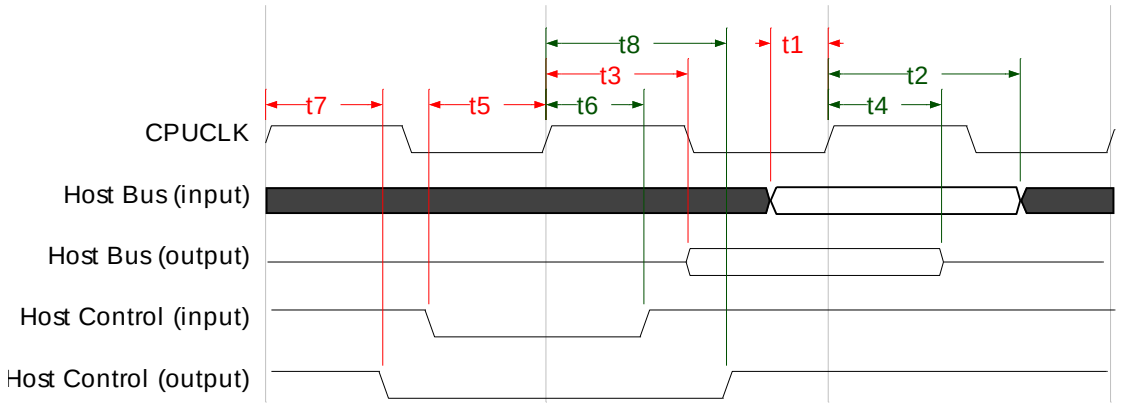


Figure 4.1 CPU Interface Timing

Table 4.3 PCI Interface Timing Parameters (All the signals are based on PCICLK rising edge)

Symbol	Parameter	Min	Max	Units	Notes
PCI Interface					
t9	FRAME#, IRDY#, DEVSEL#, TRDY#, STOP#, LOCK#, PAR, PREQ[3:0]#, AD[31:0], C/BE[3:0]# Input Setup Time	5		ns	
t10	FRAME#, IRDY#, DEVSEL#, TRDY#, STOP#, LOCK#, PAR, PREQ[3:0]#, AD[31:0], C/BE[3:0]# Input Hold Time	3		ns	
t11	FRAME#, IRDY#, DEVSEL#, TRDY#, STOP#, SERR#, PAR, PGNT[3:0]#, AD[31:0], C/BE[3:0]# Output Valid Delay	5	11	ns	
t12	FRAME#, IRDY#, DEVSEL#, TRDY#, STOP#, SERR#, PAR, PGNT[3:0]#, AD[31:0], C/BE[3:0]# Output Float Delay	8	24	ns	

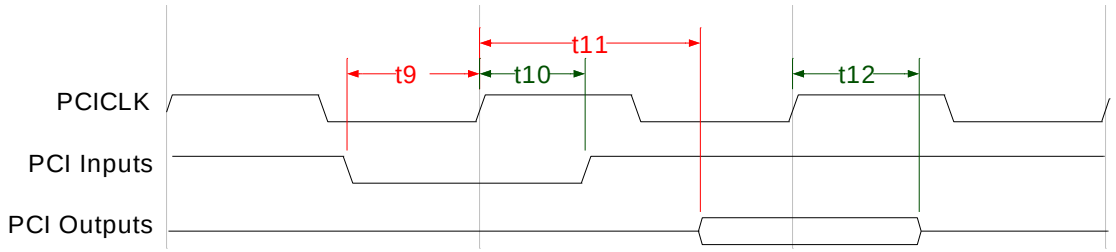


Figure 4.2 PCI Interface Timing

Table 4.4 IDE Interface Timing Parameters (All the signals are based on CPUCLK rising edge)

Symbol	Parameter	Min	Max	Units	Notes
IDE Interface					
t13	LBIDE# Output Valid Delay	6	12	ns	
t14	IDERDY# Input Setup Time	5		ns	
t15	IDERDY# Input Hold Time	3		ns	
t16	IDEDIR, IDECCS3#, IDECS1#, IDEA[2:0], IDE[1:0]RD#, IDE[1:0]WR# Output Valid Delay	6	12	ns	
t17	IDED[15:0] Input Setup Time	5		ns	
t18	IDED[15:0] Input Hold Time	3		ns	
t19	IDED[15:0] Output Valid Delay	6		ns	

Table 4.5 VESA Interface Timing Parameters (All the signals are based on ACLK rising edge)

Symbol	Parameter	Min	Max	Units	Notes
VESA Interface					
t20	LBD# , LRDY# Input Setup Time			ns	
t21	LBD# , LRDY# Input Hold Time			ns	

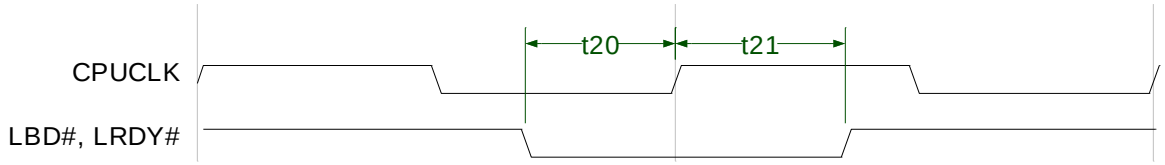


Figure 4.4 VL-Bus Timing

Table 4.6 L2 Cache Interface Timing Parameters (All the signals are based on ACLK rising edge)

Symbol	Parameter	Min	Max	Units	Notes
L2 Cache Interface					
t22	KRE[X:Y]# Output Valid Delay	2	5	ns	Note 1
t24a	KWE[X:Y]# Output Valid Delay	5	14	ns	
t24b	KWE[X:Y]# Output Valid Delay	5	14	ns	
a	KA3/KA3X, KA2/KA3Y Valid Delay	6	12	ns	
b	KA3/KA3X, KA2/KA3Y Valid Delay	3	7	ns	Note 1,3
	TA[7:0] Input Setup Time	5		ns	
	TA[7:0] Input Hold Time	3		ns	
	TA[7:0] Valid Delay from HA[27:16] Valid	6	16	ns	
	TAWE# Output Valid Delay	5	16	ns	
	DIRTY Input Setup Time	5		ns	
	DIRTY Input Hold Time	3		ns	
a	DIRTY Output Valid Delay	5	14	ns	
b	DIRTY Output Valid Delay	5	14	ns	Note 1,2
a	DIRTYWE# Output Valid Delay	5	14	ns	
b	DIRTYWE# Output Valid Delay	5	14	ns	Note 1,2

NOTE 1: Signal Timing relative to ACLK rising edge
NOTE 2: External cache write 2T (Reg. 43-42[11]=1)
NOTE 3: CPU level 2 cache burst read cycle and level 2 cache dirty line write back cycle.

Table 4.7 DRAM Interface Timing Parameters (All the signals are based on ACLK rising edge)

Symbol	Parameter	Min	Max	Units	Notes
DRAM Interface					
t39	RAS[7:0]# Output Valid Delay	5	10	ns	
t41	CAS[3:0]# Output Valid Delay	6	14	ns	
t43	MA[11:0] Output Valid Delay	6	14	ns	
t49	MRE# Output Valid Delay	5	10	ns	
t45	MWE# Output Active Delay	6	12	ns	

Table 4.8 FS-Link Interface Timing Parameters (All the signals are based on PCICLK rising edge)

Symbol	Parameter	Min	Max	Units	Notes
FS-Link Interface					
t50	HA[31:16] Input Setup Time	5		ns	
t51	HA[31:16] Input Hold Time	3		ns	
t52	HA[31:16] Output Valid Delay	6	17	ns	
t54	SADS#/SRDY# SHOLD Input Setup Time	5		ns	
t55	SADS#/SRDY# SHOLD Input Hold Time	3		ns	
t56	FADS#/FRDY#, FHOLD Output Valid Delay	6	16	ns	
t57	DEVDET# Output Valid Delay	6	16	ns	

Table 4.9 Device Detect Interface Timing Parameters

Symbol	Parameter	Min	Max	Units	Notes
Device Detect Interface					
	DEVDET# Output Active Delay from PCICLK/2 rising				
	DEVDET# Output Inactive Delay from PCICLK/2 rising				

4.4 Measurement and Test Conditions

Figure 4.9 and 4.10 define the conditions under which timing measurements are made. The component test guarantees that all timings are met with minimum clock slew rate (slowest edge) and voltage swing. The design must guarantee that minimum timings are also met with maximum clock slew rate (fastest edge) and voltage swing. In addition, the design must guarantee proper input operation for input voltage swings and slew rates that exceed the specified test conditions.

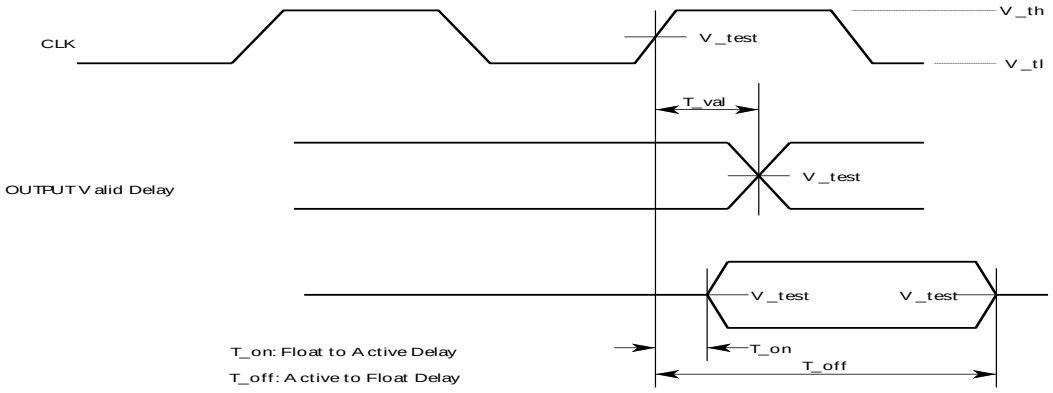


Figure 4.5 Output Timing Measurement Conditions

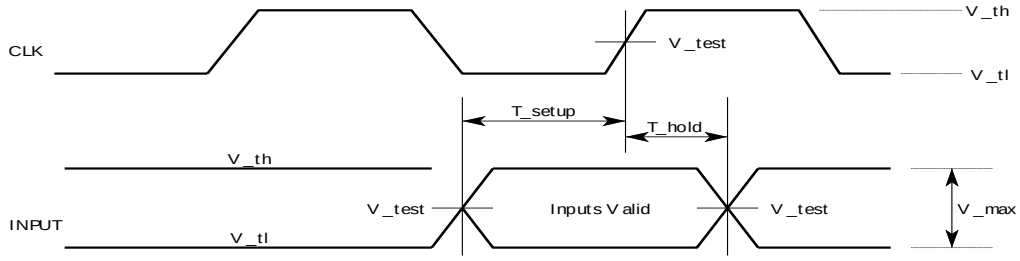


Figure 4.6 Input Timing Measurement Conditions

Table 4.10 Measure and Test Condition Parameters

Symbol	5V Signaling	Units
V_{th}	2.4	V (Note)
V_{tl}	0.4	V (Note)
V_{test}	1.5	V
V_{max}	2.0	V (Note)
Input Signal Edge Rate	1 V/ns	

NOTE:

The input test for the 5V environment is done with 400 mV of overdrive (over V_{ih} and V_{il}). Timing parameters must be met with no more overdrive than this. V_{max} specifies the maximum peak-to-peak waveform allowed for testing input timing.

Part III

1. SiS85C497 AT Bus Controller &Megacell (ATM)

1.1 Features

- **Fast- Slow Link Interface**
  - Provides the FS-Link Interface between PCI/CPU/Memory (fast machine) and ISA Bus Controller (slow machine)
- **100 % ISA Compatible**
  - ISA Master/Slave Interface
  - Directly Drives 5 ISA Slots
  - Supports ISA from 6.67 to 8.33 Mhz
  - Supports AT bus slew rate control.
- **Integrated 8-bit BIOS Timer**
- **De-turbo Function Switch**
- **Arbitration for ISA Devices**
  - Fast-Link Machine
  - DMA (include ISA master)
  - Refresh
- **XD Bus Peripheral Support**
  - Provides XD Bus data
  - Controls Lower ISA Data Byte Transceiver
  - Built-In Real Time Clock (option)
  - Coprocessor Error Reporting
- **Integrates the Functionality of Two 82C37A DMA Controllers**
  - 32-bit Paged Mapping
  - Seven Independently Programmable Channels
  - Compatible DMA Transfers
- **Integrates the Functionality of one 82C54 Timer**
  - System Timer
  - Refresh Request
  - Speaker Tone Output
- **Integrates the Functionality of two 82C59 Interrupt Controllers**
  - 14 Independently Programmable Interrupts Supported
  - IRQ[3-7], IRQ[9:12], IRQ[14-15] are Edge/Level Trigger Programmable
- **Non-Maskable Interrupts (NMI)**
  - PCI System Errors for DRAM parity checking
  - ISA Parity Errors
- **Supports SMM and PMU Features:**
  - Supports the SMI and the IRQ-SMI
  - CPU Stop Clock Function
  - Supports STPCLK# Control with Clock Throttling and Clock Scaling.
  - Four Power Saving States (normal/doze/standby/suspend)
  - Fast, Slow, and Generic System Timers

- Supports the APM control
- Supports the EXTSMI control
- System Event Monitoring and the Power Saving Control
- Supports 2 General Purpose I/O's (GPIO's) (option)
- Supports Overlaid SMRAM to A-Segment, B-Segment, or E-Segment.
- Supports I/O Restart.

- **Keyboard Controller Emulation of Fast A20GATE and CPU Reset.**
- **Supports Flash Memory**
- **Reset and Clock Circuits**
- **160-Pin PQFP**
- **0.6um Low Power CMOS Technology**

1.2 Architectural Overview

The major functions of the 85C497 component are broken up into blocks as shown in the following figure. A description of each block is provided below.

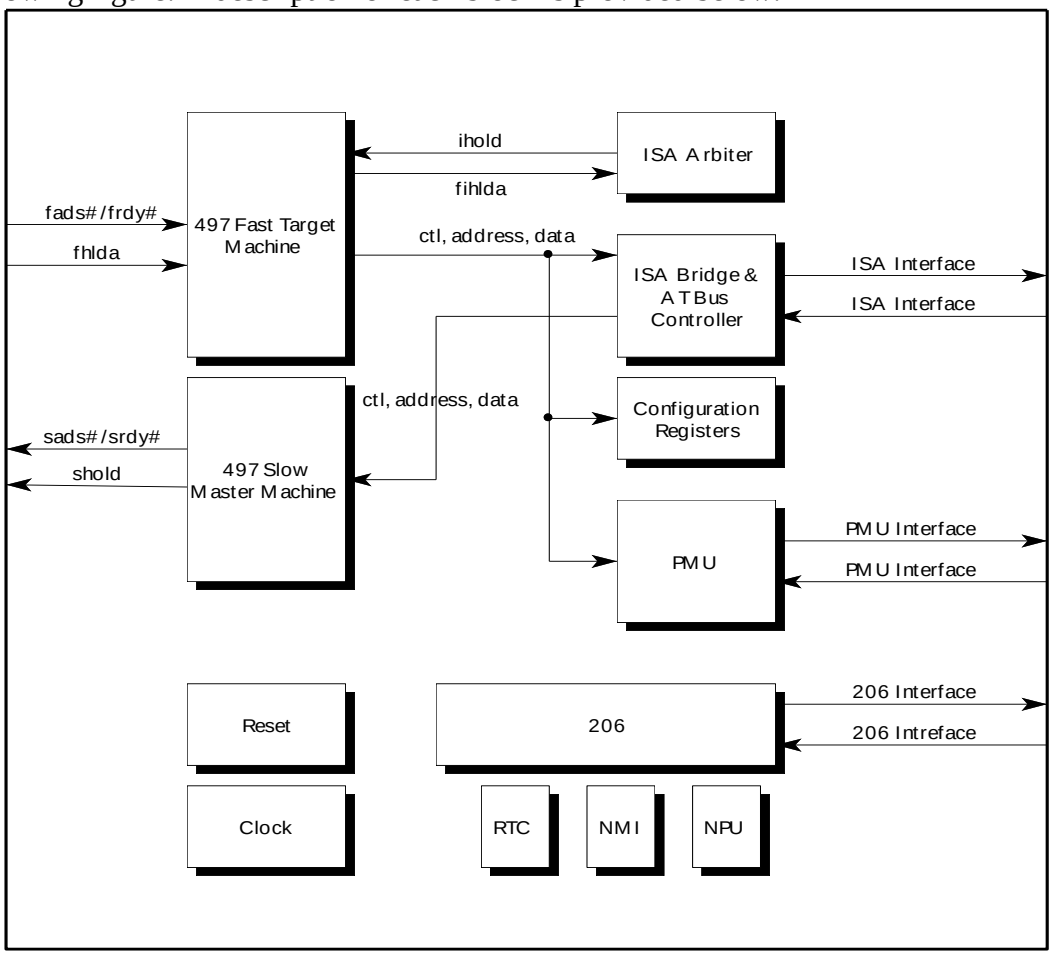


Figure 1.1 SiS85C497 Functional Block Diagram

FS-Link Bus Interface

The FS-Link Bus Interface provides the interface between the fast machine and the slow machine. The fast machines include CPU, PCI, and main memory and are running at a higher frequency (around 25 Mhz to 50 Mhz). The slow machines include ISA master, DMA, and ISA slave and are running at a lower frequency (around 8 Mhz). The 85C497 provides both a master and slave to the fast machine bus. As a slow machine master, the 85C497 runs cycles on behalf of DMA and ISA masters. The 85C497 will transfer a maximum of one dword when accessing faster machine's memory. The 85C497 does not generate I/O cycles to the FS-Link Bus as a master. As a slave, the 85C497 accepts cycles initiated by FS-Link Bus masters targeted for the 85C497's internal register set or the ISA bus. The 85C497 will accept a maximum of one dword data transaction before terminating the transaction.

SiS85C496 implements two link machines, fast master machine and slow target machine, the same as SiS85C497 fast target machine and slow master machine for the following purposes. When CPU master or PCI master wants to access ISA slave, it requests fast master machine, transfers bus cycle information to the 497 fast link target machine. The 497 fast link target machine converts cycle to AT bus controller for the ISA bus access. For ISA master, on the other hand, when it wants to access resource such as main memory, the AT bus controller requests FS-link bus to the slow link master machine, the 497 slow link master machine converts ISA cycle to the 496 slow link target machine for the resource access.

The 85C497 also provides support for system error reporting by generating a Non-Maskable-Interrupt (NMI) when SERR# is driven active.

The 85C497, as a resource, can be locked by any PCI master. In the context of locked cycles, the entire 85C497 subsystem (including the ISA bus) is considered as single resource.

The 85C497 supports the CPU and PCI interfaces and is running frequency independently from CPU or PCI's frequency. The 85C497's operating frequency ranges from 6.67 to 8.33 Mhz.

ISA Decode

The 497 contains address decoder to decode DMA initiated cycles and ISA master cycles.

ISA Bus Interface

The 85C497 ISA Bus Interface accepts those cycles from Fast Link Machine and then translates them onto the ISA bus. It also requests Slow Link Machine to generate PCI cycles on behalf of DMA or ISA master. The ISA bus interface thus contains a standard ISA Bus Controller (IBC) and a Data Buffering logic. IBC provides all the ISA control such as ISA command generation, I/O recovery control, wait-state insertion, and data buffer steering. The FS Link to/from ISA address and data bus buffering are also all integrated in 85C497. 85C497 can directly support six ISA slots without external data or address buffers.

Standard ISA bus refresh is requested by Counter 1 and then performed via the IBC. IBC generated the pertinent command and refreshes address to ISA bus. Since ISA refresh is

transparent to Fast Link Machine and DMA cycles, an arbiter is employed to resolve the possible conflicts among these cycles.

DMA

The 497 contains a seven-channel DMA controller. The channel 0 to 3 is used for 8-bit DMA devices while channel 5 to 7 is for 16-bit devices. The channels can also be programmed for any of the four transfer modes, which include single, demand, block and cascade. Except in cascade mode, each of the three active transfer modes can perform three different types of transfers which include read, write and verify. The address generation circuit in 497 cam only support 24-bit address for DMA devices.

Timer

The 497 contains 3 counter/timer. They are equivalent to 82C54 programmable interval timer. The counters use a division of 14.318 MHz OSC input as clock. The outputs of the timers are directed to key system functions. Counter 0 is connected to the interrupt controller IRQ 0 and provides a system timer interrupt for a time-of day, diskette time-put, or the other system timing function. Counter 1 generates a refresh request signal and Counter 2 generates the tone for the speaker.

Beside these three timer, the 497 provides a 16-bit timer. This timer is programmable and can be used by the system BIOS to produce delay loops. The clock source of this timer is 2.159 Mhz (derived by dividing 14.318 MHz by 7).

Interrupt Controller

The 85C497 provides an ISA compatible interrupt controller that incorporates the functionality of two 82C59 interrupt controllers. The two interrupt controllers are cascaded so that 14 external and two internal interrupts are possible. The support PCI interrupts, these interrupts are edge/level triggered programmable to each individual channel.

Power Management Unit

The 85C497 implements the power management unit to meet the Green PC's requirement.

2. Detailed Functional Description

2.1 FS-Link Interface

2.1.1 Fast Link Machine

For concurrence detecting PCI and ISA cycle, PCI target machine starts fast link machine cycle by asserting FADS#/FRDY# in the next PCI clock after FRAME# is asserted. If FADS#/FRDY# is asserted over three PCI clocks, the AT cycle will proceed normally. Others the AT cycle will be abort. After addresses, commands, byte enables and data transfer from 496 to 497 via HA[31:16], fast link machine into wait state until SADS#/SRDY# is asserted one PCI clock. If write cycle then fast link machine into idle state. In read one word cycle, it will latch data and go into idle state. In read double words, it will latch low word and then high latch word in the next PCI clock and go into idle state.

For pin counts consideration, the data communication between 496 and 497 via HA[31:16]. When FADS/FRDY# is asserted, the first cycle is turn around. The high order addresses(A[31:16]) are transferred in the second cycle, the byte enables/commands are transferred in the third cycle and the low order addresses(A[15:0]) are transferred in the forth cycle. If write cycle and the ninth bit of command is zero, low word data and high word data are transferred in the following two PCI clocks. The format of command is the same as PCI and located in the second four bits of command/byte enable. The first four bits are four byte enables. The ninth bit is the delay IRDY# which can indicate the write data whether is valid or not.

For some VL add on cards, HA[23:2] are regarded as LA[23:17] and SA[16:2], so between fast link machine turn around state and SADS#/SRDY# is asserted, the values of HA will still keep the same values as the addresses of CPU or PCI master.

There are three types of cycles are transferred from 496 to 497 by fast link machine. These cycles are AT cycle, 497 PCI configuration cycle and post memory write cycle. The behaviors of these cycles will be described in detail in the following sections.

AT Cycle:

The AT cycles include memory read/write and I/O read/write and interrupt acknowledge. The size of data are regarded as 32 bits by fast link machine. Although there are 16 bits or 8 bits devices in ISA bus, the ISA bus controller in 497 will convert the size of cycles between fast link machine and ISA bus. Because fast link machine and ISA bus are asynchronous, the signals between the two subsystems must be synchronized by a synchronizer so that can avoid logic errors making by the metastable phenomenon.

The following two figures shows that CPU reads/writes ISA target through fast-link machine.

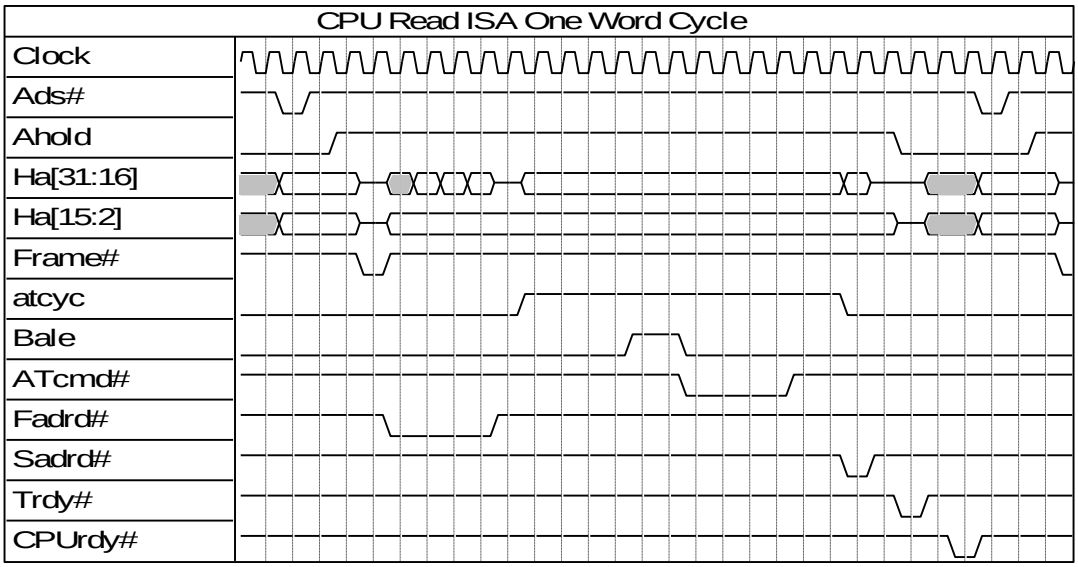


Figure 2.1 CPU Reads ISA Slave

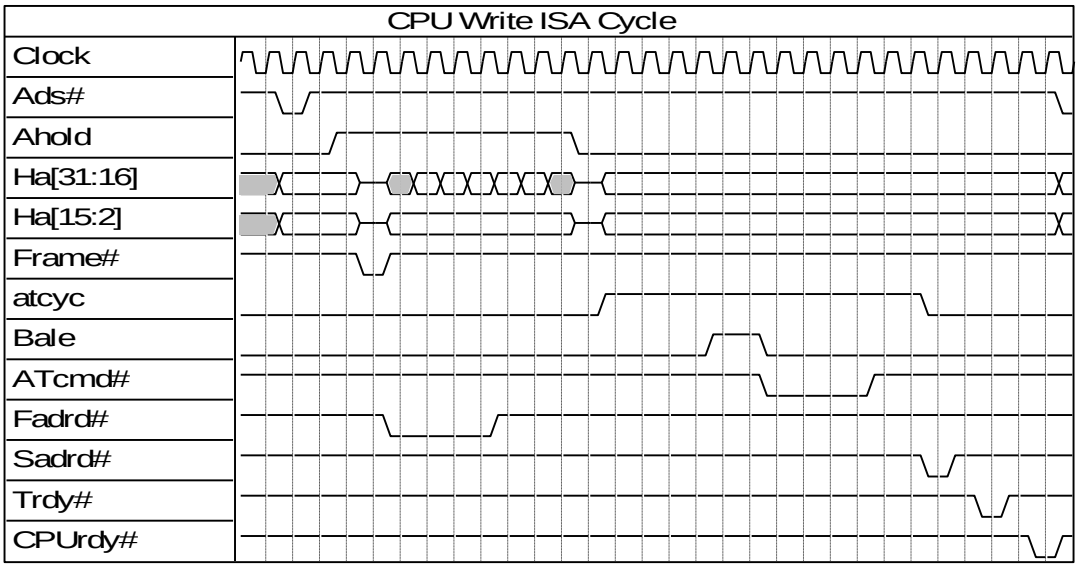


Figure 2.2 CPU Writes ISA Slave

497 Configuration Cycle:

The 497 configuration registers are physically located in 497 and logically belong to 496. So CPU or PCI master must be through fast link machine to access 497 configuration registers.

The following two figures shows that CPU reads/writes 497 configuration registers through fast-link machine.

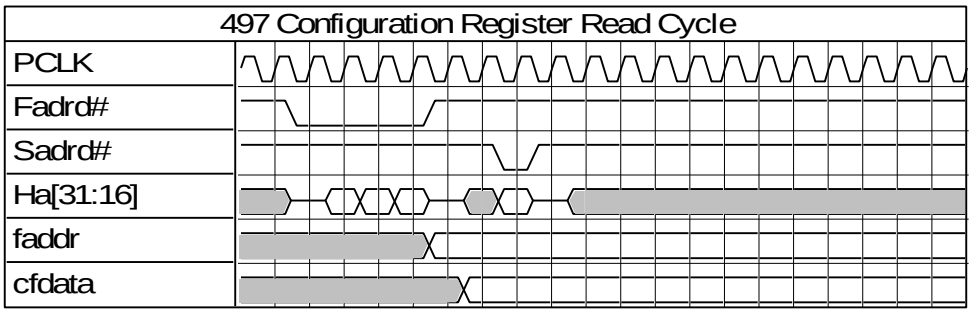


Figure 2.3 497 Configuration Register Read Cycle

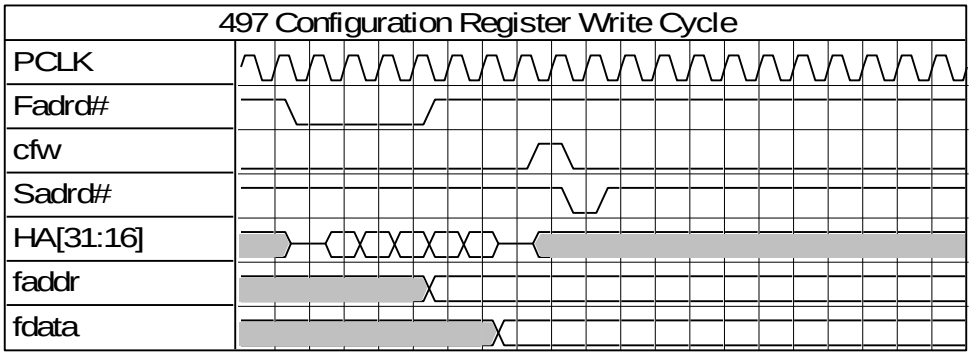


Figure 2.4 497 Configuration Register Write Cycle

Post Memory Write:

The 497 implements a 32-bit one-level post memory buffer. An AT memory write can be posted any time when the posted write buffer is empty and write posting is enabled. (bit0 of configuration register C6h is set to 1). If ISA bus is busy and another cycle accessing 497 will be pended until the cycle is completed in ISA bus. Interrupt acknowledge, Memory read, I/O read, I/O write, Configuration read and Configuration write cycles do not use the 32-bits post write buffer.

The following figure shows that CPU writes the posted write buffer through fast-link machine.

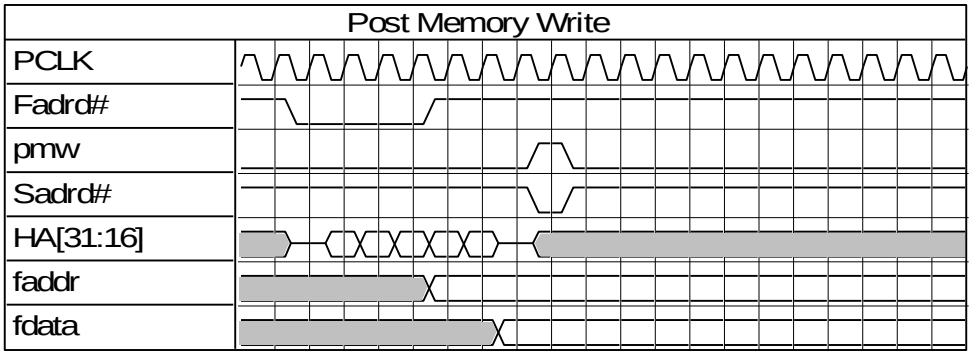


Figure 2.5 Post Memory Write Cycle

The protocol of fast link machine is illustrated below :

Cycle		Description
Idle (the first cycle)	T0	High Impedance
High Address	T1	496 F.L.M.Addr[31:16] ⇒ HA[31:16] ⇒ 497 F.L.M.
Command/Byte Enable	T2	496 F.L.M. PCI IRDY# ⇒ HA24 ⇒ 497 F.L.M. 496 F.L.M. Command[3:0] ⇒ HA[23:20] ⇒ 497 F.L.M. 496 F.L.M. Byte Enable[3:0] ⇒ HA[19:16] ⇒ 497 F.L.M.
Low Address	T3	496 F.L.M. Addr[15:0] ⇒ HA[31:16] ⇒ 497 F.L.M.
Low Word data (write)	T4	496 F.L.M. data[15:0] ⇒ HA[31:16] ⇒ 497 F.L.M.
High Word data (write)	T5	496 F.L.M.data[31:16] ⇒ HA[31:16] ⇒ 497 F.L.M.
Turn Around	T4 or T6	High Impedance
Wait		496 F.L.M. Addr[15:2] ⇒ HA[15:2] 497 F.L.M. Addr[31:16] ⇒ HA[31:16]
Last Wait (read)		497 one word ⇒ HA[31:16] ⇒ 496 F.L.M. 497 low word of double word ⇒ HA[31:16] ⇒ 496 F.L.M.
Read Double Word data		497 high word of double word ⇒ HA[31:16] ⇒ 496 F.L.M.
Idle (the last cycle)		High Impedance

NOTE: F.L.M. means Fast Link Machine.

2.1.2 Slow-Link Machine

When ISA master or DMA want to access the upper memory(onboard, VL or PCI), 497 will assert SHOLD. After arbiter asserting FHLDA to indicate that host and PCI bus are held by DMA or ISA master, 497 will assert SADS#/SRDY# one PCI clock. The first two PCI clocks are turn around. Then the high order addresses are transferred in the third cycle and the command/byte enables are transferred in the forth cycle and the low order addresses are transferred in the fifth cycle. If write cycle then the write data are transferred in the following cycle. After the need data are transferred, the slow link machine into wait state until FADS#/FRDY# is asserted by 496 to indicate the cycle is ready and if read cycle the data is valid in HA[31:16]. Because ISA master or DMA can only access the upper memory, the format of command/byte enables is the first four bits are byte enables and the fifth bit is write/read bit.

When the cycle of ISA master or DMA hits upper memory, IORDY will be asserted low until FADS#/FRDY# is asserted so that ISA master or DMA can pend its cycle after the accessing upper memory cycle is complete

Because slow link machine and ISA bus are asynchronous, the signals between the two subsystem must be synchronized by a synchronizer so that can avoid logic errors making by the metastable phenomenon.

The following two figures shows that ISA master or DMA reads/writes onboard memory through slow-link machine.

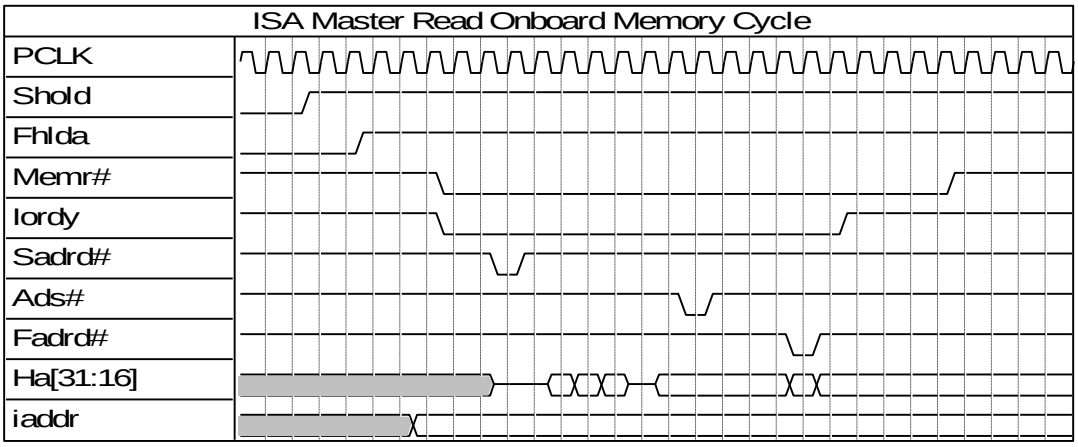


Figure 2.6 ISA Master Read Onboard Memory Cycle

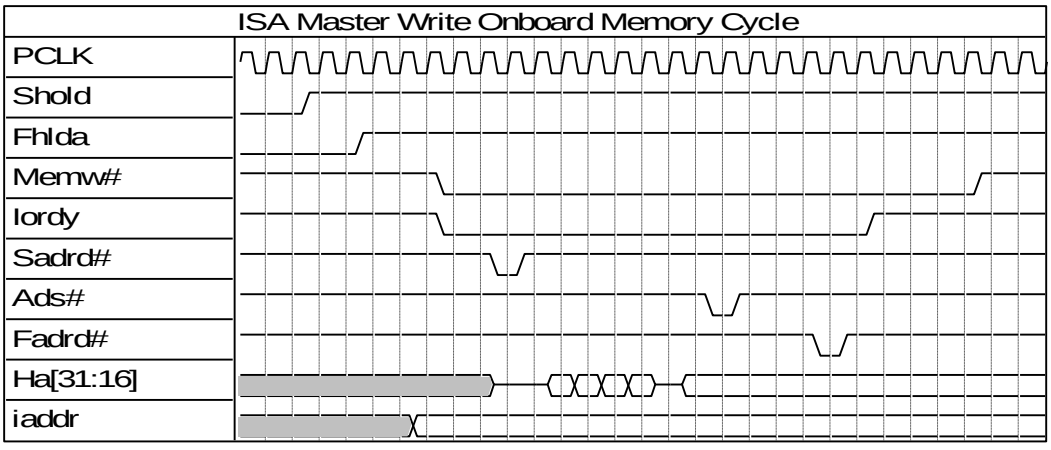


Figure 2.7 ISA Master Write Onboard Memory Cycle

The protocol of slow link machine is illustrated below :

Cycle		Description
Idle (the first cycle)	T0	497 S.L.M to inform 496 tristate HA
Idle (the second cycle)	T1	High Impedance
High Address	T2	497 S.L.M.Addr[31:16] ⇒ HA[31:16] ⇒ 496 S.L.M.
Command/Byte Enable	T3	497 S.L.M. MEMW#/MEMR# ⇒ HA20 ⇒ 496 S.L.M. 497 S.L.M. Byte Enable[3:0] ⇒ HA[19:16] ⇒ 496 S.L.M.
Low Address	T4	497 S.L.M. Addr[15:0] ⇒ HA[31:16] ⇒ 496 S.L.M.
One Word data (write)	T5	497 S.L.M. data[15:0] ⇒ HA[31:16] ⇒ 496 S.L.M.
Turn Around	T5 or T6	High Impedance
Wait		496 S.L.M. Addr[31:2] ⇒ HA[31:2]
Last Wait (read)		496 S.L.M. Data[15:0] ⇒ HA[31:16] ⇒ 497 S.L.M.
Idle (the last cycle)		High Impedance

NOTE: S.L.M. means Slow Link Machine.

2.2 ISA Interface

2.2.1 ISA Interface

The 85C497 ISA bridge accepts the cycles from Fast-Slow Link Interface and then translates the cycles into the ISA bus. It also requests the Fast-Slow Link Interface to translates the cycle into 85C496 Fast-Slow Link Interface on behalf of DMA or ISA master. The ISA bridge contains a standard AT bus controller (ATBC). ATBC provides all the ISA control parts, such as ISA command generation, wait state generation, I/O recovery time control and data byte swapping. The ISA bus refresh is requested by counter 1 and then performed via the AT bus controller. AT bus controller generates the command and refresh address to the ISA bus.

2.2.2 ISA Clock Generation

The 85C497 provides a flexible software controlled selection of the clock used for the AT bus controller. Bus clock is determined by the ISA configuration register and can be selected as 1/3 or 1/4 of the PCI clock or 7.159 MHz.

Table 2.1 SYSCLK Generation from PCICLK

PCICLK (MHz)	DIVISOR (Programmable)	SYSCLK (MHz)
25	3	8.33
33	4	8.33

2.2.3 Real Time Clock Selection

The 85C497 contains a internal Real Time Clock(RTC). If the internal RTC is enabled, then the system uses this RTC and ASRTC, DSRTC#, and RWRTC# are driven to this RTC. If the internal RTC is disabled, then ASRTC, DSRTC#, and RWRTC# drive the external RTC.

2.3 DMA Controller

Two DMA controllers are connected in such a way as to provide the user with four DMA channels (DMA1) for 8-bit transfers and three DMA channels (DMA2) for 16-bit transfers (the first 16-bit DMA channel is used for cascading). Included as part of the DMA subsystem is the Page register and High Page register which is used to supply the DMA and drive the upper address to support 32-bit address DMA. The channels can be programmed into four operating modes which include single transfer, demand transfer, block transfer and cascade mode. DMA also supports three transfer types which included read, write and verify.

2.4 Address Decoding

The 85C497 contains two decoders : one to decode PCI/CPU master cycles and DMA/ISA master cycles for accessing the 85C497 internal register, BIOS Memory Space, and utility bus encoded chip selects. The decoder decodes the address from the ISA address bus for PCI/CPU and DMA/ISA master cycles. The other decoder handles the cycles that DMA/ISA master accesses the PCI or the onboard memory.

2.4.1 85C497 I/O Addresses

These addresses are the internal, non-configuration 85C497 register locations and are shown in the 85C497 Address Decoding Table 2.8. These addressees are fixed. Noted that the PCI Configuration Registers are accessed with PCI configuration cycles. All of the registers are 8 bit registers. Accesses to these registers must be 8 bit accesses.

Table 2.8 85C497 Address Decoding

Address	Type	Names	Block
0000h	r/w	DMAS CH0 Base and Current Address	DMA
0001h	r/w	DMAS CH0 Base and Current Count	DMA
0002h	r/w	DMAS CH1 Base and Current Address	DMA
0003h	r/w	DMAS CH1 Base and Current Count	DMA
0004h	r/w	DMAS CH2 Base and Current Address	DMA
0005h	r/w	DMAS CH2 Base and Current Count	DMA
0006h	r/w	DMAS CH3 Base and Current Address	DMA
0007h	r/w	DMAS CH3 Base and Current Count	DMA
0008h	r/w	DMAS Status(r) Command(w) register	DMA
0009h	r/w	DMAS Request register	DMA
000Ah	r/w	DMAS Command(r) Single Mask Bit(w)	DMA
000Bh	wo	DMAS Mode register	DMA
000Ch	wo	DMAS Clear Byte Pointer	DMA
000Dh	wo	DMAS Master Clear	DMA
000Eh	wo	DMAS.Clear Mask register	DMA
000Fh	r/w	DMAS All Mask Register Bits	DMA
0020h	r/w	INTM Control register	INTC
0021h	r/w	INTM Mask register	INTC
0040h	r/w	Timer Counter 1 - Counter 0 Count	CTC
0041h	r/w	Timer Counter 1 - Counter 1 Count	CTC
0042h	r/w	Timer Counter 1 - Counter 2 Count	CTC
0043h	wo	Timer Counter 1 Command Mode register	CTC
0061h	r/w	Port B register	Control
0070h	r/w	NMI enable and RTC address index register	Control
0071h	r/w	RTC data index register	Control
0081h	r/w	DMA Channel 2 Page register	DMA
0082h	r/w	DMA Channel 3 Page register	DMA
0083h	r/w	DMA Channel 1 Page register	DMA
0087h	r/w	DMA Channel 0 Page register	DMA
0089h	r/w	DMA Channel 6 Page register	DMA
008Ah	r/w	DMA Channel 7 Page register	DMA
008Bh	r/w	DMA Channel 5 Page register	DMA
00A0h	r/w	INTS Control register	INTC
00A1h	r/w	INTS Mask register	INTC
00C0h	r/w	DMAM CH0 Base and Current Address	DMA
00C2h	r/w	DMAM CH0 Base and Current Count	DMA
00C4h	r/w	DMAM CH1 Base and Current Address	DMA
00C6h	r/w	DMAM CH1 Base and Current Count	DMA
00C8h	r/w	DMAM CH2 Base and Current Address	DMA
00CAh	r/w	DMAM CH2 Base and Current Count	DMA
00CCh	r/w	DMAM CH3 Base and Current Address	DMA
00CEh	r/w	DMAM CH3 Base and Current Count	DMA

00D0h	r/w	DMAM Status(r) Command(w) register	DMA
00D2h	r/w	DMAM Request register	DMA
00D4h	r/w	DMAM Command(r) Single Mask Bit(w)	DMA
00D6h	wo	DMAM Mode register	DMA
00D8h	wo	DMAM Clear Byte Pointer	DMA
00DAh	wo	DMAM Master Clear	DMA
00DCh	wo	DMAM Clear Mask register	DMA
00DEh	r/w	DMAM All Mask Register Bits	DMA
00F0h	wo	Coprocessor Error	Control
0481h	r/w	DMA CH2 High Page register	DMA
0482h	r/w	DMA CH3 High Page register	DMA
0483h	r/w	DMA CH1 High Page register	DMA
0487h	r/w	DMA CH0 High Page register	DMA
0489h	r/w	DMA CH6 High Page register	DMA
048Ah	r/w	DMA CH7 High Page register	DMA
048Bh	r/w	DMA CH5 High Page register	DMA
04D0h	r/w	INTM Level/Edge Trigger register	INTC
04D1h	r/w	INTS Level/Edge Trigger register	INTC

2.4.2 BIOS Memory Space

85C497 has a 128 Kbyte BIOS memory space. It is locatedat 000E0000h to 000FFFFFh and is aliases at FFFE0000h to FFFFFFFFh (top of 4 Gbyte) and 00FE0000h to 00FFFFFh (top of 16 Mbyte). The 128 Kbyte is always enabled an is split into two 64 Kbyte blocks (E segment and F segment). The upper 64 Kbyte is treated as ROM area and can be disabled if bit 5 of ISA BIOS Configuration Register (D0h) is 0. The lower 64 Kbyte is also treated as ROM area and can be disabled if bit 6 of ISA BIOS Configuration Register is 0. ISA master can just access BIOS in the 00E0000h to 00FFFFFh region.

85C497 supports an additional 384 Kbyte BIOS space. That is to say, the 85C497 supports up to 512kbyte BIOS space. The extra BIOS space is located at FFF80000h to FFFDFFFFh and this region can be accessed by PCI master. The ISA master can not accessed the additional BIOS space if it is located at 00F8000h to 00FDFFFFh. When ISA BIOS Configuration Register (D0h) bit 7 is set to 1, any memory access within this region is forward to ISA bus and ROMCS# will be asserted. All PCI substractively decoded cycle accessing the enabled BIOS space will be forward to ISA bus. In order to the PCI increment latency guideline, any PCI burst read access to the BIOS space will invoke “disconnect target termination” because the BIOS device has long access times and is 8 bit wide.

PCI master accessed to the enabled sections of the BIOS space (000E0000h to 000FFFFh or 00F80000h to 00FFFFFFh) will activate the encoded ROMCS# signal. The encoded ROMCS# is generated from the ISA SA and LA address bus. The encoded ROMCS# is disabled during refresh cycles.

ISA master access to the enabled sections of the BIOS space (000E0000h to 000FFFFFh) will activated the enacted ROMCS# signal. The encoded ROMCS# is generated from the ISA SA

and LA address buses. Encoded ROMCS# will not generated for the ISA master cycles access from 00F80000h to 00FFFFFFh.

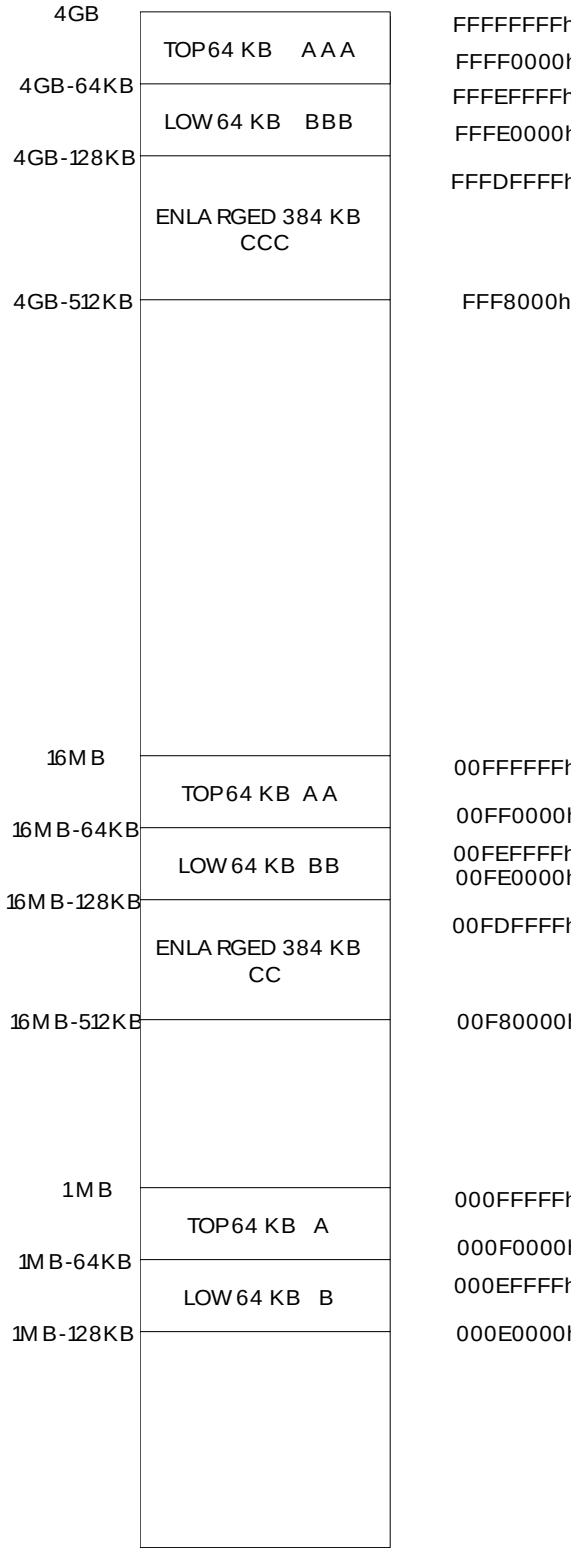


Figure 2.27 BIOS Space Decode Map

The BIOS space decode map, Figure 2.27, shows the possible BIOS spaces and the aliases throughout the memory space. BIOS is not directly accessible at AA, BB, CC address locations. These spaces, AA, BB, CC are reserved on the ISA bus since the ISA bus will see these addresses when a CPU/PCI master requests BIOS accesses at area AAA, BBB, and CCC.

2.4.3 Positive Decode to PCI

An ISA master can access PCI memory, but not I/O device residing on the PCI bus. The ISA/DMA address decoder determines several memory address regions that should be directed towards the PCI Bus. These regions are in the list below. Region "a" through "h" are fixed and can be enabled or disabled optionally. Region "i" indicates a space with a programmable upper boundary up to 16 Mb. Within this region a hole can be opened. Its size and location are programmable to allow a hole to be opened in the ISA memory space. A memory address above the 16 MB will be forwarded to be PCI Bus automatically. This is possible only during DMA cycles in which the DMA has been programmed for 32 bit addressing above 16 MB.

a. Memory : 0-640 KB
b. Memory : 640-768 KB
c. Memory : 768-800 KB
d. Memory : 800-832 KB
e. Memory : 832-864 KB
f. Memory : 864-896 KB
g. Memory : 896-960 KB
h. Memory: 960KB-1 MB
i. Memory : 1 MB-to-16 MB within which a hole can be opened.
j. Memory : > 16 MB automatically forwarded to PCI

2.5 Data Buffering

Posted write buffer contains the address, data and control posted write buffer. PCI master memory write cycles destined to ISA memory are buffered in a 32-bit Posted Write Buffer. As soon as a PCI master has posted a memory write into the posted write buffer, the buffer is scheduled to be written to the ISA Bus. Any subsequent PCI master cycles to the 85C497 (including ISA Bus) will be pending until the posted write buffer is empty. If the posted write buffer is disabled, the 85C497's response to a PCI master memory write is dependent on the state of the ISA Bus. If the ISA Bus is available and the posted write buffer is disabled, the cycle will immediately be forwarded to the ISA Bus. Prior to granting the ISA Bus to an ISA master or the DMA, the PCI master posted memory write buffer is flushed. Also, as long as the ISA master or DMA owns the ISA Bus, the posted write buffer is disabled. A PCI master write can not be posted while an ISA master or the DMA owns the ISA Bus.

2.6 85C497 Timers

2.6.1 Interval Timers

The Interval Timers in 85C497 are fully compatible to 82C54. The methods accessing the Interval Timers are the same as 82C54.

2.6.2 BIOS Timer

The BIOS Timer in the 85C497 has a time base of 2.159 Mhz. It must be programmed before it is used. Accessed to the timer can be enabled or disabled by setting ISA Configuration Register (73h~74h bit 0). The timer continues to count even if it is disabled. The I/O address of the timer is relocatable by the software. The timer includes a 32-bit register mapped in the I/O space on the location determined by the value written into the BIOS Timer Base Address Register. A write to the BIOS timer initiates a counting sequence. The timer can be initiated a 8 bit, 16 bit or 32 bit data (the upper 24bit is don't-care). The timer will start decrement until it reaches zero. The timer will stop decrementing until it is initiated again.

2.7 Interrupt Controller

There are two interrupt controllers which are fully compatible to 82C59 residing in 497. One is master and the other is slave. The INTR output of slave is connected to the IRQ 2 channel of master. The INTR output of master is connected directly to CPU.

There are two IRQ trigger modes in which this Interrupt Controller operates. One is ISA compatible mode. In this mode all the trigger mode setup is according to ICW 1 (Initial Command Word) of both master and slave. The other mode is PCI compatible mode in which the setup is according to I/O port 4D0h and 4D1h.

2.8 Power Management Unit

There are three timers used to monitor system events. The system events include video BIOS access, IRQ requests, DMA requests, Hard disk/Floppy disk access, parallel port access, serial port access, I/O traps and PCI master request. It supports SMI and SMIRQ functions. It also support STOPCLK function. There are still another two timers used for throttling mode. In this mode the STOPCLK will be asserted periodically according to the system conditions.

Major Features

-Supports Intel , Cyrix and AMD CPU
-Supports CPU SMM mode and SMI interrupt
-Supports SMIRQ for non-SMI interrupt CPU
-Supports STOPCLK interrupt
-Supports clock scaling and clock throttling
-Supports I/O trap
-Supports EXTSMI break switch
-Supports APM control
-Supports system event monitoring of each state for power saving:
  VGA Access
  HDD Access
  FLOPPY DISK Access
  PARALLEL PORT Access

SERIAL PORT Access
IRQs
DMA Request
PCI Master Request
I/O trap
KBD/MOUSE

- Supports five timers for PMU control. Slow Timer (70 us to 42.67 minutes), Fast Timer (0.3s to 75s), Generic Timer (0.6s to 150s), Clock Throttling Timer (35 micro-second base), and SMI Block Timer (35 macro-second base).
- Supports up to 2~16 SMOUT power control signals for peripheral devices

PMU Events Definition

1. System Event: These events are used to reload timers.

PCI Master Request
I/O Trap
Serial Port Access
Parallel Port Access
Hard/Floppy Disk Access
KBD/Mouse Request
DMA Request
IRQs
Video BIOS Access

2. SMI# Request Events: These events are used to generate SMI# interrupt.

DMA Request
PCI Master Request
I/O Trap
Software SMI Request
EXTSMI#
KBD/MOUSE Request
Hard/Floppy Access
Video Access Request
Parallel Port Request
Serial Port Request
Fast Timer Expire
Generic Timer Expire
Slow Timer Expire
IRQs
GPIO

3. STPCLK# Assertion Events: These events are used to assert STPCLK.

Fast Timer Expire
Generic Timer Expire
Slow Timer Expire
Software STPCLK Trigger
Throttlint_On Timer Expire

4. STPCLK# Deassertion Events: These events are used to deassert STPCLK#.

EXTSMI
IRQs
NMI#
DMA Request
PCI Master Request
Throttling_Off Timer Expire

The following diagram shows the architecture of SMI:

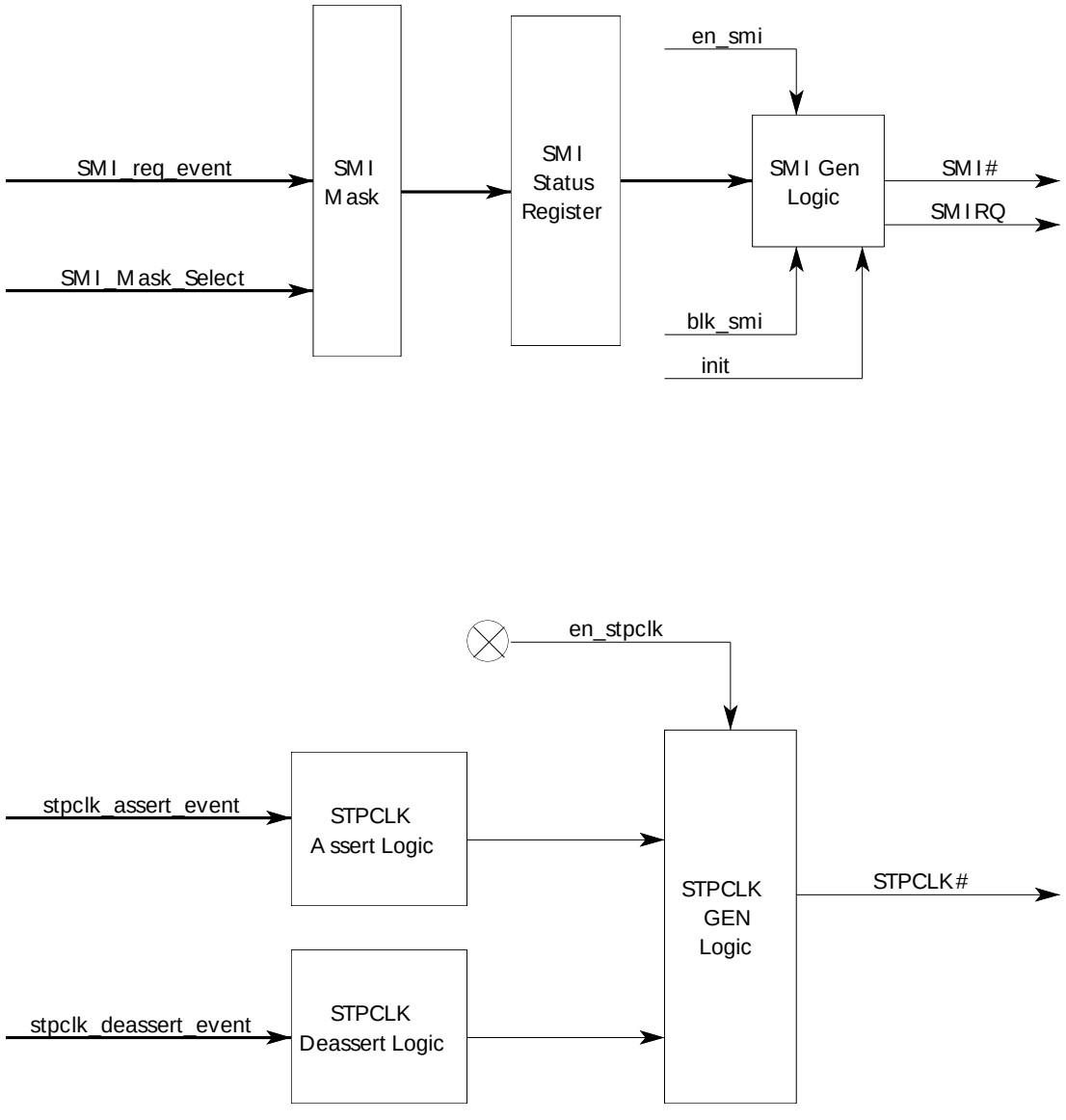


Figure 2.30 PMU Block Diagram

2.8.1 PMU Signal Description

There are six pins dedicated to the power management function on the 497: SMI#, STPCLK#/SUSP#/SMIACT#, EXTSMI#, SMOUT0/SMOUTW#, SMOUT1, DEVDET#, and three pins on the 496: SMIACT#/SMIADS#, SMI#, DEVDET#. Please refer to 496/497 Functional Pin Description sections for a detailed description of each pin.

2.8.2 Functional Description

The 496/497 power management provides four power management function.

1. SMM (System Management Mode) Function:

SMM is invoked through the assertion of SMI# pin. SMI is triggered through various hardware and software events. SMRAM is used to store the SMI code which is really the SMI interrupt handler routine. SMIRQ is an equivalent to the SMM for non-SMI CPU such as 486. IRQ10, IRQ11, IRQ12 or IRQ15 (user programmable) is generated instead of SMI# interrupt to the non-SMI CPU. Other than that, the condition of generating SMIRQ is the same as SMI.

2. Stop Clock Control Function:

The PMU uses STPCLK# to reduce the power consumption of CPU.

3. PMU timer:

There are five timers in the PMU unit. Fast timer, Slow timer and Generic timer are used to monitor the system events. These timers expire if the monitored events are idle for a period time. The system management is invoked to turn off the power of devices. Notice that although the timer intervals are different, these three timers can be used by any of the system events. SMIBLK timer is used to block internal SMI. The nested SMI is not supported by the PMU. If a SMI is asserted, the next coming SMI will be blocked until SMIBLK timer expires. Throttling timer is used for throttling mode. Two counts are used for this timer. The On_count is used to control the STPCLK# high time while the Off-count is used to control the STPCLK# low time.

4. APM (Advance Power Management) Interface:

Four APM registers are provided for the communication between OS and SMM.

2.8.3 System Event Timers

Two system event timers are used to indicate (through an SMI) that the system has been idle for a pre-programmed period of time. The Fast Timer consists of a count down timer that is decremented every 0.6 seconds. The Slow Timer consists of a count down timer that is decremented every 1.25 minutes. The Generic Timer consists of a count down timer that is decremented every 0.3 seconds. The monitor system events are all programmable. For the detailed programming information, please reference the Register Description.

The functional block is shown below:

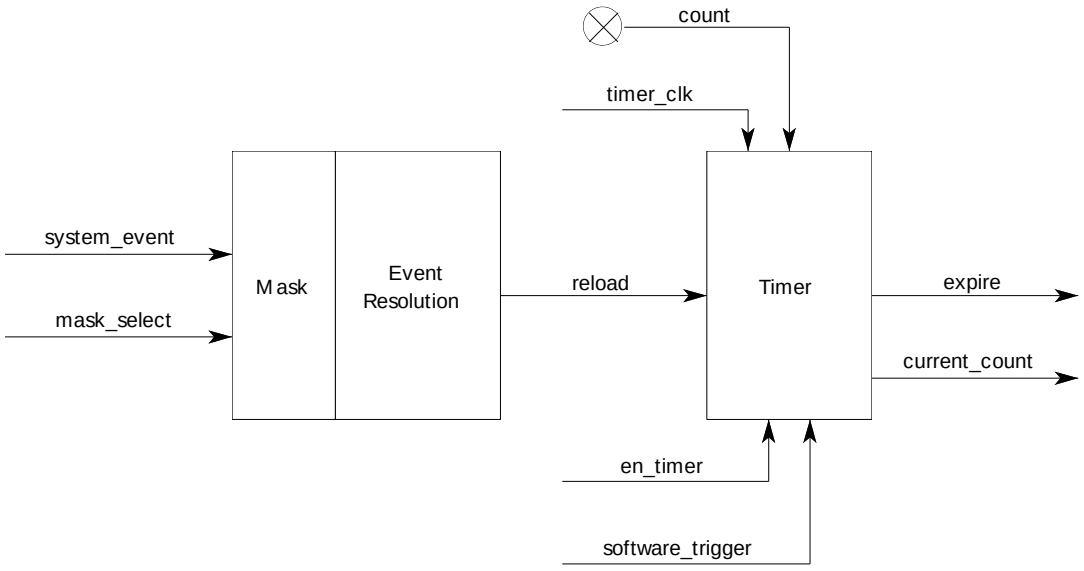


Figure 2.32 System Event Timer

The Mask is to decide which system event causes reload. The mask_select selects the system event. There are two control signals controlling the timer. The en_timer controls the output of expire. When en_timer is set, the timer will automatically reload the count and after a PCI clock, validate the output of expire. This is to prevent the timer expires right after the timer is enabled (the timers are free running internally. The expire signal is generated only when the timer is enabled). The soft_trigger provides another control for the timer. It allows the BIOS to reload timer by writing CF register 8Ch. When the timer expires, it remains in that state until the timer is disabled and re-enabled again.

An SMI# or STPCLK# interrupt is generated when any of these timers expire.

2.8.4 SMI Block Timer

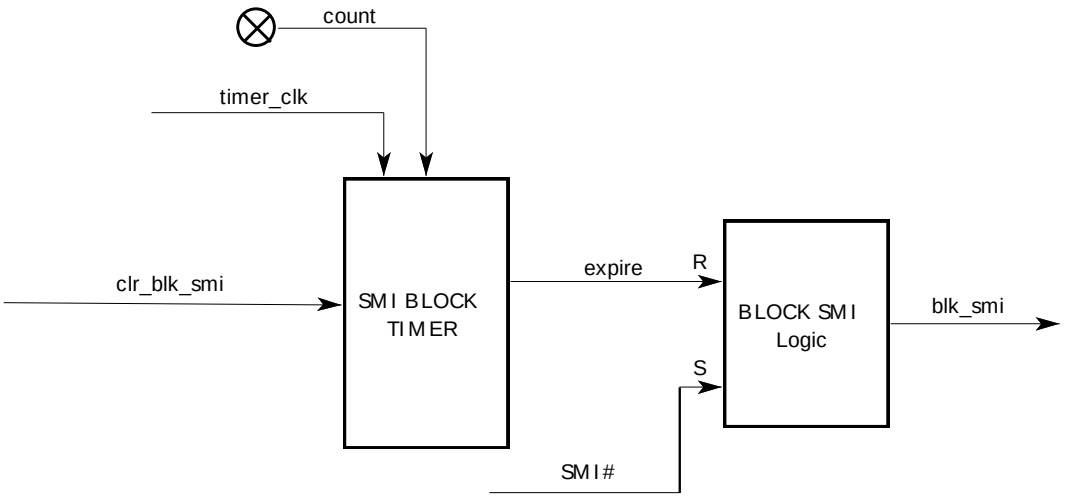


Figure 2.33 SMI Block Timer

The SMI Block timer is a 8 bits timer. The PMU doesn't support nested SMI, so when a SMI is active and the handler routine has not been executed yet, 497 will block all coming SMI in order to guarantee that only one SMI is issued. In the diagram above, when a SMI is issued, it will automatically set the blk_smi active. At the end of the SMI handler routine, BIOS will clear the blk_smi by activating clr_blk_smi. The blksmi timer will count down (in 35us time base) a predefined period. The predefined value is set according to the CPU resume time. It is different from one system to another. After the timer expires, it will inactivate the blk_smi allows the next SMI generation. An undefined count value is returned when the timers are disabled.

2.8.5 Clock Throttling Timer

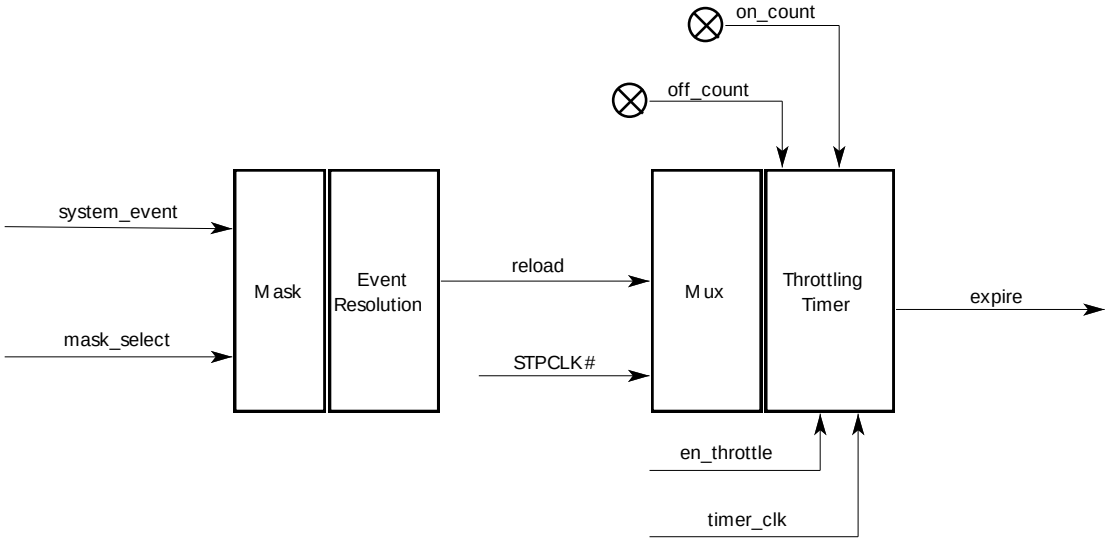


Figure 2.34 Clock Throttling Timer

Clock Throttling timer is a 8 bit timer. It has two counts. One is on_count, it specifies the period during which the STPCLK# is inactive, that is to say, the system is active (on). The other count is off_count, it specifies the period during which the STPCLK# is active, the system is in the Power Saving Mode. The counting unit for this timer is 35us.

The Mux block is to decide which value should be loaded into the timer. The select input is STPCLK#. If STPCLK# is high, Mux will select on_count, otherwise off_count. When the throttling mode is enabled, STPCLK# should be high and the on_count is loaded. During this period, the system events are allowed to reload the timer. When the timer expires, it will trigger the STPCLK#. After STPCLK# goes low, the Mux will select the off_count, after one PCI clock, the count is loaded into the timer. During this period, no reload condition is allowed. The Stop Grand Bus Cycle is not qualified to trigger timer because the counting unit is 35 us and during just one counting unit the Stop Grand Bus Cycle must have been issued. After the off_count expires, STPCLK# will be inactive. If during the off_count, a STPCLK deassertion event comes, the timer will reload the on_count and begin to count.

2.8.6 APM Ports

There are four CF registers (C8h~CBh) used as post box for APM. These four registers are read / writeable. They can be used to transfer messages between OS and SMM handler.

2.8.7 SMI Generation Logic

SMI generation logic controls the assertion of the SMI# and the recording of what event triggered SMI#. The events that trigger SMI# are either hardware events or software events. These events are listed in the beginning of this section named SMI request events. The hardware events is based on the current states of the system and cause the SMI#. Software events indicate that the OS is passing power management information to the SMI handler. Each event can be masked or unmasked.

Each SMI request event has its own select bit and status bit. The select bit is used to select the event to trigger SMI and the status bit is used to register whether this event has caused SMI or not. The status bits can be accessed by the BIOS. A 1 in the status means that this event has caused the SMI, 0 means this event did not trigger SMI. When read, the status is read. When write, the writing with 0 has no effect and writing with 1 causes the status bit to be cleared.

- **SMI Enable**

The SMI enable is to enable the whole SMI function of 497. When Enabled, 497 is able to trigger SMI. It is the responsibility of the SMI handler to make sure the enable bit are set appropriately before existing.

- **SMI Request Select Register and SMI Status Register**

SMI Request Select Register is located at CF register A2h~A3h. It select the system events to trigger SMI.

SMI Status Register is located at CF register A0h~A1h. It stores the information of which event cause SMI.

- **EXTSMI#**

The EXTSMI# is a asynchronous input. Similar to the Turbo/Deturbo input, the EXTSMI# input pin gives the system designer the capability to invoke SMM with external hardware. A typical use of the EXTSMI# input would be to connect it to a "break button" that would trigger SMI# in order to transition between the Power-On state and the Power-Saving state. A low-to-high will cause the EXTSMI# active. To implement the external debounce, when EXTSMI# transition occurs, it will automatically block the coming signals. When the EXTSMI# is recognized by the BIOS, the BIOS has to write the CF register 84h bit 1 to unblock the EXTSMI# before it exits the SMI routine.

- **SMI Block**

The PMU blocks any other SMI until handler of previous SMI has been executed. The block mechanism is described detail at the SMI Block Timer.

- **SMI Generator**

The SMI generator generates SMI# to CPU if one of programmable SMI request event is active. It will qualify blk_smi to make sure that no nested SMI and qualify en_smi to make sure that PMU function has been turned on. At last it should qualify STPCLK# and INIT to guarantee the relationship between these signals.

SMIRQ is the same as SMI except that it does not qualify STPCLK# and INIT. SMIRQ is passed to the CPU by INTR. If a system uses the SMIRQ instead SMI, it means that the CPU of this system does not support SMM. There is no relation issue between these signals because no STPCLK and INIT exist in such system.

The internal SMI is "ORed" with all SMI Status Registers' outputs. BIOS has to clear all the status bits by writing 1 to the register to stop the internal SMI.

The most important issue in SMI generator is the control of the output period of SMI. For Intel CPUs, SMI# is activated for four PCI clocks. For Cyrix CPU, the SMI# is activated for 4 PCI clocks, then switch the attribute of SMI# pin from output to input, because the CPU will recognize the input and begin to drive this pin. At the end of the handler, the CPU drives the SMI# high for one CPU clock. When 497 detects this transition, it will again switch the attribute of SMI# from input to output. For AMD Am486DXL CPU, 497 have to assert SMI# until first SMIADS# is issued by the CPU, then similarly, change the attribute of SMI# from output to input. At the end of the handler routine, like Cyrix CPU, the CPU will drive SMI# high for one CPU clock. When 497 detects this transition, it will change the attribute from input to output.

SMI# should be blocked during the active of INIT. Whenever 497 wants to assert SMI#, it must qualify INIT. If INIT is active, SMI should not be asserted. If SMI# is asserted coincidentally with the INIT, 497 has the responsibility to deassert SMI#. That is to say, INIT gets a higher priority. If an internal SMI has been blocked due to INIT, it can not drive SMI# until two CPU clocks after the inactivate of INIT.

2.8.8 STPCLK Control Logic

The STPCLK assertion logic is to decide which event cause STPCLK# and this event is not recorded. On the other hand, the STPCLK deassertion logic is to decide which event cause the inactivates of STPCLK#. The deassertion of STPCLK# must qualify CPU's Stop Grand Bus Cycle in Intel CPUs.

2.8.9 DEVDET Logic

In 496/497 architecture, PMU is located at 497 while so many system events information can only be observed from 496. A DEVDET protocol is implemented to overcome this problem.

On both 496 and 497, two identical state machines are implemented. In 496, the block diagram is shown below:

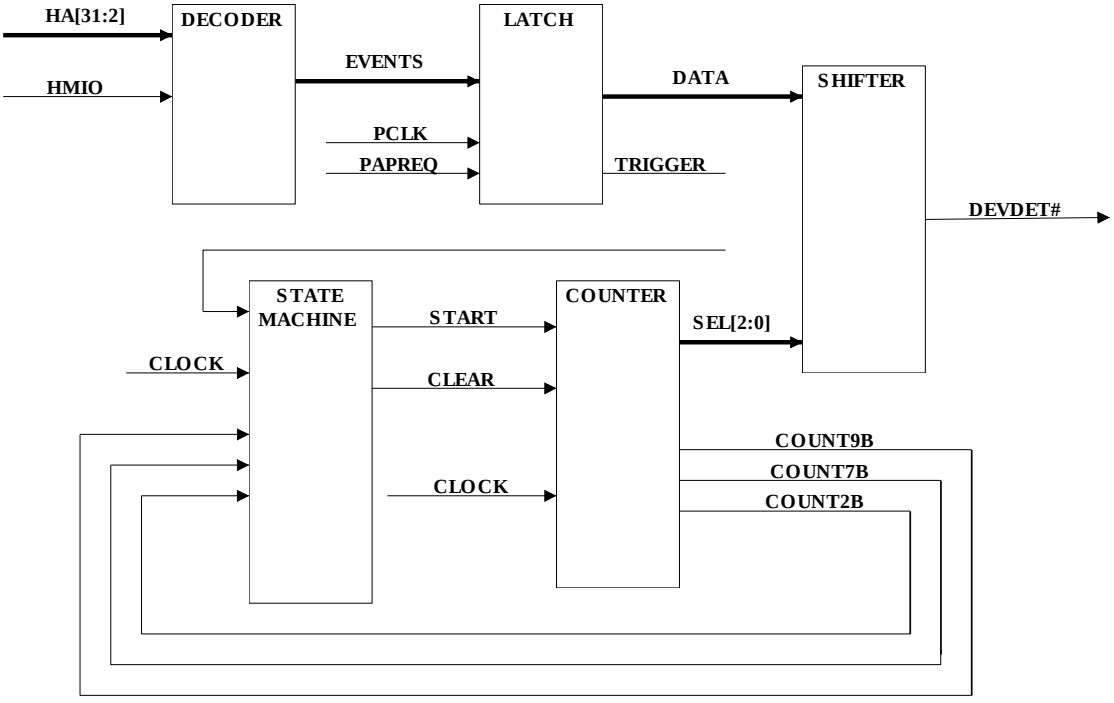


Figure 2.35 496 DEVDET Block Diagram

The decoder is used to decode the cycles issued by the CPU. Whenever a event is decoded, the latch will automatically latch it and trigger the state machine to send the information. When the state machine enters the STARTING state, the counter begins to count from 1 to 19. The state transition in the state machine is due to the count of the counter. Whenever the state

machine is active, no interruption is allowed. This means that whenever an information is detected by the decoder, it must send it to 497 without any exceptions. When the counting sequence reach 19, the state machine will enter the IDLE state. The detail state transitions are describe at the wave form below. The function of the shifter is to translate the parallel data to serial data. There are two level latches in the latch block. The first level is to latch the system events. When the state machine enters data phase, the system events will be treated as data and move to second level latch. After that, the first level latch is cleared and open to receive system events. The data in second level latch is transmitted to 497. At the turn around phase, the state machine will clear the second level latch and check the first level latch to see if any system event has come. If any, the state machine will again enter the starting phase and transmit. If none, the state machine will enter idle state and wait for trigger.

The block diagram of DEVDET located at 497 is as below:

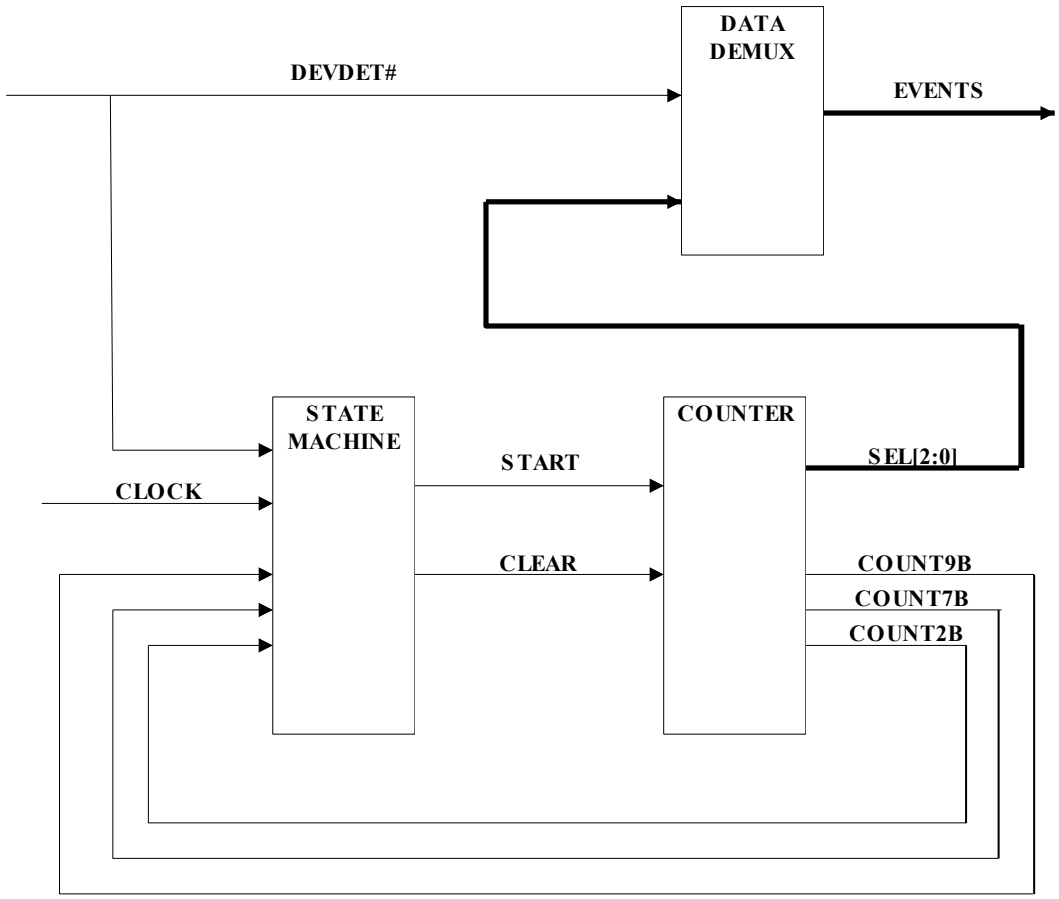


Figure 2.36 497 DEVDET Block Diagram

The state machine located at 497 uses DEVDET# to trigger itself. When the state machine has been triggered, the counter begins to count from 1 to 19, the same in 496. The state transitions in 497 is the same as those in 496 except that the state machine in 497 delays just a clock(PCLK/2). When the state machine enters data state, the DEMUX will decode the DEVDET# from serial to parallel. No latch is used because we just require that these system events active just a clock.

The DEVDET protocol is illustrated below:

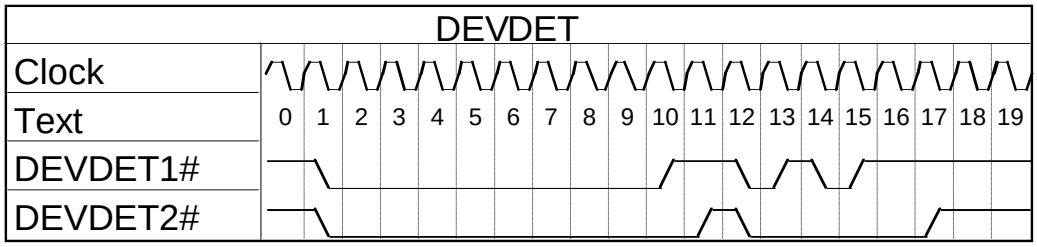


Figure 2.37 DEVDET Timing Waveform

NOTE: DEVDET1# denotes that this is the first example of DEVDET# and so is DEVDET2#.

The clock in the diagram is derived from PCI clock by dividing it with 2.

At the beginning of each transfer, DEVDET# should be low for just 9 clock. In the diagram, from 1 to 9, is the starting bits. The starting bits must have 9 bits otherwise the state machine will take it as a failure transfer. In this case, the state machine will automatically reset itself and can receive information immediately. If the nine starting bits are transmitted smoothly, the next 8 bits are data bits. In data bits 1 means a specified system event occurs, otherwise 0 means nothing happens. In this protocol, eight system events are transmitted. They are:

STPGND	(Stop Grand Bus Cycle)
PARAPORT	(Access to the Parallel Port)
SERPORT	(Access to the Serial Port)
HDSKREQ	(Access to Hard Disk)
PCIMREQ	(PCI Master Request)
IOTRAP1B	(I/O Trap 1)
IOTRAP2B	(I/O Trap 2)
VRAMREQ	(Access to the Video BIOS and VGA Port)

At the DEVDET1# waveform, there are six events transferred. They are STPGND, PARAPORT, HDAKREQ, IOTRAP1B IOTRAP2B and VRAMREQ. At the lower wave form, just one system events is transmitted, it is PARAPORT.

The data bits always has 8 bits. If the state machine enters this states, noises will probably be considered as data. In this point, the designer makes a little assumption: when the state machine enters the data phase, it means that no noise is in this transmission.

The turn around bits are used to clear the state machine and to mark the boundary between two transitions. After the turn around state, the transition can be issued immediately or the state machines enter idle state.

2.8.10 PMU Configuration Registers

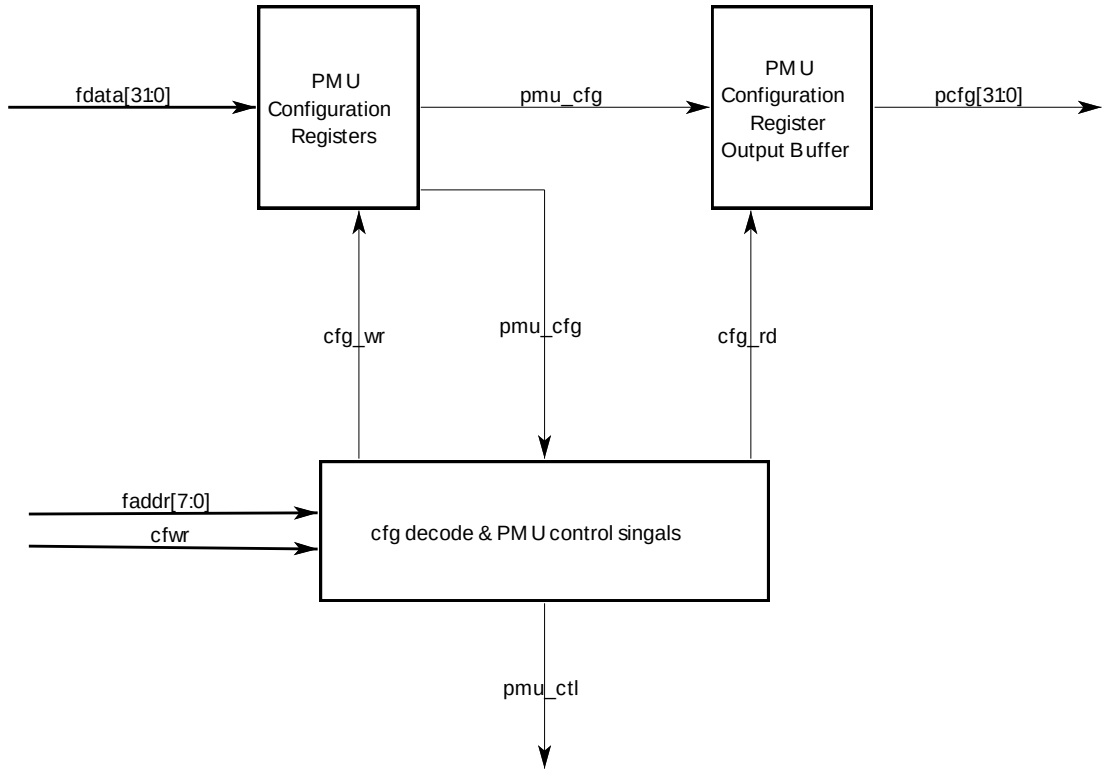


Figure 2.38 PMU Configuration Registers

The PMU configuration registers are reside in PMU block and are accessed by F-S Link via fdata[31:0] (read data). The register access control signals are generated from faddr[31:0] and cfwe decode. The output of registers (pmu_cfg) are used to control the function of PMU. The register output buffer is constructed by the tri_state buffer. In this method, a lot of multiplexers never appear; the gate count is at its minimum.

2.9 Reset and Clock

The external clocks of 497 are PCI clock and 14MHz. The system clock of ISA is generated by 497 and its speed is programmable to 1/3, 1/4 or 14MHz/2 and ranges from 6.67 to 8.33 MHz.

Both 496 and 497 receive power good reset input. 497 uses fast pwrgd (3us) to generate DRVRST, PCIRST# and internal reset and 496 uses slow pwrgd (5us) to generate INIT. 497 generates CPURST based on the states of INIT and cpurst_en bit. The detailed sequences are explained in the following section.

1) PWRGD active:

When 497 detects pwrgd is asserted (change from low to high), it does the following things:
a. triggers pd3us circuit

2) pd3us (power on reset delay 3us) active:

It takes about 3us for pd3us becomes active after pwrgd is asserted. When pd3us is active, 497 does the following things:

a. negates DRVRST
b. negates PCIRST#
c. all the internal 497 configuration registers and state machines are initialized.

3) INIT active:

Whenever the INIT signal is detected active, 497 does the following things:

a. If cpurst_en is enable(this is default state after power on in bit3 of configuration register C6h), 497 generates CPURST whenever INIT is active (combinational logic)
b. If cpurst_en is disable (this bit is disabled by BIOS during POST in the normal operation unless the CPU type is 486, or M7 and the WBAK bit in CCR2 is disabled, or if the init function is disable), the INIT is blocked and CPURST will not generated

4) PCIRST# generation:

PCIRST# is generated under two conditions:

a. PCIRST# is always generated whenever power on reset is occurred (PCIRST# has the same timing as pd3us).
b. PCIRST# can be optionally generated whenever INIT is asserted and the blkinit_en bit is disabled. (the blkinit_en bit is enabled as the default state after power on in bit2 of configuration register C6h).

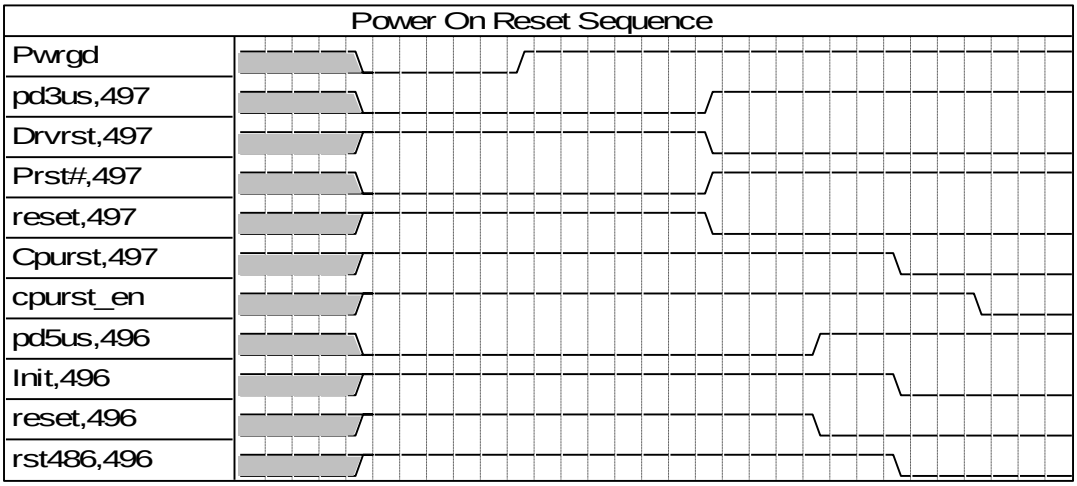


Figure 2.39 Power On Reset Sequence

3. Pin Assignment and Signal Description

This section contains a detailed description of each signal. The signals are arranged in functional groups according to the interface.

Note that the '#' symbol at the end of a signal name indicates that the active, or asserted state occurs when the signal is at a low voltage. When '#' is not present after the signal name, the signal is asserted when at the high voltage level.

The terms assertion and negation are used extensively. This is done to avoid confusion when working with a mixture of 'active-low' and 'active-high' signals. The term assert, or assertion indicates that a signal is active, independent of whether that level is represented by a high or low voltage. The term negate, or negation indicates that a signal is inactive.

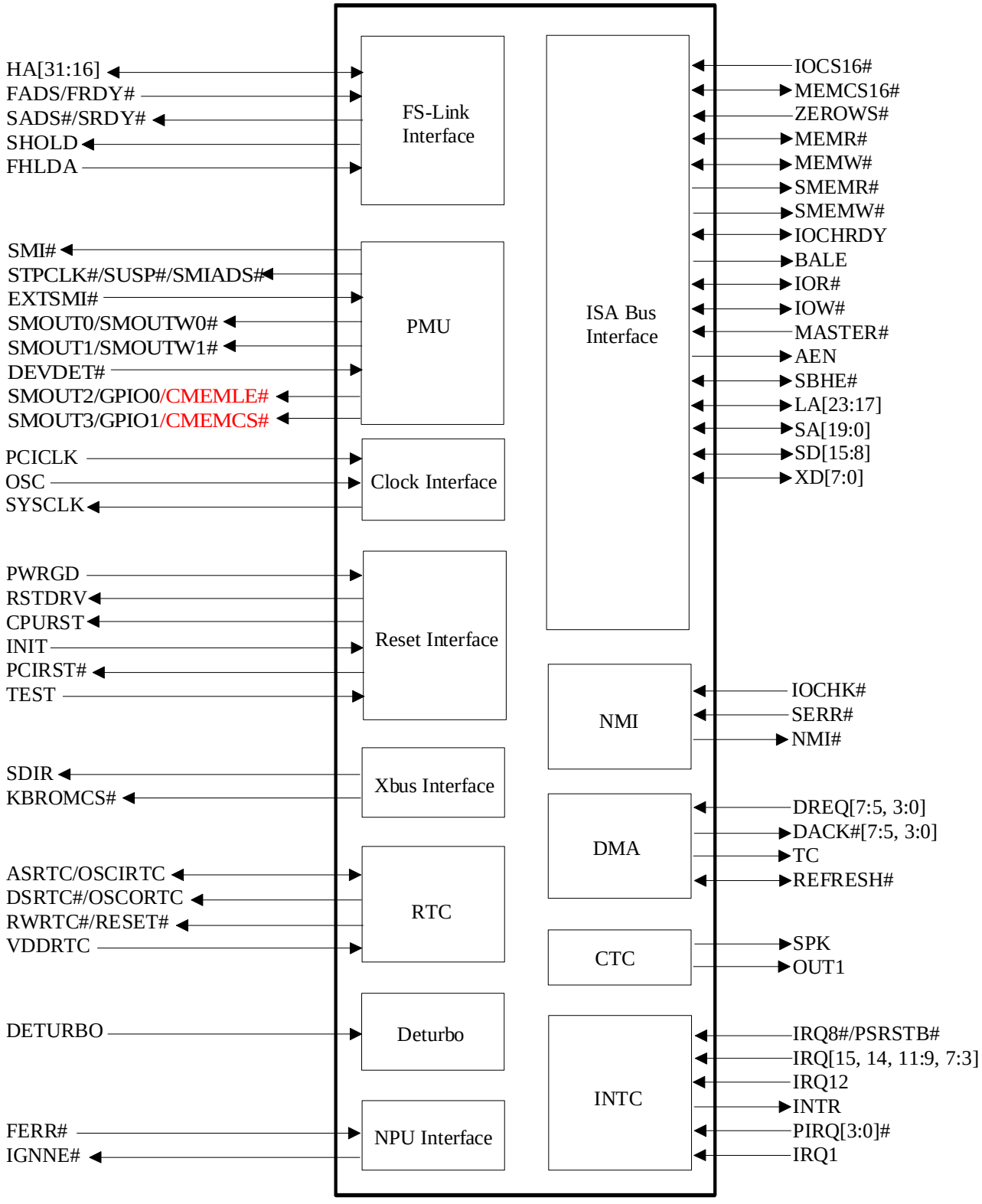


Figure 3.1 SiS85C497 Component Functional Block Diagram

3.1 Pin Assignment

85C497 Package Pinout

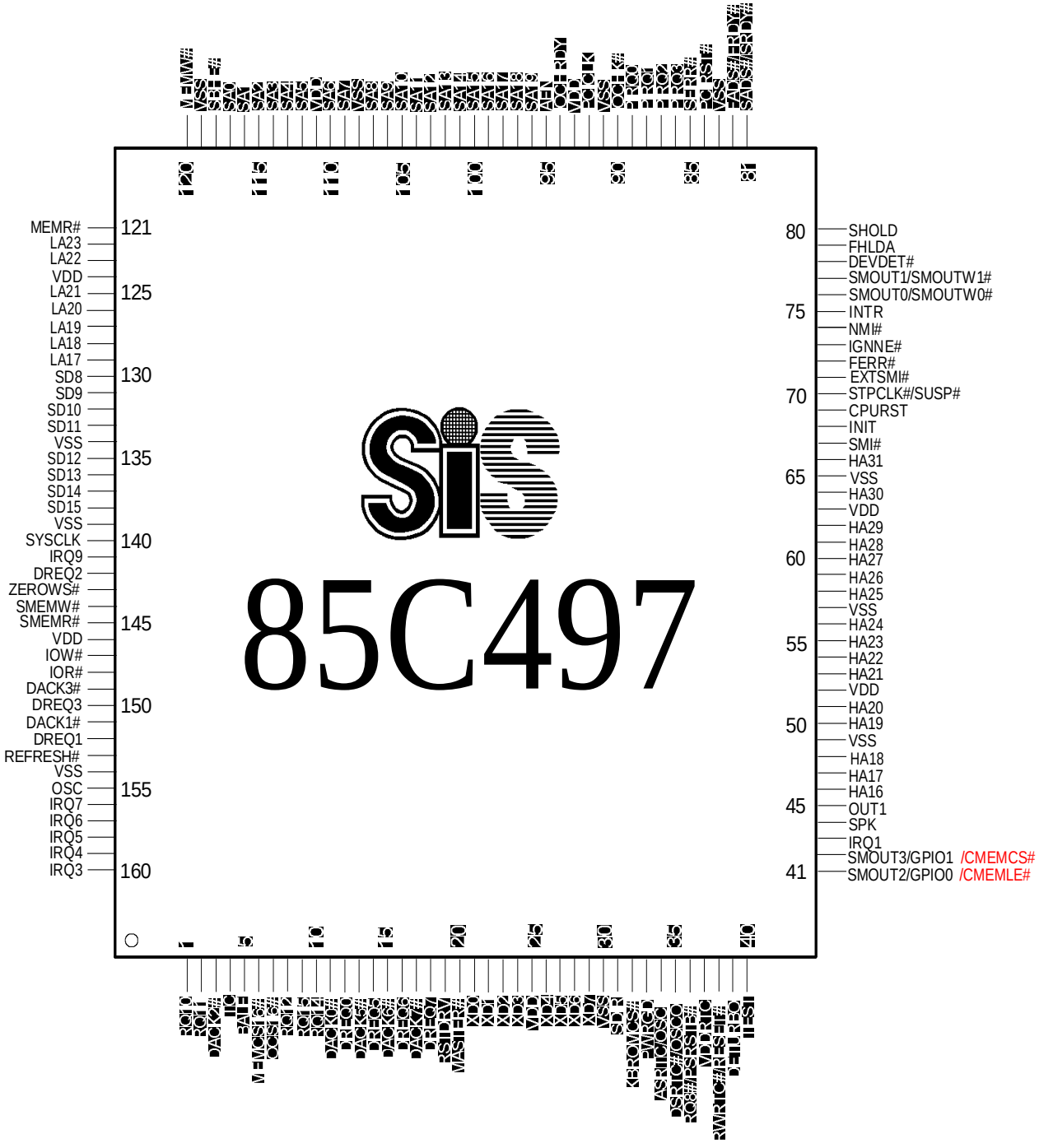


Figure 3.2 SiS85C497 Pin Assignment

85C497 Pin listing(# means active low)

1 = IRQ10	41 = SMOUT2/GPIO0 /CMEMLE#	81 = SADS#/SRDY#	121 = MEMR#
2 = IRQ11	42 = SMOUT3/GPIO1 /CMEMCS#	82 = FADS#/FRDY#	122 = LA23
3 = DACK2#	43 = IRQ1	83 = VSS	123 = LA22
4 = TC	44 = SPK	84 = PCIRST#	124 = VDD
5 = BALE	45 = OUT1	85 = SERR#	125 = LA21
6 = MEMCS16#	46 = HA16	86 = PIRQ3	126 = LA20
7 = IOCS16#	47 = HA17	87 = PIRQ2	127 = LA19
8 = IRQ12	48 = HA18	88 = PIRQ1	128 = LA18
9 = IRQ15	49 = VSS	89 = PIRQ0	129 = LA17
10 = IRQ14	50 = HA19	90 = IOCHK#	130 = SD8
11 = DACK0#	51 = HA20	91 = VSS	131 = SD9
12 = DREQ0	52 = VDD	92 = PCICLK	132 = SD10
13 = DACK5#	53 = HA21	93 = VDD	133 = SD11
14 = DREQ5	54 = HA22	94 = IOCHRDY	134 = VSS
15 = DACK6#	55 = HA23	95 = AEN	135 = SD12
16 = DREQ6	56 = HA24	96 = SA19	136 = SD13
17 = DACK7#	57 = VSS	97 = SA18	137 = SD14
18 = DREQ7	58 = HA25	98 = SA17	138 = SD15
19 = RSTDRV	59 = HA26	99 = SA16	139 = VSS
20 = MASTER#	60 = HA27	100 = SA15	140 = SYSCLK
21 = XD0	61 = HA28	101 = SA14	141 = IRQ9
22 = XD1	62 = HA29	102 = SA13	142 = DREQ2
23 = XD2	63 = VDD	103 = SA12	143 = ZEROWS#
24 = XD3	64 = HA30	104 = SA11	144 = SMEMW#
25 = VDD	65 = VSS	105 = SA10	145 = SMEMR#
26 = XD4	66 = HA31	106 = SA9	146 = VDD
27 = XD5	67 = SMI#	107 = SA8	147 = IOW#
28 = XD6	68 = INIT	108 = VSS	148 = IOR#
29 = XD7	69 = CPURST	109 = SA7	149 = DACK3#
30 = VSS	70 = STPCLK#/SUSP#	110 = SA6	150 = DREQ3
31 = SDIR	71 = EXTSMI#	111 = VDD	151 = DACK1#
32 = KBROMCS#	72 = FERR#	112 = SA5	152 = DREQ1
33 = PWRGD	73 = IGNNE#	113 = SA4	153 = REFRESH#
34 = ASRTC/OSCI	74 = NMI	114 = SA3	154 = VSS
35 = DSRTC#/OSCO	75 = INTR	115 = SA2	155 = OSC
36 = IRQ8#/PSRSTB#	76 = SMOUT0/SMOUTW0#	116 = SA1	156 = IRQ7
37 = VDDRTC	77 = SMOUT1/SMOUTW1#	117 = SA0	157 = IRQ6
38 = RWRTC#/RESET#	78 = DEVDET#	118 = SBHE#	158 = IRQ5
39 = DETURBO	79 = FHLDA	119 = VSS	159 = IRQ4
40 = TEST	80 = SHOLD	120 = MEMW#	160 = IRQ3

3.2 496 Interface (FS Link, Refresh, Deturbo, PMU)

Signal Name	Type	Pin	Description
HA[31:16]	I/O	66,64,62~58,56~53,51, 50,48~46	**FS Link Bus.** The HA[31:16] is used as the FS Link Bus to transfer information such as address, data, and command during FS link cycles.
DEVDET#	I	78	**Device Detect.** This signal is generated from 496 indicates the device detection information such of PCI master bus request and I/O port trap.
FADS#/FRDY#	I	82	**Fast Address Status/Fast Ready.** This signal services for two purposes. For Host-to-ISA or PCI-to-ISA cycles, FADS# is generated to the 497 to request the ISA Bus. When DMA or ISA master want to accesses on board or PCI memories, this pin works as ready pin indicates that the data is valid (for read) or it is ready to accept data (for write).
FHLDA	I	79	**Fast Hold Acknowledge.** This signal is generated from 496 to 497 indicates that the Host Bus is available to the ISA master or DMA.
OUT1	O	45	**Output 1.** This is an output signal from 8254 Counter/Timer and used by 496 DRAM Refresh Controller.
SADS#/SRDY#	O	81	**Slow Address Status/Slow Ready.** This signal is generated from 497 to 496 when ISA master wants to access on board or PCI memories. When Host CPU or PCI master want to access ISA slave on the other hand, this signal indicates that the data is valid (for read) or it is ready to accept the data (for write).
SHOLD	O	80	**Slow Hold Request.** This signal is asserted when an ISA master requests Host Bus. The 497 needs to own the Host Bus before it releases ISA Bus to the ISA master.

3.3 AT BUS Interface

Signal Name	Type	Pin	Description
AEN	O	95	**Address Enable.** This is driven high on ISA bus to indicate the address lines are valid in DMA or ISA master cycles. It is low otherwise.
BALE	O	5	**Bus Address Latch Enable.** This is the buffered address latch enable signal. A high pulse with half SYSCLK will be generated at the beginning of CPU/PCI ISA cycle for the external devices to latch the system address. During the CPU/PCI bus releasing (HLDA active) cycle and refresh cycle, this signal will be kept to high level status.
IOCS16#	I	7	**External 16-Bit I/O Chip Select.** This is the 16-bit I/O devices select indicating signal.
IOCHRDY	OD	94	**I/O Channel Ready.** This is the I/O channel ready signal. This input is used to extend the ISA command width for the CPU, PCI, and DMA cycle. In CPU/PCI as master cycle, the siganl will be output active low until the default wait state has been counted. In DMA/ISA master as master cycle, if it accesses upper memory, the signal will be output active low untill FADS#/FRDY# asserted.
IOR#	I/O	148	**I/O Read.** This is the I/O read signal. This signal will be an input in ISA master cycle but be an output in other cycles. In CPU/PCI ISA cycles, this signal is driven by chip internal control circuits. In DMA cycles, this signal is driven by the internal DMA controllers. In master cycles, this signal is driven by the external master devices.
IOW#	I/O	147	**I/O Write.** This is the I/O write signal. This signal will be an input in ISA master cycle but be an output in other cycles. In CPU/PCI ISA cycles, this signal is driven by chip internal control circuits. In DMA cycles, this signal is driven by the internal DMA controllers. In master cycles, this signal is driven by the external master devices.
MASTER#	I	20	**Master.** This is an active low signal from AT bus. When active, it indicates that the ISA bus master has the control of the system. The address and control signals are all driven by the ISA bus master.

Signal Name	Type	Pin	Description
LA[[23:17]	I/O	122, 123, 125, 126, 127, 128, 129	**Unlatched Address.** The LA[23:17] address lines are bi-directional. These address lines allow accesses to physical memory on the ISA Bus up to 16 MBytes. LA[23:17] are outputs when the 497 owns the ISA Bus.
MEMCS16#	OD	6	**External 16-Bit Memory Chip Select.** 16-bit memory chip select indicates a 16-bit memory transfer when assert or an 8-bit memory transfer when it is negated.
MEMR#	I/O	121	**Memory Read.** AT bus memory read command signal is an output pin during AT/DMA/refresh cycles and is an input pin in ISA master cycle.
MEMW#	I/O	120	**Memory Write.** AT bus memory write command signal is an output pin during AT/DMA/refresh cycles and is an input pin in ISA master cycle.
SA[19:0]	I/O	96-107, 110-117	**System Address Bus.** They are input pins when an external bus is in control and are output pins at all other time.
SBHE#	I/O	118	**System Byte High Enable.** Byte high enable signal indicates that the high byte has valid data on the ISA 16-bit data bus. This signal is an output except during ISA master.
SD[15:8]	I/O	138-135, 133-130	**System Data.** SD[15:8] provides the upper 8-bit data path for devices residing on the ISA Bus.
SMEMR#	O	145	**System Memory Read.** It instructs the memory device to drive data onto the data bus. It is active only when the memory being accessed is within the lowest 1 MB.
SMEMW#	O	144	**System Memory Write.** It instructs the memory devices to store the data presented on the data bus. It is active only when the memory being accessed is within the lowest 1 MB.
XD[7:0]	I/O	29-26, 24-21	**Utility Data Bus.** Peripheral Data Bus lines.
ZEROWS#	I	143	**Zero Wait States.** The system ignores the IORDY signal and terminates the AT bus cycle without additional wait state when it is asserted.

3.4 PMU Interface

Signal Name	Type	Pin	Description
EXTSMI#	I	71	**External SMI#.** The 497 generates SMI# interrupt to the CPU when the input is active.
SMI#	O	67	**System Management Interrupt.** The 497 generates SMI# interrupt to the CPU based on the status of software SMI# request, external SMI# request, system events, and I/O Trap.
SMOUT0/SMOUTW0#	O	76	**System Management Output Port 0 / System Management Write 0 Enable.** The output port is used to control peripheral for the power saving purpose. SMOUT output port can be expanded with an external latch. The SMOUT states are latched from XD[7:0] to the external latch when the SMOUTW0# is asserted.
SMOUT1/SMOUTW1#	O	77	**System Management Output Port 1 / System Management Write 1 Enable.** The output port is used to control peripheral for the power saving purpose. SMOUT output port can be expanded with an external latch. The SMOUT states are latched from XD[15:8] to the external latch when the SMOUTW1# is asserted.
STPCLK#/SUSP#	I/O	70	**Stop Clock Request / Suspend Request.** The 497 requests CPU and stops CPU's internal clock. For Intel and Cyrix CPUs, this pin is used as stopclk and suspend request respectively.
SMOUT2/GPIO0#/CMEMLE#	I/O	41	**System Management Output Port 2 / General Purpose I/O Port 0 / Configuration Memory Latch Enable.** The output port is used to control peripheral for the power saving purpose.
SMOUT3/GPIO1#/CMEMCS#	I/O	42	**System Management Output Port 3 / General Purpose I/O Port 1 / Configuration Memory Chip Select.**The output port is used to control peripheral for the power saving purpose.

3.5 Clock

Signal Name	Type	Pin	Description
OSC	I	155	**Oscillator.** OSC is the 14.318 Mhz ISA clock signal. It is used by the internal 8254 Timer, Counter 0, 1, and 2, and PMU.
PCICLK	I	92	**PCI Clock.** PCI clock is used for the 497 Fast-Slow Link State Machines. Frequencies supported by the external clock generator include 20, 25, and 33 Mhz.
SYSCLK	O	140	**System Clock.** SYSCLK is an output of the 497 component. The frequencies supported are 6.25 to 8.33 Mhz.

3.6 Reset

Signal Name	Type	Pin	Description
CPURST	O	69	**CPU Reset.** CPURST is an active high output to reset the CPU.
INIT	I	68	**Initialize.** The 497 combines this signal to the CPURST when the CPU type is other than Overdrive.
PWRGD	I	33	**Power Good.** When asserted, the power good input forces all the 497's internal registers and state machines to their default state. This input is asynchronous, but must meet setup and hold specifications for recognition in any specific clock. On the rising edge of PWRGD the 497 initiates a system "Hard Reset" by driving CPURST an PCIRST#.
PCIRST#	O	84	**PCI Reset.** PCI Bus Reset forces the PCI interfaces of each device to a known state. PCIRST# is driven during power up and when a hard reset sequence is initiated through Port 92h or fast KB reset.
RSTDRV	O	19	**Reset Drive**. 497 asserts RSTDRV when PCIRST# is asserted. RSTDRV is used to reset all the devices on ISA bus.
TEST	I	40	**Test**. The TEST signal is used for the 497 internal function test only. During normal operation, this input should be tied to ground.

3.7 XD Bus Interface

Signal Name	Type	Pin	Description
ASRTC/OSCI	O/I	34	**RTC Address Strobe / 32.768 Khz Crystal Input.** This signal is used to latch the address from XD Bus when the CPU accesses the external RTC. This pin can also be used as 32.768 Khz input.
DSRTC#/OSCO	O	35	**RTC Data Strobe / 32.768 Khz Crystal Output.** This signal is used to drive data onto XD Bus when the CPU accesses the external RTC. This pin can also be used as 32.768 Khz output.
RWRTC#/RESET#	O	38	**RTC Read Write Data / Bus Reset.** When the external RTC is used, this pin can be used as the RTC Read/Write Data.This signal is used to store the data presented on the XD bus when the CPU accesses the RTC. When the built-in RTC is used, this pin can be used as the Bus Reset. The Bus Reset is the reversed version of RSTDRV and is used to reset devices reside on VL-Bus and XD Bus.
KBROMCS#	O	32	**System ROM Chip Select.** This signal is active (low) when CPU accesses System ROM.
SDIR	O	31	**X-Data Direction.** This is the XD[7:0] external data buffer direction control signal.
VDDRTC	I	37	**Battery Power Input.** This is used for the RTC power.

3.8 Deturbo

Signal Name	Type	Pin	Description
DETURBO	I	39	**Deturbo Switch.** This is the deturbo switch.

3.9 NPU Interface

Signal Name	Type	Pin	Description
FERR#	I	72	**Numeric Coprocessor Error**. Floating point error from the CPU. It is driven active when a floating point error occurs.
IGNNE#	O	73	**Ignore Error**. It is normally in high impedance state and is asserted to inform CPU to ignore a numeric error.

3.10 NMI Interface

Signal Name	Type	Pin	Description
IOCHK#	I	90	**I/O Channel Check.** This is an active low input signal which indicates that an error has happened on the I/O bus.
NMI	O	74	**Non-Maskable Interrupt.** This interrupt is edge trigger signal to the CPU and is generated to invoke a non-maskable interrupt.
SERR#	I	85	**System Error**. SERR# can be forced active by any PCI device that detects a system error condition. Upon sampling SERR# active, 497 generates a non-maskable interrupt (NMI) to CPU.

3.11 DMA Interface

Signal Name	Type	Pin	Description
DACK[7:5, 3:0]#	O	17, 15, 13, 149, 3, 151, 11	**DMA Acknowledge.** These signals are used to notify the individual peripherals when one has been granted a DMA cycle. The active polarity of these lines is programmable. DACK1# should be pulled down with a 2.2K resistor.
DREQ[7:5, 3:0]	I	18, 16, 14, 150, 142, 152, 12	**DMA Request.** These signals are individual asynchronous channel request inputs and are used by peripheral circuits to get DMA service.
REFRESH#	I/O	153	**REFRESH**. Refresh signal is used to initiate a refresh cycle. This signal is an input in ISA bus master cycle and is an output in other cycles.
TC	O	4	**Terminal Count**. This is the DMA channel terminal count indicating signal.

3.12 CTC Interface

Signal Name	Type	Pin	Description
SPK	O	44	**Speaker Drive.** This signal is the output of CTC counter 2 and is used to drive the speaker.

3.13 INTC Interface

Signal Name	Type	Pin	Description
INTR	O	75	**CPU Interrupt Request.** INTR is high whenever a valid interrupt request is asserted.
IRQ1	I	43	**Interrupt Request of Keyboard Controller**
IRQ8#/PSRSTB#	I	36	**Interrupt Request of RTC/Power Status.**
IRQ12	I	8	**Interrupt Request.**
IRQ[15, 14, 11:9, 7:3]	I	9, 10, 2, 1, 141, 156-160	**Interrupt Request.**
PIRQ[3:0]#	I	86-89	**PCI INTA#, INTB#, INTC#, and INTD#.**

3.14 Power Pins

Signal Name	Type	Pin	Description
VSS	I	30, 49, 57, 65, 83, 91, 108, 119, 134, 139, 154	**Ground**.
VDD	I	25, 52, 63, 93, 111, 124, 146	**Power**. +5V Volt supply.

4. Electrical Characteristics

4.1 Maximum Ratings

Case Temperature under bias ....0°C to 70°C
Storage Temperature..... .....-40°C to 125°C
Voltage on any pin with respect to ground..... -0.3 Volts to Vcc+0.3 Volts
Supply voltage with respect to Vss..... .....-0.3 Volts to 7.0 Volts

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 reliability.

4.2 SiS85C497 D.C. Characteristics

Table 4.1 D.C. Specifications (V_{DD} = 5V ± 5%, T_{amb} = 0 to 70 °C)

Symbol	Parameter	Min.	Max.	Unit	Condition
V_{IL}	Input Low Voltage	-0.5	+0.8	V	
V_{IH}	Input High Voltage	+2.0	Vcc+0.5	V	
V_{T+}	TTL Schmitt Trigger, rising threshold	+1.5	+2.0	V	
V_{T-}	TTL Schmitt Trigger, falling threshold	+0.8	+1.1	V	
DV_T	TTL Schmitt Trigger, hysteresis (V_{T+} - V_{T-})	+0.4		V	
I_{IL}	Input Low Current TTL Input TTL w/pull-up	-10 -200	+10 -10	µA µA	0 < V_{IN} < Vcc
I_{IH}	Input High Current	-10	+10	µA	V_{IN} = V_{DD}
V_{OL}	Output Low Voltage		+0.4	V	@ I_{OL} max.
V_{OH}	Output High Voltage	+2.4		V	@ I_{OH} min.
I_{OL}	Output Low Current			mA	@ V_{OL} max.
I_{OH}	Output High Current			mA	@ V_{OH} min.
C_{IN}	Input Capacitance		10	pF	
C_{OUT}	Output Capacitance		10	pF	
I_{RTC}	Operating current of RTC		1.3(1)	uA	VDDRTC=2.4V VDD=0V
			2.02(1)	uA	VDDRTC=3.0V VDD=0V
			2.5(1)	uA	VDDRTC=3.3V VDD=0V

(1) Test Condition: Fosc=32768Hz, PWRGD=0V, PSRSTB=VDDRTC, all other input pins=0V

4.3 A.C. Characteristics

The A.C. Specifications given in the following tables consists of output active delays (used for control signals, from clock rising edge to active delay), output inactive delays (used for control signals, from clock rising edge to inactive delay), output valid delays (used for bus, from bus float or invalid to valid delay), output invalid delays (used for bus, from valid to float or invalid delay), input setup and input hold requirements. These specifications are given for the functional operating range of the device. Timing specifications are given in nanoseconds (ns) unless otherwise specified. The test conditions are Vcc = 5V ± 5%, Tamb = 0°C to +70°C, CL = 35 pf unless otherwise specified.

Table 4.2 Clock and Reset Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units	Notes
Clock and Reset						
t16	PWRGD input pulse width	1			ms	
t17	RSTDRV output pulse width	1			ms	
t18	PCIRST# output pulse width	1			ms	
t19	INIT input setup time (warm reset)		25		CPUCLK	
t20	CPURST output pulse width (cold reset)	1			ms	

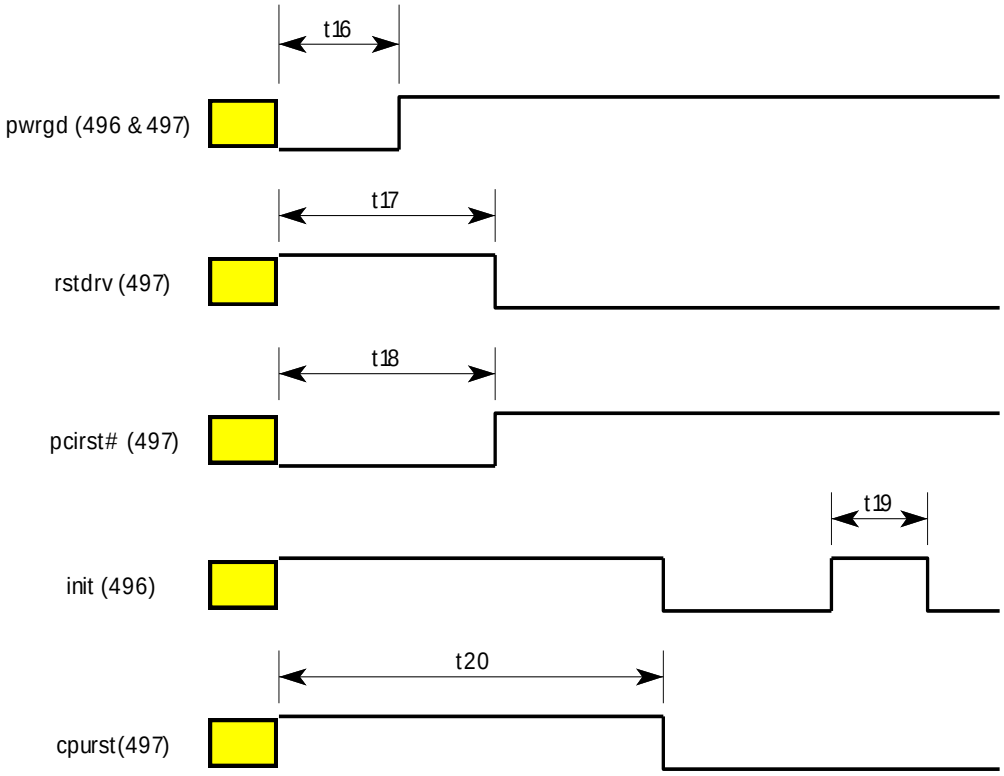


Figure 4.4 Reset Timing

Table 4.3 FS-Link Interface Timing Parameters (All the signals are based on PCICLK rising edge)

Symbol	Parameter	Min	Max	Units	Notes
t21	HA[31:16] input setup time	5		ns	
t22	HA[31:16] input hold time	3		ns	
t23	HA[31:16] output valid delay		16	ns	
t24	HA[31:16] output invalid delay		16	ns	
t25	FADS#/FRDY#, FHLDA input setup time	5		ns	
t26	FADS#/FRDY#, HHLDA input hold time	3		ns	
t27	SADS#/SRDY#, SHOLD output active delay		16	ns	
t28	SADS#/SRDY#, SHOLD output inactive delay		16	ns	

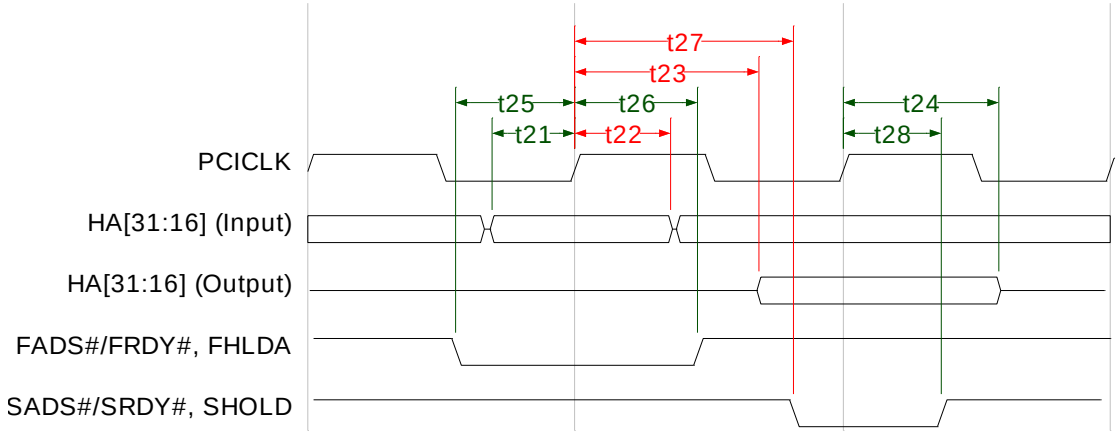


Figure 4.5 FS-Link Interface Timing

Table 4.4 PMU Interface Timing Parameters (All the signals are based on PCICLK rising edge unless otherwise specified)

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
t31	DEVDET# input setup time	7			ns	
t32	DEVDET# input hold time	5			ns	

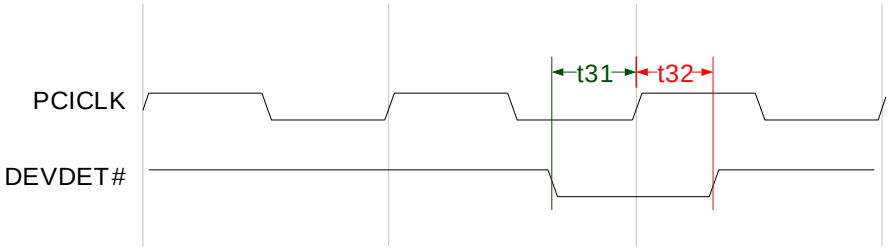


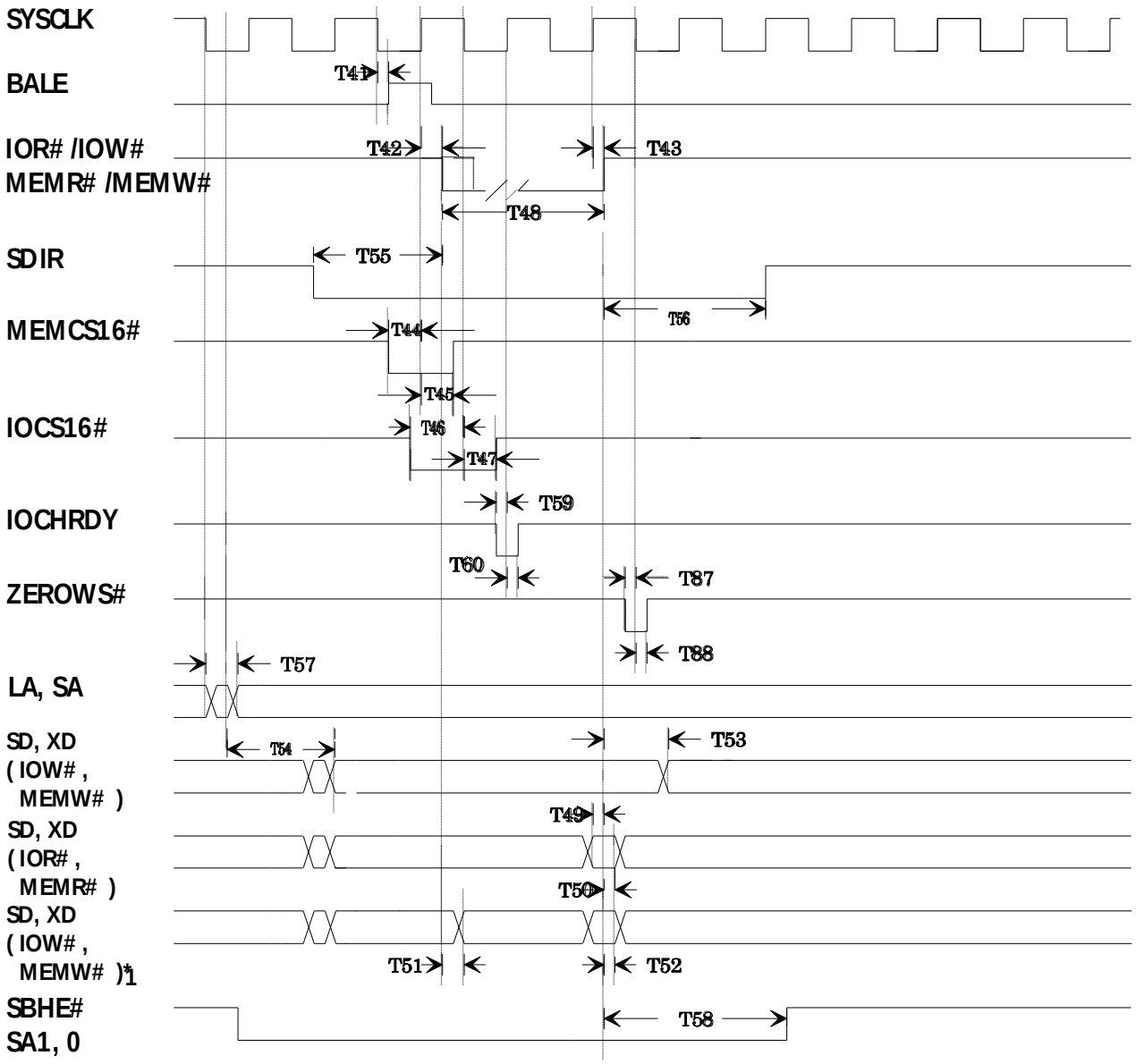
Figure 4.6 PMU Interface Timing

Table 4.6 497 As Master Timing Parameters

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
t41	BALE valid delay from SYSCLK falling			11	ns	
t42	IOR#, IOW#, MEMR#, MEMW# valid delay from SYSCLK			18	ns	
t43	IOR#, IOW#, MEMR#, MEMW# invalid delay from SYSCLK			18	ns	
t48	16 bit IOR#, IOW# pulse width		1.5		SYSCLK	
	8 bit IOR#, IOW# pulse width		4.5		SYSCLK	
	16 bit MEMR#, MEMW# *1		2		SYSCLK	
	8 bit MEMR#, MEMW#		4.5		SYSCLK	
	ROM MEMR#, MEMW# *1		2		SYSCLK	
t49	SD, XD data set up time to IOR#, MEMR# inactive	36			ns	
t50	SD, XD data hold time to IOR#, MEMR# inactive	3			ns	
t51	SD, XD valid data delay from IOW#,		22		ns	
t52	MEMW# active (for data swapping)		22			
t53	SD, XD data hold time from IOW#, MEMW# inactive in write disassembly cycle		22		ns	
t55	SDIR deassertion to IOR#, MEMR# active (16 bit)		1	2	SYSCLK	
	SDIR deassertion to IOR#, MEMR# active (8 bit)	1.5		2.5	SYSCLK	
t56	SDIR assertion delay from IOR#, MEMR# inactive		2		SYSCLK	
t58	SA0, SA1, SBHE# hold time from the negation of IOR#, IOW#, MEMW#, MEMR#	11			ns	
t87	ZEROWS# setup time to SYSCLK falling	5			ns	
t88	ZEROWS# hold time to SYSCLK falling	4			ns	
t63	AEN active to IOR# active		6		DMACLK	
t64	AEN active to IOW# active		7		DMACLK	
t65	AEN inactive from IOR# inactive		3		DMACLK	
t66	AEN inactive from IOW# inactive		4		DMACLK	
t67	BALE active to IOR# active		1.5		DMACLK	
t68	BALE active to IOW# active		2.5		DMACLK	
t69	BALE inactive from IOR# inactive		1		DMACLK	

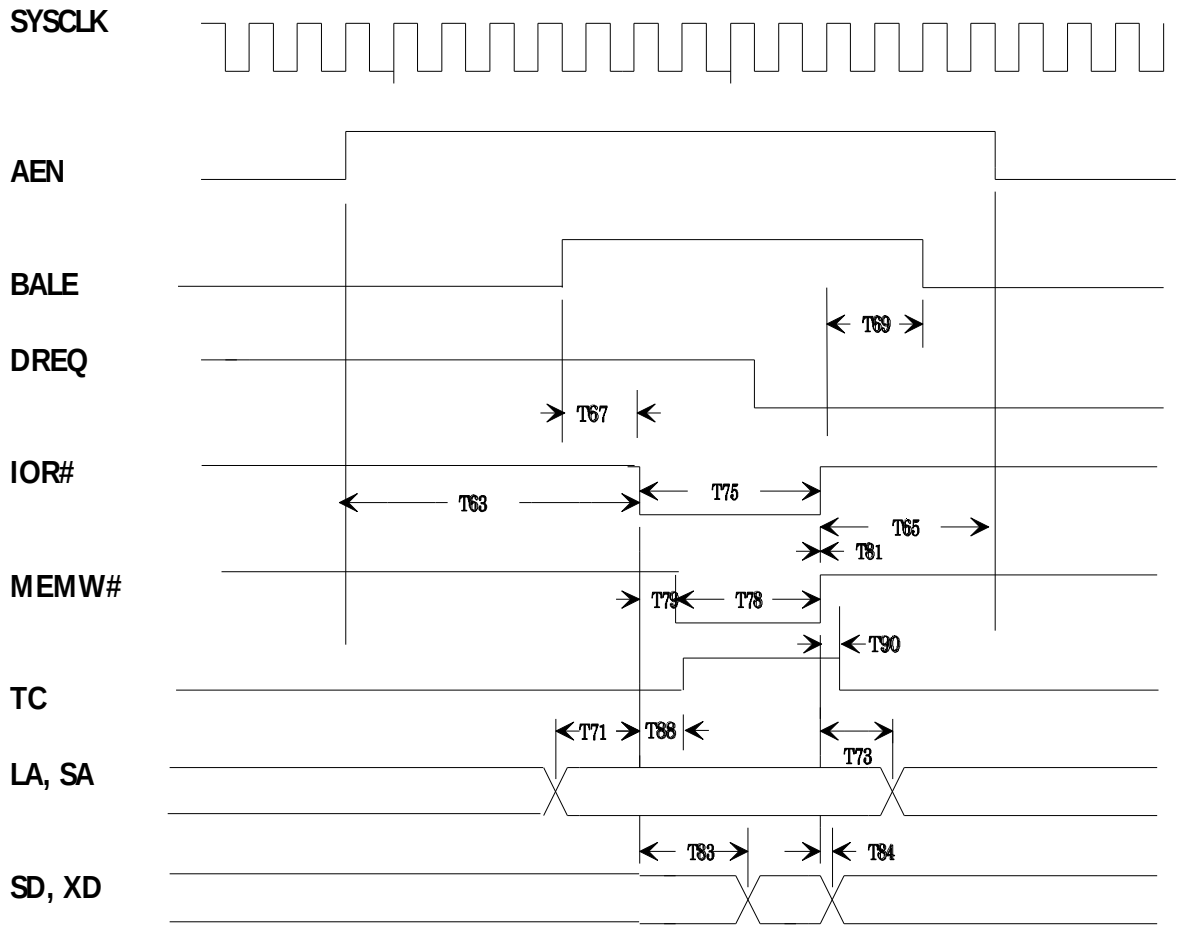
t70	BALE inactive from IOW# inactive		1		DMACLK	
t71	LA, SA, SBHE# valid set up time to IOR#		1		DMACLK	
t72	LA, SA, SBHE# valid set up time to IOW#		2		DMACLK	
t73	LA, SA, SBHE# valid hold from IOR# inactive		0.5		DMACLK	
t74	LA, SA, SBHE# valid hold from IOW# inactive		0.5		DMACLK	
t75	IOR# pusle width		4		DMACLK	
t76	IOW# pusle width		2		DMACLK	
t77	MEMR# pusle width		3		DMACLK	
t78	MEMW# pusle width		3		DMACLK	
t79	MEMW# active from IOR# active		1		DMACLK	
t80	IOW# active from MEMR# active		1		DMACLK	
t81	MEMW# inactive from IOR# inactive		1		ns	
t82	IOW# inactive from MEMR# inactive		-2.0		ns	
t83	Read data valid from IOR# active		267.5		ns	
t84	Read data valid hold from IOR# inactive		32.0		ns	
t85	Write data valid setup to IOW# inactive		162.5		ns	
t86	Write data valid hold from IOW# inactive		13.3		ns	
t87	TC active delay from IOW# active		-1.0			
t88	TC active delay from IOR# active		112.0			
t89	TC active delay from IOW# inactive		3.0			
t90	TC active delay from IOR# inactive		2.0			
t91	REFRESH# active setup to MEMR# active		2		SYSCLK	
t92	REFRESH# active hold from MEMR# inactive		0.5		SYSCLK	
t93	AEN active to REFRESH# active delay		2.0		ns	

NOTE*:* DMACLK = SYSCLK or SYSCLK/2 depends on bit 0 of ISA configuration register 01H.



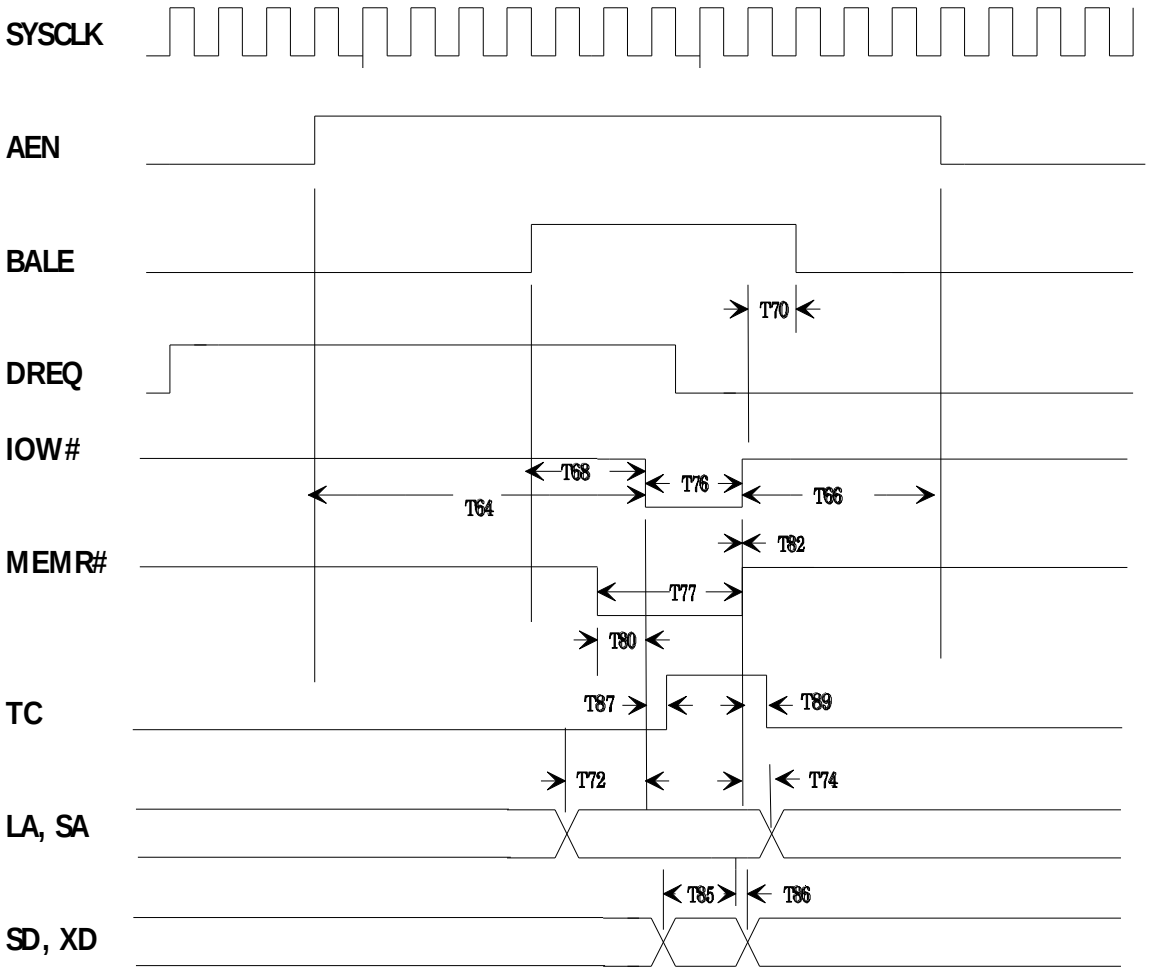
*1 IS FOR DATA SWAPPING

Figure 4.8 AT Bus Cycle



DMA cycle (IOW#, MEMR#), DMACLK = SYSCLK

Figure 4.9 DMA Cycle (IOW#, MEMR#)



DMA cycle (IOR#, MEMW#), DMACLK = SYSCLK

Figure 4.10 DMA Cycle (IOR#, MEMW#)

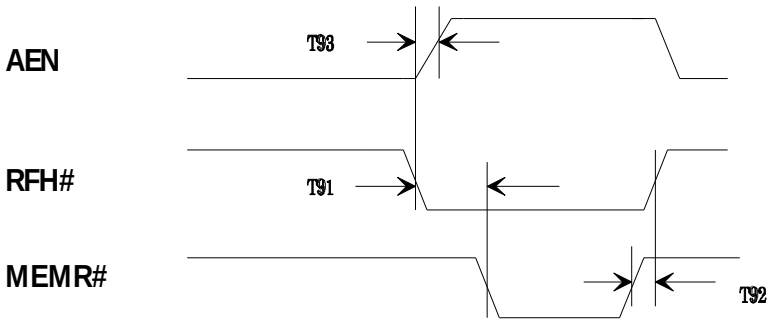


Figure 4.11 Refresh Timing

4.4 Measurement and Test Conditions

Figure 4.19 and 4.20 define the conditions under which timing measurements are made. The component test guarantees that all timings are met with minimum clock slew rate (slowest edge) and voltage swing. The design must guarantee that minimum timings are also met with maximum clock slew rate (fastest edge) and voltage swing. In addition, the design must guarantee proper input operation for input voltage swings and slew rates that exceed the specified test conditions.

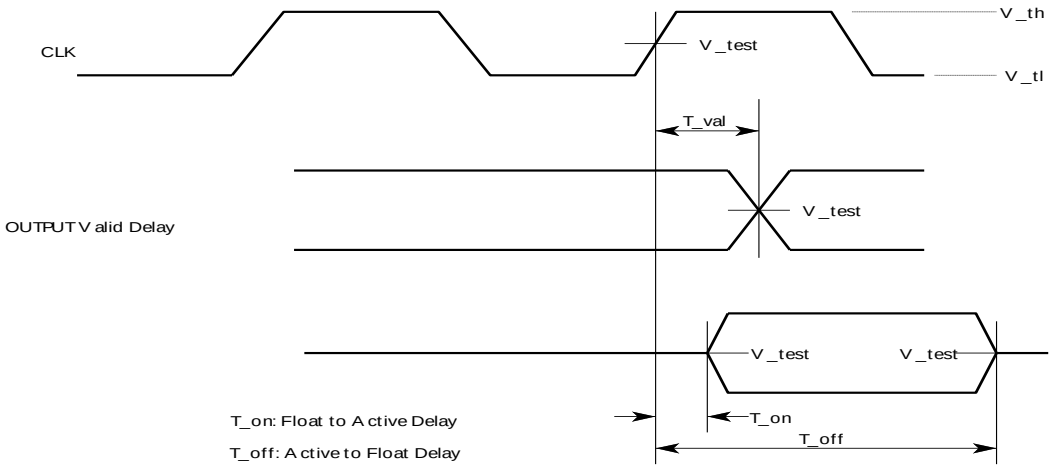


Figure 4.12 Output Timing Measurement Conditions

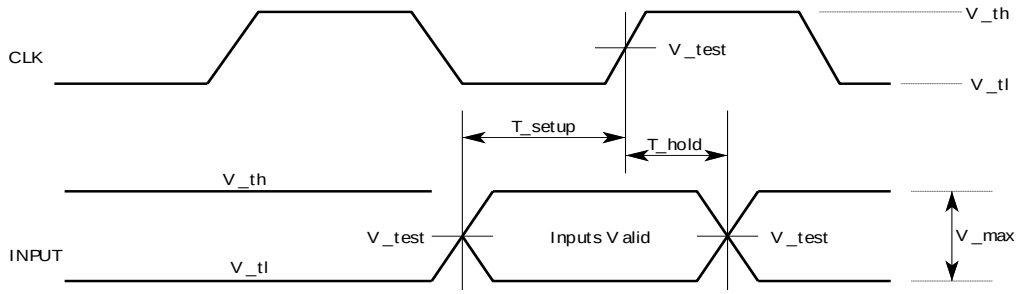


Figure 4.13 Input Timing Measurement Conditions

Table 4.10 Measure and Test Condition Parameters

Symbol	5V Signaling	Units
V_{th}	2.4	V (Note)
V_{tl}	0.4	V (Note)
V_{test}	1.5	V
V_{max}	2.0	V (Note)
Input Signal Edge Rate	1 V/ns	

NOTE:
The input test for the 5V environment is done with 400 mV of overdrive (over V_{ih} and V_{il}). Timing parameters must be met with no more overdrive than this. V_{max} specifies the maximum peak-to-peak waveform allowed for testing input timing.

Part IV

1. Register Description

The chipset is formed by two components: SiS85C496 and SiS85C497. SiS85C496, PCI, Memory, and Cache controller (85C496) equips with host interface, cache / DRAM controller, PCI bridge, IDE controller, and FS-link to ISA bus. SiS85C497 AT bus and Macrocell controller (ATM) provides ISA bridge (with FS-link), 206 macrocell and power management unit (PMU).

The SiS85C496 / 497 contains three sets of programmable registers. These registers are accessed via the CPU I/O address space and mapped to three different spaces, PCI configuration space, 85C497 I/O space (Port 22, 23), or CPU direct I/O space (206). The configuration registers in the PCI configuration and 85C497 I/O space are used for SiS85C496/497 internally to specify the configurations of major system components including CPU, PCI, VESA, ISA, cache, DRAM, IDE controller, and PMU as well as set up operating parameters. The ports in CPU direct I/O space are identical with AT compatible DMA, interrupt controllers, interval counter, and real time clock.

The SiS85C496 / 497, once receive a power-on reset, set their registers to default values. The default values provide the minimum feature set to start the system. Hence, the default setting is hardly the optimized combination. The best performance configuration is left to BIOS for complete system resource determination and properly programming.

System Hardware Configuration

Two pins in the SiS85C496/497 is used for trapping purpose to identify the hardware configurations at the power-up stage. The pin is defined to be 1 if a pull up resistor is used, and it is 0 if a pull down resistor is used. A pull up or pull down resistor of 2.2K is adequate for the trapping purpose. The definition is:

DACK0# (Pin 11, 85C497)

1: Internal RTC supported

0: External RTC supported

LBIDE# (Pin 152, 85C496)

1: Define Pin 158, 159 of 85C496 to RAS5# and RAS6#

0: Define Pin 158, 159 to HLOCK#/RC#, BOFF#

For more details, please refer to Register 67 bit 3 description.

2. PCI Configuration Control Ports

The PCI configuration control registers contains two double word registers in the CPU I/O space. They are defined to allow software to generate PCI configuration cycles using the PCI Configuration Mechanism #1 format. The general mechanism for accessing configuration space is to write a value into Configuration Address Register that specifies the PCI bus, device on the bus, and configuration register number in that device being accessed. A read or write to Configuration Data Register will then cause the bridge to translate that configuration Address Register value to the requested destination.

2.1 Configuration Address Port (0CF8h)

Register I/O Address:	0CF8h
Default Value:	00000000h
Attribute:	Read / Write

Configuration Address port is a 32-bit read / write I/O port that enables / disables configuration accesses, specifies the target PCI bus number, device number, function number and register number. A full double-word write to this port will cause host bridge (85C496) to latch the data into its CONFIG_ADDRESS register. A full double-word read to this port will cause bridge to return the current value in CONFIG_ADDRESS register.

Bit 31 **PCI Configuration Space Access Enable**
0 = Disable
1 = Enable

Bits 30:24 **Reserved**

Bits 23:16 **PCI Bus Number**

Bits 15:11 **Device Number**

Bits 10:8 **Function Number**

Bits 7:2 **Register Number**

Bits 1:0 **00b**

The Device Number of SiS85C496/497 is listed as the following table:

Device	Bus No.	Device No.	Function No.
85C496*	00h	00101b (AD 16)	000b

NOTE:
SiS85C497 is not a PCI device. It's configuration accesses are provided through SiS85C496.

2.2 Configuration Data Ports (0CFCh)

Register I/O Address:	0CFCh ~ 0CFFh
Default Value:	00000000h
Attribute:	Read / Write

This is a 32-bit read / write address range that includes four pseudo I/O ports, 0CFCh, 0CFDh, 0CFEh, and 0CFFh. A read or write within this range will cause host bridge to translate the CONFIG_DATA register value into a configuration cycle on the PCI bus. The register size in the I / O instructions, then, determine the byte-enable value on PCI bus (e.g. AL = 8 bits, AX = 16 bits, and EAX = 32 bits). For example:

```
OUT   0CF8h, 80002840h
in    AX, 0CFEh
```

will input the value of 42h and 43h from SiS85C496/497 PCI Configuration Space.

3. PCI Configuration Space Registers (00h ~ FFh)

The PCI Configuration Space Registers contains 256 registers which reside in the SiS85C496/497 configuration space. They are divided into three categories:

- **Header Registers**: a 64 bytes header region consists of fields that uniquely identify SiS85C496/497 and allow 85C496 / 85C497 to be generically controlled.
- **85C496 Specific Registers**: a 64 bytes region consists of host - to - PCI bridge / IDE controller configuration and control registers which physically reside in 85C496.
- **85C497 Specific Registers**: a 128 bytes region consists of Power Management Unit / ISA bus configuration and control registers which physically reside in 85C496.

.

3.1 PCI Configuration Header Registers (00h ~ 3Fh)

The complete definition of PCI pre-defined header registers is described in chapter 6 of *PCI Local Bus Specification, Production Version, Revision 2, (page 149).* The undefined registers should be treated as RESERVED.

Register 00h ~ 01h PCI Vendor Identification

Default Value:	1039h
Attribute:	Read Only

The register identifies the manufacture of the device. The PCI SIG allocated ID of Silicon Integrated System Corp. is 1039.

Bits 15:0 PCI Vender ID (1039 for SiS)

Register 02h ~ 03h Device Identification

Default Value:	0496h
Attribute:	Read Only

The read-only register is a product identifier allocated by SiS. The ID of 85C496/497 is 0496.

Bits 15:0 PCI Device ID (0496 for SiS85C496/497 Chipset)

Register 04h ~ 05h PCI Device Command

Default Value: 0007h
Attribute: Read/Write unless specified

The register provides coarse control over SiS85C496/497's ability to generate and respond to PCI cycles.

Bits 15:10 Reserved (Read Only)

Bit 9 Fast Back-to-Back Control
0 = Disable
1 = Enable

Bit 8 SERR# Control
0 = Disable
1 = Enable

Bit 7 Address / Data Stepping Control (Read Only)
0 = Always disabled, SiS85C496 do not been implemented

Bit 6 Parity Error Response Control
0 = Disable
1 = Enable

Bit 5 Reserved (Read Only)

Bit 4 Memory Write and Invalidate Command Control (Read Only)
0 = 85C496 does not generate Memory Write and Invalidate.

Bit 3 Special Cycle Operation Control (Read Only)
0 = 85C496 does not response to Special Cycles.

Bit 2 PCI Master Control (Read Only)
1 = 85C496 is always able to behave as a bus master.

Bit 1 Memory Space Accesses Control (Read Only)
1= 85C496 always responds to memory space accesses.

Bit 0 I/O Access Control (Read Only)
1 = 85C496 always responds to I/O space accesses.

Register 06h ~ 07h Device Status

Default Value: 0280h
Attribute: Read / Write Clear

The Status register is used to record status information for PCI bus related events. A read to this register returns the current register value and a write with value "1" clear the corresponding status bit.

Bit 15 Detected Parity Error Status
Read: 0 = No parity error is detected since this bit was cleared.
1 = Parity error is detected.
Write: 0 = No effect
1 = Clears this bit to 0

Bit 14 **Signaled System Error (SERR#) Status**

Read: 0 = No system error is signaled since this bit was cleared.
1 = System error is signaled.
Write: 0 = No effect
1 = Clears this bit to 0

Bit 13 **Received Master Abort Status**

Read: 0 = No master abort is received since this bit was cleared.
1 = Master abort is received.
Write: 0 = No effect
1 = Clears this bit to 0

Bit 12 **Received Target Abort Status**

Read: 0 = No target abort is received since this bit was cleared.
1 = Target abort is received from a PCI master.
Write: 0 = No effect
1 = Clears this bit to 0

Bit 11 **Reserved (Read only)**

Bits 10:9 **DEVSEL# Timing Status (Read Only)**

The DEVSEL# assertion timing is fixed to medium timing.
01 = Medium DEVSEL# Assertion

Bit 8 **Data Parity Detected Status**

The bit is set when
1) 85C496 acted as the bus master for the operation in which the error occurred, and
2) the Parity Error Response bit in Command Register is set.
Write: 0 = No effect
1 = Clears this bit to 0

Bit 7 **Fast Back-to-Back Capable Status (Read Only)**

1 = Always enabled

Bits 6:0 **Reserved (Read Only)**

Register 08h **Device Revision Identification**

Default Value: 02h
Attribute: Read Only

This register specifies a device specific revision identifier. The current version of 85C496/497 is 02h.

Bits 7:0 **Revision ID =** 02h

Register 09h ~ 0Bh Device Class Code

Default Value: 060000h
Attribute: Read Only

The Class Code register is read-only and is used to identify the generic function of the device and a specific register-level programming interface. Since SiS85C496 and SiS85C497 share a configuration space, the class code is chosen to show as a PCI host bridge.

Bits 23:0 PCI Class Code Register
Base Class : PCI Bridge = 06h
Sub-Class : Host Bridge = 0000h

Register 0Eh Device Header Type

Default Value: 00h
Attribute: Read Only

This register defines the layout of register 10h to 3Fh. The 00h is used to defined a single function device.

Bits 7:0 PCI Header Type Register
00h : Single Function Device

3.2 85C496 Specific Registers (40h ~ 7Fh)

The 85C496 consists of a set of configuration registers to specify CPU configuration, DRAM configuration, external cache configuration, PCI configuration, IDE controller configuration and operating parameters.

Register 40h CPU Configuration

Default Value: 00h
Attribute: Read / Write

The register is defined to configure 85C496 circuitry for various CPU types. This register must be set as soon as possible after the CPU type is determined.

Bit 7 Reserved

Bit 6 CPU Burst Write Enable (P24T/P24D/M7/Cx 5x86/**Enhanced Am486)**
0 = Disable
1 = Enable
For the other CPU types, this bit must be disabled.

Bit 5 CPU Internal Cache Write Back Mode Enable
0 = Disable
1 = Enable

Bits 4:2 **Host Processor Type Selection (the three bits must be set immediately after powerup and match the setting in Reg 81[4:2])**
000 = Intel i486DX/DX2/AMD Am486DX/DX2/DX4
001 = Intel i486 SL-Enhanced/ i486 DX4
010 = Intel P24D/P24T/AMD Enhanced Am486 DX2/DX4/Cyrix Cx 5x86
011 = AMD Am486DXL
101 = Cyrix Cx486DX / Cx486DX2
Others = Reserved

Bits 1:0 **DRAM Speed**
00 = Slowest (50Mhz)
01 = Slower (40Mhz)
10 = Faster (33Mhz)
11 = Fastest (25Mhz)

Register 41h DRAM Configuration

Default Value: 00h
Attribute: Read / Write

The register defines the DRAM parity check, type, refresh mode, and access timing.

Bit 7 **DRAM Parity Check Enable**
0 = Disable
1 = Enable

Bits 6:5 **DRAM Type:**
(If more than one DRAM type is installed, set to the smallest type)
00 = 256K/512K × 32/36 bit type
01 = 1M/2M × 32/36 bit type
10 = 4M/8M/16M/32M × 32/36 bit type
11 = Reserved

Bit 4 **DRAM Slow Refresh Mode Enable**
0 = Refresh occurred at normal frequency
1 = Refresh occurred at one quarter of normal frequency

Bit 3 **DRAM RAS# to CAS# Delay**
0 = 3T (T : CPU Clock)
1 = 2T

Bit 2 **DRAM Write Cycle Post Enable**
0 = Disable, 1 wait state
1 = Enable, 0 wait state

Bit 1 **DRAM Write Cycle CAS# Pulse Width**
0 = 2T (T : CPU Clock)
1 = 1T

Bit 0 **DRAM CAS Precharge Time**
0 = 2T (T : CPU Clock)
1 = 1T

Register 42h ~ 43h Cache Configure

Default Value: 0000h
Attribute: Read / Write
The register defines Cache access timing, external cache size, and attributes.

Bit 15 **CPU burst order mode select**
0 = Toggle mode (Intel compatible burst order mode)
1 = Linear burst order mode

Bits 14:12 **Reserved**

Bit 11 **Level 2 Cache Single Write/Burst Write Cycle**
0 = 3T/2T (T : CPU Clock)
1 = 2T/1T

Bit 10 **Level 2 Cache Burst Read Subsequent Cycle Wait State**
0 = 1T (T : CPU Clock)
1 = 2T

Bit 9 **Level 2 Cache / DRAM Single Read or Burst Read Lead-off Cycle Wait State**
0 = 3T (T : CPU Clock)
1 = 2T

Bit 8 **Level 2 Cache Write Back / Write Through Mode**
0 = Write Through Mode
1 = Write Back Mode

Bit 7:5 **Level 2 Cache Size**
001 = 64 KByte
010 = 128 KByte
011 = 256 KByte
100 = 512 KByte
101 = 1 MByte
Others: reserved

Bit 4 **Level 2 Cache Interleave Control**
0 = Single bank, non-interleave mode
1 = Double bank, interleave mode

Bit 3 **Level 2 Cache Dirty Bit Select**
0 = Enable Dirty Pin Input
1 = Force Dirty Status as Always Dirty

Bit 2 **Level 2 Cache Tag Width Select (Multiplexed Pin 128 Select)**
0 = 8 Bit Tag (Pin 128 as TA7)
1 = 7 Bit Tag (Pin 128 as Dirty)

Bit 1 **Level 2 Cache Test (Always Hit) Enable**
0 = Normal operation
1 = External cache always hit.

Bit 0 **External Cache Enable**
0 = Disable
1 = Enable

Register 44h ~ 45h Shadow Configure

Default Value: 0000h
Attribute: Read / Write

The register defines the shadow area size and attributes for memory 0C0000h ~ 0FFFFFh.

Bits 15:12 Reserved

Bit 11 Internal Cache Cacheable Area Control
(Level 2 Cache Not Affected)
0 = The shadowed L2-cacheable area is cacheable by internal (L1) cache
1 = Non-cacheable by internal (L1) cache
This bit should be enabled for CPU with write-back L1 cache

Bit 10 PCI, ISA Master Access Shadow RAM Area Enable
0 = Disable
1 = Enable

Bit 9 Shadow RAM Read Control
0 = Read Disable
Reading to 0C0000h ~ 0FFFFFh are directed to Extension Bus (PCI, ISA).
1 = Read Enable
Reads to the areas designated by bit 7 to 0 are directed to the onboard DRAM.

Bit 8 Shadow RAM Write Control
0 = Write Enable
Writes to the areas designated by bit 7 to 0 are directed to the onboard DRAM.
1 = Write Disable
Writing to C0000h ~ FFFFFh are directed to Extension Bus (PCI, ISA).

Bit 7 0F8000h-0FFFFFh Shadow RAM Enable
0 = Disable
1 = Enable

Bit 6 0F0000h-0F7FFFh Shadow RAM Enable
0 = Disable
1 = Enable

Bit 5 0E8000h-0EFFFFh Shadow RAM Enable
0 = Disable
1 = Enable

Bit 4 0E0000h-0E7FFFh Shadow RAM Enable
0 = Disable
1 = Enable

Bit 3 0D8000h-0DFFFFh Shadow RAM Enable
0 = Disable
1 = Enable

Bit2 0D0000h-0D7FFFh Shadow RAM Enable
0 = Disable
1 = Enable

Bit 1 **0C8000h-0CFFFFh Shadow RAM Enable**
0 = Disable
1 = Enable

Bit 0 **0C0000h-0C7FFFh Shadow RAM Enable**
0 = Disable
1 = Enable

NOTE: Bit 11 defines whether a shadowed and cacheable (by external cache) area can be cached by CPU internal cache. This bit is specially defined for Intel P24T, P24D during write-back mode.

Register 46h **Cacheable Control**

Default Value: 0000h
Attribute: Read / Write

Bit 7 **0F8000h-0FFFFFh RAM Cacheable (Internal and External Cache)**
Bit 6 **0F0000h-0F7FFFh RAM Cacheable (Internal and External Cache)**
Bit 5 **0E8000h-0EFFFFh RAM Cacheable (Internal and External Cache)**
Bit 4 **0E0000h-0E7FFFh RAM Cacheable (Internal and External Cache)**
Bit 3 **0D8000h-0DFFFFh RAM Cacheable (Internal and External Cache)**
Bit 2 **0D0000h-0D7FFFh RAM Cacheable (Internal and External Cache)**
Bit 1 **0C8000h-0CFFFFh RAM Cacheable (Internal and External Cache)**
Bit 0 **0C0000h-0C7FFFh RAM Cacheable (Internal and External Cache)**
0 = Non-cacheable
1 = Cacheable

Register 47h **85C496 Address Decoder**

Default Value: 00h
Attribute: Read / Write

Bits [3:1] specifies ISA address forwarding. When a memory access to the memory area defined by the following table, if a bit is written with 0, the access will be forward to PCI bus first, and ,then, forward to ISA bus if none of the PCI devices responses to the access within three PCI clock period. If a bit is written with 1, the memory cycle forward to PCI bus only, which allows PCI write cycles posted for a better performance.

Bit 4 enables concurrent circuitry for CPU to main memory and PCI masters to PCI slaves cycles.

Bits 7:5 **Reserved**

Bit 4 **Concurrent Access Control** (CPU to Memory and PCI to PCI)
0 = Disable
1 = Enable

Bit 3 **CPU Memory Access to FFF80000h ~ FFFDFFFFh forward to**
0 = PCI and ISA Bus
1 = PCI Bus

Bit 2 **CPU Memory Access to B0000h ~ BFFFFh forward to**
0 = PCI and ISA Bus
1 = PCI Bus

Bit 1 **CPU Memory Access to A0000h ~ AFFFFh forward to**
0 = PCI and ISA Bus
1 = PCI Bus

Bit 0 **DRAM A, B, D, E Segment Relocate Enable**
0 = Disable
1 = Enable, if total DRAM installed is 8Mbyte (include) or less,
and segment D and E is not shadowed, then the DRAM area A0000h ~ BFFFFh and D0000h ~ EFFFFh are appended to the top of DRAM. Accesses to memory space above physically memory boundary will be mapped to memory segment A, B, D, and E.

Register 48h ~ 4Fh **DRAM Boundary**

Default Value: 00h for each register
Attribute: Read / Write

85C496 provides eight logical banks which are translated from four physical double-side banks. Each register records the accumulated DRAM size including the present and previous banks.

Bits 7:0 **DRAM Bank Boundary Address A[27:20]**
00h = 0Mbyte
01h = 1Mbyte
... ...
80h = 128Mbyte

The following table shows a possible combination of the Boundary register values of a system equipped with various SIMM.

Physical Bank	SIMM Type	Memory Size in Logical Bank	Boundary Reg Value
0	Single Side 1Mbyte	0 ← 1M 1 ← 0M	48 ← 01 49 ← 01
1	Double Side 8Mbyte	2 ← 4M 3 ← 4M	4A ← 5 4B ← 9
2	Double Side 2Mbyte	4 ← 1M 5 ← 1M	4C ← 0A 4D ← 0B
3	Single Side 16Mbyte	6 ← 16M 7 ← 0 M	4E ← 1B 4F ← 1B

Register 50h ~ 51h Exclusive Area 0 Setup

Default Value: 0000h
Attribute: Read / Write

An exclusive area could be either an on-board non-cacheable area or a PCI memory hole. If the area is defined as a non-cacheable area, L1 and L2 cache will not cache memory located within this area. If the area is defined as a PCI hole, a memory access within this area will be forward to PCI bus. The area can be disabled by setting area size to 0. The starting address of the exclusive area is defined by the bit [11:0]. The defined address must be aligned with the address size.

Bit 15 Exclusive Area 0 Select
0 = Non-cacheable Area
1 = PCI Memory Hole

Bits 14:12 Area Size
000 = 0 KByte (Exclusive Area 0 Disabled)
001 = 64 KByte
010 = 128 KByte
011 = 256 KByte
100 = 512 KByte
101 = 1 MByte
110 = 2 MByte
111 = 4 MByte

Bits 11:0 Exclusive Area 0 Base Address A[27:16]

Register 52h ~ 53h Exclusive Area 1 Setup

Default Value: 0000h
Attribute: Read / Write

An exclusive area could be either an on-board non-cacheable area or a PCI memory hole. If the area is defined as a non-cacheable area, L1 and L2 cache will not cache memory located within this area. If the area is defined as a PCI hole, a memory access within this area will be forward to PCI bus. The area can be disabled by setting area size to 0. The starting address of the exclusive area is defined by the bit [11:0]. The defined address must be aligned with the address size.

Bit 15 Exclusive Area 1 select
0 = Non-cacheable Area
1 = PCI Memory Hole

Bits 14:12 Area Size
000 = 0 KByte (Exclusive Area 1 Disabled)
001 = 64 KByte
010 = 128 KByte
011 = 256 KByte
100 = 512 KByte
101 = 1 MByte

110 = 2 MByte
111 = 4 MByte

Bits 11:0 Exclusive Area 1 Base Address A[27:16]

Register 54h ~ 55h Exclusive Area 2 Setup

Default Value: 0000h
Attribute: Read / Write

An exclusive area could be either an on-board non-cacheable area or a PCI memory hole. If the area is defined as a non-cacheable area, L1 and L2 cache will not cache memory located within this area. If the area is defined as a PCI hole, a memory access within this area will be forward to PCI bus. The area can be disabled by setting area size to 0. The starting address of the exclusive area is defined by the bit [11:0]. The defined address must be aligned with the address size.

Bit 15 Exclusive Area 2 select
0 = Non-cacheable Area
1 = ISA Memory Hole

Bits 14:12 Area Size
000 = 0 KByte (Exclusive Area 1 Disabled)
001 = 64 KByte
010 = 128 KByte
011 = 256 KByte
100 = 512 KByte
101 = 1 MByte
110 = 2 MByte
111 = 4 MByte

Bits 11:8 Reserved

Bits 7:0 Exclusive Area 2 Base Address A[23:16]

Register 56h PCI / Keyboard Configure

Default Value: 00h
Attribute: Read / Write

The register defines the arbitration protocol, post and burst of PCI Bus as well as emulated keyboard ports.

Bits 7:6 PCI Arbitration Scheme Select:
00 = Weakest CPU Request
01 = Weaker CPU Request
10 = Stronger CPU Request
11 = Strongest CPU Request

Bit 5 CPU -to- PCI Memory Post Write Buffer Enable
0 = Disable
1 = Enable

Bit 4 **CPU -to- PCI Memory Burst Write Enable**
0 = Disable
1 = Enable

Bit 3 **Reserved**

Bit 2 **Keyboard Reset / INIT CPU Command Pending during SMI Control**
0 = Disable
1 = Enable

Bit 1 **Emulated PS/2 Keyboard 92h Port Control**
0 = Disable
1 = Enable

Bit 0 **Emulated Keyboard Fast Reset Command Latency**
0 = 2 μs
1 = 6 μs

Register 57h **Output Pin Configuration**

Default Value: 00h
Attribute: Read / Write
The register defines output buffer driving capability and multiplexed output pin selection. Bit 7 must turns on when a PCI master with burst read / write ability presents.
Bit 2:0 provides various combinations of the following output signals.

Bit 7 **PCI Master Burst Read/Write Main Memory Control.**
0 = Disable
1 = Enable

Bit 6 **CAS[3:0]# Output Buffer Driving Capability Control**
0 = 12 mA
1 = 24 mA

Bit 5 **MA[10:0] Output Buffer Driving Capability Control**
0 = 12 mA
1 = 24 mA

Bit 4 **MWE# Output Buffer Driving Capability Control**
0 = 12 mA
1 = 24 mA

Bit 3 **Multiplexed Pin 207, 204 Select**
0 = Pin 207: LBD#, PIN 204: LRDY#
1 = Pin 207: PREQ2#, PIN 204: PGNT2#

Bits 2:0 **Multiplexed Pin 157, Pin 126, Pin 127 Select**

The possible combinations of output pins and functions are shown in the following table.

Output pin combinations:

Bits 2:0	Pin 157	Pin 126	Pin 127
000	RAS7#	DIRTY	DIRTYWE#
100	MA11	DIRTY	DIRTYWE#
010	RAS7#	MA11	DIRTYWE#
110	MA11	MA11	DIRTYWE#
0X1	RAS7#	PREQ3#	PGNT3#
1X1	MA11	PREQ3#	PGNT3#

Function combinations:

Bits 2:0	Largest DRAM Type	Maximum DRAM Bank No.	8 bit Tag + 1 bit Dirty	PCI Master No.
000	4Mx36	8	Yes	3
100	16Mx36	7	Yes	3
010	16Mx36	8	No	3
110	16Mx36	7	No	3
0X1	4Mx36	8	No	4
1X1	16Mx36	7	No	4

Register 58h ~ 59h **Build-in IDE Controller / VESA Bus Configuration**

Default Value: 0000h
Attribute: Read / Write

The register defines the wait state, channel address and power management of the built-in IDE controller as well as VESA local bus access timing.

Bit 15 **IDE Controller Post Write Buffer Enable**

0 = Disable

1 = Enable

Bits 14:13 **Dual prefetch buffers control**

0X : Disable Channel 0 and Channel 1 prefetch buffers

10 : Enable Channel 0 and Channel 1 prefetch buffers

11 : Enable Channel 0 prefetch buffer and Disable Channel 1 prefetch buffer

Note: *"X" means don't care.*

Bits 12:10 **IDE Controller Address Setup Time Select**

000 = 8 CPU Clocks

001 = 1 CPU Clocks

010 = 2 CPU Clocks

011 = 3 CPU Clocks

100 = 4 CPU Clocks

101 = 5 CPU Clocks

110 = 6 CPU Clocks
111 = 7 CPU Clocks

Bit 9 **IDE Controller Channel Select**
0 = Channel 0 on I/O Ports 1FXh
Channel 1 on I/O Ports 17Xh
1 =Channel 0 on I/O Ports 17Xh
Channel 1 on I/O Ports 1FXh

Bit 8 **IDE Controller Enable**
0 = Disable
1 = Enable

Bit 7 **I/O ports 1Fx & 3F6 Control**
0 = Enable
1 = Disable

Bit 6 **I/O ports 17x & 376 Control**
0 = Enable
1 = Disable

Bit 5 **Reserved**

Bit 4 **PMU DEVDET Generation Mode**
0 = Synchronous Mode
1 = Asynchronous Mode

Bit 3 **Reserved**

Bit 2 **CPU INIT queued by Stop Grant Mode Control**
0 = Disable
1 = Enable

Bit 1 **VESA Local Device RAMDAC I/O Write Cycle Shadow Control**
0 = Normal operation (No shadow)
1 = CPU I/O Writes to 03C6h-03C9h will be duplicated to PCI Bus even when VESA local device claims this cycle

Bit 0 **VESA Local Device LBD# Sampling Point**
0 = End of T3
1 = End of T2

Register 5Ah **SMRAM Remapping Configuration**

Default Value: 00h
Attribute: Read / Write

The register defines A20M# behavior and SMRAM remapping during System Management Mode (SMM). If bit [5] is enabled, A20M# is always deasserted during SMM to ensure that the SMI handler is able to access the full 4 GByte memory space.

85C496/497 provides a 64KByte memory segment located at A0000h ~ AFFFFh or B0000h ~ BFFFFh as System Management RAM. During SMM, memory accesses to logical address to 60000h ~ 6FFFFh or E0000h ~ EFFFFh are mapped to Segment A or B depending on the

setting in bit[4:3]. When bit[2] is enabled, system is forced into SMM. A SMI handler can be moved into SMRAM after this bit is enabled.

Bit 7 **ISA Bus Master Request Control (this bit must be written with 1)**

0 = Disable

1 = Enable

Bit 6 **Reserved**

Bit 5 **SMM A20M# Assertion Control**

0 = Disable, A20M# is not altered in SMM

1 = Enable, A20M# is deasserted in SMM and return to previous status after SMM.

Bits 4:3 **SMRAM Remapping Mode**

00 = Logical accesses to 60000h ~ 6FFFFh will be directed to physical memory,
A0000h ~ AFFFFh

01 = Logical accesses to 60000h ~ 6FFFFh will be directed to physical memory,
B0000h ~ BFFFFh

10 = Logical accesses to E0000h ~ EFFFFh will be directed to physical memory
, A0000h ~ AFFFFh

11 = Logical accesses to E0000h ~ EFFFFh will be directed to physical memory
, B0000h ~ BFFFFh

Bit 2 **SMRAM Initialization Mode Enable**

0 = Disable

1 = Enable, SMRAM remapping will be enabled according to the setting in bit[4:3] even when not in SMM.

A SMI handler code can be moved into SMRAM if this bit is enabled.

Bit 1 **SMRAM Remapping Enable**

0 = Disable, SMRAM will not be remapped regardless the status of SMI mode.

1 = Enable

Bit 0 **Reserved**

Register 5Bh **Programmable I/O Traps Configure**

Default Value: 00h
Attribute: Read / Write

A programmable I/O trap is a 1 to 128 byte programmable I/O port address. If the trap is enabled, accesses to the programmed address will be monitored by SiS 85C496/497. Once an access happens, depending on the Reg 90h, 92h, 94h and A0h, a timer reload, or a SMI could be triggered.

Bits 7:5 **Programmable I/O Trap 1 Area**

000 = 1 Byte
001 = 2 Byte
010 = 4 Byte
011 = 8 Byte
100 = 16 Byte
101 = 32 Byte
110 = 64 Byte
111 = 128 Byte

Bit 4 **Programmable I/O Trap 1 Enable**
0 = Disable
1 = Enable

Bits 3:1 **Programmable I/O Trap 0 Area**
000 = 1 Byte
001 = 2 Byte
010 = 4 Byte
011 = 8 Byte
100 = 16 Byte
101 = 32 Byte
110 = 64 Byte
111 = 128 Byte

Bit 0 **Programmable I/O Trap 0 Enable**
0 = Disable
1 = Enable

Register 5Ch ~ 5Dh **Programmable I/O Trap 0 Base**

Default Value: 0000h
Attribute: Read / Write

Bits 15:0 **Programmable I/O Trap 0 Address A[15:0]**

Register 5Eh ~ 5Fh **Programmable I/O Trap 1 Base**

Default Value: 0000h
Attribute: Read / Write

Bits 15:0 **Programmable I / O Trap 1 Address A[15:0]**

Register 60h, 61h **IDE Controller Channel 0 Configuration**

Default Value: 0000h
Attribute: Read / Write

Bits 15:12 **IDE Controller Channel 0 Slave Drive Recovery Time**

Bits 11:8 **IDE Controller Channel 0 Master Drive Recovery Time**

Bits 7:4 IDE Controller Channel 0 Slave Drive Command Active Time

Bits 3:0 IDE Controller Channel 0 Master Drive Command Active Time

IDE Controller Channel Wait state Table

B3:B0	CPU Clocks
0000	16
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Register 62h~63h IDE Controller Channel 1 Configuration

Default Value: 0000h
Attribute: Read / Write

Bits 15:12 IDE Controller Channel 1 Slave Drive Recovery Time
Bits 11:8 IDE Controller Channel 1 Master Drive Recovery Time
Bits 7:4 IDE Controller Channel 1 Slave Drive Command Active Time
Bits 3:0 IDE Controller Channel 1 Master Drive Command Active Time

IDE Controller Channel Wait state Table

B3:B0	CPU Clocks
0000	16
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11

1100	12
1101	13
1110	14
1111	15

Register 64 ~ 65h Exclusive Area 3 setup

Default Value: 00h
Attribute: Read / Write

Bit 15 **Exclusive Area 3 designated as**
0 = Onboard non-cacheable memory area
1 = PCI/ISA Bus non-postable memory area

Bits 14:12 **Exclusive Area 3 Size**
000 = 0 KByte (Exclusive Area 3 Disabled)
001 = 64 KByte
010 = 128 KByte
011 = 256 KByte
100 = 512 KByte
101 = 1 MByte
110 = 2 MByte
111 = 4 MByte

Bits 7:0 **Exclusive Area 3 Starting Address A[23:16]**

Register 66h EDO DRAM Configuation

Default Value: 00h
Attribute: Read / Write

Bit 7 **Bank 7 DRAM type selection**
Bit 6 **Bank 6 DRAM type selection**
Bit 5 **Bank 5 DRAM type selection**
Bit 4 **Bank 4 DRAM type selection**
Bit 3 **Bank 3 DRAM type selection**
Bit 2 **Bank 2 DRAM type selection**
Bit 1 **Bank 1 DRAM type selection**
Bit 0 **Bank 0 DRAM type selection**
0 = FPM (Fast page mode) DRAM
1 = EDO (Extended data-out) DRAM

Register 67h Miscellaneous Control

Default Value: 00h
Attribute: Read / Write

Bit 7 **Emulated Keyboard fast reset and A20M# control I/O write cycle flow through to ISA bus control**
0 = Disable
1 = Enable

Bit 6 **Emulated CPU A20M# control**
0 = Enable
1 = Disable

Bit 5 **Emulated CPU fast reset control**
0 = Enable
1 = Disable

Bit 4 **Emulated Keyboard 60h/64h Port Control**
0 = Enable
1 = Disable

Bit 3 **This bit and the external pull down resistor on LBIDE# determine whether pin 158 and 159 are configured to support two banks of DRAM, PCI-to-PCI bridge on PCI bus, or one DRAM bank and CPU reset command from external keyboard controller.**

Detect LBIDE#	Bit 3	Pin 158	Pin 159
1 (High)	X	RAS6#	RAS5#
0 (Low)	0	BOFF#	HLOCK#
0 (Low)	1	RAS6#	RC#

Note: *"X" means don't care.*

Bits 2:1 **Reserved**

Bit 0 **FPM/EDO type DRAM detection control**
0 = Disable
1 = Enable

Register 68h~69h Asymmetry DRAM Configuation

Default Value: 00h
Attribute: Read / Write

Bits 15:14 **Bank 7 asymmetry type DRAM selection**

Bits 13:12 **Bank 6 asymmetry type DRAM selection**

Bits 11:10 **Bank 5 asymmetry type DRAM selection**

Bits 9:8 **Bank 4 asymmetry type DRAM selection**

Bits 7:6 **Bank 3 asymmetry type DRAM selection**

Bits 5:4 **Bank 2 asymmetry type DRAM selection**

Bits 3:2 **Bank 1 asymmetry type DRAM selection**

Bits 1:0 **Bank 0 asymmetry type DRAM selection**

00 = DRAM type is specified by Reg. 41[6:5]

01 = DRAM type is 1M by 36/32 bit, 12 row addresses by 8 column addresses, overrides Reg. 41[6:5]

10 = DRAM type is 2M by 36/32 bit, 12 row addresses by 9 column addresses, overrides Reg. 41[6:5]

11 = DRAM type is 4M by 36/32 bit, 12 row addresses by 10 column addresses, overrides Reg. 41[6:5]

3.3 85C497 Specific Registers (80h ~ FFh)

The 85C497 consists of a set of configuration registers to specify, power management unit configuration, ISA bridge related configuration, and operating parameters.

Register 80h **PMU Configuration**

Default Value: 00h
Attribute: Read / Write

The register consists the major switches of power management functions.
Bit [7] controls the generation of system management interrupt. If the bit is disabled, the system is blocked from entering system management mode regardless the other register status.
Bit [6] controls the generation of STPCLK#. STPCLK# could be asserted when the bit is enabled.

Bit [5] controls the generation of Software STPCLK# Generation. STPCLK# can be generated by enabling a write-only bit, Reg 84h bit [0] when bit [5] is enabled.

Bit [4] chooses the source of entering the system management mode. For CPUs with system management features, SMI# can be chosen as SMM trigger source. For the other CPUs without SMI features, IRQ can be used to force system entering SMM.

Bit [2] controls the generation of Software SMI# Generation. SMI# can be generated by enabling a write-only bit, Reg 9Eh bit [0] when bit [3] is enabled.

Bit [1] controls the break switch. When this bit is enabled, pressing break switch triggers SMI#.

Bit [0] controls the clock throttling mode. If the bit is enabled, STPCLK# is periodically asserted and deasserted depending on the On timer and Off timer setting (Reg 8E and 8F).

Bit 7 **SMM Enable**

0 = Disable

1 = Enable

This bit is used to enable/disable the SMI# generation.

Bit 6 **Stop Clock Enable**

0 = Disable

1 = Enable

This bit is used to enable /disable the of STPCLK# generation.

Bit 5 **Software STPCLK# Enable**
0 = Disable
1 = Enable

Bit 4 **System Management Request Selection**
0 : By SMI#
1 : By IRQ
This bit is used to select how to enter the system management mode

Bit 3 **Reserved and must be written with 1**

Bit 2 **Software SMI Generation Control**
0 = Disable Soft-SMI
1 = Enable Soft-SMI

Bit 1 **Break Switch Control**
0 = Disable Break Switch
1 = Enable Break Switch

Bit 0 **Clock Throttling Enable**
0 = Disable
1 = Enable

Register 81h **PMU CPU Type Configuration**

Default Value: 00h
Attribute: Read / Write

The register is defined to configure 85C496 circuitry for various CPU types. This register must be set as soon as possible after the CPU type is determined.

Bit 7 selects different deturbo function. When the bit is enabled. pressing deturbo switch triggers SMI. The status of Deturbo Switch can be observed on Reg C7 bit 0.

Bit 7 **Deturbo Switch Trigger Select**
0 = Hold CPU
1 = SMI

Bits 6:5 **Reserved**

Bits 4:2 **Host Processor Type Selection (this three bits must be set immediately after powerup and match the setting in Reg 40[4:2])**
000 = Intel i486DX/DX2/AMD Am486DX/DX2/DX4
001 = Intel i486 SL-Enhanced/ i486 DX4
010 = Intel P24D/P24T/AMD Enhanced Am486 DX2/DX4/Cyrix Cx 5x86
101 = Cyrix Cx486DX / Cx486DX2
Others = Reserved

Bits 1:0 **Select SMI IRQ for Non-SMI CPU**
00 = IRQ 10
01 = IRQ 11
10 = IRQ 12
11 = IRQ 15

Register 82h Port 22h Mirror

Default Value: 00h
Attribute: Read Only

This register stores the current value of I/O port 22h.

Bits 7:0 CPU I/O Port 22h Mirror

Register 83h Port 70h Mirror

Default Value: 00h
Attribute: Read Only

This register stores the current value of I/O port 70h.

Bits 7:0 CPU I/O Port 70h Mirror

Register 84h Soft-STPCLK# Assertion / Break SW BLK CLR

Default Value: 00h
Attribute: Write Only

Bit [1] is used to clear break switch blocking feature in PMU. After break switch is pressed, Bit [1] must be written with 1 and the break switch is ready for next action. Bit [0] is write-only. When the bit is enabled, a STPCLK# is asserted.

Bits 7:3 Reserved

Bit 2 Deturbo Switch Blocking Clear

0: No effect

1: Clear Deturbo Switch Blocking

This bit must be written with 1 after the deturbo switch is press in order to re-enable it.

Bit 1 Break Switch Blocking Clear

0: No effect

1: Clear Break Switch Blocking

This bit must be written with 1 after the break switch is press in order to re-enable it.

Bit 0 STPCLK# Assertion

0: No effect

1: Assert the STPCLK# immediately

Register 85h STPCLK# Event Control

Default Value: 00h
Attribute: Read / Write

The Bit [7:5] selects the conditions to assert STPCLK# when a chosen event happens.
The Bit [4:0] selects the conditions to deassert STPCLK# when a chosen event happens

Bit 7 STPCLK# will be asserted if the Fast Timer times out
0 = Disable
1 = Enable

Bit 6 STPCLK# will be asserted if the Generic Timer times out
0 = Disable
1 = Enable

Bit 5 STPCLK# will be asserted if the Slow Timer times out
0 = Disable
1 = Enable

Bit 4 STPCLK# will be deasserted if the Break Switch is pressed
0 = Disable
1 = Enable

Bit 3 STPCLK# will be deasserted if an selected IRQ Event happens (refers Reg 86h, 87h)
0 = Disable
1 = Enable

Bit 2 STPCLK# will be deasserted if a NMI# Event happens
0 = Disable
1 = Enable

Bit 1 STPCLK# will be deasserted if a DMA Request happens
0 = Disable
1 = Enable

Bit 0 STPCLK# will be deasserted if a PCI Master Request happens
0 = Disable
1 = Enable

Register 86h ~ 87h STPCLK# Deassertion IRQ Selection

Default Value: 0000h
Attribute: Read / Write

The selected IRQs will cause STPCLK# be deasserted if Reg 85h bit 3 is set.

Bit 15 IRQ 15
Bit 14 IRQ 14
Bit 13 IRQ 13

Bit 12 **IRQ 12**
Bit 11 **IRQ 11**
Bit 10 **IRQ 10**
Bit 9 **IRQ 9**
Bit 8 **IRQ 8**
Bit 7 **IRQ 7**
Bit 6 **IRQ 6**
Bit 5 **IRQ 5**
Bit 4 **IRQ 4**
Bit 3 **IRQ 3**
Bit 2 **IRQ 2**
Bit 1 **IRQ 1**
Bit 0 **IRQ 0**
0 = Disable
1 = Enable

Register 88h **Timer Control**

Default Value: 00h
Attribute:0 Read / Write

Bit [5:3] controls the timer behavior when STPCLK# is asserted. If one of the bits is disabled, the corresponding timer keeps running after STPCLK# is asserted; otherwise; the timer is disabled after STPCLK# is asserted and reload when system resumes.

Bit [2:0] are timer control bits. A disabled timer keeps running till it times out, but no SMI# is generated. Once a timer is enabled, the timer is reload to a pre-defined value in Reg 89h, 8Ah, and 8Bh and starts counting down.

Bits 7:6 **Reserved**

Bit 5 **Slow Timer STPCLK# Blocking Control**
0 = Timer keeps running when STPCLK# is asserted.
1 = Timer is disabled when STPCLK#is asserted and reload after STPCLK# is deasserted.

Bit 4 **Generic Timer STPCLK# Blocking Control**
0 = Timer keeps running when STPCLK# is asserted.
1 = Disable Generic Timer when STPCLK# is asserted and enables Generic Timer after STPCLK# is deasserted.

Bit 3 **Fast Timer STPCLK# Blocking Control**
0 = Timer keeps running when STPCLK# is asserted.
1 = Timer is disabled when STPCLK#is asserted and reload after STPCLK# is deasserted.

Bit 2 **Slow Timer Enable**
0 = Disable
1 = Enable

Bit 1 **Generic Timer Enable**
0 = Disable
1 = Enable

Bit 0 **Fast Timer Enable**
0 = Disable
1 = Enable

Register 89h **Fast Timer Count**

Default Value: 00h
Attribute: Read / Write

When the register is written, the count of Fast Timer is set according to the register value. When the register is read, the return value is the current count of Fast Timer.

Bits 7:0 **Timer Counter, the minimum value is 2**
The timer-expire interval is translated by the follow equation:
Timer-Expire Interval = (Timer Counter - 1) × 0.6 seconds
The interval is allowed between 0.6 seconds and 150 seconds.

Register 8Ah **Generic Timer Count**

Default Value: 00h
Attribute: Read / Write
When the register is written, the count of Generic Timer is set according to the register value. When the register is read, the return value is the current count of Generic Timer.

Bits 7:0 **Timer Counter, the minimum value is 2**
The timer-expire interval is translated by the follow equation:
Timer-Expire Interval = (Timer Counter - 1) × 0.3 second
The interval is allowed between 0.3 seconds and 75 seconds.

Register 8Bh **Slow Timer Count**

Default Value: 00h
Attribute: Read / Write

When the register is written, the count of Slow Timer is set according to the register value. When the register is read, the return value is the current count of Slow Timer.

Bits 7:0 **Timer Counter, the minimum value is 2**

The timer-expire interval is translated by the follow equation:

Timer-Expire Interval = (Timer Counter - 1) × Time Base

(Please refer to Register 96h~97h [15:14])

Register 8Ch **Timers Reset**

Default Value: 00h
Attribute: Write Only

Once a bit is written with 1, the correspond timer will be reload a pre-defined value.

Bits 7:3 **Reserved**

Bit 2 **Fast Timer Reload**

0 = No effect
1 = Reload Fast Timer

Bit 1 **Generic Timer Reload**

0 = No effect
1 = Reload Generic Timer

Bit 0 **Slow Timer Reload**

0 = No effect
1 = Reload Slow Timer

Register 8Dh **RMSMIBLK Timer Count**

Default Value: 00h
Attribute: Read / Write

A consecutive SMI will be blocked until the RMSMIBLK Timer times out. For Intel SL-enhanced CPU, the minimum counter value is 2.

Bits 7:0 **Timer Counter, the minimum value is 2**

The timer-expire interval is translated by the follow equation:

Timer-Expire Interval = (Timer Counter - 1) × 35 μs

The interval is allowed between 35*μs and 9 ms*.

Register 8Eh **Clock Throttling On Timer Count**

Default Value: 00h
Attribute: Read / Write

The timer starts counting down when STPCLK# is deasserted. Once it times out, STPCLK# is asserted.

Bits 7:0 **Timer Counter, the minimum value is 2**

The timer-expire interval is translated by the following equation:

Timer-Expire Interval = (Timer Counter - 1) × 35 μs

The interval is allowed between 35*μs and 9 ms*.

Register 8Fh Clock Throttling Off Timer Count

Default Value: 00h
Attribute: Read / Write

The timer starts counting down when STPCLK# is asserted. Once it times out, STPCLK# is deasserted.

Bits 7:0 Timer Counter, the minimum value is 2
The timer-expire interval is translated by the follow equation:
Timer-Expire Interval = (Timer Counter - 1) × 35 μs
The interval is allowed between 35*μs and 9 ms*.

Register 90h ~ 91h Clock Throttling On Timer Reload Condition

Register Address: 90h ~ 91h
Default Value: 00h
Attribute: Read / Write

The register defines the conditions which cause timer to reload. If a selected request occurs before the timer times out, it will be reload.

Bits 15:10 Reserved

Bit 9 Video Access Request (Memory Segment A0000h ~ BFFFFh , I/O Port 3B0h ~ 3B7h, 3C0h ~ 3CFh, 3D0h ~ 3DFh)
0 = Disable
1 = Enable

Bit 8 IRQ Request
0 = Disable
1 = Enable

Bit 7 DMA Request
0 = Disable
1 = Enable

Bit 6 Keyboard and Mouse Event Request (IRQ 1, 3, 4 and 12)
0 = Disable
1 = Enable

Bit 5 Hard Disk / Floppy Disk Access
0 = Disable
1 = Enable

Bit 4 Parallel Port Access (Port 278h-27Fh, 378h-37Fh and 3BCh-3BFh)
0 = Disable
1 = Enable

Bit 3 **Serial Port Access (Port 2F8h-2FFh, 3F8h-3FFh,2E8h-2EFh and 3E8h-3EFh)**
0 = Disable
1 = Enable

Bit 2 **I/O Address Trap 1 Access**
0 = Disable
1 = Enable

Bit 1 **I/O Address Trap 0 Access**
0 = Disable
1 = Enable

Bit 0 **PCI Master Request**
0 = Disable
1 = Enable

Register 92h ~ 93h Fast Timer Reload Condition

Default Value: 00h
Attribute: Read / Write

The register defines the conditions which cause timer to reload. If a selected request occurs before the timer times out, it will be reload.

Bits 15:10 **Reserved**

Bit 9 **Video Access Request (Memory Segment A0000h ~ BFFFFh , I/O Port 3B0h ~ 3B7h, 3C0h ~ 3CFh, 3D0h ~ 3DFh)**
0 = Disable
1 = Enable

Bit 8 **IRQ Request**
0 = Disable
1 = Enable

Bit 7 **DMA Request**
0 = Disable
1 = Enable

Bit 6 **Keyboard and Mouse Event Request (IRQ 1, 3, 4 and 12)**
0 = Disable
1 = Enable

Bit 5 **Hard Disk / Floppy Disk Access**
0 = Disable
1 = Enable

Bit 4 **Parallel Port Access (Port 278h-27Fh, 378h-37Fh and 3BCh-3BFh)**
0 = Disable
1 = Enable

Bit 3 **Serial Port Access (Port 2F8h-2FFh, 3F8h-3FFh,2E8h-2EFh and 3E8h-3EFh)**
0 = Disable
1 = Enable

Bit 2 **I/O Address Trap 1 Access**
0 = Disable
1 = Enable

Bit 1 **I/O Address Trap 0 Access**
0 = Disable
1 = Enable

Bit 0 **PCI Master Request**
0 = Disable
1 = Enable

Register 94h ~ 95h Generic Timer Reload Condition

Default Value: 00h
Attribute: Read / Write

The register defines the conditions which cause timer to reload. If a selected request occurs before the timer times out, it will be reload.

Bits 15:10 **Reserved**

Bit 9 **Video Access Request (Memory Segment A0000h ~ BFFFFh , I/O Port 3B0h ~ 3B7h, 3C0h ~ 3CFh, 3D0h ~ 3DFh)**
0 = Disable
1 = Enable

Bit 8 **IRQ Request**
0 = Disable
1 = Enable

Bit 7 **DMA Request**
0 = Disable
1 = Enable

Bit 6 **Keyboard and Mouse Event Request (IRQ 1, 3, 4 and 12)**
0 = Disable
1 = Enable

Bit 5 **Hard Disk / Floppy Disk Access**
0 = Disable
1 = Enable

Bit 4 **Parallel Port Access (Port 278h-27Fh, 378h-37Fh and 3BCh-3BFh)**
0 = Disable
1 = Enable

Bit 3 **Serial Port Access (Port 2F8h-2FFh, 3F8h-3FFh,2E8h-2EFh and 3E8h-3EFh)**
0 = Disable
41 = Enable

Bit 2 **I/O Address Trap 1 Access**
0 = Disable
1 = Enable

Bit 1 **I/O Address Trap 0 Access**
0 = Disable
1 = Enable

Bit 0 **PCI Master Request**
0 = Disable
1 = Enable

Register 96h ~ 97h Slow Timer Reload Condition

Default Value: 0000h
Attribute: Read / Write

The register defines the conditions which cause timer to reload. If a selected request occurs before the timer times out, it will be reload.

Bits 15:14 **Slow Timer Time Base**
00 = 70 us
01 = 1.2 s
10 = 4.8 s
11 = 9.7 s

Bits 13:10 **Reserved**

Bit 9 **Video Access Request (Memory Segment A0000h ~ BFFFFh , I/O Port 3B0h ~ 3B7h, 3C0h ~ 3CFh, 3D0h ~ 3DFh)**
0 = Disable
1 = Enable

Bit 8 **IRQ Request**
0 = Disable
1 = Enable

Bit 7 **DMA Request**
0 = Disable
1 = Enable

Bit 6 **Keyboard and Mouse Event Request (IRQ 1, 3, 4 and 12)**
0 = Disable
1 = Enable

Bit 5 **Hard Disk / Floppy Disk Access**
0 = Disable
1 = Enable

Bit 4 **Parallel Port Access (Port 278h-27Fh, 378h-37Fh and 3BCh-3BFh)**
0 = Disable
1 = Enable

Bit 3 **Serial Port Access (Port 2F8h-2FFh, 3F8h-3FFh,2E8h-2EFh and 3E8h-3EFh)**
0 = Disable
1 = Enable

Bit 2 **I/O Address Trap 1 Access**
0 = Disable
1 = Enable

Bit 1 **I/O Address Trap 0 Access**
0 = Disable
1 = Enable

Bit 0 **PCI Master Request**
0 = Disable
1 = Enable

Register 98h ~ 99h Fast Timer Reload IRQ Selection

Default Value: 0000h
Attribute: Read / Write

The register defines the IRQs which cause timer to reload.

Bit 15 **IRQ 15**
Bit 14 **IRQ 14**
Bit 13 **IRQ 13**
Bit 12 **IRQ 12**
Bit 11 **IRQ 11**
Bit 10 **IRQ 10**
Bit 9 **IRQ 9**
Bit 8 **IRQ 8**
Bit 7 **IRQ 7**
Bit 6 **IRQ 6**
Bit 5 **IRQ 5**
Bit 4 **IRQ 4**
Bit 3 **IRQ 3**
Bit 2 **IRQ 2**
Bit 1 **IRQ 1**
Bit 0 **IRQ 0**
0 = Disable
1 = Enable

Register 9Ah ~ 9Bh Generic Timer Reload IRQ Selection

Default Value: 0000h
Attribute: Read / Write

The register defines the IRQs which cause timer to reload.

Bit 15 IRQ 15
Bit 14 IRQ 14
Bit 13 IRQ 13
Bit 12 IRQ 12
Bit 11 IRQ 11
Bit 10 IRQ 10
Bit 9 IRQ 9
Bit 8 IRQ 8
Bit 7 IRQ 7
Bit 6 IRQ 6
Bit 5 IRQ 5
Bit 4 IRQ 4
Bit 3 IRQ 3
Bit 2 IRQ 2
Bit 1 IRQ 1
Bit 0 IRQ 0
0 = Disable
1 = Enable

Register 9Ch ~ 9Dh Slow Timer Reload IRQ Selection

Default Value: 0000h
Attribute: Read / Write

The register defines the IRQs which cause timer to reload.

Bit 15 IRQ 15
Bit 14 IRQ 14
Bit 13 IRQ 13
Bit 12 IRQ 12
Bit 11 IRQ 11
Bit 10 IRQ 10
Bit 9 IRQ 9
Bit 8 IRQ 8
Bit 7 IRQ 7
Bit 6 IRQ 6
Bit 5 IRQ 5

Bit 4 **IRQ 4**

Bit 3 **IRQ 3**

Bit 2 **IRQ 2**

Bit 1 **IRQ 1**

Bit 0 **IRQ 0**

0 = Disable

1 = Enable

Register 9Eh **Soft-SMI Generation / RMSMIBLK Trigger**

Default Value: 00h

Attribute: Write Only

Bit [1] control the state of Resume_from_SMI_Blocking (RMSMIBLK) signal. SiS 85C496/497 does not allow nested SMI. Therefore, a SMI Mask is automatically enabled when SMI# is asserted. The SMI mask and must be removed by enabling the RMSMIBLK timer before RSM instruction. Note that RMSMIBLK is used only inside SMI subroutine. Bit [0] triggers SMI# immediately when it is written with 1.

Bits 7:2 **Reserved**

Bit 1 **RMSMIBLK Timer start**

0 = No effect

1 = Start Remove-SMI-Blocking Timer counting down

(Detail of this timer is described in Reg 8Dh)

Bit 0 **SMI# is asserted immediately**

0 = No effect

1 = Assert the SMI

Register A0h ~ A1h **SMI Request Status**

Default Value: 0000h

Attribute: Read / Write Clear

The register shows the requests which trigger the current SMI. Write to the bits with 1s will clear the corresponding requests.

Bit 15 **Deturbo Switch Request**

Read: 0: No Request

1: Request

Write: 0: No effect

1: Clear

Bit 14 **Video Access Request (Memory Segment A0000h ~ BFFFFh , I/O Port 3B0h ~ 3B7h, 3C0h ~ 3CFh, 3D0h ~ 3DFh)**

Read: 0 = No Request

1 = Request

Write: 0 = No Effect

1 = Clear

Bit 13 **IRQ Request**
Read: 0 = No Request
1 = Request
Write: 0 = No Effect
1 = Clear

Bit 12 **Fast Timer Expire Request**
Read: 0 = No Request
1 = Request
Write: 0 = No Effect
1 = Clear

Bit 11 **Generic Timer Expire Request**
Read: 0 = No Request
1 = Request
Write: 0 = No Effect
1 = Clear

Bit 10 **Slow Timer Expire Request**
Read: 0 = No Request
1 = Request
Write: 0 = No Effect
1 = Clear

Bit 9 **Serial Port Request**
Read: 0 = No Request
1 = Request
Write: 0 = No Effect
1 = Clear

Bit 8 **Parallel Port Request**
Read: 0 = No Request
1 = Request
Write: 0 = No Effect
1 = Clear

Bit 7 **Hard Disk/Floppy Disk Request**
Read: 0 = No Request
1 = Request
Write: 0 = No Effect
1 = Clear

Bit 6 **Keyboard/Mouse Request**
Read: 0 = No Request
1 = Request
Write: 0 = No Effect
1 = Clear

Bit 5 **Break Switch Request**
Read: 0 = No Request
1 = Request
Write: 0 = No Effect
1 = Clear

Bit 4 **Software SMI Request**
Read: 0 = No Request
1 = Request
Write: 0 = No Effect
1 = Clear

Bit 3 **Programmable I/O Trap_1 Request**
Read: 0 = No Request
1 = Request
Write: 0 = No Effect
1 = Clear

Bit 2 **Programmable I/O Trap_0 Request**
Read: 0 = No Request
1 = Request
Write: 0 = No Effect
1 = Clear

Bit 1 **PCI Master Request**
Read: 0 = No Request
1 = Request
Write: 0 = No Effect
1 = Clear

Bit 0 **DMA Request**
Read: 0 = No Request
1 = Request
Write: 0 = No Effect
1 = Clear

Register A2h ~ A3h SMI Request Status Selection

Default Value: 0000h
Attribute: Read / Write

The register selects the events which will be moniterd and assert SMI# when present.

Bit 15 **Reserved**

Bit 14 **Video Access Request (Memory Segment A0000h ~ BFFFFh , I/O Port 3B0h ~ 3B7h, 3C0h ~ 3CFh, 3D0h ~ 3DFh)**
0 : Ignored
1 : Selected

Bit 13 **IRQ Request**
0 : Ignored
1 : Selected

Bit 12 **Fast Timer Expire Request**
0 : Ignored
1 : Selected

Bit 11 **Generic Timer Expire Request**
0 : Ignored
1 : Selected

Bit 10 **Slow Timer Expire Request**
0 : Ignored
1 : Selected

Bit 9 **Serial Port Request**
0 : Ignored
1 : Selected

Bit 8 **Parallel Port Request**
0 : Ignored
1 : Selected

Bit 7 **Hard Disk/Floppy Disk Request**
0 : Ignored
1 : Selected

Bit 6 **Keyboard/Mouse Request**
0 : Ignored
1 : Selected

Bit 5 **Break Switch Request**
0 : Ignored
1 : Selected

Bit 4 **Software SMI Request**
0 : Ignored
1 : Selected

Bit 3 **Programmable I/O Trap_1 Request**
0 : Ignored
1 : Selected

Bit 2 **Programmable I/O Trap_0 Request**
0 : Ignored
1 : Selected

Bit 1 **PCI Master Request**
0 : Ignored
1 : Selected

Bit 0 **DMA Request**
0 : Ignored

1 : Selected

Register A4h ~ A5h SMI Request IRQ Selection

Default Value: 0000h
Attribute: Read / Write Clear

Bit 15 IRQ 15
Bit 14 IRQ 14
Bit 13 IRQ 13
Bit 12 IRQ 12
Bit 11 IRQ 11
Bit 10 IRQ 10
Bit 9 IRQ 9
Bit 8 IRQ 8
Bit 7 IRQ 7
Bit 6 IRQ 6
Bit 5 IRQ 5
Bit 4 IRQ 4
Bit 3 IRQ 3
Bit 2 IRQ 2
Bit 1 IRQ 1
Bit 0 IRQ 0
0 = Disable
1 = Enable

Register A6h ~ A7h Clock Throttling On Timer Reload IRQ Selection

Default Value: 0000h
Attribute: Read / Write Clear

The register defines the IRQs which cause timer to reload.

Bit 15 IRQ 15
Bit 14 IRQ 14
Bit 13 IRQ 13
Bit 12 IRQ 12
Bit 11 IRQ 11
Bit 10 IRQ 10
Bit 9 IRQ 9
Bit 8 IRQ 8
Bit 7 IRQ 7
Bit 6 IRQ 6

Bit 5 **IRQ 5**
Bit 4 **IRQ 4**
Bit 3 **IRQ 3**
Bit 2 **IRQ 2**
Bit 1 **IRQ 1**
Bit 0 **IRQ 0**
0 = Disable
1 = Enable

Register A8h **GPIO Control**

Default Value: 00h
Attribute: Read / Write

Bit [7:6] Controls the GPIO function. When these bits are enabled, Reg D4 bit 3 is override to 0. The input source for GPIOs can be either a stable level signal or a bouncing signal with arbitrary active level. These attributes are defined in bit [5:2]

Bit 7 **Enable GPIO 0**
0 = Disable
1 = Enable
Bit 6 **Enable GPIO 1**
0 = Disable
1 = Enable
Bit 5 **Set Active Level of GPIO 0**
0 = Active Low
1 = Active High
Bit 4 **Set Active Level of GPIO 1**
0 = Active Low
1 = Active High
Bit 3 **De-bounce Control of GPIO 0**
0 = Disable de-bounce circuitry
1 = Enable de-bounce circuitry
Bit 2 **De-bounce Control of GPIO 1**
0 = Disable de-bounce circuitry
1 = Enable de-bounce circuitry
Bit 1 **GPIO 0 Function Control**
0 = De-assert STPCLK#
1 = Triggert SMI
Bit 0 **GPIO 1 Function Control**
0 = De-assert STPCLK#
1 = Triggert SMI

Register A9h GPIO SMI Request Status

Default Value: 00h
Attribute: Read / Write Clear

Bits 7:2 Reserved

Bit 1 GPIO 0 Switch Request

Read: 0: No Request
1: Request
Write: 0: No effect
1: Clear

Bit 0 GPIO 1 Request

Read: 0: No Request
1: Request
Write: 0: No effect
1: Clear

Register AAh GPIO DeBounce Count

Default Value: 00h
Attribute: Read / Write

Bits 7:4 De Bounce Counts for GPIO 0. The minimum value is 2

The timer-expire interval is translated by the follow equation:
Timer-Expire Interval = (Timer Counter - 1) × 0.15s
The interval is allowed between 0.15s and 2.25s.

Bits 3:0 De Bounce Counts for GPIO 1. The minimum value is 2

The timer-expire interval is translated by the follow equation:
Timer-Expire Interval = (Timer Counter - 1) × 0.15s
The interval is allowed between 0.15s and 2.25s

Register C0h PCI INTA# -to-IRQ Link

Default Value: 00h
Attribute: Read / Write

The link register sets up the connection between PCI interrupt line, INTA#, and interrupt request pin, IRQ.

Bit 7 Link Enable

0 = Disable
1 = Enable

Bits 6:4 Reserved

Bits 3:0 Link Selection

0000 : Reserved
0001 : Reserved
0010 : Reserved
0011 : IRQ3
0100 : IRQ4
0101 : IRQ5
0110 : IRQ6
0111 : IRQ7
1000 : Reserved
1001 : IRQ9
1010 : IRQ10
1011 : IRQ11
1100 : IRQ12
1101 : Reserved
1110 : IRQ14
1111 : IRQ15

Register C1h PCI INTB# -to-IRQ Link

Default Value: 00h
Attribute: Read / Write

The link register sets up the connection between PCI interrupt line, INTB#, and interrupt request pin, IRQ.

Bit 7 Link Enable
0 = Disable
1 = Enable

Bits 6:4 Reserved

Bits 3:0 Link Selection
0000 : Reserved
0001 : Reserved
0010 : Reserved
0011 : IRQ3
0100 : IRQ4
0101 : IRQ5
0110 : IRQ6
0111 : IRQ7
1000 : Reserved
1001 : IRQ9
1010 : IRQ10
1011 : IRQ11
1100 : IRQ12
1101 : Reserved
1110 : IRQ14
1111 : IRQ15

Register C2h PCI INTC# -to-IRQ Link

Default Value: 00h
Attribute: Read / Write

The link register sets up the connection between PCI interrupt line, INTC#, and interrupt request pin, IRQ.

Bit 7 Link Enable
0 = Disable
1 = Enable

Bits 6:4 Reserved

Bits 3:0 Link Selection
0000 : Reserved
0001 : Reserved
0010 : Reserved
0011 : IRQ3
0100 : IRQ4
0101 : IRQ5
0110 : IRQ6
0111 : IRQ7
1000 : Reserved
1001 : IRQ9
1010 : IRQ10
1011 : IRQ11
1100 : IRQ12
1101 : Reserved
1110 : IRQ14
1111 : IRQ15

Register C3h PCI INTD# -to-IRQ Link

Default Value: 00h
Attribute: Read / Write

The link register sets up the connection between PCI interrupt line, INTD#, and interrupt request pin, IRQ.

Bit 7 Link Enable
0 = Disable
1 = Enable

Bits 6:4 Reserved

Bits 3:0 Link Selection
0000 : Reserved
0001 : Reserved
0010 : Reserved
0011 : IRQ3
0100 : IRQ4

0101 : IRQ5
0110 : IRQ6
0111 : IRQ7
1000 : Reserved
1001 : IRQ9
1010 : IRQ10
1011 : IRQ11
1100 : IRQ12
1101 : Reserved
1110 : IRQ14
1111 : IRQ15

RegisterC4h ~ C5h ISA Interrupt Active Level Configuration

Default Value: 0000h
Attribute: Read / Write

The register defines the active level of IRQ channels.

Bit 15 IRQ 15
Bit 14 IRQ 14
Bit 13 No effect
Bit 12 IRQ 12
Bit 11 IRQ 11
Bit 10 IRQ 10
Bit 9 IRQ 9
Bit 8 No effect
Bit 7 IRQ 7
Bit 6 IRQ 6
Bit 5 IRQ 5
Bit 4 IRQ 4
Bit 3 IRQ 3
Bit 2 No effect
Bit 1 No effect
Bit 0 No effect

0 = High Active
1 = Low Active

Register C6h 85C497 Post / INIT Configuration

Default Value: 00h
Attribute: Read / Write

Bit [3:2] controls the INIT routing to CPU and PCI Reset.

Bit [1] selects the interrupts of interrupt controller as PC compatible or programmable.

Bit [0] controls the PCI to ISA post buffer.

Bits 7:4 **Reserved**

Bit 3 **INIT to CPU Reset Enable**

0 = Enable, INIT is directed to CPU Reset

1 = Disable

Bit 2 **INIT to PCI Reset Enable**

0 = Disable

1 = Enable, INIT is directed to PCI Reset

Bit 1 **INTC Compability Select**

0 = ISA compatible, all interrupt channel in each 8259 are set to the same attributes.

1 = PCI compatible, each indivial channel can be set to different attributes.

Bit 0 **PCI to ISA Posting Enable**

0 = Disable

1 = Enable

Register C7h **Deturbo Switch Status**

Default Value: 00h
Attribute: Read

Bits 7:1 **Reserved**

Bit 0 **Deturbo Switch Status**

0 = De-asserted

1 = Asserted

Register C8h ~ CBh **Mail Box**

Default Value: 00000000h
Attribute: Read / Write

The registers are generic Flip-flops. They can be used to store information without any side effects. For example, APM information can be easily passed to SMI handler through these registers.

Bits 31:0 **Free storage.**

Read / Write to these register will have no effect on SiS85C496 / 497 , CPU and external circuit. These registers are provided for BIOS programmer as a temporary storge space which can be access anytime they feel like to do so.

Register D0h **ISA BIOS Configuration**

Default Value: 78h
Attribute: Read / Write

The register specifies the size and type of system BIOS.
Bit 1 controls the auto-detect function of Reg D1. The bit should be set to 1 after BIOS POST.
Bit 0 provides write protection of Reg D1.

Bit 7 **Memory Space FFFDFFFFh ~ FFFA0000h as ROM Area**
0 = Disable
1 = Enable

Bit 6 **Memory Space 0EFFFFh ~ 0E0000h and FFFEFFFFh ~ FFFE0000h as ROM Area**
0 = Disable
1 = Enable

Bit 5 **Memory Space 0FFFFFh ~ 0F0000h and FFFFFFFFh ~ FFFF0000h as ROM Area**
0 = Disable
1 = Enable

Bit 4 **Flash EPROM One-shot Write Enable**
0 = Disable
1 = Enable
This bit is enabled after power-on. After the bit is disabled, it can not be enabled until next power-on.

Bit 3 **Flash EPROM Write Enable**
0 = Write protect
1 = Write enable
When bit 4 is enabled, the Flash EPROM write attribute is controlled by this bit.

Bit 2 **Reserved**

Bit 1 **Register D1 Auto-detect Blocking Control**
0 = Enable auto-detect
1 = Disable auto-detect

Bit 0 **Register D1 Write Enable**
0 = Disable
1 = Enable

Register D1h ISA Address Decoder

Default Value: FFh
Attribute: Read Only or Read / Write (depends on Reg D0h bit0 = 1)

The register is a read-only ISA address decoding status. When an ISA Master accesses an enabled area, the access is translated by ISA decoder into a PCI cycle according to the register status. The value is set by 85C497.

When register D0 bit 0 is enabled, this register can be manually porgrammed.

Bit 7 **F0000h ~ FFFFh Memory Enable**

Bit 6 **E0000h ~ EFFFFh Memory Enable**
Bit 5 **D8000h ~ DFFFFh Memory Enable**
Bit 4 **D0000h ~ D7FFFh Memory Enable**
Bit 3 **C8000h ~ CFFFFh Memory Enable**
Bit 2 **C0000h ~ C7FFFh Memory Enable**
Bit 1 **A0000h ~ BFFFFh Memory Enable**
Bit 0 **00000h ~ 9FFFFh Memory Enable**
0 : Disable
1 : Enable

Register D2h ~ D3h **Exclusive Area 2 Base Address**

Default Value: 0000h
Attribute: Read / Write

The register setting must be identical with Reg 54h ~ 55h.

Bit 15 **Exclusive Area 2 select**
0 = Non-cacheable Area
1 = ISA Memory Hole

Bits 14:12 **Area Size**
000 = 0 KByte (Exclusive Area 1 Disabled)
001 = 64 KByte
010 = 128 KByte
011 = 256 KByte
100 = 512 KByte
101 = 1 MByte
110 = 2 MByte
111 = 4 MByte

Bits 11:8 **Reserved**
Bits 7:0 **Exclusive Area 2 Base Address A[23:16]**

Register D4h **Miscellaneous Configuration**

Default Value: 00h
Attribute: Read / Write

Bit [3] and bit [5] are both multiplexing control for pin 41, 42. If Reg A8 bit [7:6] are enabled, pin 41 and 42 are forced to be GPIO and the Reg D4h bit 3 is override. If bit [3] is set to 1, bit [5] selects the functions to be either external CMOS page select or SMOUT. When bit [5] is set to 1, Bit [4:0] of CPU direct I/O port C00h are used to select 32 pages of external DS 1255 CMOS RAM and port 8XX are mapped to bytes of these pages as EISA convention.

Bit 7 **Reserved**
Bit 6 **Extended CMOS RAM Page Enable**

0 = Access 114 bytes general purpose RAM
1 = Access to extended 128 bytes general purpose RAM

Bits 5,3 **Pin 41, 42 Secondary Multiplexing Select**
X0 = Pin 41 is GPIO0, Pin 42 is GPIO1
01 = Pin 41 is SMOUT2, Pin 42 is SMOUT3
11 = Pin 41 is CMEMLE# (I/O writing to Port C00h)
Pin 42 is CMEMCS# (access I/O Port 800h-8FFh)

Bit 4 **Reserved**

Bit 2 **SA[2:0] and SBHE# Output Buffer Driving Capability Control**
0 = 24mA
1 = 12mA

Bit 1 **IOR# and IOW# Output Buffer Driving Capability Control**
0 = 24mA
1 = 12mA

Bit 0 **Reserved**

4. 85C497 I/O Control Ports

ATM (SiS85C497) contains a set of configuration registers which are programmed through 85C497 I/O Index register (22h) and Data register (23h). The port 23h is a read / writable register. The port 22h is write-only and mirrored on register PCI configuration register 82h.

5. 85C497 I/O Configuration Registers

Two sets of registers are presented in this section: index 01 is a set of configuration registers defined for the built-in 206 Macrocell and the rest registers (index 70 to 75) are configuration registers for ISA timing, system management mode output (SMOUT), and a timer.

Register 70h **ISA Bus Clock Selection**

Default Value: 00h
Attribute: Read / Write

The register selects the ISA bus clock frequency as 7.159 MHz, 1/4, or 1/3 PCI clock.

Bits 7:6 **Bus Clock Frequency Selection**
00: BUSCLK = 7.159 MHz
01: BUSCLK = 1/4 PCICLK
10: BUSCLK = 1/3 PCICLK
11: Reserved

Bits 5:0 **Reserved**

Register 71h **ISA Bus Timing Control**

Default Value: 01h
Attribute: Read / Write

The register defines the ISA bus I/O cycle command recovery time and I/O and memory wait state.

Bits 7:6 **16-Bit I / O Cycle Command Recovery Time Selection**
00: 5 BUSCLK
01: 4 BUSCLK
10: 3 BUSCLK
11: 2 BUSCLK

Bits 5:4 **8-Bit I/O Cycle Command Recovery Time Selection**
00: 8 BUSCLK
01: 5 BUSCLK
10: 4 BUSCLK
11: 3 BUSCLK

Bit 3 **Reserved**

Bit 2 **16-Bit Memory, I/O Wait State Selection**
0: 2 wait states
1: 1 wait states

Bit 1 **8-Bit Memory, I/O Wait State Selection**
0: 5 wait states
1: 4 wait states

Bit 0 **Reserved**

Register 72h SMOUT

Default Value: FFh
Attribute: Read / Write

SMOUT pins are output pins which could be used as peripheral control outputs. 85C496/497 provides either 2 or 8 system management outputs depending on Reg 75 bit [3] setting and extra application circuitry. Writing to the register changes the value of corresponding SMOUT pins.

Bits 7:0 **SMOUT [7:0]**

Register 73h ~ 74h BIOS Timer

Default Value: 0000h
Attribute: Read / Write

The BIOS timer is a 8-bit width and each count translates to a 5 μs delay. When the access bit (Reg 73 Bit 0) is enabled, the timer starts counting down immediately after a write to the I/O port defined in Reg 73 Bit 15:2. The timer keeps counting until the count reaches zero and freezes till the next write to the port. A read to the port returns the current timer counts.

Bits 15:2 **BIOS Timer Base Address A[15:2]**

Bit 1 **Reserved**

Bit 0 **BIOS Timer Access Enable**
0 = Disable

1 = Enable

Register 75h DMA / Deturbo Control

Default Value: 00h
Attribute: Read / Write

Bit [7] control DMA addressing mode

Bit [6:4] configures deturbo mode.

Bit [3] is a output pin multiplexing control. If the bit is set to 0, SMOUT[1:0] are available. If the bit is set to 1 and proper application circuit is applied, SMOUT[7:0] are available.

Bit 7 DMA 32-Bit Address Mode
0 = Disable
1 = Enable

Bit 6 Deturbo Mode Enable
Enabled deturbo mode allows software and hardware deturbo switch to assert / desassert deturbo.
0 = Disable
1 = Enable

Bit 5 Software Deturbo Switch On / Off
0 = Off
1 = On

Bit 4 Deturbo Hold Time
0 = hold 4us every 12us
1 = hold 8us every 12us

Bit 3 SMOUT0 / SMOUTW0# Multiplexing Select
0 = Pin 76 is SMOUT0
1 = Pin 76 is SMOUTW0#

Bit 2 SMOUT1 / SMOUTW1# Multiplexing Select
0 = Pin 77 is SMOUT1
1 = Pin 77 is SMOUTW1#

Bits 1:0 Reserved

Register 76h SMOUT

Default Value: FFh
Attribute: Read / Write

SMOUT pins are output pins which could be used as peripheral control outputs. 85C496/497 provides either 2 or 8 system management outputs depending on Reg 75 bit 2 setting and extra application circuitry. Writing to the register changes the value of corresponding SMOUT pins.

Bits 7:0 SMOUT[15:8]

Register 01h Built-in 206 Timing Control

Default Value: C0h
Attribute: Read / Write

These registers control the DMA clock rate, CPU read or write cycle length, DMA command width of the built-in 206 macrocell.

Bits 7:6 CPU to 206 R/W Cycle Wait State
00 = 1 wait state
01 = 2 wait states
10 = 3 wait states
11 = 4 wait states

Bits 5:4 16-bit DMA Wait State
00 = 1 wait state
01 = 2 wait states
10 = 3 wait states
11 = 4 wait states

Bits 3:2 8-bit DMA Wait State
00 = 1 wait state
01 = 2 wait states
10 = 3 wait states
11 = 4 wait states

Bit 1 DMAMEMR# Assertion Timing
0 = DMAMEMR# is delayed by one clock than XIOR#
1 = DMAMEMR# and XIOR# are asserted at the same clock

Bit 0 DMA Clock
0 = DMA Clock is equal to ISA Clock / 2
1 = DMA Clock is equal to ISA Clock

Register 4D0h~4D1h Interrupt Edge / Level Control

Default Value: 0000h
Attribute: Read / Write

The register specifies the interrupts to be triggered by either the signal edge or the logic level. A 0 bit indicates an edge sensitive interrupt, and a 1 is for level sensitive.

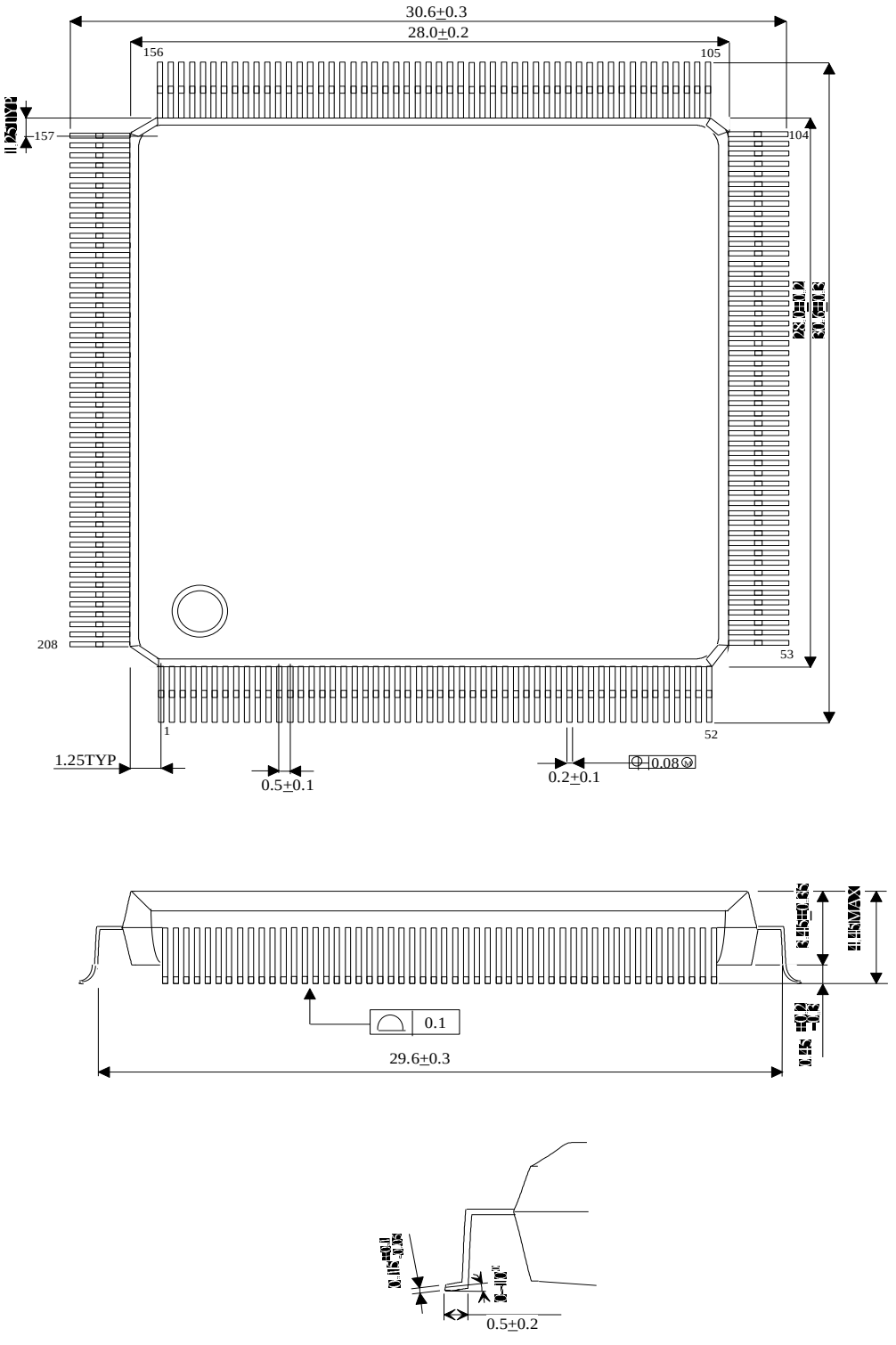
Bit 15 IRQ 15
Bit 14 IRQ 14
Bit 13 Reserved
Bit 12 IRQ 12
Bit 11 IRQ 11
Bit 10 IRQ 10
Bit 9 IRQ 9
Bit 8 Reserved

Bit 7	**IRQ 7**
Bit 6	**IRQ 6**
Bit 5	**IRQ 5**
Bit 4	**IRQ 4**
Bit 3	**IRQ 3**
Bit 2	**Reserved**
Bit 1	**Reserved**
Bit 0	**Reserved**
	0 = Edge sensitive
	1 = Level sensitive

6. MECHANICAL DIMENSION

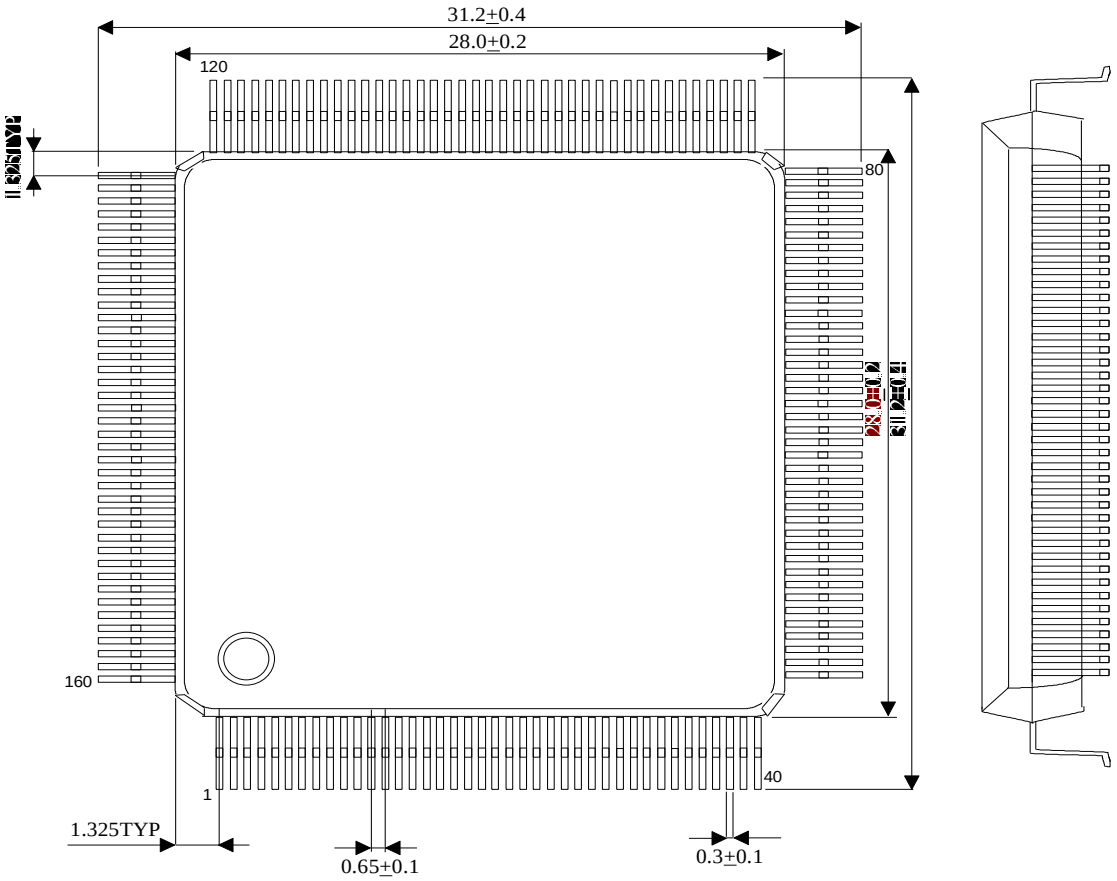
6.1 SiS85C496 (208 pins)

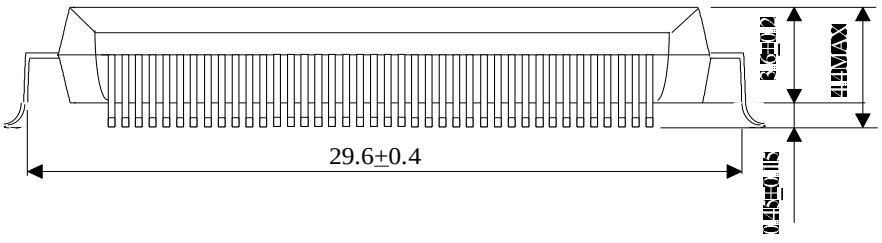
QFP208-P (208-Pin Plastic Flat Package) Unit:mm

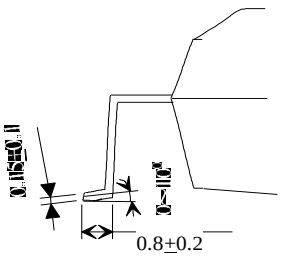


6.2 SiS85C497 (160 pins)

QFP160-P (160-Pin Plastic Flat Package) Unit:mm







COPYRIGHT NOTICE

TRADEMARKS

DISCLAIMER

RESTRICTED RIGHTS LEGEND

Table Of Contents

PART I 1

1. SIS85C496/497 OVERVIEW 1

PART II 3

1. SIS85C496 PCI & CPU MEMORY CONTROLLER 3

1.1 Features 3

1.2 Architectural Overview 4

1.2.1 Host Interface Operations 5
1.2.2 Cache Operations 5
1.2.3 DRAM Memory Operations 6
1.2.4 PCI Bridge Operations 6
1.2.5 FS-Link Operations 7

2. DETAILED FUNCTIONAL DESCRIPTION 7

2.1 CPU/Cache Interface 7

2.1.1 L2 Cache Controller 8
2.1.2 L1/L2 Cache Mode Configurations 16
2.1.3 Snoop Operations 16

2.2 DRAM Interface 18

2.2.1 DRAM Organization 18
2.2.2 DRAM Size Configuration 19
2.2.3 DRAM Address Translation 25

2.3 PCI Interface 25

2.3.1 Functional Overview 26
2.3.2 Host-to-PCI Cycles 27
2.3.3 Host-to-PCI Configuration Cycles 28
2.3.4 PCI Cycle Transaction Termination 28
2.3.5 Exclusive Accesses 30
2.3.6 Parity Support 31
2.3.7 Conditions for Generating SERR# Signal 31
2.3.8 PCI Arbitration 31
2.3.9. CPU Back off support for PCI Bus PCI-to-PCI Bridge 33

2.4 FS-Link Interface 34

2.4.1 Fast Link Machine 34
2.4.2 Slow-Link Machine 38

2.5 IDE Controller 39

2.6 VESA Device Interface 40

2.7 Multiplexed pins 41

2.8 Clocks **41**
2.9 Reset Logic **41**
3. PIN ASSIGNMENT AND SIGNAL DESCRIPTION **43**
3.1 Pin Assignment **45**
3.2 Signal Description **47**
3.2.1 Host CPU Interface 47
3.2.2 497 Interface (FS Link, Refresh, Deturbo, PMU) 50
3.2.3 PCI Interface 51
3.2.4 IDE Interface 53
3.2.5 Cache Interface 54
3.2.6 DRAM Interface 55
3.2.7 Clock 56
3.2.8 Reset 56
3.2.9 Power Pins 56
4. ELECTRICAL CHARACTERISTICS **57**
4.1 Maximum Ratings **57**
4.2 SiS85C496 D.C. Characteristics **57**
4.3 A.C. Characteristics **57**
4.4 Measurement and Test Conditions **62**
PART III **63**
1. SIS85C497 AT BUS CONTROLLER &MEGACELL (ATM) **63**
1.1 Features **63**
1.2 Architectural Overview **64**
2. DETAILED FUNCTIONAL DESCRIPTION **67**
2.1 FS-Link Interface **67**
2.1.1 Fast Link Machine 67
2.1.2 Slow-Link Machine 70
2.2 ISA Interface **72**
2.2.1 ISA Interface 72
2.2.2 ISA Clock Generation 72
2.2.3 Real Time Clock Selection 73
2.3 DMA Controller **73**
2.4 Address Decoding **73**
2.4.1 85C497 I/O Addresses 73
2.4.2 BIOS Memory Space 75
2.4.3 Positive Decode to PCI 77
2.5 Data Buffering **77**

2.6 85C497 Timers 78
2.6.1 Interval Timers 78
2.6.2 BIOS Timer 78
2.7 Interrupt Controller 78
2.8 Power Management Unit 78
2.8.1 PMU Signal Description 81
2.8.2 Functional Description 81
2.8.3 System Event Timers 81
2.8.4 SMI Block Timer 83
2.8.5 Clock Throttling Timer 83
2.8.6 APM Ports 84
2.8.7 SMI Generation Logic 84
2.8.8 STPCLK Control Logic 86
2.8.9 DEVDET Logic 86
2.8.10 PMU Configuration Registers 89
2.9 Reset and Clock 89

3. PIN ASSIGNMENT AND SIGNAL DESCRIPTION 91
3.1 Pin Assignment 93
3.2 496 Interface (FS Link, Refresh, Deturbo, PMU) 95
3.3 AT BUS Interface 96
3.4 PMU Interface 98
3.5 Clock 99
3.6 Reset 99
3.7 XD Bus Interface 100
3.8 Deturbo 100
3.9 NPU Interface 100
3.10 NMI Interface 101
3.11 DMA Interface 101
3.12 CTC Interface 101
3.13 INTC Interface 102
3.14 Power Pins 102

4. ELECTRICAL CHARACTERISTICS 103
4.1 Maximum Ratings 103
4.2 SiS85C497 D.C. Characteristics 103
4.3 A.C. Characteristics 104
4.4 Measurement and Test Conditions 111

PART IV 112

1. REGISTER DESCRIPTION 112

2. PCI CONFIGURATION CONTROL PORTS 113

2.1 Configuration Address Port (0CF8h) 113

2.2 Configuration Data Ports (0CFCh) 114

3. PCI CONFIGURATION SPACE REGISTERS (00H ~ FFH) 114

3.1 PCI Configuration Header Registers (00h ~ 3Fh) 114

3.2 85C496 Specific Registers (40h ~ 7Fh) 117

3.3 85C497 Specific Registers (80h ~ FFh) 134

4. 85C497 I/O CONTROL PORTS 160

5. 85C497 I/O CONFIGURATION REGISTERS 160

6. MECHANICAL DIMENSION 166

6.1 SiS85C496 (208 pins) 166

6.2 SiS85C497 (160 pins) 167

COPYRIGHT NOTICE 168