RC900

CAM451 Cache Memory

Hardware reference manual

PN: 99111393

PN: 99200389

RC Computer

CAM451 Cache Memory

Keywords:
CAM451, CPU452, RC900 System Board

Abstract:
This document is the hardware description of the CAM451 Cache Memory for insertion in the 80386 PGA socket on the CPU452 System Board. CAM451 contains 32Kbytes cache memory, a cache controller, the 80386 microprocessor, and busseparation devices.

Date:
October 1988

Author:
Albert Jensen

Table of Contents

1. General Information

The CAM451 is a cache memory and i80386 microprocessor for insertion in the 80386 PGA socket on the CPU452 system board. Its purpose is to reduce memory access time thus increasing overall system speed and to increase the available bandwidth on the CPU452 system board bus.

The CAM451 contains a 8K by 32 bit wide cache memory, a cache controller, the i80386 CPU, and drivers for bus isolation. A block diagram is shown in figure 1.

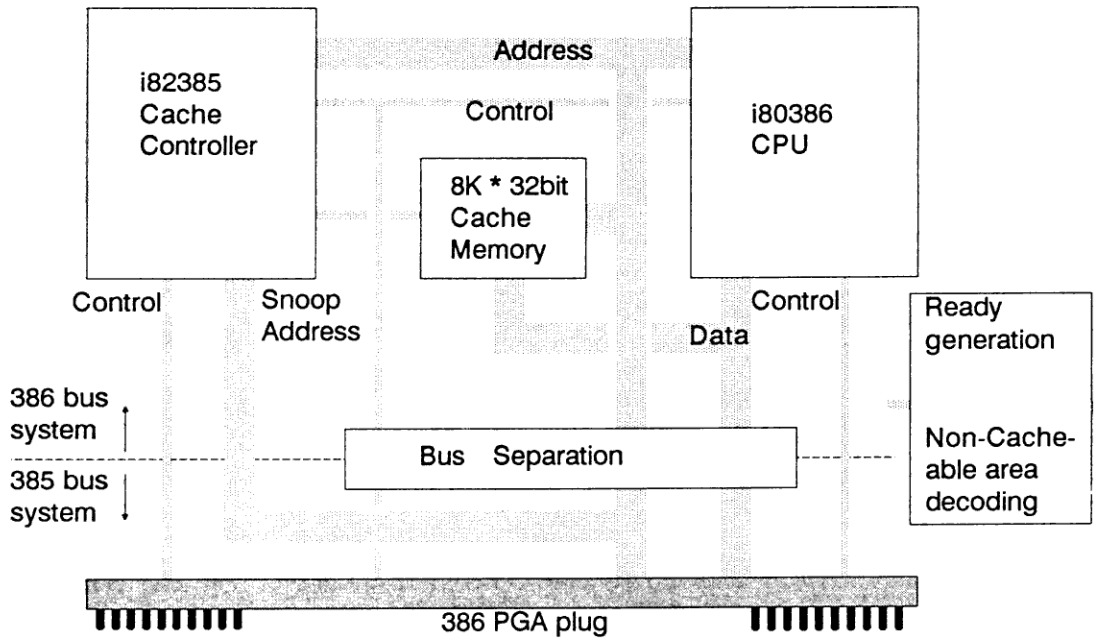


Figure 1. CAM451 Block Diagram

This document describes the CAM451 board in five major sections. The first one gives a short description and specification for the board. The second one contains a functional description of the board elements. The third one contains a description of the testing requirements. The fourth one contains the layout of the CAM451 printed circuit board. The fifth one contains the logic diagrams. Finally an appendix containing data on the i82385 cache controller is provided.

1.1 Short Description

The first purpose of CAM451 is to increase the available bus bandwidth on CPU452. As shown in figure 1 above the CAM451 contains two isolated bus systems comprising both address, data, and control busses. The upper bus system is called the *386 bus system.* It is used by the i80386 CPU and its primary purpose is memory transfers with the cache. With a 20MHz system clock its maximum throughput is 10M Dwords or 40Mbytes per second.

The lower bus system is called the *385 bus system.* It is controlled from the CAM451 by the i82385 cache controller and by DMA or intelligent devices attached to the CPU452 board. This bus system has a maximum throughput similar to the 386 bus system. Assuming a high isolation between the two bus systems, inserting CAM451 thus almost doubles the available bus bandwidth.

The second purpose of CAM451 is to reduce the memory access time by reducing the amount of wait states being inserted into 80386 cycles. When operating from the CAM451 cache memory (*memory read hits*), no wait states are inserted. If the required memory date is unavailable in the cache (*memory read miss*), it has to be fetched from system memory. This will require initially two wait stetes to be inserted, but consecutive memory read miss cycles will alter between none and one wait state being inserted, because the i80386 CPU will be put in pipeline mode.

When memory writes are performed, system memory is accessed, but without passing on system memory wait states to the 80386 CPU. Memory write wait states will thus only be incurred during consecutive write cycles (e.g. REP STOSW) or if a memory read miss is incurred during the next state. Most memory write cycles therefore will proceed without incurring wait states.

As shown in figure 1, CAM451 contains six elements. The 80386 plug is inserted into the i80386 PGA socket on the CPU452 board. All connections to CAM451 will normally flow through this socket. The removed i80386 is instead inserted into the PGA socket on the CAM451. To the CPU452, CAM451 will therefore appear as the original i80386 CPU but with significantly reduced bus bandwidth requirements. Its function is transparent to both software and hardware.

The central element in implementing this transparency is the i82385 cache controller. It monitors the 80386 cycles and decide whether they should be forwarded to the CPU452 via the 385 bus or should be processed locally by accessing the cache memory.

The cache memory is a 35nS 8K by 32 bit static RAM. It is controlled by the 82385 controller and feeds data to the 80386 CPU.

The bus separation drivers and latches connect the 386 and 385 bus systems. Both address, data, and control signals are passed through here. It is controlled by the 82385 cache controller.

Finally some random logic is provided for compatiblity purposes:

1. To disable I/O write posting and to generate the READY signals for the i80386 CPU and the i82385 cache controller.
2. To latch the cache miss signal, whenever data is recorded as not present in the cache. This flag may be read by software and is used primarily for testing purposes.
3. To decode the memory cycles, that are mapped as non-cacheable.
4. To control the address line 20 signal for the i82385 cache controller in accordance with the CPU452 scheme.

1.2 Specifications

1.2.1 Performance

Clock speed
20MHz

Bus bandwidth
The CAM451 provides an isolation between the CPU452 bus system and the local CAM451 386 bus. This almost doubles the bus bandwidth available, because few transfers are necessary between the two bus systems. Each bus system has a maximum capacity of 10M transfers (of one to four bytes) per second.

Wait state insertion
Insertion of wait states is reduced significantly. During memory write cycles usually no wait states are incurred. During memory read cycles the *cache hit rate* is above 90% in which no wait states need to be inserted.

Bus system sizes
32 bit address bus
32 bit data bus
A control bus

Cache RAM
35nS 8K by 32 bit

Generation of performance evaluation signals
Signals are generated for sampling by an external device to determine system performance regarding wait state insertion and cache miss cycles.

1.2.2 Environmental

Operating temperature	: 0 to 40 degrees C^{o}.
Relative humidity	: 20% to 80% (non condensing).
Altitude	: 0 to 10,000 feet.

1.2.3 Physical

Width	: 100 mm
Length	: 150 mm
Heigth	: 18 mm
Weight	: 100 grams

1.2.4 Power requirements

Dissipation	: 12 W max.
Vcc	: +5V +/-5% (2.4A max.) Note that 2 W come from the i80386 CPU. The CPU452 power dissipation is reduced correspondingly.

2. Functional Description

This section describes in detail each of the elements that comprises the CAM451. They are the following:

2.1 i80386 CPU

2.2 i82385 Cache Controller

2.3 Bus Separation

2.4 Random logic

2.5 Cache Memory

2.6 80386 Plug

2.1 i80386 CPU

The i80386 CPU is the primary processing unit of the CPU452/CAM451 combination. It formerly occupied the 80386 PGA plug on the CPU452 board now occupied by the CAM451. For a more detailed description of the i80386 refer to the CPU452 Hardware Reference Manual or the Intel datasheet on this IC.

i80386 signals are passed to the 80386 Plug in three different ways:

1. **Signals connected directly to the plug:**
   - *BusSize16* — Used for selecting 16bit bus size. It is strapped inactive on CPU452.
   - *Reset*
   - *Interrupt*
   - *NonMaskableInterrupt*
   - *CoprocessorRequest (PEREQ)*
   - *CoprocessorBusy (BUSY)*
   - *CoprocessorError (ERROR)*
   - *Clock2* — Basic processor clock.

2. **Signals passing through the bus separator:**
   - *Data 0..31* — Data lines. They are fed to the cache memory also.
   - *Address 2..31* — Address lines. They are monitored by the cache controller also and used for addressing the cache memory.
   - *Write/Read, Data/Code, Memory/IO* — Cycle definition signals. They are monitored by the cache controller and the wait state monitor.

3. **Signals fed to the cache controller and regenerated there:**
   - *ByteEnable0..3* — Enable signals for each of the four byte wide busses comprising the 32bit data bus.
   - *NextAddress* — Signal controlling whether the 80386 should enter pipelined mode or not.
   - *Lock* — Signal controlling whether the cycle is locked or not.
   - *AddressStrobe* — Signal used for latching address and cycle definition signals.

In addition to these two other groups exist. *Hold* and *HoldAcknowledge* are unconnected because the connection to the CPU452 bus is managed by the cache controller. Therefore it is also this controller that generates the bus arbitration signals. Finally the i80386 *Ready* signal is generated by random logic gating 385 bus ready signals with the cache controller ready signal. See section 2.4.

2.2 i82385 Cache Controller

The bus separation and control of the cache memory is performed by the i82385 cache controller designed specifically for use with the i80386 CPU. It allows data residing in the cache to be accessed without involving the CPU452.

The i82385 operates only on memory accesses. I.e. I/O transfers, Halt, and Interrupt Acknowledge cycles are forwarded to the system bus.

Memory reads operate in one of two ways:

1. If the data requested already reside in the cache (Cache Read Hit), it is forwarded to the i80386 without interference with the system bus and without inserting wait states.
2. If the data requested does not reside in the cache (Cache Read Miss), it has to be read from system memory. In this case the address is put on the 385 bus one CLK cycle after the i80386 request. System memory is accessed and returns the data. It also controls the insertion of wait states. The first Cache Read Miss requires at least two wait states to be inserted - one from the cache controller and one from the system memory. During this cycle the cache controller puts the i80386 in pipelined mode to hide subsequent wait states. Consecutive Cache Read Miss cycles therefore will alter between none and and at least one wait state inserted. System memory should be set up to always insert at least one wait state.
   When data arrive on the 386 bus, it is supplied to the i80386 and written in the cache. The cache directory is also updated. Even if the i80386 requests only a byte or a word, the whole DWORD send by system memory is written in the cache.

Memory writes are *posted through*, signifying that data is written to both the cache and to the system memory. This requires a system bus transfer. This transfer however is managed by the i82385 and the i80386 therefore may proceed before this write cycle terminates. Only with consecutive write cycles or a Cache Read Miss immediately after the write cycle, wait states may have to be inserted to slow down the i80386. In most cases however, the write cycle will be performed without the insertion of i80386 wait states even though system memory may require the insertion of several wait states.

Memory writes also proceed in two ways:

1. If the data to be written already reside in the cache (Cache Write Hit), the address is forwarded to the system memory and to the cache and the bytes indicated by the i80386 are written in both memories.
2. If the data to be written does not reside in the cache (Cache Write Miss), the cycle proceeds as indicated above except for that data is not written in the cache.

To maintain cache consistency with the transfers taking place at the 385 bus, the i82385 monitors this bus. Whenever a match is found, i.e. whenever an external device performs an write to system memory data that also reside in the cache, the cache data is invalidated. This means that the next time this data is requested by the i80386, it has to be fetched from system memory. System bus monitoring by the i82385 is performed interleaved with i80386 requests and thus does not slow down i80386 cycles.

The cache controller address monitoring is based on an internal cache directory. This directory has 1024 entries. Each entry consists of a 17 bit tag field (the upper 17 address bits), a tag valid bit, and 8 DWORD valid bits. Comparison works as shown in figure 2 below.

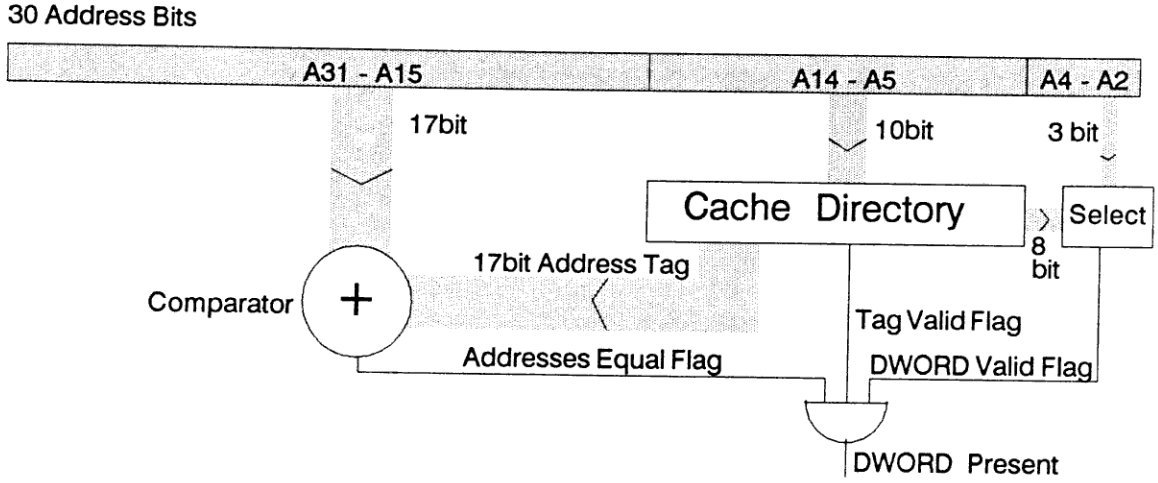


Figure 2. Cache Controller address monitoring.

The cache regards the system memory as 2^{17} *pages* of 8K DWORD's each. Each page is composed of 1024 *sets* and each set of 8 consecutive *DWORD's*. The cache controller allows addressing of up to 1024 different sets and within each set up to 8 consecutive DWORD's. A change of the directory tag will therefore change the validity of not 1 but 8 DWORD's.

The cache controller signals are put into 9 groups:

1. **Interface with the i80386 CPU:**

Clock2	The master clock.
Reset	
ByteEnable0..3	Enable signals for each of the four byte wide busses comprising the 32bit data bus.
Address 2..30	Address lines. They are monitored by the cache controller to determine if the data is cached. A20 is gated for compatibility with the CPU452.
Address 31	In most 386 cache systems, the i80387 FPA would reside on the 386 local bus, thus not requiring any 385 bus accesses, when it is used. With the CAM451 however, the i80387 resides on the system board - i.e. outside the CAM451.This is done for mechanical reasons. This however signifies that i80387 utilization requires 385 bus accesses, because data have to be transferred to it. The cache controller intercepts i80387 data transfers by monitoring A31 during I/O transfers. A31 on i82385 is therefore tied low, so that i80387 data transfers are always led to the 385 bus.

Write/Read, Data/Code, Memory/IO
Cycle definition signals. They are monitored by the cache controller and the wait state monitor.

Lock Signal controlling whether the cycle is locked or not. When the i80386 signals that a cycle is locked, all memory requests are forwarded to the 385 bus - i.e. are handled as if they were Cache Miss cycles.

AddressStrobe Signal used for latching address and cycle definition signals.

NextAddress Signal controlling whether the i80386 should enter pipelined mode or not.
The i80386 CPU contains an instruction pipeline allowing it to prefetch instructions. This facility is used by the cache controller whenever Cache Miss cycles occur. The i80386 is forced into pipelined mode to allow overlapping of memory access cycles. This reduces the number of wait states to be inserted, because the i82385 can decode the next cycle and its address in advance.

ReadyInput, ReadyOutput, BusReadyEnable
These signals are fed to the Ready decoding logic described in section 2.4 below.

2. **Access decoding:**

BusSize16 Used for selecting 16bit bus size. It is strapped inactive on CPU452.

LocalBusAccess This signal is used to indicate that the cycle is to performed locally. It is activated only during I/O access to port 2Ch.

NonCacheableAccess
This signal is used as an indication that the present cycle is non-cacheable - i.e. that it should be forwarded to the 385 bus without cache interference. This signal is activated whenever a memory access is decoded as non-cacheable. (See section 2.4 below).

3. **385 bus signals.** These signals are the processed equivalents of the corresponding i80386 signals:

BusAddressStrobe
Signal used for latching address and cycle definition signals. It is issued only when the cache controller requires 385 bus cycles to be performed.

BusByteEnable0..3
Enable signals for each of the four byte wide busses comprising the 32bit data bus. They are activated when the cache controller requires 385 bus cycles to be performed. In this case they are replicas of the i80386 *ByteEnable0..3* signals exept during Cache Read Miss cycles in which the cache controller requires all four bytes for updating the cache memory.

BusLock Signal controlling whether the cycle is locked or not. When the i80386 signals that a cycle is locked, all memory requests are forwarded to the 385 bus - i.e. are handled as if they were Cache Miss cycles.

BusNextAddress Signal controlling whether the i80386 should enter pipelined mode or not. It is strapped active on the CPU452 to allow entering pipelined mode whenever the cache controller decides so.

BusReady Ready signal from the 385 bus. It is sampled by the cache controller only when it itself did start a 385 bus cycle. It is used by CPU452 devices to insert wait states and to signal completion of the 385 bus cycle.

4. **Bus separation control:**
LoadStrobe, DataOutEnable, BusTransmit/Receive
These signals are used to control the separation of the 386 and 385 data busses. This is described in more detail in section 2.3 below.
BusAddressClockPulse, BusAddressOutputEnable
These signals are used to latch and enable the address and cycle defintion signals coming from the 386 bus. This bus separation is described in more detail in section 2.3 below.

5. **385 address bus monitoring:**
SystemAddress2..31
These lines are connected to the 385 address bus lines to allow monitoring of CPU452 activity. This is done to insure cache coherency as described above. SA31 is tied low just like A31 is.
SnoopAddressStrobe, SnoopAddressEnable
These lines are used to signal, that a valid address is available on the 385 address bus. This is the case when the 385 bus is put into a HOLD cycle (by another bus master) and the *DataWriteEnable* signal from CPU452 indicates that a memory write is in progress.

6. **385 bus arbitration:**
BusHold, BusHoldAcknowledge
These lines have the equivalent function as did the i80386 *Hold* and *HoldAcknowledge* have before. They are used for arbitration on the 385 bus.

7. **Cache control signals:**
CacheAddressLatchEnable, CacheTransmit/Receive, ChipSelect0..3, CacheOutputEnable, CacheWriteEnable
These signals are used for controlling the cache memory. For more details see section 2.5 below.

8. **Status and control:**
Miss This may be understood as an additional cycle definition signal. When active, it indicates that the cache memory access was a miss - i.e. that a system memory access had to be performed. It is latched. For more details see section 2.4 below.

WriteBufferStatus
: This signal indicates that the bus separator contains write data, that has not been written to the system memory. This signal is used for preventing posting of I/O writes.

Flush
: This signal is used to flush the cache directory. I.e. to invalidate all directory entries. It is used for disabling the cache through connection to bit 3 of the NCA register.

9. Cache controller configuration:

2Way/Direct
: This signal is used to select one of two operating cache modes - two way associative or direct mapped cache. It is strapped for direct mode operation.

Master/Slave
: The cache controller may be used in a multiprocessor system in which several cache controllers exist. Hovwever since only one controller is used, it is strapped into the master configuration.

For more details on the operation of the i82385 cache controller see appendix A.

2.3 Bus Separation

The bus separator is used to separate the 386 and the 385 bus systems. It is controlled by the cache controller and consist of a data bus separator, a address bus separator, and a cycle definition signal separator.

Address and cycle definition separation is performed with 74F374 registers, that are controlled by the *BusAddressClockPulse* and the *BusAddressOutputEnable* signals from the cache controller.

The 386 address bus and the cycle definition signals are loaded one CLK after *AddressStrobe* goes active or later if required so by a 385 access cycle already in progress. This operation is performed independently of if the cycle requires a 385 bus transfer. Not until that is determined and the *BusAddressStrobe* goes active are the drivers enabled. This is done with a timing so that this information becomes available at the same time it would from the i80386 CPU.

Data separation is performed with 74F646 bus transceivers, that are controlled by the cache controller.

During write cycles, data is latched with *LoadDataStrobe* to allow write termination without i80386 intervention. The i80386 data is latched at Ø2 of *BusAddressStrobe*. After *BusAddressStrobe* data will therefore be available 1 CLK2 + 26nS + 13nS = 64nS later. Valid delay timing for i80386 is 63nS.

BusTransmit/Receive controls the transceiver direction. The transmit direction is set when *BusAddressStrobe* goes active and is removed 1 CLK2 after *BusReady* is sampled.

DataOutputEnable controls the output. During Cache Read Miss cycles, the drivers are opened 1 CLK2 after *BusAddressStrobe* termination. Valid delay is therefore 1 CLK2 + 12.5nS + 17nS = 54.5nS. Data setup requirements for the i80386 is 11nS and for the cache memory it is 41nS. A wait state therefore has to be inserted by the system memory.

During memory write cycles, the drivers are opened 1 CLK2 after *BusAddressStrobe* activation. Data will therefore be available 1 CLK2 + 12.5nS + 17nS = 54.5nS after *BusAddressStrobe* activation. The corresponding i80386 data valid delay is 63nS. I.e. memory write cycles could be performed without wait state insertion. However a wait state will always be inserted by system memory because this is a Cache Read Miss requirement as described above.

2.4 Random Logic

The random logic has the following functions:

1. To disable I/O write posting and to generate the READY signals for the i80386 CPU and the i82385 cache controller.
2. To latch the cache miss signal, whenever data is recorded as not present in the cache. This flag may be read by software and is used primarily for testing purposes.
3. To decode the memory cycles, that are mapped as non-cacheable and possibly prevent write-updates of the cache RAM.
4. To control the address line 20 signal for the i82385 cache controller in accordance with the CPU452 scheme.

2.4.1 Ready Generation and Disabling I/O Write Posting

The local 386 bus *Ready* signal is isolated from the *Ready* signal available on the 385 bus. The latter signal is usually disabled by the cache controller, and the cache controller instead provides its own *Ready* signal to terminate Cache Read Hit cycles. However during cycles involving the 385 bus - I/O transfers, memory write, and Cache Read Miss cycles - the 385 bus *Ready* signal is enabled, and the cycle is terminated when the CPU452 device indicates so.

The i82385 cache controller posts not only memory writes, but also I/O writes. This may give I/O recovery or timing problems in some programs and this facility is therefore disabled. This is done with a PAL that upholds the *Ready* signal for the i80386 until the i82385 write buffer is empty during I/O write cycles. The performance penalty is neglectible because of the few number of I/O writes.

BusReady setup requirements are changed. The *Ready* decoding introduces a 11.8nS delay, so that the total setup time is 23.8nS. This however is fulfilled by all *BusReady* sources on the CPU452 board.

2.4.2 Cache Miss Latch

CAM451 also contains circuitry to latch the precense of Cache Miss cycles. This flag is available at bit 4 during an input from the I/O port 2Ch. It will be set if a miss occured since the last read to this address. The miss flag is set 1 CLK after *BusAddressStrobe* goes active if the cycle definition signals indicate that a memory access is in progress and that *Miss* is active. The flag is reset again when the access to I/O address 2Ch terminates.

2.4.3 Non-Cacheable Area Control

Some areas of the system memory are non-cacheable for different reasons. However in order to maximize performance some of these areas may be made fully or partially cacheable. This definition of non-cacheable access is controlled by

writing to I/O port 2Ch. Four bits are used and they may also be read. The functionality is as follows:

Memory Area	Bit:3210	Description
All memory	0xxx	All memory non-cacheable. This will be the situation at system startup. After BIOS-POST bit 3 is set to 1 making the system memory cacheable by removing the FLUSH signal for 82385.
	1pqr	System memory cacheable in the following areas: 0..9FFFFh (0..640Kb) 100000..FBFFFF (1M..15.75Mb) The remaining areas depend on bits 0..2 (pqr) and are defined below.
A0000..BFFFF	xxxx	Videobuffer area. Always non-cacheable.
C0000-C7FFF	1x0x	EGA BIOS area. Non-cacheable.
	1x1x	EGA BIOS read cycles cached. This will be the normal setting since the EGA BIOS may be read but not written. After BIOS-POST, bit 1 is 1 if the POST determined that the EGA controller in use is the native one. Otherwise it will be 0.
C8000-CFFFF	1xx0	Adapter BIOS area. Non-cacheable. This will be the setting also after BIOS-POST.
	1xx1	Adapter BIOS read cycles cached. This will be a possible setting with adapters residing in ROM that do not use bank switching techniques.
D0000-DFFFF	xxxx	Adapter BIOS or device area. Always non-cacheable.
E0000-EFFFF	10xx	System memory (or device area). Non-cacheable. (Default).
	11xx	System memory cacheable. If page E is used as system memory, this will be the normal setting.
F0000-FFFFF and FC0000-FFFFFF	1xxx	BIOS area. Read cycles cached. This will be the normal setting.

After RESET, only bit 3 is defined (0) indicating that the cache is disabled. A typical setting after BIOS-POST would be 1010 indicating that the Adapter BIOS area C8000..CFFFFh is non-cacheable, the EGA BIOS is read cached, page E is not cached, and the rest of the memory (except for D0000..DFFFFh) is cacheable. Making the BIOS areas read cacheable means that they will not be written during accidential writes.

Note that whenever bit 0, 1, or 2 are changed with the cache enabled (bit 3 set to 1), the cache should first be disabled. This is very important because memory that is made cacheable could reside in the cache though changed in system memory creating cache incoherence. An example of a valid NCA register access sequence is shown below.

```
    MOV AL,0
    OUT 2Ch,AL      ;Disable cache (flush its directory)
    JMP ENABLE_CACHE ;Wait for cache to be correctly flushed
ENABLE_CACHE:
    MOV AL,0Fh
    OUT 2Ch,AL      ;Enable cache with new value.
```

Also note that interference problems may occur between the cache and speed switching. This is caused by incapability of the cache controller to operate below 12MHz. When the CPU is slowed down to e.g. 6.6MHz this should be done through the SPEEDSWITCH hook in the BIOS. This code will disable the cache before the CPU is slowed down and will not enable it again before the CPU again is running at 20MHz. If the hook is not used for slowing down, the same operation should be performed with great care. Otherwise the system may crash!

In PAT259, the address lines A15..A25 are decoded and gated with bit 0..3 from the register included in PAT261. It provides a *NonCacheableAccess* signal for the cache controller. This signal is sampled in the cache controller 1 CLK after *AddressStrobe* goes active.

If the accessed memory area is read-enabled but write-disabled as could be the case with the BIOS area, the write-update is prevented by generating a *CacheWriteInhibit* signal. This signal is latched together with the address for the cache RAM's and is used to disable these thus preventing the update.

PAT261 may put the four bits back on the data bus if selected by the -CSEL signal from PAT262 and the -WW/R being low. Its register is loaded as directed by the PAT262 signals -CSEL and -WW/R.

PAT262 contains the state machine used for decoding an I/O access to CAM451, for monitoring wait state insertion, for disabling posted I/O write cycles, and for latching cache miss cycles.

I/O address 2Ch is decoded and generates the *LocalBusAccess* signal for the cache controller and used by the PAT262 state machine for generating the -CSEL signal. This signal is used as an indication of an access to CAM451 and for terminating the cycle by activating READY.

The *LocalBusAccess* signal indicates to the cache controller, that the input cycle is a local 386 bus access. It is sampled by the controller 1 CLK after the address and cycle definition signals become available.

2.4.4 A20 Gating

For PC compability reasons address line 20 is gated on CPU452. The same has to be done on CAM451. The CPU452 *GateA20* signal is supplied through an unused pin in the 386 plug and is used for gating the i82385 A20 signal.

2.4.5 PAL Listings and Description

CAM451 contains three PAL's.

PAT259 decodes the address lines A15..A25, the -W/R signal, and 4 signals latched in a register in PAT261. It generates the -NCA signal for the cache controller to indicate an area as non-cacheable and the -CWINH signal used for inhibiting cache-updates in write-disabled memory areas.

PAT261 contains a four bit register, that captures D0..D3 from the 80386 data bus during output to I/O port 2Ch. Data is captured as indicated at the timing diagrams at page 23. The register content is put on the data bus during input from I/O port 2Ch. This is controlled by the -CSEL and -WW/R signals generated by the PAT262 state machine.

PAT261 also latches the MISS signal when a cache miss cycle occurs. It is put on the 80386 data bus bit 4, when -CSEL goes active and -WW/R is low indicating an input from I/O port 2Ch. The MISS_FLAG is reset after any access to this I/O port.

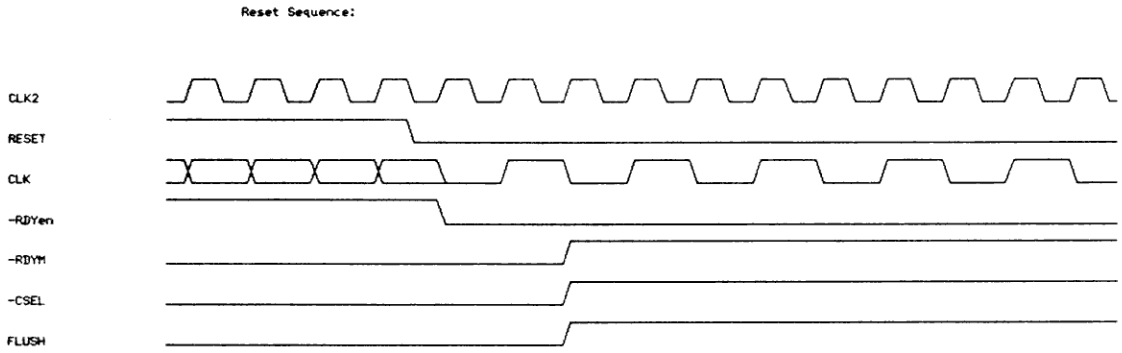
After RESET, bits 0, 1, 2, and 4 are undefined. Only bit 3 passed from PAT262 is defined as 0.

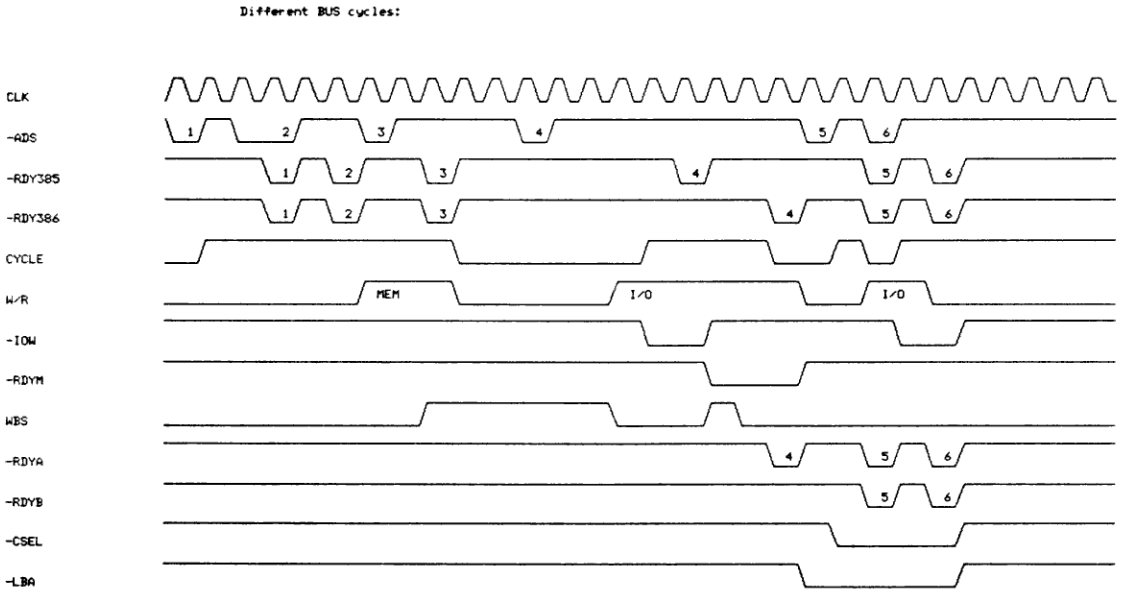
PAT262 is the state machine running in parallel with the state machines contained in 80386 and 82385. It generates various signals identifying the present state:

-SCLK	the system clock running synchronously with the internal 80386 and 82385 SCLK.
RDYM	
-RDYA	used for supplying a 386 READY signal.
-RDYB	used for supplying a 385 READY signal.
-WW/R	is a latched copy of -W/R.

PAT262 also generates the CAM451 select signal -CSEL and the -FLUSH signal which is a copy of Q3 in PAT261.

The next page shows PAT262 signal interdependence during normal 80386 cycles and during pipelined cycles. The subsequent three pages are the definitions for PAT259..262.





Buscycles:

1: non pipelined MEM- or IO-read

2: pipelined MEM- or IO-read

3: MEM write (posting allowed)

4: IO write (posting overridden)

5: Cache control register Read

6: do. Write

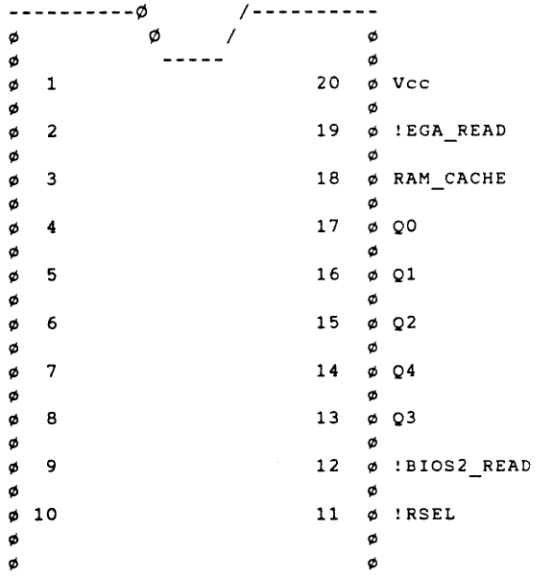
PAT262 timing

```
ABEL(tm) Version 2.10a  -  Document Generator                14-Aug-88 03:48 PM
CAM451 Non_Cacheable_Address Decoder
Albert Jensen, Regnecentralen 130888
Equations for Module PAT259

Device PAT259tx

equations
        enable !WINH=   (1);
        enable RAM_CACHE=       (0);
        enable !EGA_READ_ALLOW=(0);
        enable !BIOS2_READ_ALLOW=(0);
        enable A24= (0);
        enable A25= (0);
        enable !NCA= (1);
        NCA =(0C0000 > Address) & (Address >= 0A0000)
            §(0C8000 > Address) & (Address >= 0C0000) & !EGA_READ_ALLOW
            §(0D0000 > Address) & (Address >= 0C8000) & !BIOS2_READ_ALLOW
            §(0E0000 > Address) & (Address >= 0D0000)
            §(0F0000 > Address) & (Address >= 0E0000) & !RAM_CACHE;
        WINH=((0C8000 > Address) & (Address >= 0C0000)
            §(0D0000 > Address) & (Address >= 0C8000)
            §(100000 > Address) & (Address >= 0F0000)
            §(1000000 > Address) & (Address >= 0FC0000)) & !RD;

- Reduced Equations:

    enable !WINH = (1);

    enable RAM_CACHE = (0);

    enable !EGA_READ_ALLOW = (0);

    enable !BIOS2_READ_ALLOW = (0);

    enable A24 = (0);

    enable A25 = (0);

    enable !NCA = (1);

    !NCA  = !(A17 & !A18 & A19 & !A20 & !A21 & !A22 & !A23 & !A24 & !A25
              § A16 & !A17 & A18 & A19 & !A20 & !A21 & !A22 & !A23 & !A24 &
            !A25
              § !A16 & A17 & A18 & A19 & !A20 & !A21 & !A22 & !A23 & !A24 &
            !A25 & !RAM_CACHE
              § !A15 & !A16 & !A17 & A18 & A19 & !A20 & !A21 & !A22 & !A23 &
            !A24 & !A25 & !EGA_READ_ALLOW
              § A15 & !A16 & !A17 & A18 & A19 & !A20 & !A21 & !A22 & !A23 &
            !A24 & !A25 & !BIOS2_READ_ALLOW);

    !WINH = !(A18 & A19 & A20 & A21 & A22 & A23 & !A24 & !A25 & !RD
              § A16 & A17 & A18 & A19 & !A20 & !A21 & !A22 & !A23 & !A24 &
            !A25 & !RD
              § !A15 & !A16 & !A17 & A18 & A19 & !A20 & !A21 & !A22 & !A23 &
            !A24 & !A25 & !RD
              § A15 & !A16 & !A17 & A18 & A19 & !A20 & !A21 & !A22 & !A23 &
            !A24 & !A25 & !RD);

Chip diagram for Module PAT259
Device PAT259tx

                          P16L8

                    ----------Ø       /----------
                    ø          Ø     /          ø
                    ø                -----      ø
                !RD ø  1                       20  ø Vcc
                    ø                          ø
                A15 ø  2                       19  ø !NCA
                    ø                          ø
                A16 ø  3                       18  ø !BIOS2_READ_ALLO
                    ø                          ø
                A17 ø  4                       17  ø !EGA_READ_ALLOW
                    ø                          ø
                A18 ø  5                       16  ø RAM_CACHE
                    ø                          ø
                A19 ø  6                       15  ø !WINH
                    ø                          ø
                A20 ø  7                       14  ø A25
                    ø                          ø
                A21 ø  8                       13  ø A24
                    ø                          ø
                A22 ø  9                       12  ø
                    ø                          ø
                GND ø 10                       11  ø A23
                    ø                          ø
                    ø                          ø
                    ----------------------------
```

```
ABEL(tm) Version 2.10a  -  Document Generator              25-Aug-88 01:43 PM
CAM451 Non-Cacheable-Address Register
Albert Jensen, Regnecentralen 150288
Equations for Module PAT261

Device PAT261tx

equations
        enable BIOS2_READ =  (1);
        enable Q3         = CSEL & WWR;
        enable RAM_CACHE  =  (1);
        enable EGA_READ   =  (1);
        Q0:= Q0 & !sel
             § sel & D0;
        Q1:= Q1 & !sel
             § sel & D1;
        Q2:= Q2 & !sel
             §  sel & D2;
        Q3 = !FLUSH;
        Q4:= Q4 & !(CSEL & WWR & RDY386)        "Set when MISS
             § MISS;
        BIOS2_READ=   Q0;
        EGA_READ  =   Q1;
        RAM_CACHE =   Q2;

- Reduced Equations:

    enable !BIOS2_READ = (1);

    enable Q3 = (CSEL & WWR);

    enable RAM_CACHE = (1);

    enable !EGA_READ = (1);

    Q0 := !(!CSEL & !Q0
           § !D0 & !Q0
           § !Q0 & !RDY386
           § !Q0 & WWR
           § CSEL & !D0 & RDY386 & !WWR);

    Q1 := !(!CSEL & !Q1
           § !D1 & !Q1
           § !Q1 & !RDY386
           § !Q1 & WWR
           § CSEL & !D1 & RDY386 & !WWR);

    Q2 := !(!CSEL & !Q2
           § !D2 & !Q2
           § !Q2 & !RDY386
           § !Q2 & WWR
           § CSEL & !D2 & RDY386 & !WWR);

    Q3 = !(FLUSH);

    Q4 := !(!MISS & !Q4 § CSEL & !MISS & RDY386 & WWR);

    !BIOS2_READ = !(Q0);

    !EGA_READ = !(Q1);

    RAM_CACHE = !(!Q2);
```

Chip diagram for Module PAT261

Device PAT261tx



```
ABEL(tm) Version 2.10a  -  Document Generator              24-Aug-88 01:19 PM
Local Ready- and FLUSH- generation and Suppression of posted IOwrite.
Albert Jensen, Regnecentralen 240888
Equations for Module PAT262

Device PAT262tx

equations
"State machine
     SCLK:=     !SCLK & !RESET                             "Phase tracking.
                & !(CYCLE & RDYM & CSEL)                   "SCLK=0 just after RESET
                § RESET & !RD                              "
                § RESET & RD & !SCLK;                      " test initialization.
                                                           "
   !CYCLE:=     READY & !IOW & SCLK & !ADS & !RESET        "Indicates an active
                § RDYM & !WBS & SCLK & !RESET              " bus cycle.
                § CSEL & !RDYA & SCLK & !RESET             " CYCLE is set at ADS
                §!CYCLE & !ADS & !RESET                    " and cleared at ready
                §!CYCLE & !SCLK & !RESET                   " if no pipelining.
                § CSEL & RDYM & !RESET;                    "
                                                           "
   !IOW:=       !IOW & (!ADS                               "Is set when an IOwrite
                      § (!READY & CYCLE & !RDYA)           " is initiated.
                      § RD § !IO § !SCLK)                  "Inhibits RDYO to 386
                § IOW & READY & SCLK                       "Held until IOwrite is
                § RESET;                                   " posted.
                                                           "
    RDYM:=      READY & IOW & SCLK & !CSEL                 "Set when 385 has posted
                § RDYM & !CSEL & WBS & !IOW                " the IOwrite.
                § RDYM & CSEL & CYCLE                      "Held until the IOwrite
                § RDYM & !SCLK                             " is done then assures
                § RESET;                                   " that RDYA is set.

    CSEL:=      LBA & ADS & SCLK & READY & !IOW            "Selection of Local Bus
                § LBA & ADS & SCLK & !CYCLE                " register
                § CSEL & !RDYM & !RDYB                     "
                § CSEL & RDYM & CYCLE                      "CSEL and RDYM are
                § CSEL & !SCLK                             " coincident after RESET
                § RESET;                                   " only.
                                                           "
    RDYA:=      RDYM & !CSEL & !WBS & SCLK                 "RDY386
                § CSEL & !RDYM & SCLK & !RDYA              " generated after a
                § RDYA & !SCLK & !RDYM                     " Local Bus Access and
                § RDYA & !SCLK & !CSEL                     " at the completion of
                § RESET;                                   " a posted IOwrite.
                                                           "
    RDYB:=      CSEL & !RDYM & SCLK & !RDYB & !RESET       "RDY385
                § RDYB & !SCLK & !RESET;                   " generated after a
                                                           " Local Bus Access.
"FLUSH latching
   !FLUSH:=     RESET                                      "FLUSH is set after
                § CYCLE & RDYM & CSEL                      " RESET at coincidence
                § !FLUSH & !SCLK                           " of CSEL and RDYM.
                § !FLUSH & !CSEL                           "
                § !FLUSH & !RDYA & !RDYM                   " Then altered according
                § !FLUSH & !IOW & !RDYM                    " to D3 on LBA-writes
                § D3 & SCLK & CSEL & IOW & RDYA;           " only.

- Reduced Equations:

    !SCLK  := !(!RD & RESET
                § !CSEL & !RESET & !SCLK
                § !CYCLE & !RESET & !SCLK
                § RD & RESET & !SCLK
                § !RDYM & !RESET & !SCLK);

    CYCLE  := !(!ADS & !CYCLE & !RESET
                § CSEL & RDYM & !RESET
                § !CYCLE & !RESET & !SCLK
                § CSEL & !RDYA & !RESET & SCLK
                § RDYM & !RESET & SCLK & !WBS
                § !ADS & !IOW & READY & !RESET & SCLK);

    IOW  := !(RESET
              § !ADS & !IOW
              § !IO & !IOW
              § !IOW & RD
              § !IOW & !SCLK
              § IOW & READY & SCLK
              § CYCLE & !IOW & !RDYA & !READY);

    !RDYM  := !(RESET
                § RDYM & !SCLK
                § CSEL & CYCLE & RDYM
                § !CSEL & !IOW & RDYM & WBS
                § !CSEL & IOW & READY & SCLK);

    !CSEL  := !(RESET
                § CSEL & !SCLK
                § CSEL & CYCLE & RDYM
                § CSEL & !RDYB & !RDYM
                § ADS & !CYCLE & LBA & SCLK
                § ADS & !IOW & LBA & READY & SCLK);

    !RDYA  := !(RESET
                § !CSEL & RDYA & !SCLK
                § RDYA & !RDYM & !SCLK
                § CSEL & !RDYA & !RDYM & SCLK
                § !CSEL & RDYM & SCLK & !WBS);

    !RDYB  := !(RDYB & !RESET & !SCLK
                § CSEL & !RDYB & !RDYM & !RESET & SCLK);

    FLUSH  := !(RESET
                § !CSEL & !FLUSH
                § !FLUSH & !SCLK
                § CSEL & CYCLE & RDYM
                § !FLUSH & !IOW & !RDYM
                § !FLUSH & !RDYA & !RDYM
                § CSEL & D3 & IOW & RDYA & SCLK);
```

```
Chip diagram for Module PAT262

Device PAT262tx

                                  P16R8

              ----------ø         /----------
                        ø    ø  ____/        ø
                        ø                    ø
           CLK ø  1                    20    ø  Vc
        !READY ø  2                    19    ø  !R
           !IO ø  3                    18    ø  IO
           !RD ø  4                    17    ø  CY
           WBS ø  5                    16    ø  !R
            D3 ø  6                    15    ø  !C
          !LBA ø  7                    14    ø  !S
         RESET ø  8                    13    ø  !R
          !ADS ø  9                    12    ø  FL
           GND ø 10                    11    ø  OE
               ø                             ø
              -------------------------------
```

2.5 Cache Memory

The cache memory consists of 4 8Kbytes static RAM's with 35nS access time. They are organized as 8K by 32bit and their data pins are connected to the local 386 data bus.

The cache memory is addressed from the local 386 address bus through 74F373 latches.

To speed up cache read cycles, the cache RAM outputs are usually enabled 1 CLK after memory read cycles start. However if the cycle turns out to be a Cache Read Miss, the outputs are disabled again after 1 CLK, and 1/2 CLK later data from the 385 bus is gated to the local 386 data bus through the 74F646 transceivers. This delay related to the switch in bus mastership is removed with consecutive Cache Read Miss cycles, because the cache controller puts the i80386 into pipelined mode. This allows the bus mastership decision to be made in time, without having to incur this extra state.

Cache update cycles can be inhibited when the CWINH signal from PAT259 is active. It will be latched together with the address and will disable all cache RAM's.

2.6 80386 Plug

The interface between the CPU452 and the CAM451 goes through the 386 PGA plug. All signals corresponding to the i80386 signals are led through here.

To supply extra signals for CAM451 three unused pins are used. These are B6, C6, and C7. Through these pins the following signals are fed:

SA20	The gated address line 20 from the 385 address bus.
-DWE	The data-write-enable generated by 82C302 during memory writes.
GATEA20	The CPU452 signal used for gating address line 20.

3. Testing Requirements

Memory testing has to be changed with the insertion of *CAM451*. The reason is that the *CAM451* functions transparently to software. An appropriate strategy for testing system memory therefore has to be developed and a strategy for testing the cache memory and its controller has to be developed.

3.1 System Memory Testing

Most systems perform a memory test by writing a data pattern to system memory and afterwards comparing this pattern with what it should be. This test however however may give an erroneous result if the cache memory is not taken into account. If the block written and subsequently compared is smaller than 32Kbytes, the data written will reside in the cache, and therefore the cache and not the system memory is actually tested.

To circumvent the caching interference, memory testing should work with blocks greater than 32Kbytes. Preferably 64Kbytes. This will insure the first data written to be flushed, before it is read again. Patterns in location *N* and *N*+32Kbyte should differ.

It should be noted that this procedure tests not only the system memory but also partially the cache memory and its directory, since test results are passed through here. To insure complete testing of the cache, a special test procedure should be included also as described below.

3.2 Cache Memory Testing

Two strategies for testing cache memory exist depending on whether the test routine itself resides in cache memory or not. To enhance testing, the non-cached test routine strategy is chosen. The steps to be performed are the following:

1. The test routine should be loaded into page E of system memory. This is done by e.g. a REP MOVSW operation. A branch to the test routine should then be performed.
2. Next the cache directory for the rest of the cache should be defined by performing a read to 32Kbytes.
3. An appropriate test pattern should be written into the cache.
4. This test pattern should be compared by reading the same locations. Now only cache data are used because of read hits. Step 3 and 4 may be repeated with other test patterns.
5. A branch back to the main POST should be performed to resume testing or report any errors.

This routine will propably test the cache directory and the cache memory itself. A possibility however exists that the cache directory functions improperly and that read accesses that should be hits are signalled as misses - i.e. the comparison performed in step 4 is done not to the cache but to system memory. To circumvent this possibility, the -MISS output from 82385 is latched and is monitored by the test routine. If this latched MISS gets active during step 4 it would indicate a cache directory malfunction. The MISS flag is available as bit 4 at I/O port 2Ch. It is reset after each access to this port.

4. Using CAM451

Software interface with CAM451 goes through I/O port 2Ch. Bit 0-3 on this port defines cache function and bit 4 is a *cache miss* flag set whenever a cache miss access occurs and reset whenever red. It is primarily used for testing purposes.

Bit	R/W	Function
0	R/W	Enable read-caching of adapter BIOS area (C8000..CFFFFh) if cache is enabled.
1	R/W	Enable read-caching of EGA BIOS area (C0000..C7FFFh) if cache is enabled.
2	R/W	Enable caching of system memory E0000..EFFFF if cache is enabled.
3	R/W	Master cache enable. Cache should be disabled (bit 3 set to 0) between every change of bit 0..2. This will flush the cache.
4	RO	Cache Miss flag. Set on every cache miss access. Reset after red.

Memory Area	Bit:3210	Description
All memory	0xxx	All memory non-cacheable. This will be the situation at system startup. After BIOS-POST bit 3 is set to 1 making the system memory cacheable by removing the FLUSH signal for 82385.
	1pqr	System memory cacheable in the following areas: 0..9FFFFh (0..640Kb) 100000..FBFFFF (1M..15.75Mb) The remaining areas depend on bits 0..2 (pqr) and are defined below.
A0000..BFFFF	xxxx	Videobuffer area. Always non-cacheable.
C0000-C7FFF	1x0x	EGA BIOS area. Non-cacheable.
	1x1x	EGA BIOS read cycles cached. This will be the normal setting since the EGA BIOS may be read but not written. After BIOS-POST, bit 1 is 1 if the POST determined that the EGA controller in use is the native one. Otherwise it will be 0.
C8000-CFFFF	1xx0	Adapter BIOS area. Non-cacheable. This will be the setting also after BIOS-POST.
	1xx1	Adapter BIOS read cycles cached. This will be a possible setting with adapters residing in ROM that do not use bank switching techniques.
D0000-DFFFF	xxxx	Adapter BIOS or device area. Always non-cacheable.

E0000-EFFFF	10xx	System memory (or device area). Non-cacheable. (Default).
	11xx	System memory cacheable. If page E is used as system memory, this will be the normal setting.
F0000-FFFFF and FC0000-FFFFFF	1xxx	BIOS area. Read cycles cached. This will be the normal setting.

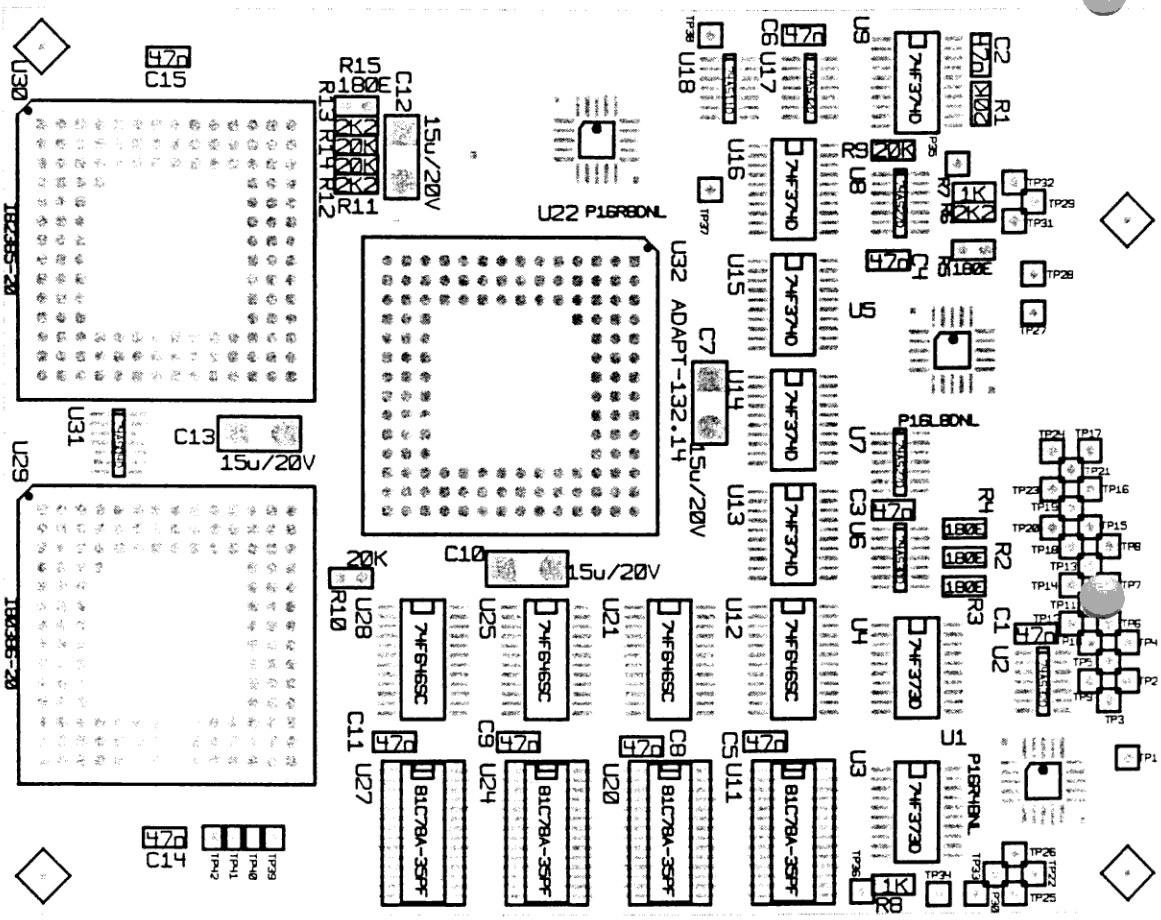
After RESET, bit 3 is 0. Bits 0-2 are undefined. A typical setting after BIOS-POST would be 1010 indicating that the Adapter BIOS area C8000..CFFFFh is non-cacheable, the EGA BIOS is read cached, page E is not cached, and the rest of the memory (except for D0000..DFFFFh) is cacheable. Making the BIOS areas read cacheable means that they will not be written during accidential writes.

Note that whenever bit 0, 1, or 2 are changed with the cache enabled (bit 3 set to 1), the cache should first be disabled. This is very important because memory that is made cacheable could reside in the cache though changed in system memory creating cache incoherence. An example of a valid NCA register access sequence is shown below.

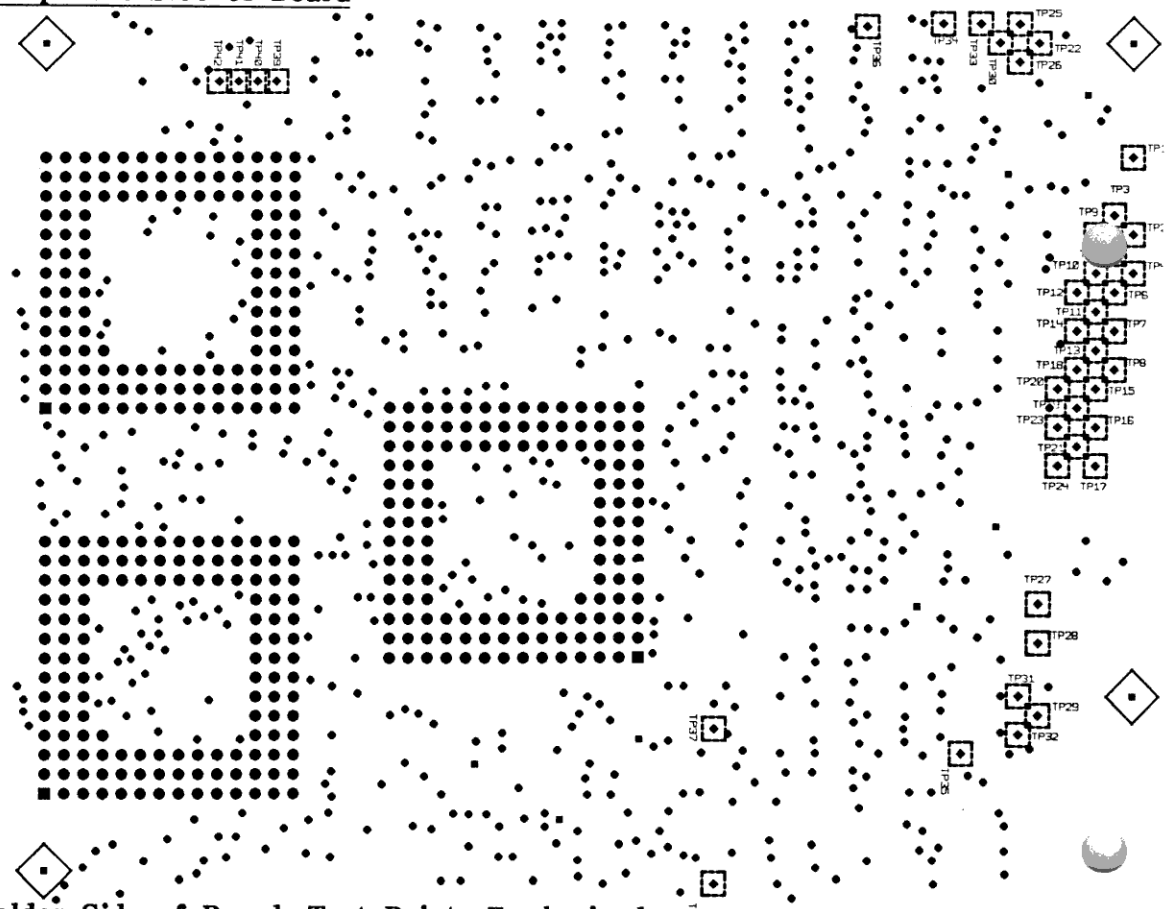
```
    MOV AL,0
    OUT 2Ch,AL          ;Disable cache (flush its directory)
    JMP ENABLE_CACHE    ;Wait for cache to be correctly flushed
ENABLE_CACHE:
    MOV AL,0Fh
    OUT 2Ch,AL          ;Enable cache with new value.
```

Also note that interference problems may occur between the cache and speed switching. This is caused by incapability of the cache controller to operate below 12MHz. When the CPU is slowed down to e.g. 6.6MHz this should be done through the SPEEDSWITCH hook in the BIOS. This code will disable the cache before the CPU is slowed down and will not enable it again before the CPU again is running at 20MHz. If the hook is not used for slowing down, the same operation should be performed with great care. Otherwise the system may crash!

5. Printed Circuit Board Layout



Component Side of Board



Solder Side of Board. Test Points Emphasized.

6. Logic Diagrams

This section contains the logic diagrams for the CAM451. Each diagram has a page number in its lower rigth corner. This number is different from the page numbers in this document. Each signal has numbers associated with it designating which other diagram pages that use this signal.

Figure 3 below contains the CAM451 block diagram and numbers corresponding to the relevant diagram pages.

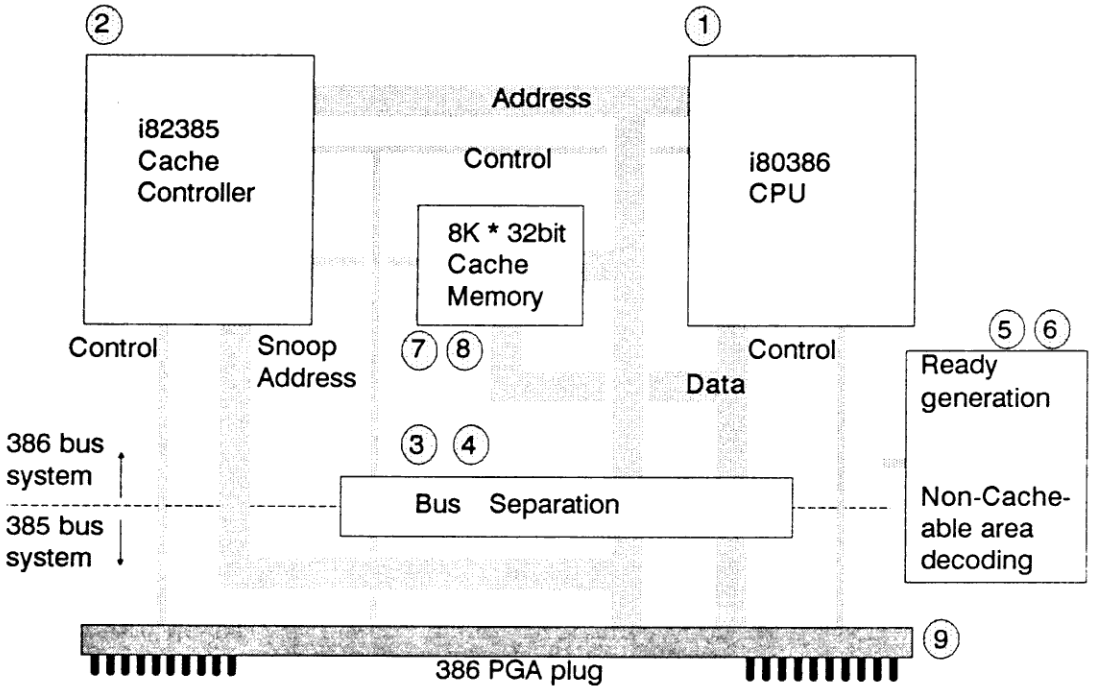
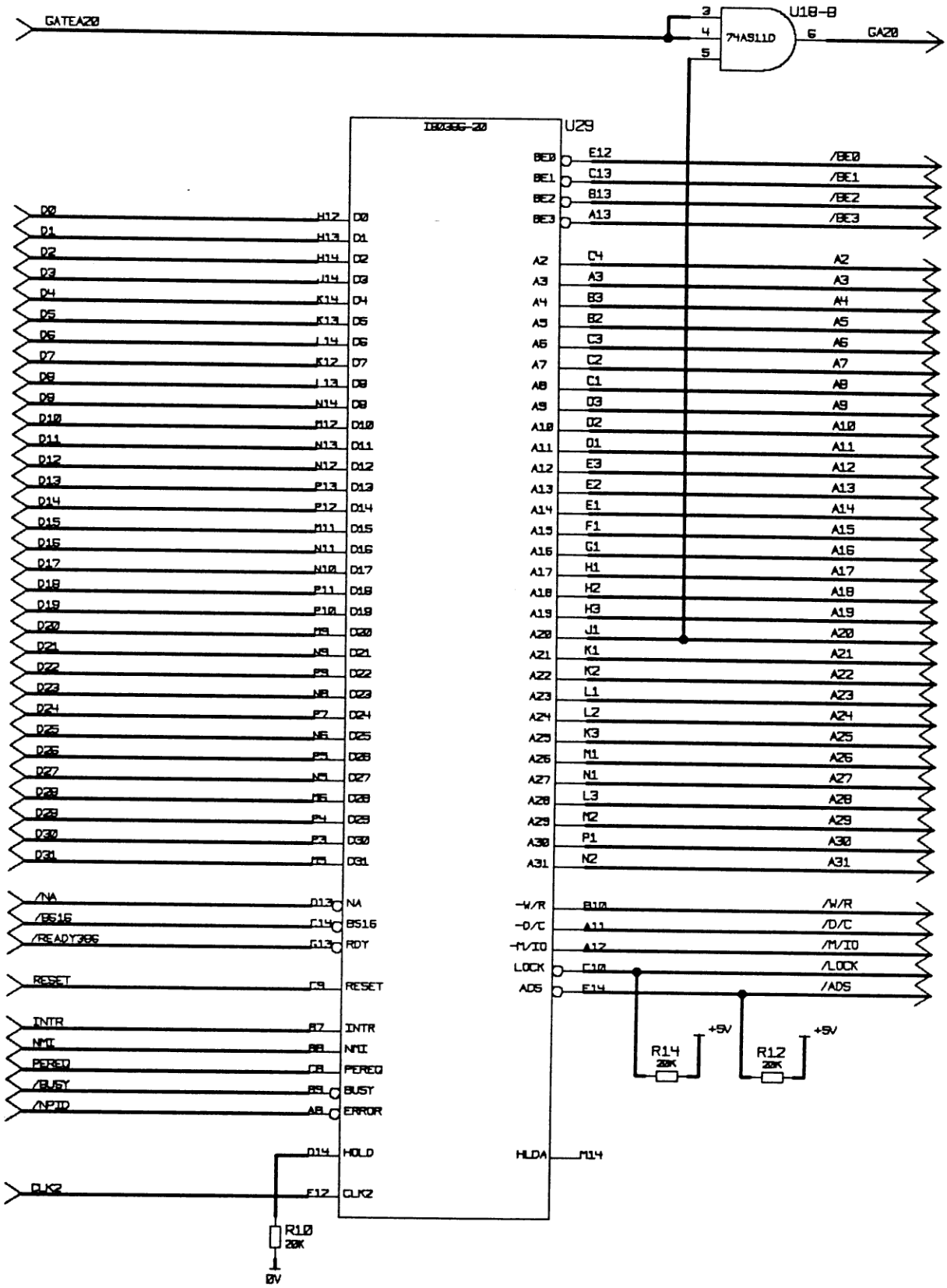
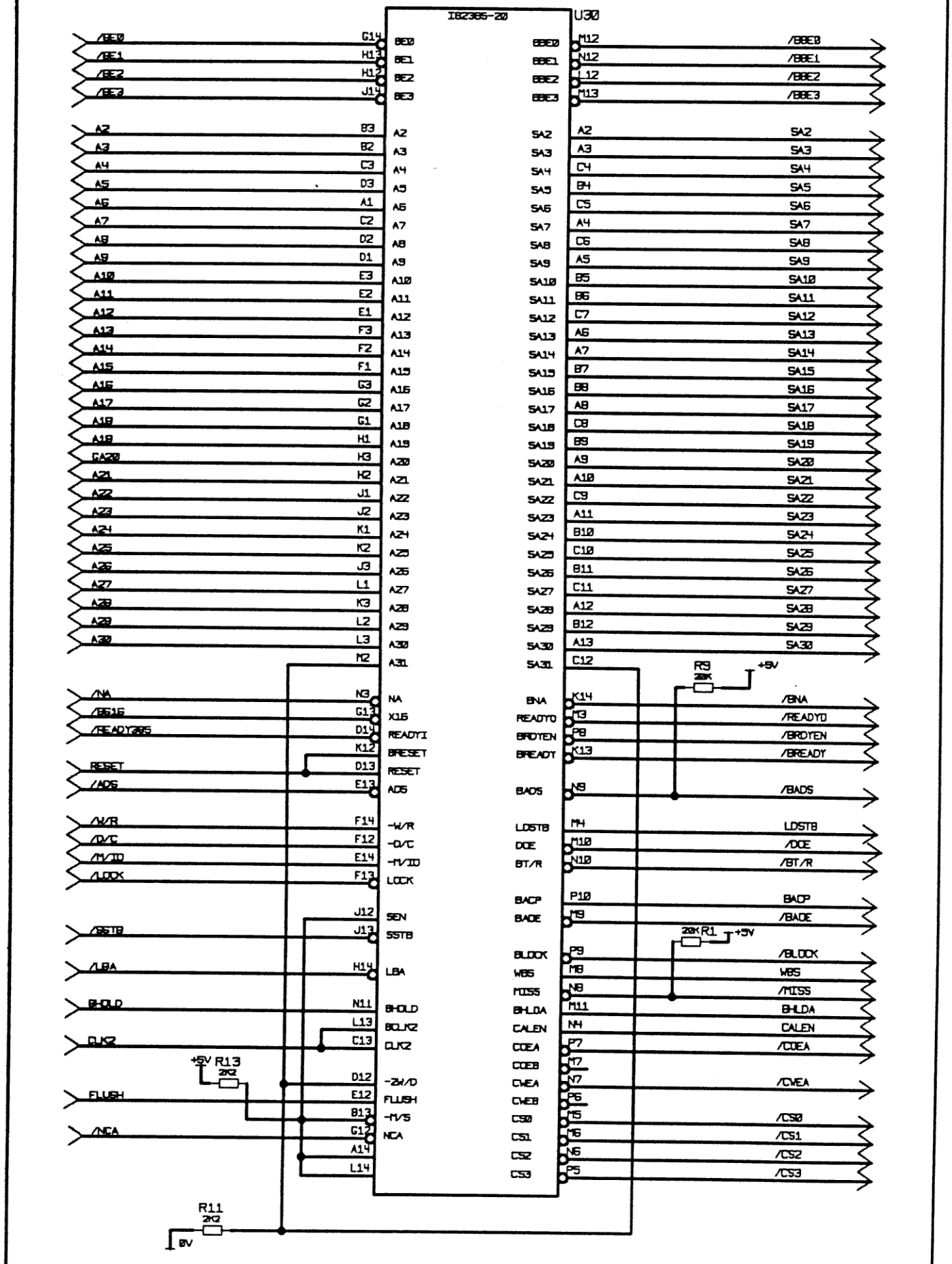


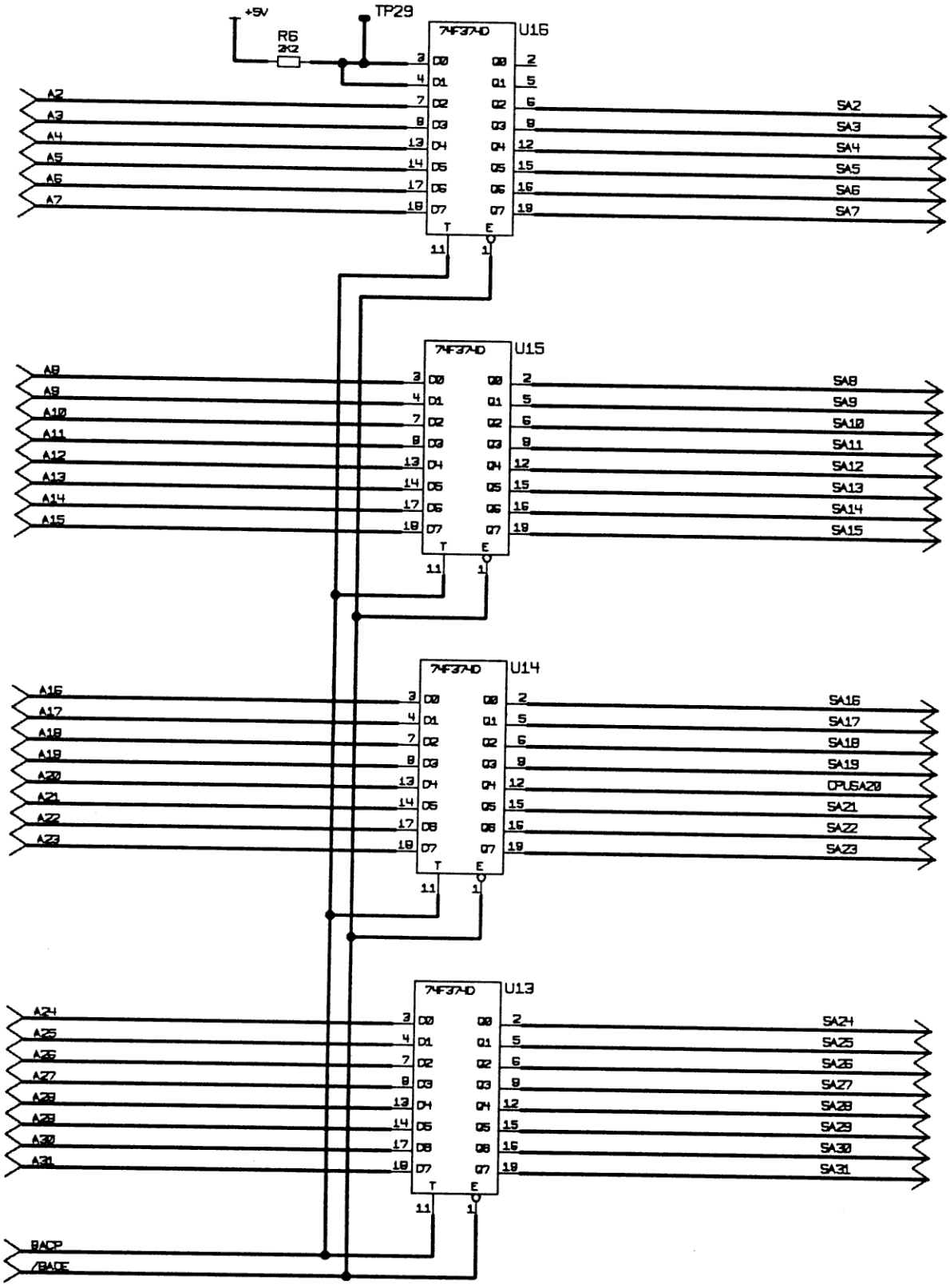
Figure 3. CAM451 Block Diagram with logic diagram references.



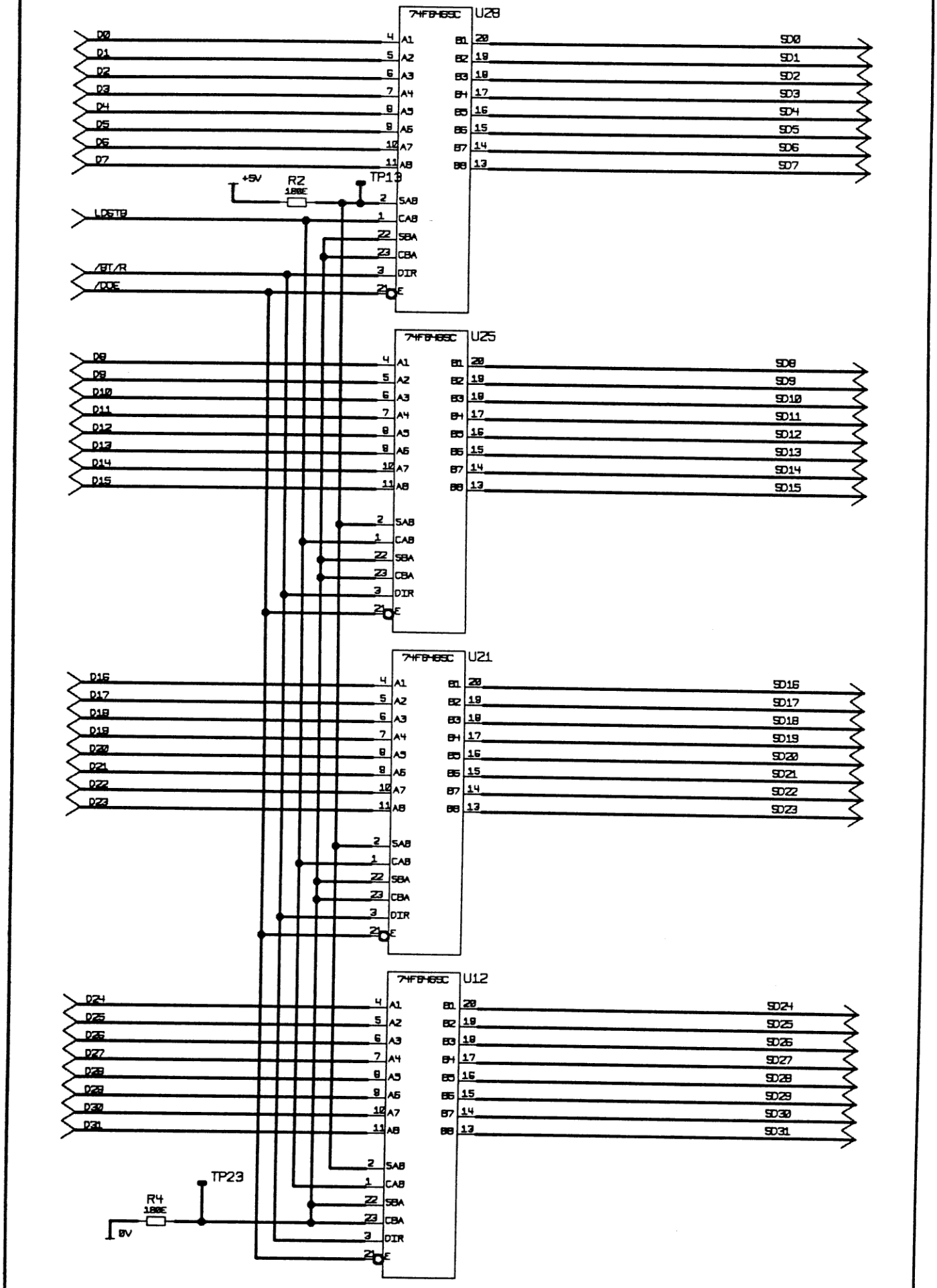
RC900 CAM451	CPU	DATE: 1988.10.13 DESIGNER: AFJ SHEET: 1



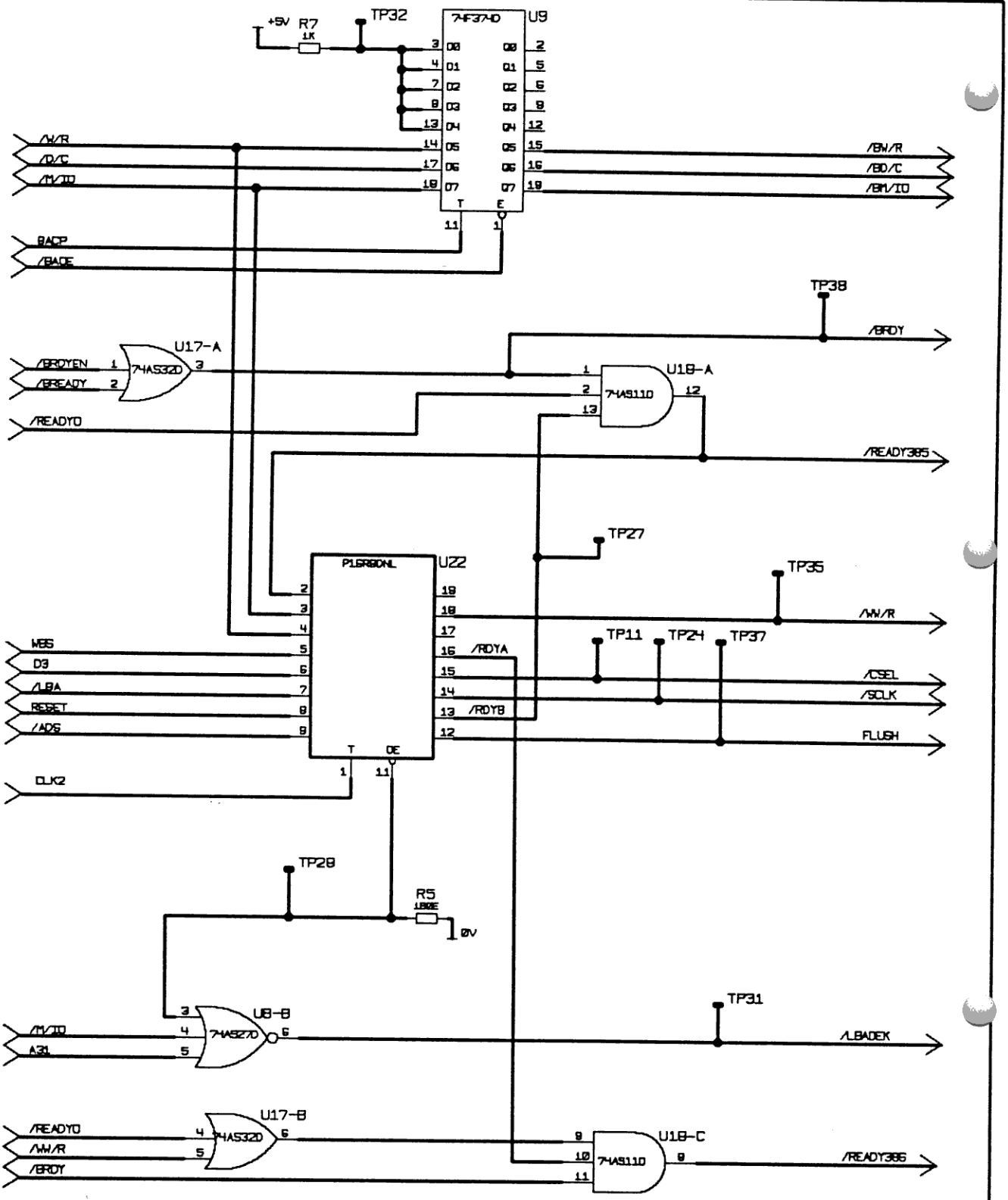
RC900 CAM451	CACHE CONTROLLER	DATE: 1988.10.13 DESIGNER: AFJ SHEET: 2



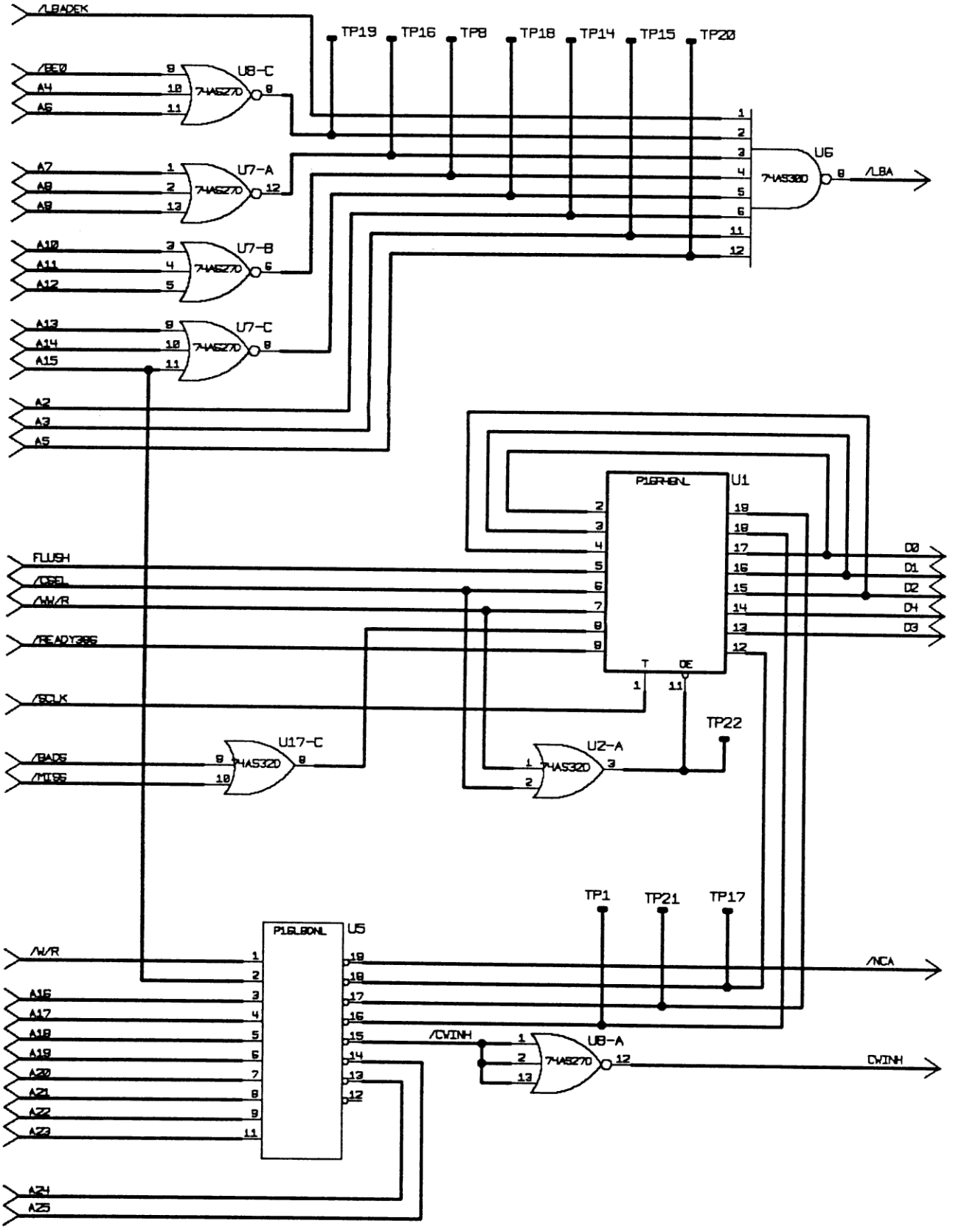
RC900 CAM451	ADDRESS BUS SEPARATOR	DATE: 1988.10.13 DESIGNER: AFJ SHEET: 3



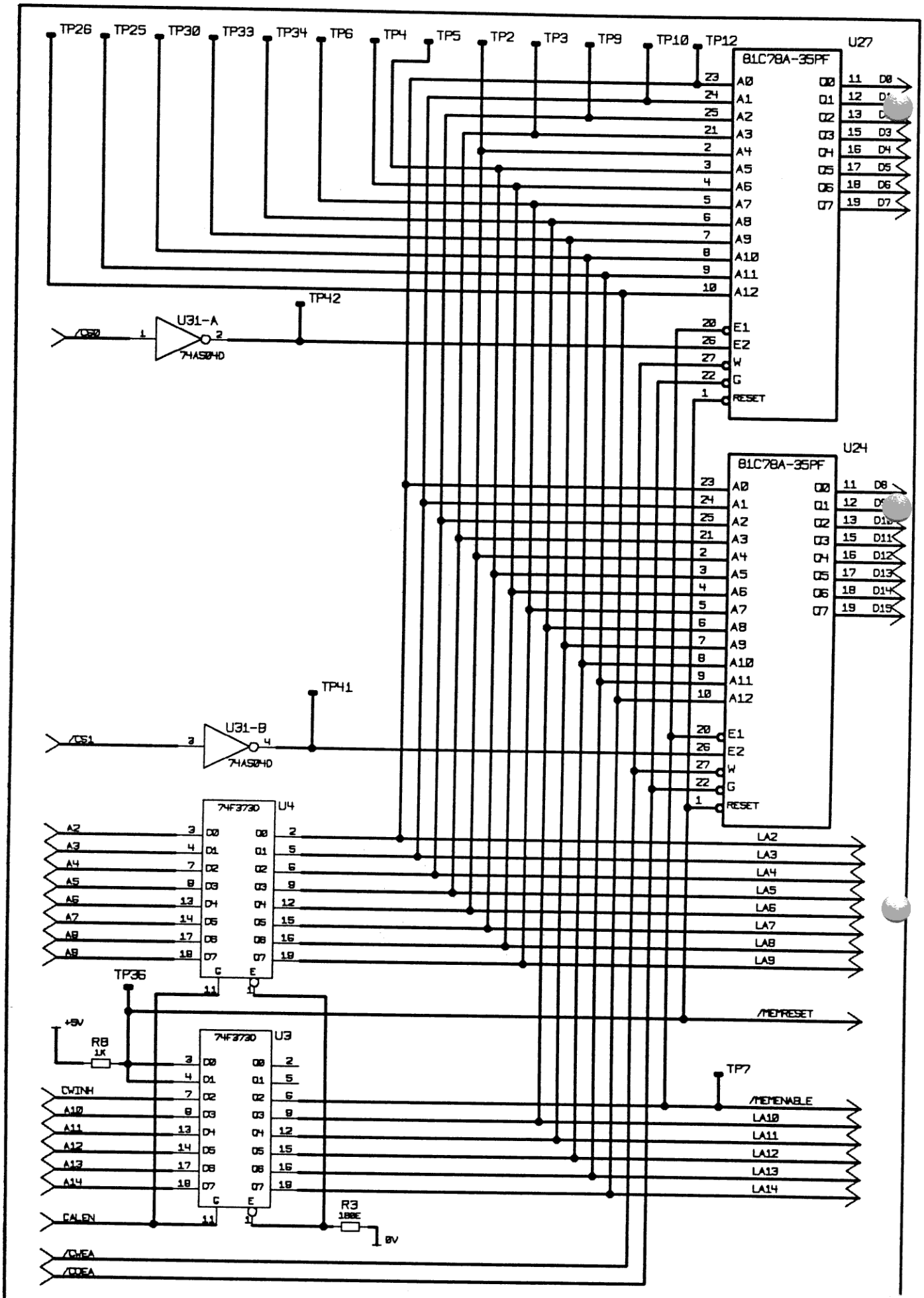
RC900 CAM451	DATA BUS SEPARATOR	DATE: 1988.10.13 DESIGNER: AFJ SHEET: 4



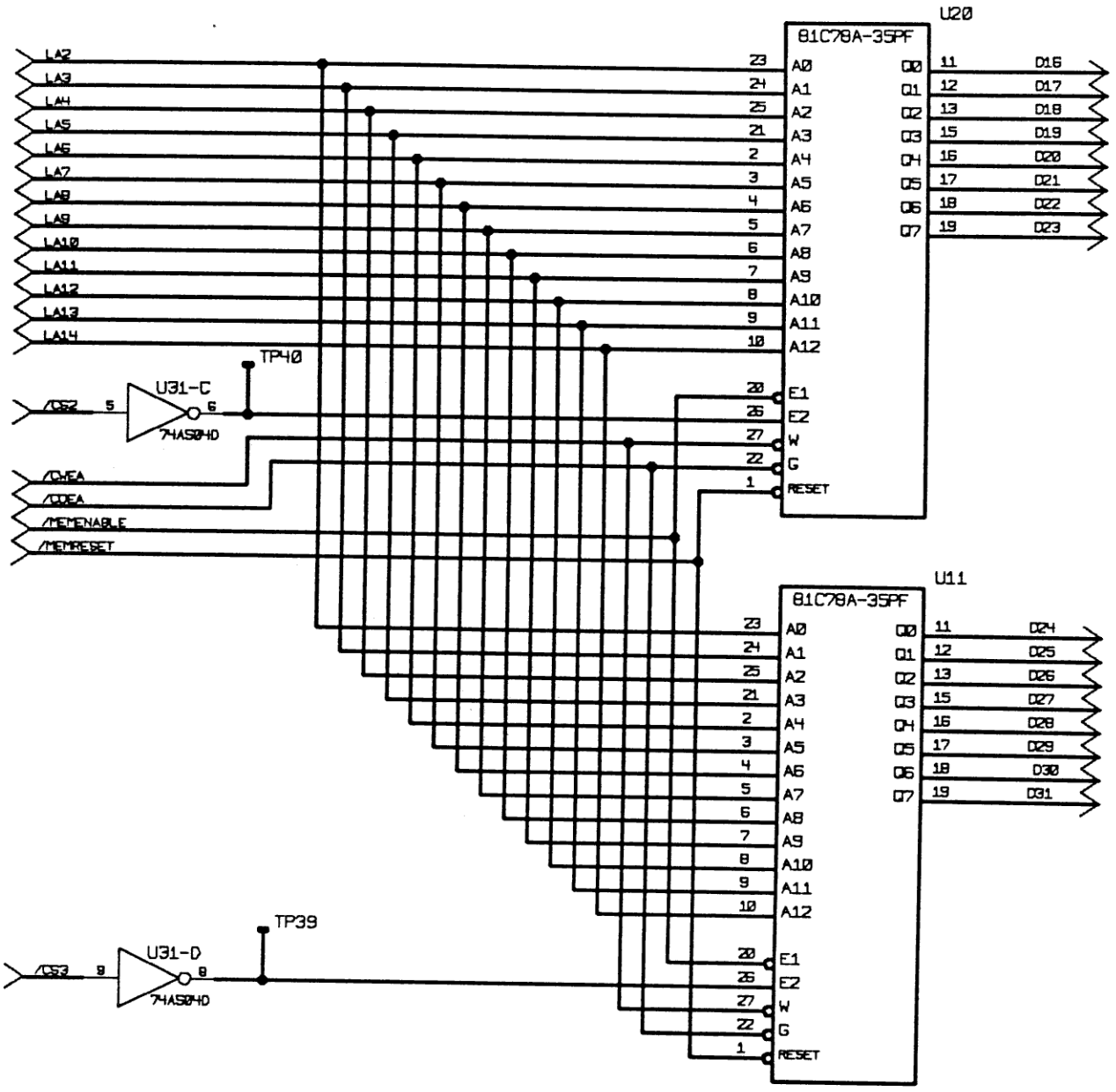
RC900 CAM451	CONTROL SIGNAL SEPARATOR 386 STATE MACHINE	DATE: 1988.10.13 DESIGNER: AFJ SHEET: 5



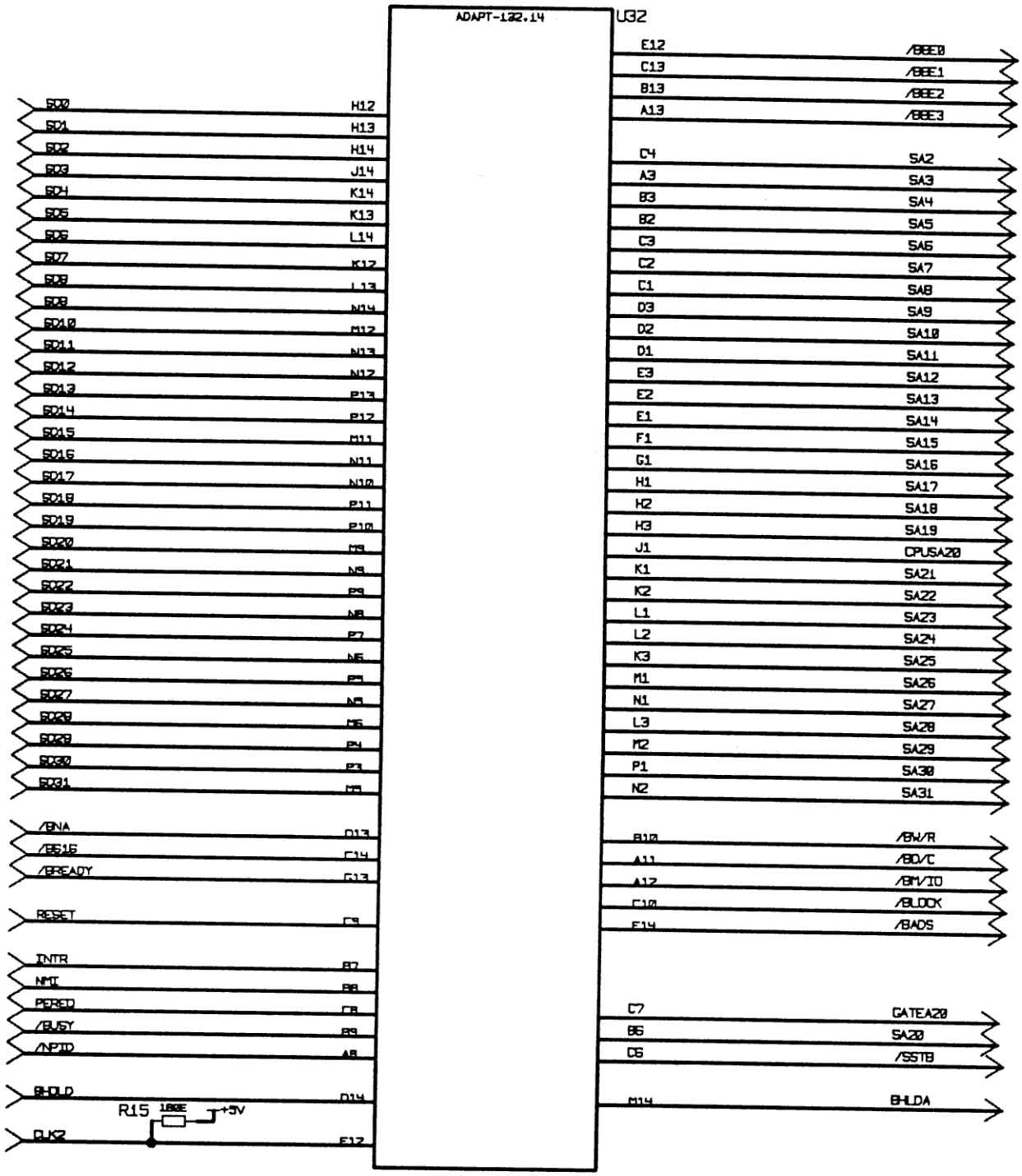
RC900 CAM451	NON CACHEABLE MEMORY AREA DECODING	DATE: 1988.10.13 DESIGNER: AFJ SHEET: 6



RC900 CAM451	8K*16 CACHE RAM	DATE: 1988.10.13 DESIGNER: AFJ SHEET: 7



RC900 CAM451	8K*16 CACHE RAM	DATE: 1988.10.13 DESIGNER: AFJ SHEET: 8



RC900 CAM451	80386 PLUG	DATE: 1988.10.13 DESIGNER: AFJ SHEET: 9

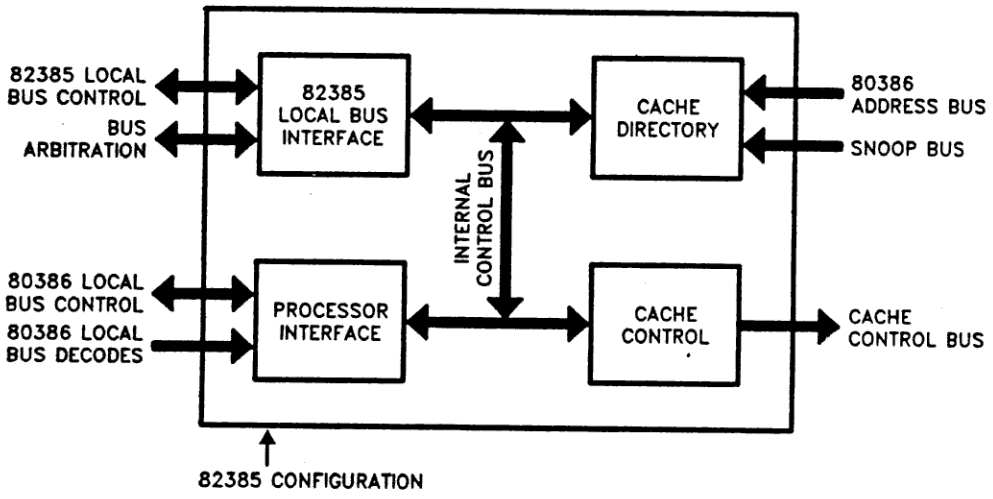
ADVANCE INFORMATION

82385
HIGH PERFORMANCE
32-BIT CACHE CONTROLLER

- **Improves 80386 System Performance**
  - **— Reduces Average CPU Wait States to Nearly Zero**
  - **— Zero Wait State Read Hit**
  - **— Zero Wait State Posted Writes**
  - **— Allows Other Masters to Access the System Bus More Readily**
- **Hit Rates up to 99%**
- **Optimized as 80386 Companion**
  - **— Simple 80386 Interface**
  - **— Part of 386-Based Compute Engine Including 80387 Numerics Coprocessor and 82380 Integrated System Peripheral**
  - **— 16 MHz and 20 MHz Operation**
- **Software Transparent**
- **Synchronous Dual Bus Architecture**
  - **— Bus Watching Maintains Cache Coherency**
- **Maps Full 80386 Address Space (4 Gigabytes)**
- **Flexible Cache Mapping Policies**
  - **— Direct Mapped or 2-Way Set Associative Cache Organization**
  - **— Supports Non-Cacheable Memory Space**
  - **— Unified Cache for Code and Data**
- **Integrates Cache Directory and Cache Management Logic**
- **High Speed CHMOS III Technology**
- **132-Pin PGA Package**

The 82385 Cache Controller is a high performance 32-bit peripheral for Intel's 80386 Microprocessor. It stores a copy of frequently accessed code and data from main memory in a zero wait state local cache memory. The 82385 enables the 80386 to run at its full potential by reducing the average number of CPU wait states to nearly zero. The dual bus architecture of the 82385 allows other masters to access system resources while the 80386 operates locally out of its cache. In this situation, the 82385's "bus watching" mechanism preserves cache coherency by monitoring the system bus address lines at no cost to system or local throughput.

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



290143-1

82385 Internal Block Diagram

July 1987
Order Number: 290143-001

1.0 82385 FUNCTIONAL OVERVIEW

The 82385 Cache Controller is a high performance 32-bit peripheral for Intel's 80386 microprocessor. This chapter provides an overview of the 82385, and of the basic architecture and operation of an 80386/82385 system.

1.1 82385 OVERVIEW

The main function of a cache memory system is to provide fast local storage for frequently accessed code and data. The cache system intercepts 80386 memory references to see if the required data resides in the cache. If the data resides in the cache (a hit), it is returned to the 80386 without incurring wait states. If the data is not cached (a miss), the reference is forwarded to the system and the data retrieved from main memory. An efficient cache will yield a high "hit rate" (the ratio of cache hits to total 80386 accesses), such that the majority of accesses are serviced with zero wait states. The net effect is that the wait states incurred in a relatively infrequent miss are averaged over a large number of accesses, resulting in an average of nearly zero wait states per access. Since cache hits are serviced locally, a processor operating out of its local cache has a much lower "bus utilization" which reduces system bus bandwidth requirements, making more bandwidth available to other bus masters.

The 82385 Cache Controller integrates a cache directory and all cache management logic required to support an external 32 Kbyte cache. The cache directory structure is such that the entire physical address range of the 80386 (4 Gigabytes) is mapped into the cache. Provision is made to allow areas of memory to be set aside a non-cacheable. The user has two cache organization options: direct mapped and 2-way set associative. Both provide the high hit rates necessary to make a large, relatively slow main memory array look like a fast, zero wait state memory to the 80386.

A good hit rate is an essential ingredient of a successful cache implementation. Hit rate is the measure of how efficient a cache is in maintaining a copy of the most frequently requested code and data. However, efficiency is not the only factor for performance consideration. Just as essential are sound cache management policies. These policies refer to the handling of 80386 writes, preservation of cache coherency, and ease of system design. The 82385's "posted write" capability allows the majority of 80386 writes, including non-cacheable and I/O writes, to run with zero wait states, and the 82385's "bus watching" mechanism preserves cache coherency with no impact on system performance. Physically, the 82385 ties directly to the 80386 with virtually no external logic.

1.2 SYSTEM OVERVIEW I: BUS STRUCTURE

A good grasp of the bus structure of an 80386/82385 system is essential in understanding both the 82385 and its role in an 80386 system. The following is a description of this structure.

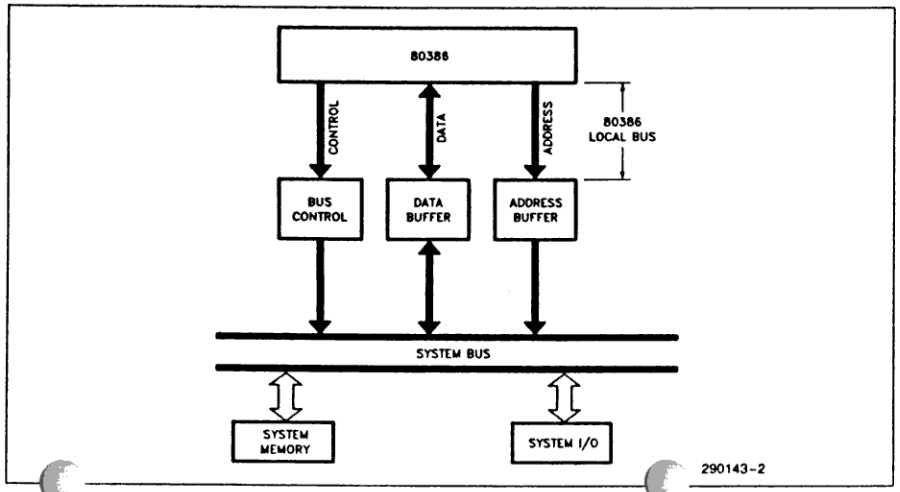


Figure 1-1. 80386 System Bus Structure

1.2.1 80386 Local Bus/82385 Local Bus/System Bus

Figure 1-1 depicts the bus structure of a typical 80386 system. The "80386 Local Bus" consists of the physical 80386 address, data, and control busses. The local address and data busses are buffered and/or latched to become the "system" address and data busses. The local control bus is decoded by bus control logic to generate the various system bus read and write commands.

The addition of an 82385 Cache Controller causes a separation of the 80386 bus into two distinct busses: the actual 80386 local bus and the "82385 Local Bus" (Figure 1-2). The 82385 local bus is designed to look like the front end of an 80386 by providing 82385 local bus equivalents to all appropriate 80386 signals. The system ties to this "80386-like" front end just as it would to an actual 80386. The 80386 simply sees a fast system bus, and the system sees an 80386 front end with low bus bandwidth requirements. The cache subsystem is transparent to both. Note that the 82385 local bus is not simply a buffered version of the 80386 bus, but rather is distinct from, and able to operate in parallel with the 80386 bus. Other masters residing on either the 82385 local bus or system bus are free to manage system resources while the 80386 operates out of its cache.

1.2.2 Bus Arbitration

The 82385 presents the "80386-like" interface which is called the 82385 local bus. Whereas the 80386 provides a Hold Request/Hold Acknowledge bus arbitration mechanism via its HOLD and HLDA pins, the 82385 provides an equivalent mechanism via its BHOLD and BHLDA pins. (These signals are described in section 3.7.) When another master requests the 82385 local bus, it issues the request to the 82385 via BHOLD. Typically, at the end of the current 82385 local bus cycle, the 82385 will release the 82385 local bus and acknowledge the request via BHLDA. The 80386 is of course free to continue operating on the 80386 local bus while another master owns the 82385 local bus.

1.2.3 Master/Slave Operation

The above 82385 local bus arbitration discussion is strictly true only when the 82385 is programmed for "Master" mode operation. The user can, however, configure the 82385 for "Slave" mode operation. (Programming is done via a hardware strap option.) The roles of BHOLD and BHLDA are reversed for an 82385 in slave mode; BHOLD is now an output indicating a request to control the bus, and BHLDA is an input indicating that a request has been granted. An 82385 programmed in slave mode drives the 82385 local bus only when it has requested and subsequently been granted bus control. This allows multiple 80386/82385 subsystems to reside on the same 82385 local bus (Figure 1-3).

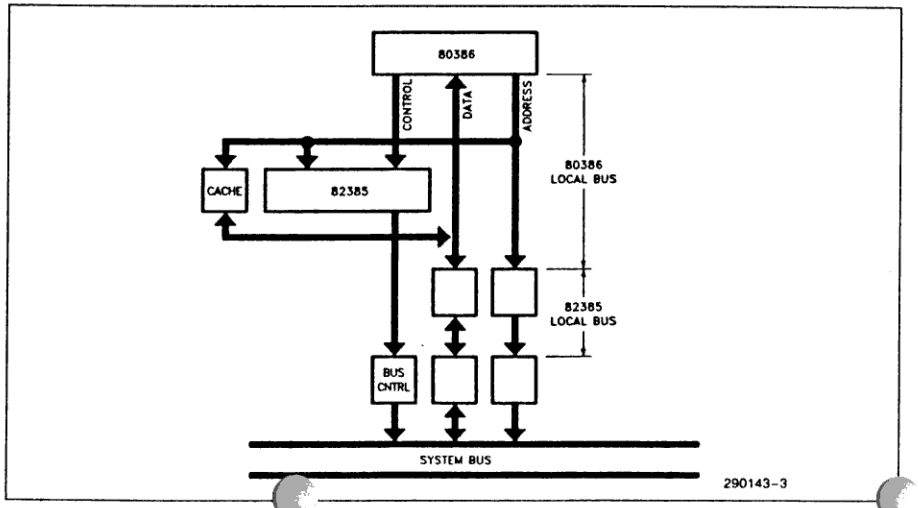


Figure 1-2. 80386/82385 System Bus Structure

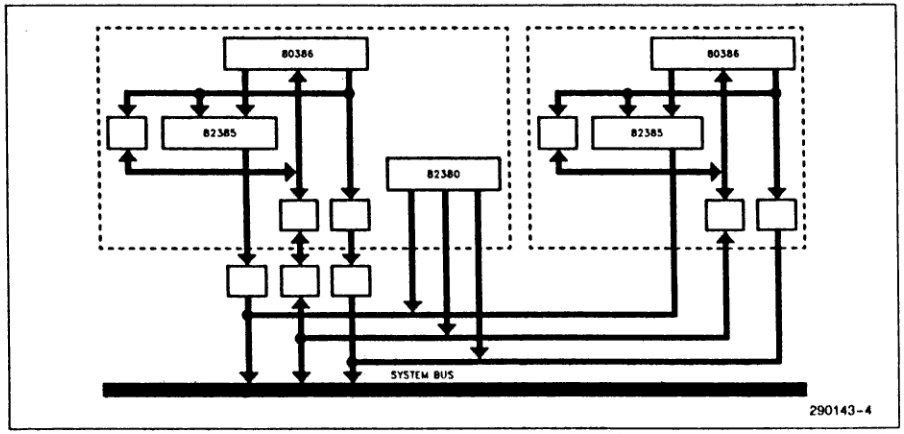


Figure 1-3. Multi-Master/Multi-Cache Environment

1.2.4 Cache Coherency

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

The 82385 preserves cache coherency via "bus watching" (also called snooping), a technique that neither impacts performance nor restricts memory mapping. An 82385 that is not currently bus master monitors system bus cycles, and when a write cycle by another master is detected (a snoop), the system address is sampled and used to see if the referenced location is duplicated in the cache. If so (a snoop hit), the corresponding cache entry is invalidated, which will force the 80386 to fetch the up-to-date data from main memory the next time it accesses this modified location. Figure 1-4 depicts the general form of bus watching.

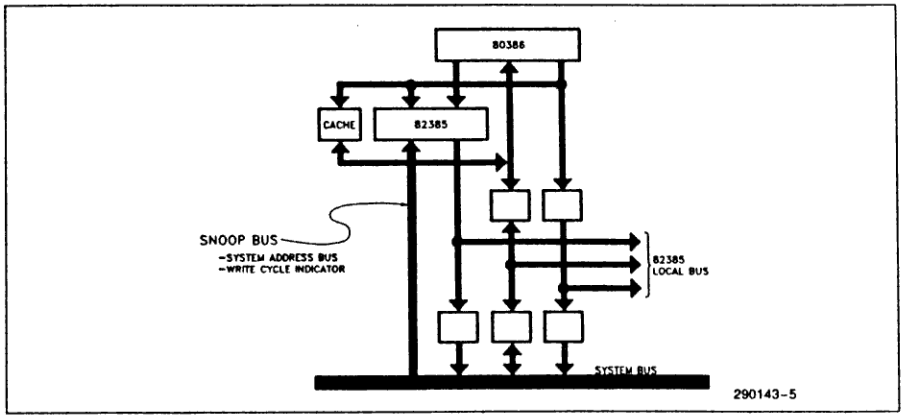


Figure 1-4. 82385 Bus Watching—Monitor System Bus Write Cycles

1.3 SYSTEM OVERVIEW II: BASIC OPERATION

This discussion is an overview of the basic operation of an 80386/82385 system. Items discussed include the 82385's response to all 80386 cycles, including interrupt acknowledges, halts, and shutdowns. Also discussed are non-cacheable and local accesses.

1.3.1 80386 Memory Code and Data Read Cycles

1.3.1.1 READ HITS

When the 80386 initiates a memory code or data read cycle, the 82385 compares the high order bits of the 80386 address bus with the appropriate addresses (tags) stored in its on-chip directory. (The directory structure is described in chapter 2.) If the 82385 determines that the requested data is in the cache, it issues the appropriate control signals that direct the cache to drive the requested data onto the 80386 data bus, where it is read by the 80386. The 82385 terminates the 80386 cycle without inserting any wait states.

1.3.1.2 READ MISSES

If the 82385 determines that the requested data is not in the cache, the request is forwarded to the 82385 local bus and the data retrieved from main memory. As the data returns from main memory, it is directed to the 80386 and also written into the cache. Concurrently, the 82385 updates the cache directory such that the next time this particular piece of information is requested by the 80386, the 82385 will find it in the cache and return it with zero wait states.

The basic unit of transfer between main memory and cache memory in a cache subsystem is called the line size. In an 82385 system, the line size is one 32-bit aligned doubleword. During a read miss, all four 82385 local bus byte enables are active. This ensures that a full 32-bit entry is written into the cache. (The 80386 simply ignores what it did not request.) In any other type of 80386 cycle that is forwarded to the 82385 local bus, the logic levels of the 80386 byte enables are duplicated on the 82385 local bus.

The 82385 does not actively fetch main memory data independently of the 80386. The 82385 is essentially a passive device which only monitors the address bus and activates control signals. The read miss is the only mechanism by which main memory data is copied into the cache and validated in the cache directory.

In an isolated read miss, the number of wait states seen by the 80386 is that required by the system memory to respond with data plus the cache comparison cycle (hit/miss decision). The cache system must determine that the cycle is a miss before it can begin the system memory access. However, since misses most often occur consecutively, the 82385 will begin 80386 address pipelined cycles to effectively "hide" the comparison cycle beyond the first miss (refer to section 4.1.3).

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

1.3.2 80386 Memory Write Cycles

The 82385's "posted write" capability allows the majority of 80386 memory write cycles to run with zero wait states. The primary memory update policy implemented in a posted write is the traditional cache "write through" technique, which implies that main memory is always updated in any memory write cycle. If the referenced location also happens to reside in the cache (a write hit), the cache is updated as well.

Beyond this, a posted write latches the 80386 address, data, and cycle definition signals, and the 80386 local bus cycle is terminated without any wait states, even though the corresponding 82385 local bus cycle is not yet completed, or perhaps not even started. A posted write is possible because the 82385's bus state machine, which is almost identical to the 80386 bus state machine, is able to run 82385 local bus cycles independently of the 80386. The only time the 80386 sees wait states in a write cycle is when a previously latched write has not yet been completed on the 82385 local bus. An 80386 write can be posted even if the 82385 does not currently own the 82385 local bus. In this case, an 82385 in master mode will run the cycle as soon as the current master releases the bus, and an 82385 in slave mode will request the bus and run the cycle when the request is acknowledged. The 80386 is free to continue operating out of its cache (on the 80386 local bus) during this time.

1.3.3 Non-Cacheable Cycles

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

Table 1-1. 82385 Response to 80386 Cycles

80386 Bus Cycle Definition					82385 Response when Decoded as Cacheable			82385 Response when Decoded as Non-Cacheable			82385 Response when Decoded as an 80386 Local Bus Access		
M/IO#	D/C#	W/R#	80386 Cycle		Cache	Cache Directory	82385 Local Bus	Cache	Cache Directory	82385 Local Bus	Cache	Cache Directory	82385 Local Bus
0	0	0	INT ACK	N/A	—	—	INT ACK	—	—	INT ACK	—	—	IDLE
0	0	1	UNDEFINED	N/A			UNDEFINED			UNDEFINED			IDLE
0	1	0	I/O READ	N/A	—	—	I/O READ	—	—	I/O READ	—	—	IDLE
0	1	1	I/O WRITE	N/A	—	—	I/O WRITE	—	—	I/O WRITE	—	—	IDLE
1	0	0	MEM CODE READ	HIT	CACHE READ	—	IDLE	—	—	MEM CODE READ	—	—	IDLE
				MISS	CACHE WRITE	DATA VALIDATION	MEM CODE READ						
1	0	1	HALT/ SHUTDOWN	N/A	—	—	HALT/ SHUTDOWN	—	—	HALT/ SHUTDOWN	—	—	IDLE
1	1	0	MEM DATA READ	HIT	CACHE READ	—	IDLE	—	—	MEM DATA READ	—	—	IDLE
				MISS	CACHE WRITE	DATA VALIDATION	MEM DATA READ						
1	1	1	MEM DATA WRITE	HIT	CACHE WRITE	—	MEM DATA WRITE	—	—	MEM DATA WRITE	—	—	IDLE
				MISS	—	—	MEM DATA WRITE						

NOTES:
- A dash (—) indicates that the cache and cache directory are unaffected. This table does not reflect how an access affects the LRU bit.
- An "IDLE" 82385 Local Bus implies that this bus is available to other masters.
- The 82385's response to 80387 accesses is the same as when decoded as an 80386 Local Bus access.
- The only other operations that affect the cache directory are:
  1. RESET or Cache Flush—all tag valid bits cleared.
  2. Snoop Hit—corresponding line valid bit cleared.

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

The 82385 allows the system designer to define areas of main memory as non-cacheable. The 80386 address bus is decoded and the decode output is connected to the 82385's non-cacheable access (NCA#) input. This decoding is done in the first 80386 bus state in which the non-cacheable cycle address becomes available. Non-cacheable read cycles resemble cacheable read miss cycles, except that the cache and cache directory are unaffected. Non-cacheable writes, like all writes, are posted.

The 82385 defines certain cycles as non-cacheable without using its non-cacheable access input. These include I/O cycles, interrupt acknowledge cycles, and halt/shutdown cycles. I/O reads and interrupt acknowledge cycles execute as any other non-cacheable read. I/O write cycles and Halt/Shutdown cycles, as with other non-cacheable writes, are posted. During a halt/shutdown condition, the 82385 local bus duplicates the behavior of the 80386, including the ability to recognize and respond to a BHOLD request. (The 82385's bus watching mechanism is functional in this condition.)

1.3.3.1 16-BIT MEMORY SPACE

The 82385 does not cache 16-bit memory space (as decoded by the 80386 BS16# input), but does make provisions to handle 16-bit space as non-cacheable. (There is no 82385 equivalent to the 80386 BS16# input.) In a system without an 82385, the 80386 BS16# input need not be asserted until the last state of a 16-bit cycle for the 80386 to recognize it as such (unless NA# is sampled active earlier in the cycle.) The 82385, however, needs this information earlier, specifically at the end of the first 80386 bus state in which the address of the 16-bit cycle becomes available. The result is that in a system without an 82385, 16-bit devices can inform the 80386 that they are 16-bit devices "on the fly," while in a system with an 82385, devices decoded as 16-bit (using the 80386 BS16#) must be located in address space set aside for 16-bit devices. If 16-bit space is decoded according to 82385 guidelines (as described later in the data sheet), then the 82385 will handle 16-bit cycles just like the 80386 does, including effectively locking the two halves of a non-aligned 16-bit transfer from interruption by another master.

1.3.4 80386 Local Bus Cycles

80386 Local Bus Cycles are accesses to resources on the 80386 local bus rather than to the 82385 itself. The 82385 simply ignores these accesses: they are neither forwarded to the system nor do they affect the cache. The designer sets aside memory and/or I/O space for local resources by decoding the 80386 address bus and feeding the decode to the 82385's local bus access (LBA#) input. The designer can also decode the 80386 cycle definition signals to keep specific 80386 cycles from being forwarded to the system. For example, a multi-processor design may wish to capture and remedy an 80386 shutdown locally without having it detected by the rest of the system. Note that in such a design, the local shutdown cycle must be terminated by local bus control logic. The 80387 Numerics Coprocessor is considered an 80386 local bus resource, but it need not be decoded as such by the user since the 82385 is able to internally recognize 80387 accesses via the M/IO# and A31 pins.

1.3.5 Summary of 82385 Response to All 80386 Cycles

Table 1-1 summarizes the 82385 response to all 80386 bus cycles, as conditioned by whether or not the cycle is decoded as local or non-cacheable. The table describes the impact of each cycle on the cache and on the cache directory, and whether or not the cycle is forwarded to the 82385 local bus. Whenever the 82385 local bus is marked "IDLE", it implies that this bus is available to other masters.

1.3.6 Bus Watching

As previously discussed, the 82385 "qualifies" an 80386 bus cycle in the first bus state in which the address and cycle definition signals of the cycle become available. The cycle is qualified as read or write, cacheable or non-cacheable, etc. Cacheable cycles are further classified as hit or miss according to the results of the cache comparison, which accesses the 82385 directory and compares the appropriate directory location (tag) to the current 80386 address. If the cycle turns out to be non-cacheable or a 80386 local bus access, the hit/miss decision is ignored. The cycle qualification requires one 80386 state. Since the fastest 80386 access is two states, the second state can be used for bus watching.

When the 82385 does not own the system bus, it monitors system bus cycles. If another master writes into main memory, the 82385 latches the system address and executes a cache look-up to see if the altered main memory location resides in the cache. If so (a snoop hit), the cache entry is marked invalid in the cache directory. Since the directory is at most only being used every other state to qualify 80386 accesses, snoop look-ups are interleaved between 80386 local bus look-ups. The cache directory is time multiplexed between the 80386 address and the latched system address. The result is that all snoops are caught and serviced without slowing down the 80386, even when running zero wait state hits on the 80386 local bus.

1.3.7 Cache Flush

The 82385 offers a cache flush input. When activated, this signal causes the 82385 to invalidate all data which had previously been cached. Specifically, all tag valid bits are cleared. (Refer to the 82385 directory structure in chapter 2.) Therefore, the cache is effectively empty and subsequent cycles are misses until the 80386 begins repeating the new accesses (hits). The primary use of the FLUSH input is for diagnostics and multi-processor support.

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

2.0 82385 CACHE ORGANIZATION

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

2.1 DIRECT MAPPED CACHE

2.1.1 Direct Mapped Cache Structure and Terminology

Figure 2-1 depicts the relationship between the 82385's internal cache directory, the external cache memory, and the 80386's 4 Gigabyte physical address space. The 4 Gigabytes can conceptually be thought of as cache "pages" each being 8K doublewords (32 Kbytes) deep. The page size matches the cache size. The cache can be further divided into 1024 (0 thru 1023) sets of eight doublewords (8 x 32 bits). Each 32-bit doubleword is called a "line." The unit of transfer between the main memory and cache is one line.

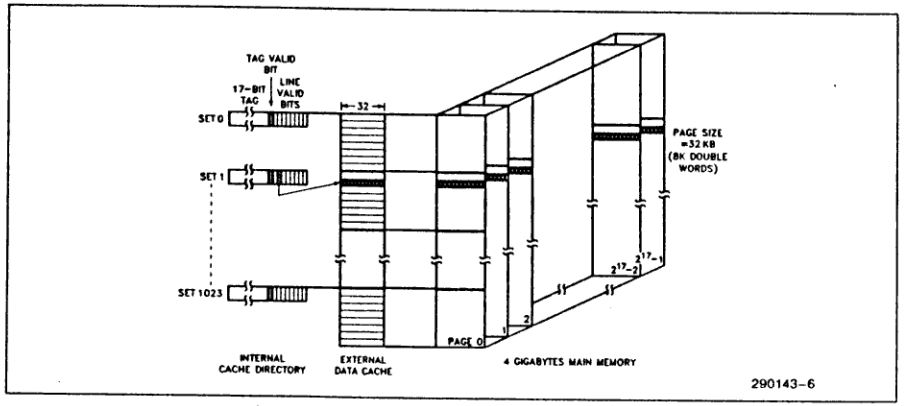


Figure 2-1. Direct Mapped Cache Organization

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

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

The 82385 sees the 80386 address bus (A2–A31) as partitioned into three fields: a 17-bit "tag" field (A15–A31), a 10-bit "set-address" field (A5–A14), and a 3-bit "line select" field (A2–A4). (See Figure 2-2.) The lower 13 address bits (A2–A14) also serve as the "cache address" which directly selects one of 8K doublewords in the external cache.

2.1.2 Direct Mapped Cache Operation

The following is a description of the interaction between the 80386, cache, and cache directory.

2.1.2.1 READ HITS

When the 80386 initiates a memory read cycle, the 82385 uses the 10-bit set address to select one of 1024 directory entries, and the 3-bit line select field to select one of eight line valid bits within the entry. The 13-bit cache address selects the corresponding doubleword in the cache. The 82385 compares the 17-bit tag field (A15–A31 of the 80386 access) with the tag stored in the selected directory entry. If the tag and upper address bits match, and if both the tag and appropriate line valid bits are set, the result is a hit, and the 82385 directs the cache to drive the selected doubleword onto the 80386 data bus. A read hit does not alter the contents of the cache or directory.

2.1.2.2 READ MISSES

A read miss can occur in two ways. The first is known as a "line" miss, and occurs when the tag and upper address bits match and the tag valid bit is set, but the line valid bit is clear. The second is called a "tag" miss, and occurs when either the tag and upper address bits do not match, or the tag valid bit is clear. (The line valid bit is a "don't care" in a tag miss.) In both cases, the 82385 forwards the 80386 reference to the system, and as the returning data is fed to the 80386, it is written into the cache and validated in the cache directory.

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

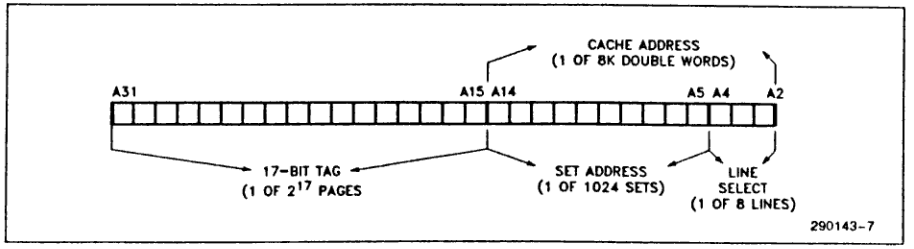


Figure 2-2. 80386 Address Bus Bit Fields—Direct Mapped Organization

2.1.2.3 OTHER OPERATIONS THAT AFFECT THE CACHE AND CACHE DIRECTORY

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

When an 80386/82385 system "wakes up" upon reset, all tag valid bits are clear. At this point, a read miss is the only mechanism by which main memory data is copied into the cache and validated in the cache directory. Assume an early 80386 code access seeks (for the first time) line 9 of page 2. Since the tag valid bit is clear, the access is a tag miss, and the data is fetched from main memory. Upon return, the data is fed to the 80386 and simultaneously written into line 9 of the cache. The set directory entry is updated to show this line as valid. Specifically, the tag and appropriate line valid bits are set, the remaining seven line valid bits cleared, and a binary 2 written into the tag. Since code is sequential in nature, the 80386 will likely next want line 10 of page 2, then line 11, and so on. If the 80386 sequentially fetches the next six lines, these fetches will be line misses, and as each is fetched from main memory and written into the cache, its corresponding line valid bit is set. This is the basic flow of events that fills the cache with valid data. Only after a piece of data has been copied into the cache and validated can it be accessed in a zero wait state read hit. Also, a cache entry must have been validated before it can be subsequently altered by a write hit, or invalidated by a snoop hit.

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

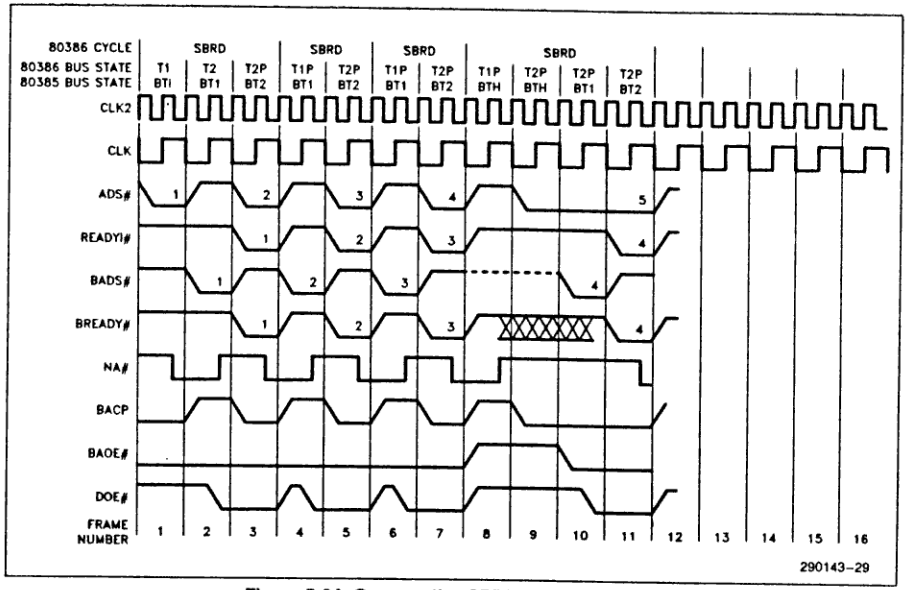


Figure 5-3A. Consecutive SBRD Cycles—(N = 0)

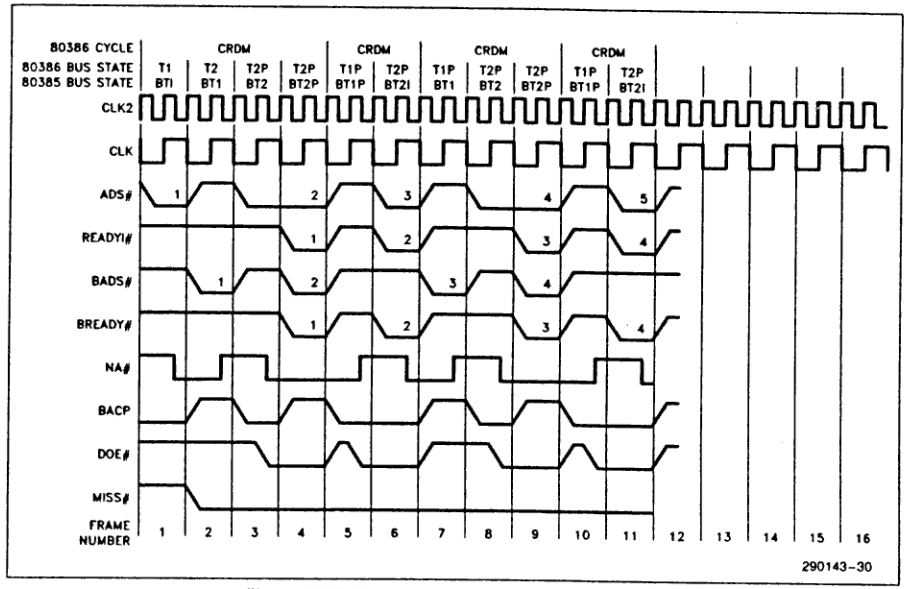


Figure 5-3B. Consecutive CRDM Cycles—(N = 1)

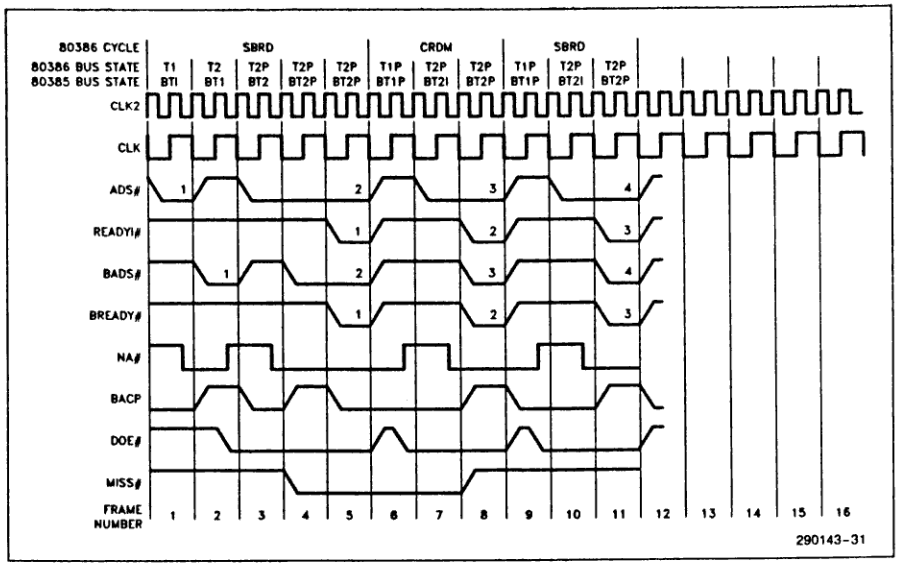


Figure 5-3C. SBRD, CRDM, SBRD—(N = 2)

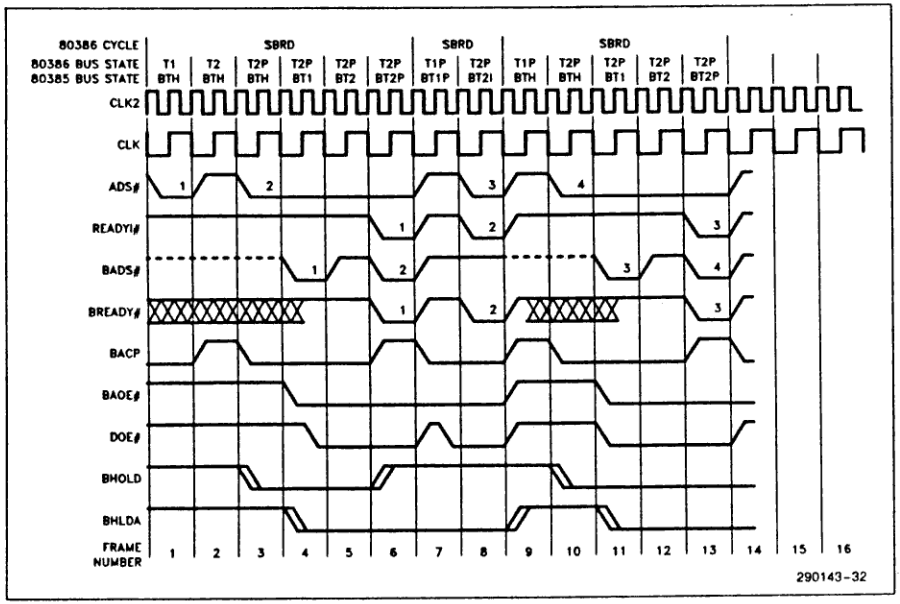


Figure 5-3D. SBRD Cycles Interleaved with BTH States—(N = 1)

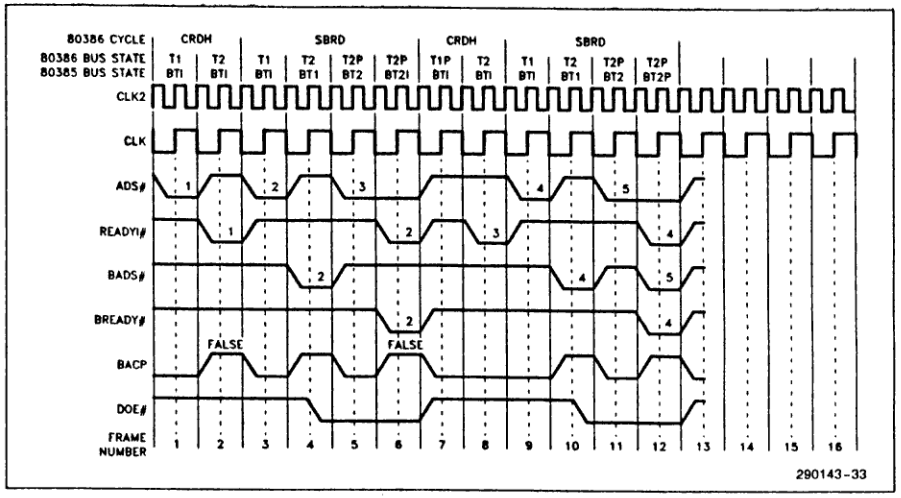


Figure 5-3E. Interleaved SBRD/CRDH Cycles—(N = 1)

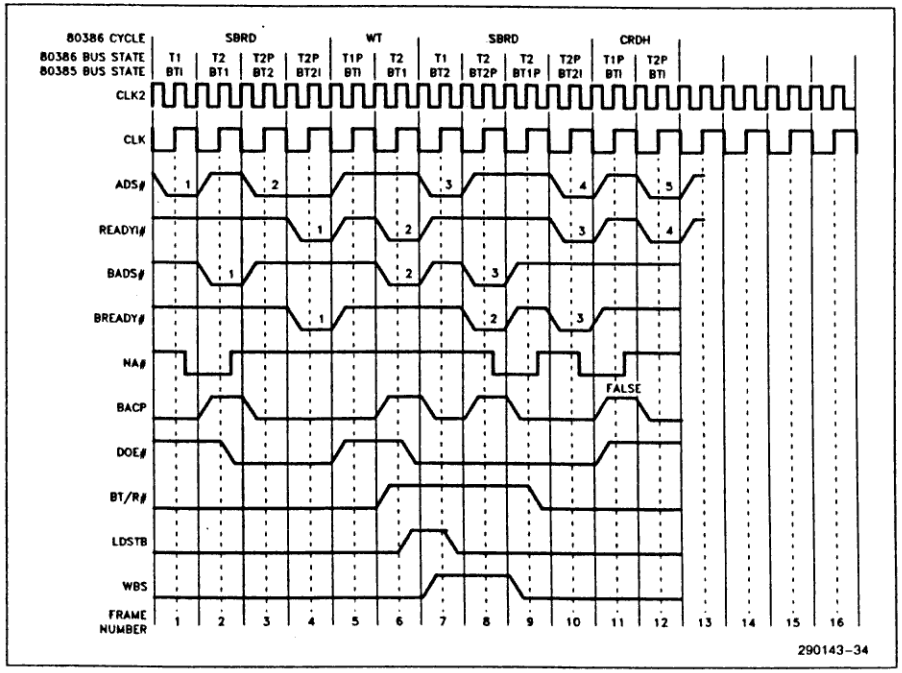


Figure 5-3F. SBRD, WT, SBRD, CRDH—(N = 1)

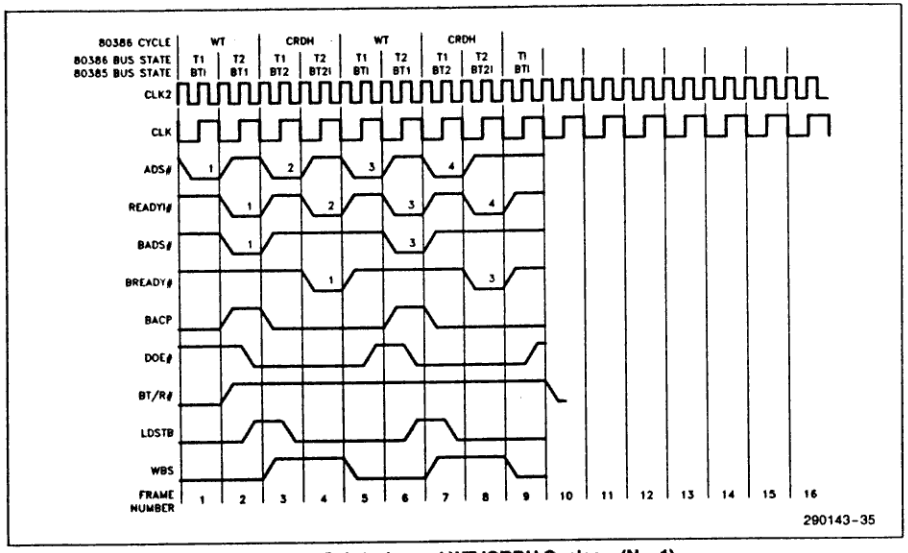


Figure 5-3G. Interleaved WT/CRDH Cycles—(N = 1)

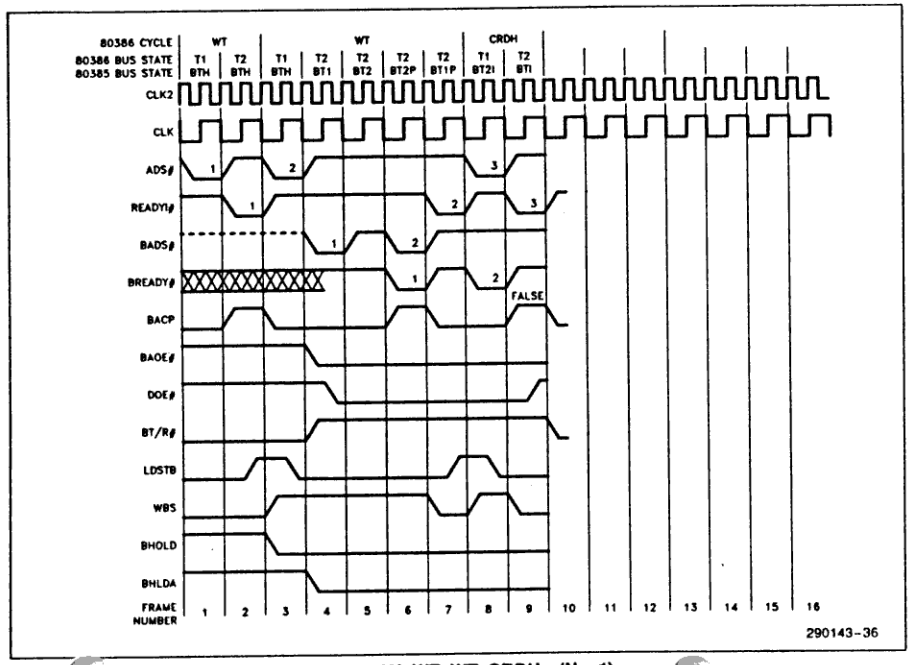


Figure 5-3H. WT, WT, CRDH—(N = 1)

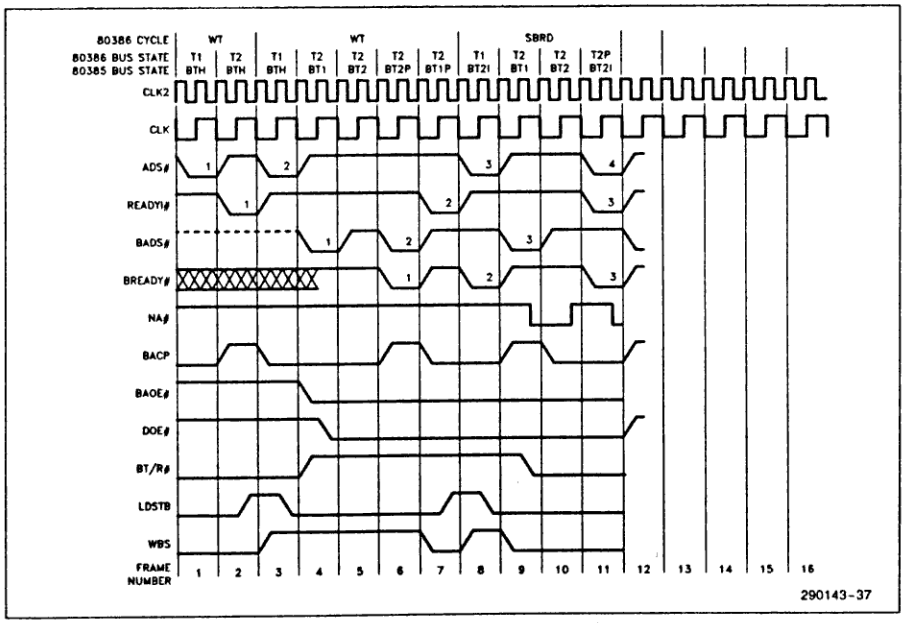


Figure 5-3I. WT, WT, SBRD—(N = 1)

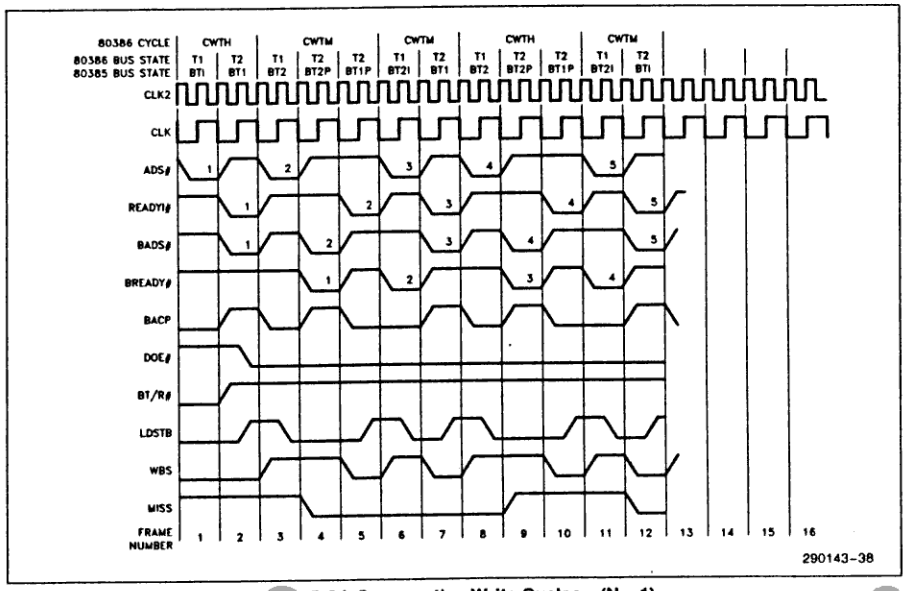


Figure 5-3J. Consecutive Write Cycles—(N = 1)

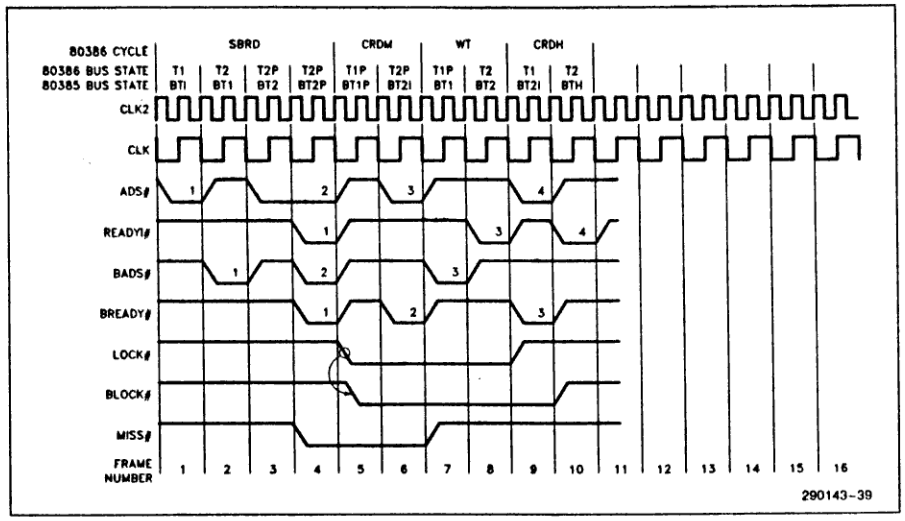


Figure 5-3K. LOCK# /BLOCK# in Non-Cacheable or Miss Cycles—(N = 1)

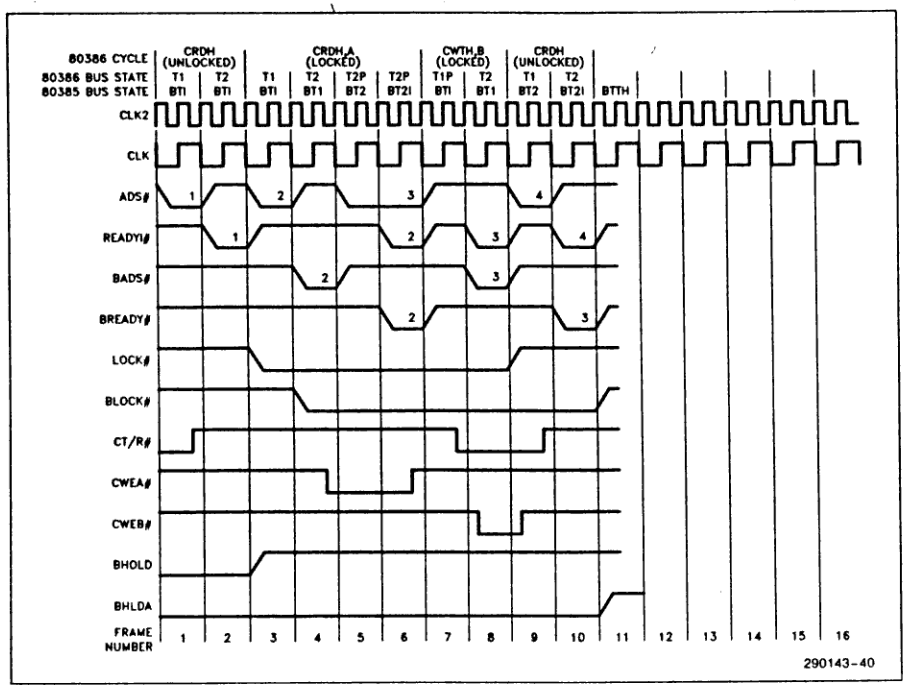


Figure 5-3L. LOCK# /BLOCK# in Cache Read Hit Cycle—(N = 1)

Table 8-1. 82385 PGA Pinout—Functional Grouping

Pin/Signal		Pin/Signal		Pin/Signal		Pin/Signal	
M2	A31	C12	SA31	C1	V_{CC}	B1	V_{SS}
L3	A30	A13	SA30	C14	V_{CC}	B14	V_{SS}
L2	A29	B12	SA29	M1	V_{CC}	M14	V_{SS}
K3	A28	A12	SA28	N13	V_{CC}	N1	V_{SS}
L1	A27	C11	SA27	P1	V_{CC}	N2	V_{SS}
J3	A26	B11	SA26	P3	V_{CC}	N14	V_{SS}
K2	A25	C10	SA25	P11	V_{CC}	P2	V_{SS}
K1	A24	B10	SA24	P13	V_{CC}	P4	V_{SS}
J2	A23	A11	SA23	E13	ADS#	P12	V_{SS}
J1	A22	C9	SA22			P14	V_{SS}
H2	A21	A10	SA21	F14	W/R#		
H3	A20	A9	SA20	F12	D/C#	N9	BADS#
H1	A19	B9	SA19	E14	M/IO#	M12	BBE0#
G1	A18	C8	SA18	F13	LOCK#	N12	BBE1#
G2	A17	A8	SA17			L12	BBE2#
G3	A16	B8	SA16	N3	NA#	M13	BBE3#
F1	A15	B7	SA15			P9	BLOCK#
F2	A14	A7	SA14	G13	X16#		
F3	A13	A6	SA13	G12	NCA#	K14	BNA#
E1	A12	C7	SA12	H14	LBA#		
E2	A11	B6	SA11	D14	READYI#	N4	CALEN
E3	A10	B5	SA10	M3	READYO#	P7	COEA#
D1	A9	A5	SA9			M7	COEB#
D2	A8	C6	SA8	E12	FLUSH	N7	CWEA#
C2	A7	A4	SA7	M8	WBS	P6	CWEB#
A1	A6	C5	SA6	N8	MISS#	M5	CS0#
D3	A5	B4	SA5			M6	CS1#
C3	A4	C4	SA4	D12	2W/D#	N6	CS2#
B2	A3	A3	SA3	B13	M/S#	P5	CS3#
B3	A2	A2	SA2	M10	DOE#		
G14	BE0#	J12	SEN	M4	LDSTB	N5	CT/R#
H13	BE1#	J13	SSTB#				
H12	BE2#			N11	BHOLD	P8	BRDYEN#
J14	BE3#	A14	RESERVED	M11	BHLDA	K13	BREADY#
		L14	RESERVED			P10	BACP
C13	CLK2					M9	BAOE#
D13	RESET					N10	BT/R#
K12	BRESET						
L13	BCLK2						

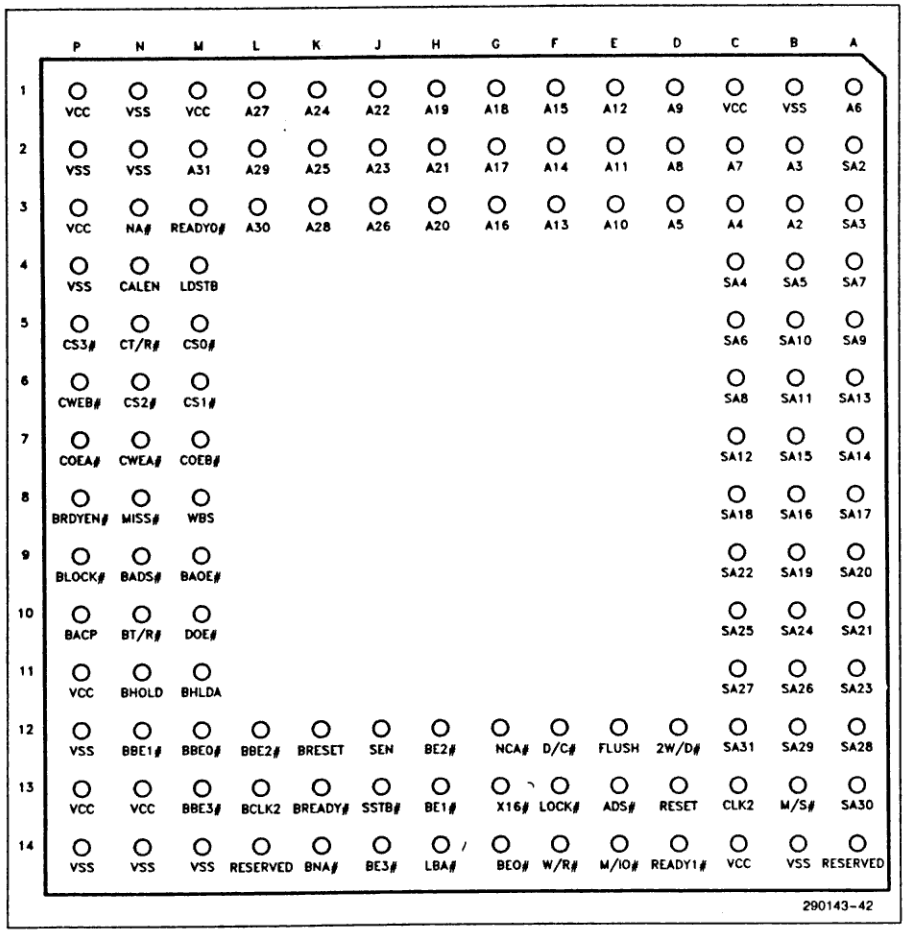


Figure 8-1. 82385 PGA Pinout—View from TOP Side

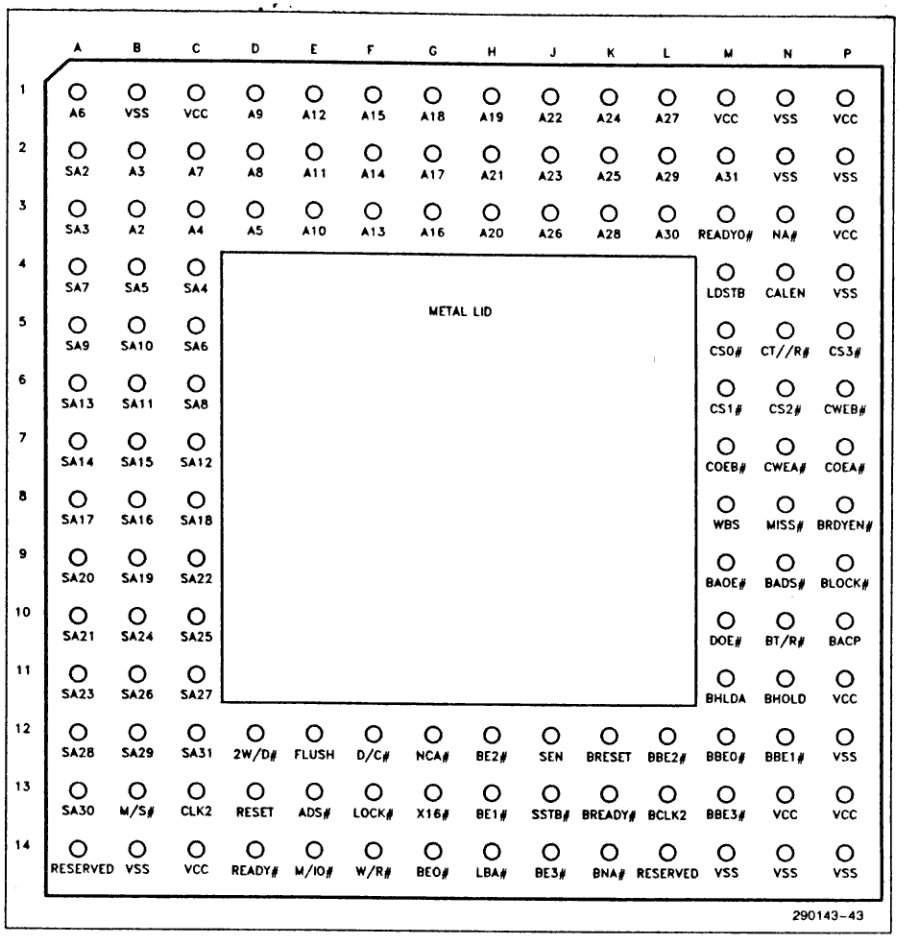


Figure 8-2. 82385 PGA Pinout—View from PIN Side

APPENDIX

82385 Signal Summary

Signal Group/Name	Signal Function	Active State	Input/ Output	Tri-State Output?
80386 INTERFACE				
RESET	386 Reset	High	I	—
A2–A31	386 Address Bus	High	I	—
BE0#–BE3#	386 Byte Enables	Low	I	—
CLK2	386 Clock	—	I	—
READYO#	Ready Output	Low	O	No
BRDYEN#	Bus Ready Enable	Low	O	No
READYI#	386 Ready Input	Low	I	—
ADS#	386 Address Status	Low	I	—
M/IO#	386 Memory / I/O Indication	—	I	—
W/R#	386 Write/Read Indication	—	I	—
D/C#	386 Data/Control Indication	—	I	—
LOCK#	386 Lock Indication	Low	I	—
NA#	386 Next Address Request	Low	O	No
CACHE CONTROL				
CALEN	Cache Address Latch Enable	High	O	No
CT/R#	Cache Transmit/Receive	—	O	No
CS0#–CS3#	Cache Chip Selects	Low	O	No
COEA#, COEB#	Cache Output Enables	Low	O	No
CWEA#, CWEB#	Cache Write Enables	Low	O	No
LOCAL DECODE				
LBA#	386 Local Bus Access	Low	I	—
NCA#	Non-Cacheable Access	Low	I	—
X16#	16-Bit Access	Low	I	—
STATUS AND CONTROL				
MISS#	Cache Miss Indication	Low	O	Yes
WBS	Write Buffer Status	High	O	No
FLUSH	Cache Flush	High	I	—
82385 INTERFACE				
BREADY#	385 Ready Input	Low	I	—
BNA#	385 Next Address Request	Low	I	—
BLOCK#	385 Lock Indication	Low	O	Yes
BADS#	385 Address Status	Low	O	Yes
BBE0#–BBE3#	385 Byte Enables	Low	O	Yes

82385 Signal Summary (Continued)

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

Table 5-1. Pin State During RESET and Initialization

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

NOTE:
1. In Master Mode, BAOE# is low and BACP emits a pulse stream during reset. As the 80386 address and cycle definition

3.0 82385 PIN DESCRIPTION

The 82385 creates the 82385 local bus, which is a functional 80386 interface. To facilitate understanding, 82385 local bus signals go by the same name as their 80386 equivalents, except that they are preceded by the letter "B". The 82385 local bus equivalent to ADS# is BADS#, the equivalent to NA# is BNA#, etc. This convention applies to bus states as well. For example, BT1P is the 82385 local bus state equivalent to the 80386 T1P state.

3.1 80386/82385 INTERFACE SIGNALS

These signals form the direct interface between the 80386 and 82385.

3.1.1 80386/82385 Clock (CLK2)

CLK2 provides the fundamental timing for an 80386/82385 system, and is driven by the same source that drives the 80386 CLK2 input. The 82385, like the 80386, divides CLK2 by two to generate an internal "phase indication" clock. (See Figure 3-1.) The CLK2 period whose rising edge drives the internal clock low is called PHI1, and the CLK2 period that drives the internal clock high is called PHI2. A PHI1-PHI2 combination (in that order) is known as a "T" state, and is the basis for 80386 bus cycles.

3.1.2 80386/82385 Reset (RESET)

This input resets the 82385, bringing it to an initial known state, and is driven by the same source that drives the 80386 RESET input. A reset effectively flushes the cache by clearing all cache directory tag valid bits. The falling edge of RESET is synchronized to CLK2, and used by the 82385 to properly establish the phase of its internal clock. (See Figure 3-2.) Specifically, the second internal phase following the falling edge of RESET is PHI2.

3.1.3 80386/82385 Address Bus (A2–A31), Byte Enables (BE0#–BE3#), and Cycle Definition Signals (M/IO#, D/C#, W/R#, LOCK#)

The 82385 directly connects to these 80386 outputs. The 80386 address bus is used in the cache directory comparison to see if data referenced by 80386 resides in the cache, and the byte enables inform the 82385 as to which portions of the data bus are involved in an 80386 cycle. The cycle definition signals are decoded by the 82385 to determine the type of cycle the 80386 is executing.

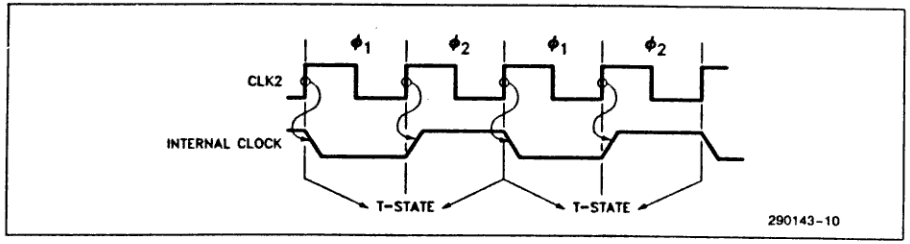


Figure 3-1. CLK2 and Internal Clock

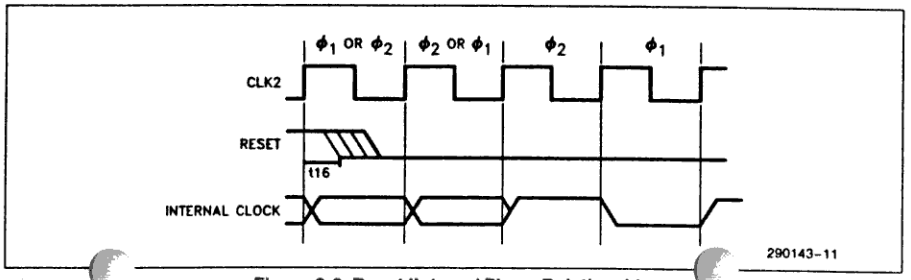


Figure 3-2. Reset/Internal Phase Relationship

3.1.4 80386/82385 Address Status (ADS#) and Ready Input (READYI#)

ADS# is an 80386 output, and tells the 82385 that new address and cycle definition information is available. READYI# is an input to both the 80386 (via the 80386 READY# input pin) and 82385, that indicates the completion of an 80386 bus cycle. ADS# and READYI# are used to keep track of the 80386 bus state.

3.1.5 80386 Next Address Request (NA#)

This 82385 output controls the pipelining of the 80386. It can be tied directly to the 80386 NA# input, or it can be logically "AND"ed with other 80386 local bus next address requests.

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

The 82385 directly terminates all but two types of 80386 bus cycles with its READYO# output. 80386 local bus cycles must be terminated by the local device being accessed. This includes devices decoded using the 82385 LBA# signal and 80387 accesses.

The other cycles not directly terminated by the 82385 are 82385 local bus reads, specifically cache read misses and non-cacheable reads. (Recall that the 82385 forwards and runs such cycles on the 82385 bus.) In these cycles the signal that terminates the 82385 local bus access is BREADY#, which is gated through to the 80386 local bus such that the 80386 and 82385 local bus cycles are concurrently terminated. BRDYEN# is used to gate the BREADY# signal to the 80386.

3.2 CACHE CONTROL SIGNALS

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

3.2.1 Cache Address Latch Enable (CALEN)

This signal controls the latch (typically an F or AS series 74373) that resides between the low order 80386 address bits and the cache SRAM address inputs. (The outputs of this latch are the "cache address" described in the previous chapter.) When CALEN is high the latch is transparent. The falling edge of CALEN latches the current outputs which remain applied to the cache data memory until CALEN returns to an active high state.

3.2.2 Cache Transmit/Receive (CT/R#)

This signal defines the direction of an optional data transceiver (typically an F or AS series 74245) between the cache and 80386 data bus. When high, the transceiver is pointed towards the 80386 local data bus (the SRAMs are output enabled). When low, the transceiver points towards the cache data memory. A transceiver is required if the cache is designed with SRAMs that lack an output enable control. A transceiver may also be desirable in a system that has a heavily loaded 80386 local data bus. These devices are not necessary when using SRAMs which incorporate an output enable.

3.2.3 Cache Chip Selects (CS0#–CS3#)

These active low signals tie to the cache SRAM chip selects, and individually enable the four bytes of the 32-bit wide cache. CS0# enables D0–D7, CS1# enables D8–D15, CS2# enables D16–D23, and CS3# enables D24–D31. During read hits, all four bytes are enabled regardless of whether or not all four 80386 byte enables are active. (The 80386 ignores what it did not request.) Also, all four cache bytes are enabled in a read miss so as to update the cache with a complete line (double word). In a write hit, only those cache bytes that correspond to active byte enables are selected. This prevents cache data from being corrupted in a partial doubleword write.

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

COEA# and COEB# are active low signals which tie to the cache SRAM output enables and respectively enable cache bank A or B to drive the 80386 data bus. In a two-way set associative cache, either COEA# or COEB# is active during a read hit, depending on which bank is selected. In a direct mapped cache, both are activated, so the designer is free to use either one.

CWEA# and CWEB# are active low signals which tie to the cache SRAM write enables, and respectively enable cache bank A or B to receive data from the 80386 data bus (80386 write hit or read miss update). In a two-way set associative cache, one or the other is enabled in a read miss or write hit. In a direct mapped cache, both are activated, so the designer is free to use either one.

If the cache is implemented with SRAMs that do not have output enables, then a transceiver between the cache memory and 80386 data bus is required. In this case, the output enable of each bank must be "AND"ed with the corresponding write enable to provide the transceiver enable signal. For example, COEA# and CWEA# are "AND"ed to enable the transceiver between cache bank A and the 80386 data bus (chapter 4, Figures 4-4B and 4-4D). The various cache configurations supported by the 82385 are described in chapter 4.

3.3 80386 LOCAL BUS DECODE INPUTS

These 82385 inputs are generated by decoding the 80386 address and cycle definition lines. These active low inputs are sampled at the end of the first state in which the address of a new 80386 cycle becomes available (T1 or first T2P).

3.3.1 80386 Local Bus Access (LBA#)

This input identifies an 80386 access as directed to a resource (other than the cache) on the 80386 local bus. (The 80387 Numerics Coprocessor is considered an 80386 local bus resource, but LBA# need not be generated as the 82385 internally decodes 80387 accesses.) The 82385 simply ignores these cycles. They are neither forwarded to the system nor do they affect the cache or cache directory. Note that LBA# has priority over all other types of cycles. If LBA# is asserted, the cycle is interpreted as an 80386 local bus access, regardless of the cycle type or status of NCA# or X16#. This allows any 80386 cycle (memory, I/O, interrupt acknowledge, etc.) to be kept on the 80386 local bus if desired.

3.3.2 Non-Cacheable Access (NCA#)

This active low input identifies an 80386 cycle as non-cacheable. The 82385 forwards non-cacheable cycles to the 82385 local bus and runs them. The cache and cache directory are unaffected.

NCA# allows a designer to set aside a portion of main memory as non-cacheable. Potential applications include memory-mapped I/O and systems where multiple masters access dual ported memory via different busses. Another possibility makes use of the 80386 D/C# output. The 82385 by default implements a unified code and data cache, but driving NCA# directly by D/C# creates a data only cache. If D/C# is inverted first, the result is a code only cache.

3.3.3 16-Bit Access (X16#)

X16# is an active low input which identifies 16-bit memory and/or I/O space, and the decoded signal that drives X16# should also drive the 80386 BS16# input. 16-bit accesses are treated like non-cacheable accesses: they are forwarded to and executed on the 82385 local bus with no impact on the cache or cache directory. In addition, the 82385 locks the two halves of a non-aligned 16-bit transfer from interruption by another master, as does the 80386.

3.4 82385 LOCAL BUS INTERFACE SIGNALS

The 82385 presents an "80386-like" front end to the system, and the signals discussed in this section are 82385 local bus equivalents to actual 80386 signals. These signals are named with respect to their 80386 counterparts, but with the letter "B" appended to the front.

Note that the 82385 itself does not have equivalent output signals to the 80386 data bus (D0–D31), address bus (A2–A31), and cycle definition signals (M/IO#, D/C#, W/R#). The 82385 data bus (BD0–BD31) is actually the system side of a latching transceiver, and the 82385 address bus and cycle definition signals (BA2–BA31, BM/IO#, BD/C#, BW/R#) are the outputs of an edge-triggered latch. The signals that control this data transceiver and address latch are discussed in section 3.5.

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

BBE0#–BBE3# are the 82385 local bus equivalents to the 80386 byte enables. In a cache read miss, the 82385 drives all four signals low, regardless of whether or not all four 80386 byte enables are active. This ensures that a complete line (doubleword) is fetched from main memory for the cache update. In all other 82385 local bus cycles, the 82385 duplicates the logic levels of the 80386 byte enables. The 82385 tri-states these outputs when it is not the current bus master.

3.4.2 82385 Bus Lock (BLOCK#)

BLOCK# is the 82385 local bus equivalent to the 80386 LOCK# output, and distinguishes between locked and unlocked cycles. When the 80386 runs a locked sequence of cycles (and LBA# is negated), the 82385 forwards and runs the sequence on the 82385 local bus, regardless of whether any locations

referenced in the sequence reside in the cache. A read hit will be run as if it is a read miss, but a write hit will update the cache as well as being completed to system memory. In keeping with 80386 behavior, the 82385 does not allow another master to interrupt the sequence. BLOCK# is tri-stated when the 82385 is not the current bus master.

3.4.3 82385 Bus Address Status (BADS#)

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

3.4.4 82385 Bus Ready Input (BREADY#)

82385 local bus cycles are terminated by BREADY#, just as 80386 cycles are terminated by the 80386 READY# input. In 82385 local bus read cycles, BREADY# is gated by BRDYEN# onto the 80386 local bus, such that it terminates both the 80386 and 82385 local bus cycles.

3.4.5 82385 Bus Next Address Request (BNA#)

BNA# is the 82385 local bus equivalent to the 80386 NA# input, and indicates that the system is prepared to accept a pipelined address and cycle definition. If BNA# is asserted and the new cycle information is available, the 82385 begins a pipelined cycle on the 82385 local bus.

3.5 82385 BUS DATA TRANSCEIVER AND ADDRESS LATCH CONTROL SIGNALS

The 82385 data bus is the system side of a latching transceiver (typically an F or AS series 74646), and the 82385 address bus and cycle definition signals are the outputs of an edge-triggered latch (F or AS series 74374). The following is a discussion of the 82385 outputs that control these devices. An important characteristic of these signals and the devices they control is that they ensure that BD0–BD31, BA2–BA31, BM/IO#, BD/C#, and BW/R# reproduce the functionality and timing behavior of their 80386 equivalents.

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

These signals control the latching data transceiver. BT/R# defines the transceiver direction. When high, the transceiver drives the 82385 data bus in write cycles. When low, the transceiver drives the 80386 data bus in 82385 local bus read cycles. DOE# enables the transceiver outputs.

The rising edge of LDSTB latches the 80386 data bus in all write cycles. The interaction of this signal and the latching transceiver is used to perform the 82385's posted write capability.

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

These signals control the latch that drives BA2–BA31, BM/IO#, BW/R#, and BD/C#. In any 80386 cycle that is forwarded to the 82385 local bus, the rising edge of BACP latches the 80386 address and cycle definition signals. BAOE# enables the latch outputs when the 82385 is the current bus master and disables them otherwise.

3.6 STATUS AND CONTROL SIGNALS

3.6.1 Cache Miss Indication (MISS#)

This output accompanies cacheable read and write miss cycles. This signal transitions to its active low state when the 82385 determines that a cacheable 80386 access is a miss. Its timing behavior follows that of the 82385 local bus cycle definition signals (BM/IO#, BD/C#, BW/R#) so that it becomes available with BADS# in BT1 or the first BT2P. MISS# is floated when the 82385 does not own the bus, such that multiple 82385's can share the same node in multi-cache systems. (As discussed in Chapter 7, this signal also serves a reserved function in testing the 82385.)

3.6.2 Write Buffer Status (WBS)

The latching data transceiver is also known as the "posted write buffer." WBS indicates that this buffer contains data that has not yet been written to the system even though the 80386 may have begun its next cycle. It is activated when 80386 data is

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

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

3.6.3 Cache Flush (FLUSH)

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

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

3.7 BUS ARBITRATION SIGNALS (BHOLD AND BHLDA)

In master mode, BHOLD is an input that indicates a request by a slave device for bus ownership. The 82385 acknowledges this request via its BHLDA output. (These signals function identically to the 80386 HOLD and HLDA signals.)

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

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

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

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

These signals select the configurations supported by the 82385. They are hardware strap options and must not be changed dynamically. 2W/D# (2-Way/Direct Mapped Select) selects a two-way set associative cache when tied high, or a direct mapped cache when tied low. M/S# (Master/Slave Select) chooses between master mode (M/S# high) and slave mode (M/S# low).

4.0 80386 LOCAL BUS INTERFACE

The following is a detailed description of how the 82385 interfaces to the 80386 and to 80386 local bus resources. Items specifically addressed are the interfaces to the 80386, the cache SRAMs, and the 80387 Numerics Coprocessor.

The many timing diagrams in this and the next chapter provide insight into the dual pipelined bus structure of an 80386/82385 system. It's important to realize, however, that one need not know every possible cycle combination to use the 82385. The interface is simple, and the dual bus operation invisible to the 80386 and system. To facilitate discussion of the timing diagrams, several conventions have been adopted. Refer to Figure 4-2A, and note that 80386 bus cycles, 80386 bus states, and 82385 bus states are identified along the top. All states can be identified by the "frame numbers" along the bottom. The cycles in Figure 4-2A include a cache read hit (CRDH), a cache read miss (CRDM), and a write (WT). WT represents any write, cacheable or not. When necessary to distinguish cacheable writes, a write hit goes by CWTH and a write miss by CWTM. Non-cacheable system reads go by SBRD. Also, it is assumed that system bus pipelining occurs even though the BNA# signal is not shown. When the system pipeline begins is a function of the system bus controller.

80386 bus cycles can be tracked by ADS# and READYI#, and 82385 cycles by BADS# and BREADY#. These four signals are thus a natural choice to help track parallel bus activity. Note in the timing diagrams that 80386 cycles are numbered using ADS# and READYI#, and 82385 cycles using BADS# and BREADY#. For example, when the address of the first 80386 cycle becomes available, the corresponding assertion of ADS# is marked "1", and the READYI# pulse that terminates the cycle is marked "1" as well. Whenever an 80386 cycle is forwarded to the system, its number is forwared as well so that the corresponding 82385 bus cycle can be tracked by BADS# and BREADY#.

The "N" value in the timing diagrams is the assumed number of main memory wait states inserted in a non-pipelined 82386 bus cycle. For example, a non-pipelined access to N = 2 memory requires a total of four bus states, while a pipelined access requires three. (The pipeline advantage effectively hides one main memory wait state.)

A.C. SPECIFICATION TABLES

Functional Operating Range: V_{CC} = 5V ±5%; T_{CASE} = 0°C to +85°C

Table 9-2. A.C. Specifications

Symbol	Parameter	82385-16		82385-20		Units	Notes
		Min	Max	Min	Max		
t1	Operating Frequency	12	16	12	20	MHz	
t2	CLK2 Period	31.25	41.67	25	41.67	ns	
t3	CLK2 High Time	9		8		ns	
t4	CLK2 Low Time	9		8		ns	
t5	CLK2 Fall Time		8		8	ns	
t6	CLK2 Rise Time		8		8	ns	
t7	A(2–31), BE(0–3)#, Lock Setup Time	25		19		ns	(Note 1)
t8	A(2–31), BE(0–3)#, Lock Hold Time	3		3		ns	
t9	W/R#, M/IO#, D/C#, ADS# Setup Time	28		21		ns	(Note 1)
t10	W/R#, M/IO#, D/C#, ADS# Hold Time	5		5		ns	
t11	READYI# Setup	21		12		ns	(Note 1)
t12	READYI# Hold	4		4		ns	
t13	LBA#, NCA#, X16# Setup Time	16		10		ns	
t14	LBA#, NCA#, X16# Hold Time	4		4		ns	
t15	RESET, BRESET Setup	13		12		ns	
t16	RESET, BRESET Hold	4		4		ns	
t17	NA# Delay	15	42	15	34	ns	(Note 1) C_L = 25 pF
t18	READYO# Delay	4	31	4	28	ns	(Note 1) C_L = 25 pF
t19	BRDYEN# Delay	4	31	4	28	ns	C_L = 40 pF
t21a	CALEN Delay	3	25	3	19	ns	(Note 2) C_L = 40 pF
t21b	CALEN Rising Delay	3	38	3	33	ns	(Notes 1, 3)
t22a	CWEA#, CWEB# Delay	14	31	12	25	ns	(Notes 1, 4) C_L = 75 pF
t22b	CWEA#, CWEB# Pulse Width	40		30		ns	(Notes 1, 5)
t23	CS(0–3)# Delay	14	38	12	33	ns	(Notes 1, 6) C_L = 50 pF
t24	CT/R# Delay	14	38	12	33	ns	(Notes 1, 7) C_L = 75 pF
t25a	COEA#, COEB#, Falling Delay	1	24	1	18	ns	(Note 8) C_L = 75 pF
t25b	COEA#, COEB#, Falling Delay	1	30	1	23	ns	(Notes 1, 9)
t25c	COEA#, COEB#, Rising Delay	5	19	5	15	ns	(Note 10)
t26	CS(0–3)# Active to CWEA#, CWEB# Rising	40		30		ns	(Notes 1, 5)
t27	CWEA#, CWEB# Falling to CS(0–3)# Falling Delay	0		0		ns	

A.C. SPECIFICATION TABLES (Continued)

Functional Operating Range: V_{CC} = 5V ±5%; T_{CASE} = 0°C to +85°C

Table 9-2. A.C. Specifications (Continued)

Symbol	Parameter	82385-16		82385-20		Units	Notes
		Min	Max	Min	Max		
t28	CWEA#, CWEB# Rising to CALEN Rising and CS(0–3)# Falling Delay	0		0		ns	
t31	SA(2–31) Setup	25		19		ns	
t32	SA(2–31) Hold	3		3		ns	
t33	BADS# Valid Delay	6	38	6	28	ns	(Note 1) C_L = 75 pF
t34	BADS# Float Delay	6	35	6	30	ns	
t55	BLOCK#, BBE(0–3)# Valid Delay	4	36	4	30	ns	(Note 1) C_L = 75 pF
t56	MISS# Valid Delay	4	43	4	35	ns	(Note 1) C_L = 75 pF
t57	MISS#, BBE(0–3)#, BLOCK# Float Delay	4	40	4	32	ns	
t58	WBS Delay	4	45	4	37	ns	(Note 1) C_L = 75 pF
t35	BNA# Setup	11		9		ns	
t36	BNA# Hold	15		15		ns	
t37a	BREADY# Setup	31		26		ns	(Notes 1, 11)
t37b	BREADY# Setup	21		12		ns	(Note 12)
t38	BREADY# Hold	4		4		ns	
t40	BACP Delay	4	23	4	18	ns	C_L = 60 pF
t41	BAOE# Delay	4	23	4	18	ns	
t43a	BT/R#, DOE# Delay	2	25	2	17	ns	C_L = 50 pF
t43b	DOE# Rising Delay	4	24	4	17	ns	
t43c	LDSTB Delay	2	33	2	26	ns	C_L = 50 pF
t44	SEN, SSTB# Setup	15		11		ns	
t45	SEN, SSTB# Hold	5		5		ns	
t46	BHOLD Setup	26		17		ns	(Note 13)
t47	BHOLD Hold	5		5		ns	(Note 13)
t48	BHLDA Delay	6	33	5	28	ns	(Note 13) C_L = 75 pF
t49	BHLDA Setup	20		17		ns	(Note 14)

A.C. SPECIFICATION TABLES (Continued)

Functional Operating Range: V_{CC} = 5V ±5%; T_{CASE} = 0°C to +85°C

Table 9-2. A.C. Specifications (Continued)

Symbol	Parameter	82385-16		82385-20		Units	Notes
		Min	Max	Min	Max		
t50	BHLDA Hold	5		5		ns	(Note 14)
t51	BHOLD Delay	6	33	6	28	ns	(Note 14) C_L = 75 pF
t59	FLUSH Setup	21		16		ns	
t60	FLUSH Hold	5		5		ns	
t61	FLUSH Setup to RESET Low	31		26		ns	
t62	FLUSH Hold from RESET Low	31		26		ns	

NOTES:
1. Frequency dependent specifications.
2. All cycles except cache write hit. CALEN triggers by PHI2 in TIP state.
3. The end of cache write hit cycles. Triggered by PHI1.
4. CWE# transitions by PHI1 in cache write hit cycles. CWE# transitions by PHI2 in cache read miss cycles.
5. Used for cache data memory (SRAM) specifications.
6. In cache write hit cycles, CS(0-3)# transition high by PHI2 and low by PHI1. In cache read miss cycles, CS(0-3)# transition high by PHI1 and low by PHI2.
7. In cache write hit cycles, CT/R# transitions low by PHI2. In cache read hit cycles, CT/R# goes high by PHI2. In cache read miss cycles, CT/R# goes low by PHI1.
8. Direct mapped configuration.
9. Two way set associative configuration.
10. COE# switches high by PHI1 at the end of a cache read hit cycle.
11. Cache read miss cycles.
12. Non-cacheable read cycles and system write cycles.
13. Master mode configuration. BHOLD is an input and BHLDA is an output.
14. Slave mode configuration. BHOLD is an output and BHLDA is an input.

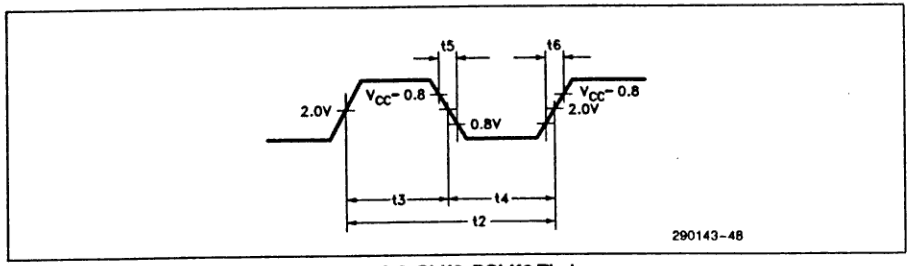


Figure 9-2. CLK2, BCLK2 Timing

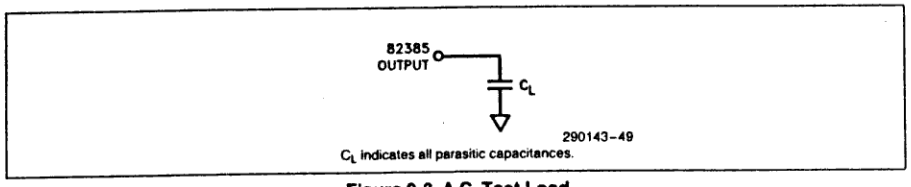
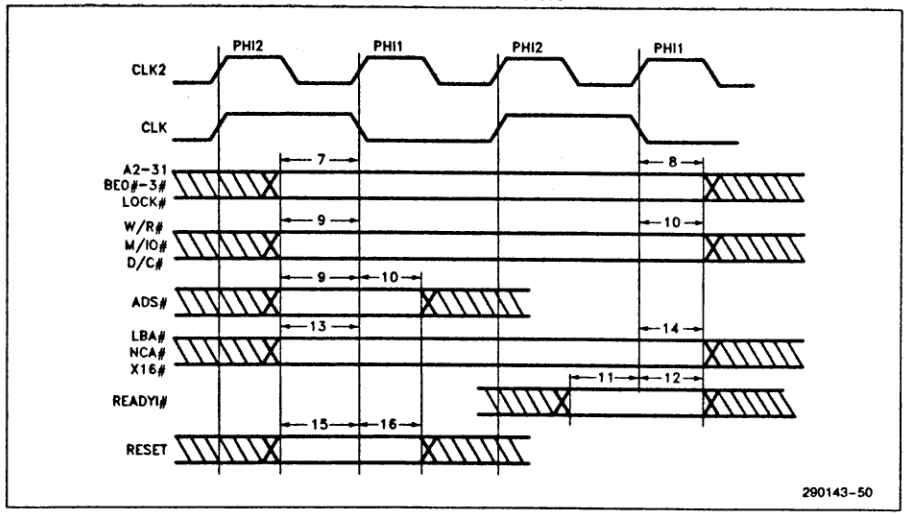
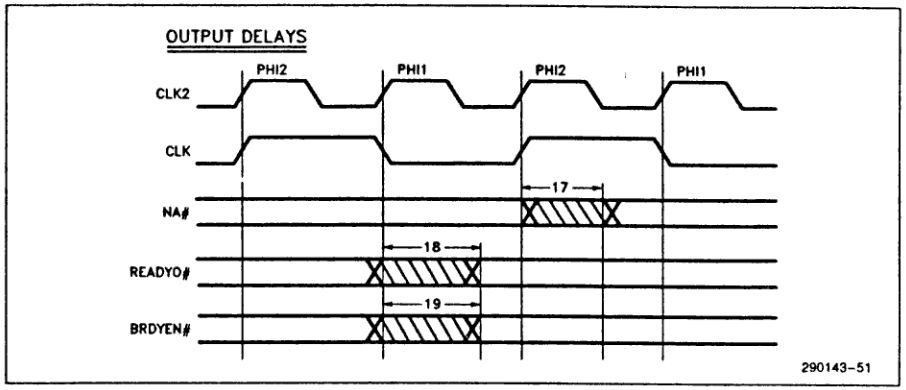


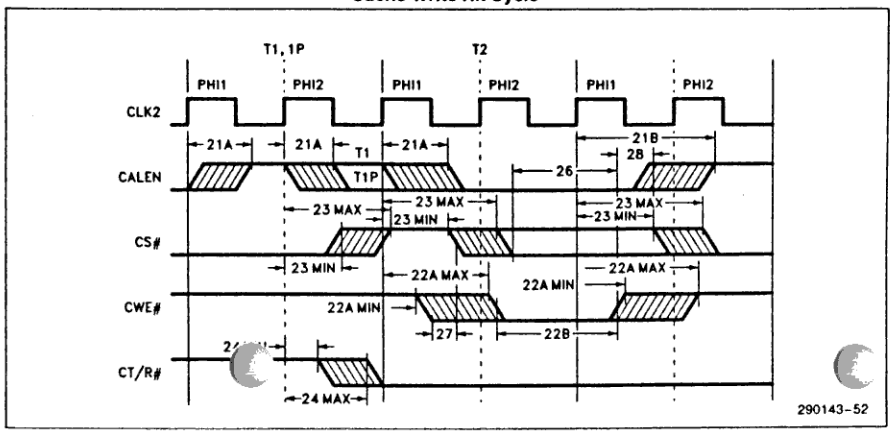
Figure 9-3. A.C. Test Load

386 Interface Parameters

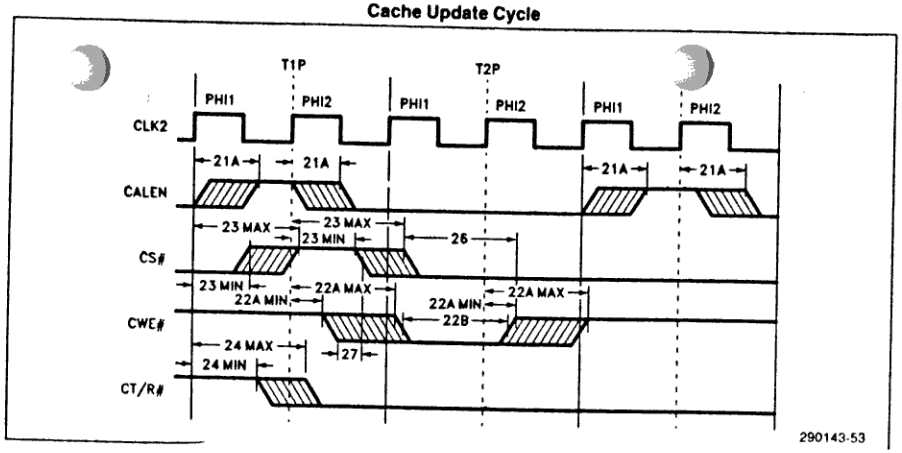




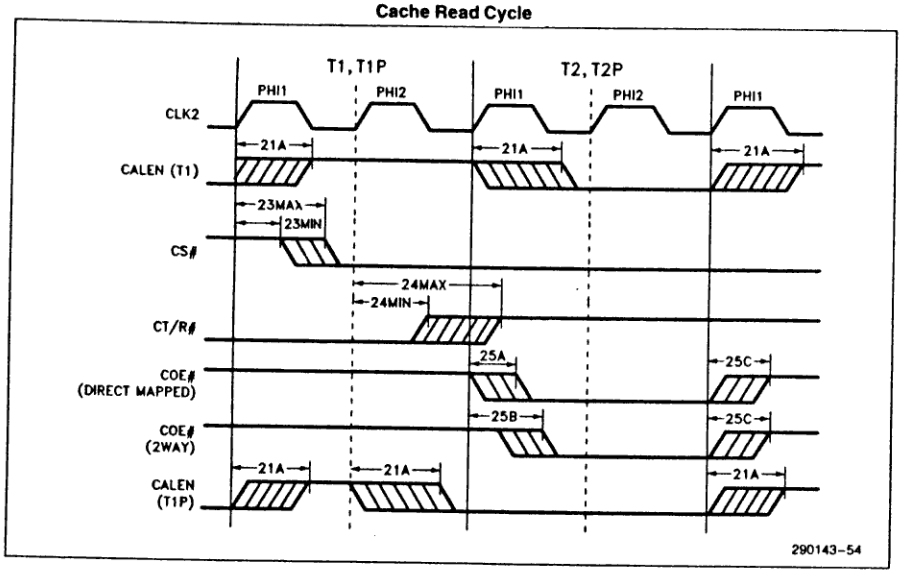
Cache Write Hit Cycle



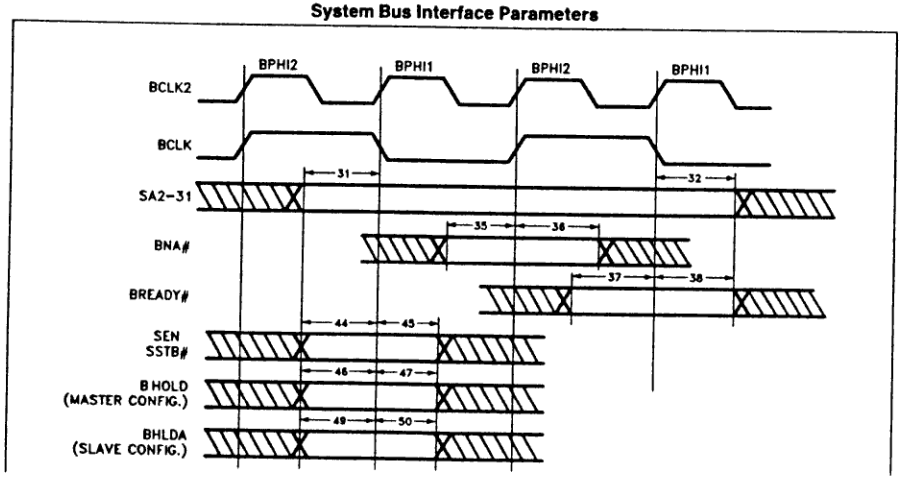
Cache Update Cycle



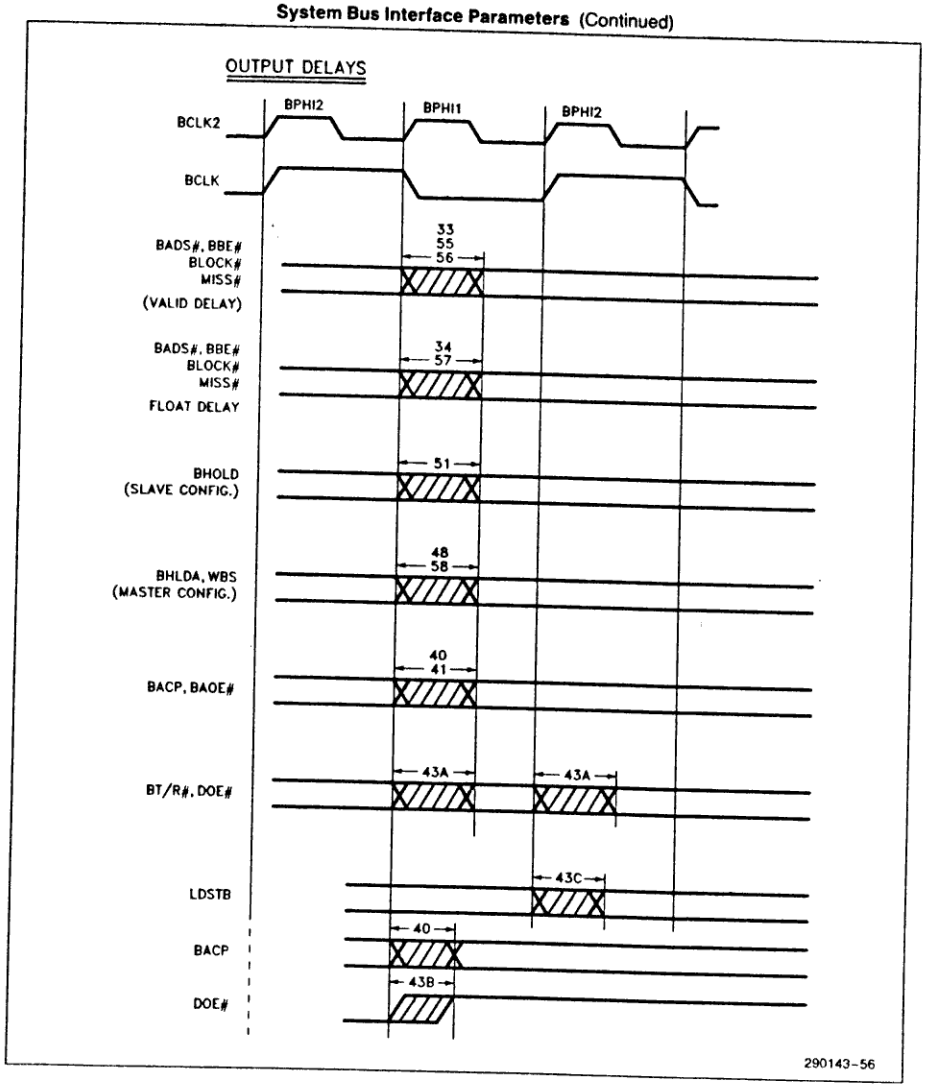
Cache Read Cycle



System Bus Interface Parameters



System Bus Interface Parameters (Continued)



Appendix B. CAM451 Components List

Table 1 below shows the components list and their associated parts numbers for CAM451.

Item	Code no	Description	Quantity					
1	116141	32-BIT CACHE CONTROLLER 20MHz	1	U30				
2	116142	32-BIT MICROPROCESSOR 20MHz	1	U29				
3	1319086	PGA SOCKET-ADAPTER 132.14	1	U32				
4	5132046	PAL OCTAL 16-INPUT AND-OR PLCC	1	U5				
5	5132050	PAL QUAD 16-INPUT REG. AND-OR PLC	1	U1				
6	5132054	PAL OCTAL 16-INPUT REG. AND-OR PLC	1	U22				
7	5152004	HEX INVERTER SOIC	1	U31				
8	5152011	TRIPLE 3 INPUT POSITIVE AND GATESOI	1	U18				
9	5152027	TRIPLE 3 INPUT POSITIVE NOR GATESOI	2	U7	U8			
10	5152030	8 INPUT NAND GATE SOIC	1	U6				
11	5152032	QUAD 2 INPUT POSITIVE OR GATES SOI	2	U2	U17			
12	5154015	STATIC RAM 8K * 8BIT 35ns SOIC	4	U11	U20	U24	U27	
14	5172373	OCTAL LATCH WITH 3-STATE OUTPUT	2	U3	U4			
15	5172374	OCTAL D-TYPE REG. 3-STATE OUTPUT	5	U9	U13	U14	U15	U16
16	5172646	8-BIT X-CEIVER TS	4	U12	U21	U25	U28	
17	5878052	CAP 47n 50DO SM01 p10 X7R CER.	11	C1	C2	C3	C4	C5
				C6	C8	C9	C11	C14
				C15				
18	6001016	ELLYT TANTAL SMD 15u/20V	4	C7	C10	C12	C13	
19	6104071	RES METF SM01 180E 5% .25W	5	R2	R3	R4	R5	R15
20	6104089	RES METF SM01 1K 5% .25W	2	R7	R8			
21	6104097	RES METF SM01 2K2 5% .25W	3	R6	R11	R13		
22	6104120	RES METF SM01 20K 5% .25W	5	R1	R9	R10	R12	R14

Table 1. CAM451 components list and associated parts numbers.

References

(1) CPU452 Hardware Reference Manual
Regnecentralen a-s.

(2) Intel Microprocessor and Peripheral Handbook.
Vol. I. Microprocessors.
Datasheets on i80386-20 and i82385-20.

(3) CAM451 Cache Memory Test Program Manual
Regnecentralen a-s.