82C836

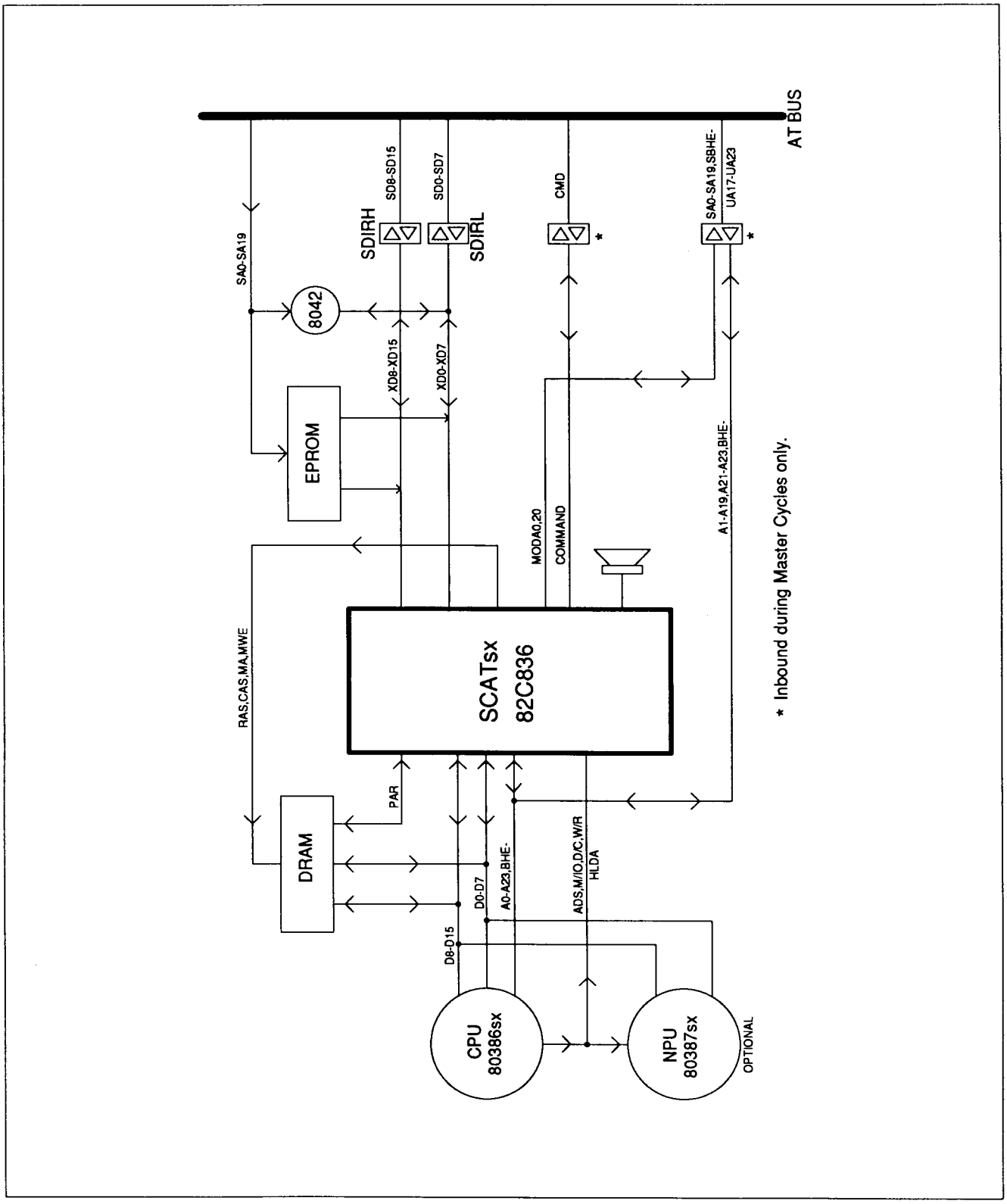
The 82C836, also known as SCATsx, is a VLSI device that incorporates most of the motherboard logic required to build a low cost, highly integrated, IBM PC AT compatible computer using the 80386sx. It is designed to be used in conjunction with other Chips and Technologies controllers such as the 82C45X VGA Controller, the 82C601 Multifunction Controller, the 82C765 Floppy Disk Controller, and the 82C710 Integrated Floppy Disk and Multifunction Controller. When used with these devices, the 82C836 acts as the heart of a highly integrated system that significantly reduces motherboard size, component count, and the need for many I/O channel slots. Figure 1 shows a block diagram of the basic system architecture.

Features

The 82C836 provides the following features:

- 80386sx control logic and clocks to support CPU speeds of up to 25 MHz with zero (or one) wait states
- A 146818-compatible real time clock with 114 bytes of CMOS RAM
- Two 8237-compatible DMA controllers
- Two 8259-compatible interrupt controllers
- An 8254-compatible programmable interval timer
- An 82284-compatible clock generation and READY interface
- An 82288-compatible bus controller
- A DRAM controller that supports up to 8 MB of DRAM (up to 16 MB with the addition of an external 74F538-type decoder)
- A memory controller that provides shadow RAM and support for either 8-bit or 16-bit BIOS ROM
- Support for fast local cache RAM via external cache controller
- A DRAM refresh controller
- Power management features
- Four EMS page registers (LIM EMS 4.0 and 3.2 compatible)
- Interface logic for an 80387sx numeric coprocessor
- Interface logic for an 8042 keyboard controller
- Fast Gate A20 and Fast CPU Reset logic
- Compact packaging in a single 160-pin plastic flat pack (160 PQFP)

Figure 1 *SCATsx Basic System Architecture*



Architectural Overview

The major address and data buses are described in Table 1. Various memory and I/O "resources" are accessible from these buses. I/O resources contained within the 82C836 include: Internal Configuration Registers (ICRs), two DMA controllers, two interrupt controllers, a Real-Time Clock (RTC) with CMOS RAM, timer registers and EMS page registers.

Table 1 *Address and Data buses*

82C836 Pins	Bus Name	Description
D15-D0	CPU Data	A 16-bit bidirectional bus for data transfer to or from the CPU. Also used during DMA and master cycles for data transfer to or from DRAM.
MD15-MD0	Memory Data	A 16-bit bidirectional bus for data transfer to or from local DRAM. Should be connected to the D-bus either directly or through bidirectional buffers, depending on system size and speed. Same as D15-D0 in Figure 1.
XD15-XD0	X-Bus	A 16-bit bidirectional bus for data transfer to or from on-board peripherals, ROM, or AT Bus (buffered).
SD15-SD0	AT Bus Data	The main 16-bit bidirectional bus for transferring data to or from add-on cards.
A23-A0	CPU Address	A 24-bit address bus, driven primarily by the CPU (80386sx). Driven by the 82C836 during DMA, and refresh cycles.
SA19-SA0	AT Bus Addr	The main 20-bit address bus for addressing I/O and memory resources on the AT bus.
UA23-UA17	Unlatch Addr	An unlatched address bus providing the high-order address bits for memory resources on the AT Bus.
MA9-MA0	Row/Col Addr	A multiplexed address bus driven by the 82C836 for DRAM row & column addressing.

Typical on-board I/O resources external to the 82C836 include: keyboard controller, optional numeric coprocessor, and optional Real-Time Clock. The SCATsx XD-Bus, subject to loading limitations, can also support an on-board video controller, floppy/hard disk controller, communications ports, parallel port, and/or game port normally residing on the AT Bus.

On-board memory resources external to the 82C836 include the local DRAM and BIOS EPROM.

In general, all memory and I/O resources, whether on-board or on the AT Bus, are either 16-bit or 8-bit resources. 16-bit resources support 16-bit data transfer on all 16 bits of the respective data bus, as well as 8-bit data transfer to or from an odd address on data bits 8-15, or 8-bit data transfer to or from an even address on data bits 0-7. Note that 16-bit resources, including the CPU itself, always use data bits 0-7 for even-addressed byte transfers and data bits 8-15 for odd-addressed byte transfers.

8-bit resources, in contrast, always use data bits 0-7, regardless of even or odd addresses, and can transfer only 8 bits at a time. This disparity between 8-bit resources and 16-bit resources gives rise to two special cases in data transfer:

Byte swapping — Whenever a 16-bit resource (or the CPU) transfers a data byte to or from an 8-bit resource at an odd address, the data on bits 8-15 for the 16-bit resource must be transferred to or from bits 0-7 for the 8-bit resource. The 82C836 performs this byte swapping as needed during CPU I/O or memory read/write cycles, DMA cycles, and Master cycles, including DMA or Master cycles in which both the data source and the data destination reside on the AT Bus.

Bus Conversion — Whenever the CPU attempts to transfer a 16-bit even-addressed word to or from an 8-bit resource, the 82C836 converts the 16-bit CPU cycle into two consecutive 8-bit cycles. The CPU is delayed by means of wait states until both 8-bit cycles have been completed. Bus conversion by the 82C836 can occur on either I/O or memory cycles, but only on CPU initiated cycles, not on DMA or Master cycles.

The list below summarizes all the possible cycle types in a SCATsx AT-compatible architecture.

- CPU initiated local cycles (no command on AT Bus)
  - Local DRAM read or write
  - Interrupt Acknowledge
  - Halt/Shutdown
- CPU initiated AT Bus Cycles (command generated on AT Bus)
  - On-board ROM read or write
  - On-board I/O read or write
  - AT Bus memory read or write
  - AT Bus I/O read or write
- DMA
  - Memory to I/O (simultaneous mem read and I/O write)
  - I/O to memory (simultaneous I/O read and mem write)
- Master
  - Memory read or write (may access on-board DRAM & ROM)
  - I/O read or write
- Refresh
  - Normal (system initiated)
  - Master initiated
- Additional variations on the above types
  - Resource: 8/16 bit, memory or I/O, on-board or AT Bus
  - Data transfer size: byte or word
  - Address: even or odd, first 1 MB or not
  - Delay or speed-up via IOCHRDY or 0WS

The major signal groups in a SCATsx AT-compatible architecture are listed below.

- AT Bus Interface Signals, 8-bit Extension
  - XD0-7, SDIRL (SD0-7)
  - A0-19, MODA0 (SA0-19)
  - HLDA (AEN), REFRESH-
  - ALE
  - LOMEGCS- (SMEMR-, SMEMW-)
  - XIOR-, XIOW-
  - IOCHRDY, 0WS-
  - IOCHCK-, IRQ3-7, IRQ9
  - DREQ1-3, DACK[1-3]-, TC

  - XRST (RESETDRV), BUSCLK, OSC2
- AT Bus Interface Signals, 16-bit Extension
  - XD8-15, SDIRH (SD8-15)
  - A17-23, MODA20, BHE-, MEMCS16-, IOCS16-
  - MASTER-
  - XMEMR-, XMEMW-
  - IRQ10-12, IRQ14-15
  - DREQ0, DREQ5-7, DACK[0,5-7]-
- Other 82C836 Signals
  - D0-15
  - ADS-, READY-, M/IO-, D/C-, W/R-, NA-
  - HOLD, INTR, NMI, PWRGOOD, CPURST
  - CXIN, PROCCLK, OSC1
  - MA0-9, RAS0-3, CASL-, CASH-, MWE-, PARL, PARH
  - ROMCS-, 8042CS-, RESET2-, 32KHZ/IRQ8, PS/RTCCS-, SPKOUT
  - NPBUSY-, NPERR-, BUSY-

Table 2 *Bus Ownership*

HLDA	AEN	MASTER-	REFRESH-	Bus Owner
L	L	H	H	System CPU
H	H	H	H	DMA Controller
H	L	L	H	Add-On Card Bus Master
H	H	H	L	Refresh, Initiated by System
H	L	L	L	Refresh, Initiated by Master

Note: Signals shown above are 82C836 signals. Certain closely related AT Bus equivalent signals are shown in parentheses in cases where the correlation between 82C836 and AT Bus signals may not be readily apparent.

Table 2 lists the key signals that signify bus ownership. Additional information on signal functions and timing relationships is contained in *System Timing Relationships* and *AC Characteristics* in this data book.

Note: While MASTER- is inactive, AEN should follow HLDA.
When MASTER- is active, AEN should be forced low.

Throughout this data sheet, a minus sign (-) at the end of a signal name signifies that the signal is active-low, i.e., "true" or "asserted" when low.

Functional Description

This section of the data sheet provides functional descriptions of the following features of the 82C836.

- Internal configuration registers (introduction)
- Reset strap options
- Battery back-up and power up/down
- Operational power management

- Clock generation
- System reset and clock synchronization
- Bus control, arbitration and basic timing
- Memory interface
- Numeric coprocessor interface
- Keyboard/mouse interface
- Real-time clock interface
- Programmable interval timer
- Interrupt controller
- DMA controller

Internal Configuration Registers (ICRs)

The 82C836 contains an extensive and versatile set of internal configuration registers for enabling or disabling various optional modes and features. The internal configuration registers are accessed using I/O ports 22H and 23H. To read or write an internal configuration register, the register number (index) should be output to port 22H, and the contents of the register then read or written at port 23H. Each read or write to port 23H must be preceded by an output to port 22H, even if the index value written to port 22H is unchanged from the last access.

The internal configuration registers are frequently mentioned throughout this data sheet and are discussed in detail in "Internal Configuration Registers." The abbreviation "ICR" is used throughout this data sheet to refer to internal configuration registers.

Reset Strap Options

The 82C836 features reset strap options without having to use extra pins. The 82C836 uses DACK- lines to implement the strap options. The DACK- lines are normally outputs, but during power-up they are used as input signals. These lines are sampled during the high state of XRST after PWRGOOD goes high. Later, they convert to outputs for normal functions. A 4.7K ohm pull-down or pull-up resistor pulls the signals low (L) or high (H) for strap option sampling. Table 3 describes the pins that are used for strap options.

Table 3 *Pins for Strap Options*

Signal	Strap Function	Description
DACK0-	SENSE0	Determines ICR 45H bit 0
DACK1-	SENSE1	Determines ICR 45H bit 1
DACK2-	VIDEO	Video on SD (H) or XD bus (L); also determines ICR 45H bit 2
DACK3-	NA/STCYC	NA- (H) or STCYC- (L) function
DACL5-	EXRTC-	Internal RTC (H) or external (L)
DACK6-	Reserved	(Should be pulled up)
DACK7-	ROM16-	8-bit ROM (H) or 16-bit ROM (L)

SENSE0 and SENSE1 are general purpose and can be used according to specific system requirements. DACK2- also can be used as a general purpose SENSE2 if ICR 44H is appropriately programmed to specify where the video controller resides. (See "Internal Configuration Registers" for further details.)

Battery Backup and Power-Up/Down

Most applications require the Real Time Clock to remain active whenever the system power is turned off. To accomplish this, the user must provide an alternate source of power to the 82C836. This alternate source of power is normally provided by connecting a battery to the V_{CC} supply pin of the device. A means should be provided for switching between the system power supply and the battery. A circuit such as the one shown in Figure 2 may be used to eliminate power drain on the battery when system power is available.

The circuit shown here allows reliable transitions between system and battery power without undue battery power drain. The battery voltage should not be high enough to cause a net increase in back-up voltage when normal power is turned off. Otherwise, the 82C836 and 32 KHz oscillator will continue to be powered by the battery even when the power supply is turned on, and battery drain will actually increase when power is on due to reverse conduction through the PNP transistor (2N3906).

A similar precaution is needed during V_{CC} ramp up or ramp down. The V_{CC} "changeover" threshold, determined by the 2.4K and 470 ohm resistors, should be high enough so that the 2N3906 cannot remain on or turn on when V_{CC} is less than about 4VDC. Otherwise, depending on battery voltage, the 2N3906 may again reverse conduct and load down the battery momentarily as V_{CC} is ramping up or down.

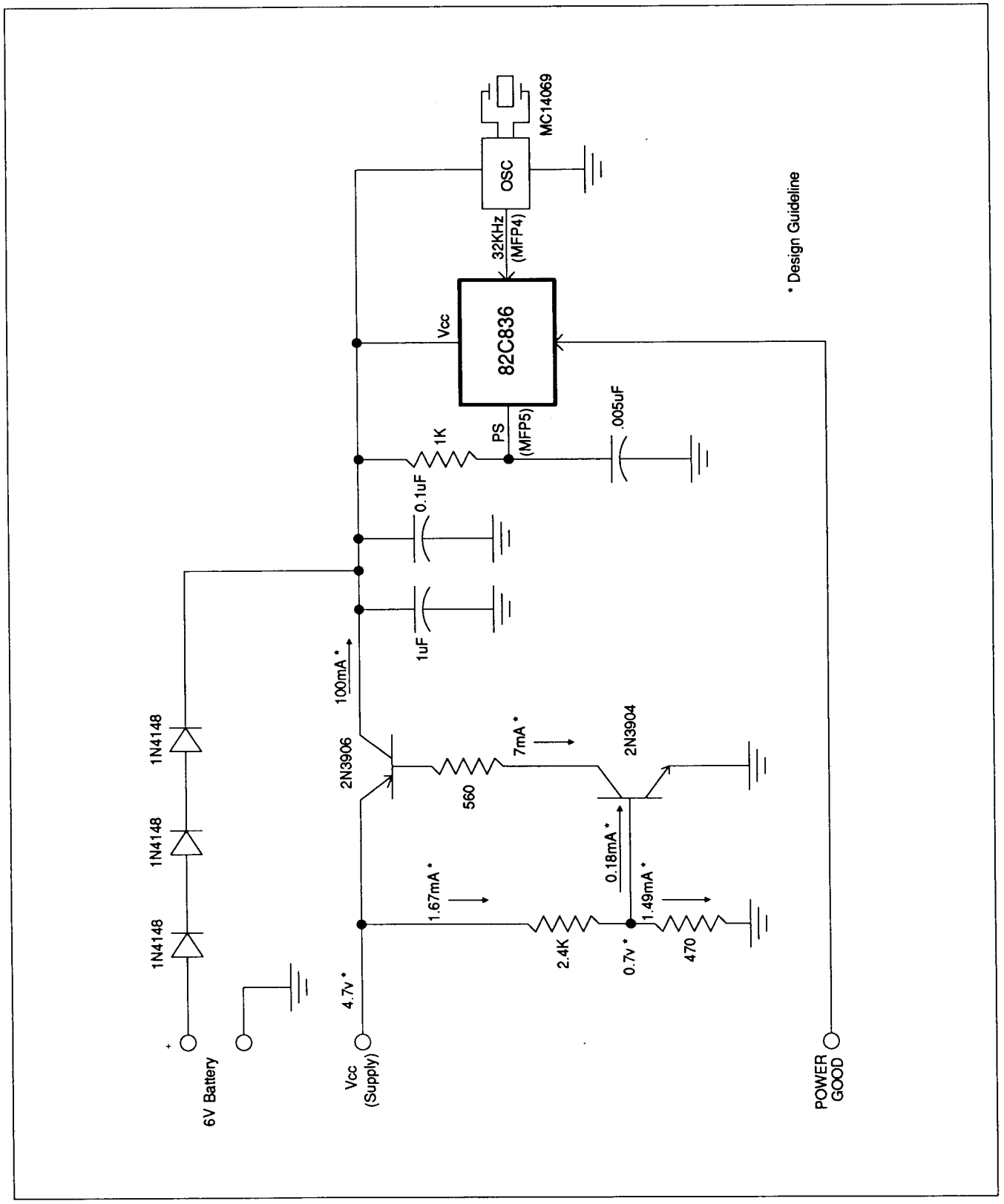
The user should also ensure that the Vin maximum specification for the 82C836 is never exceeded when powering the system up or down. Failure to observe this specification may result in damage to the device. The three diodes in series with the battery provide roughly 1.8 VDC voltage drop between the battery and the 82C836, as well as protecting the battery from possible explosion in the event of accidental polarity reversal. (Safety agencies generally require a minimum of two series diodes or equivalent.)

A pin is provided on the 82C836 to protect the contents of the Real Time Clock and reduce power consumption whenever the system is powered down. This pin (PWRGOOD) should be low whenever the system power supply is not within specifications for proper operation of the system. This signal may be generated by circuitry either in the power supply or on the system board. The PWRGOOD input disables all unnecessary inputs during the time the system is powered down to prevent noise on the inactive pins from causing increased ICC. This pin must therefore be active (high) for the remainder of the device to operate properly when system power is applied.

As in most AT-compatible architectures, the reset and power good functions normally available separately on an MC146818 are combined in the SCATsx. The Power Good input serves not only as a power down control pin, but also as a general hardware reset input. It does not reset the Real Time Clock.

Another pin is provided to initialize the Real Time Clock (RTC) whenever system or battery power is first applied to the 82C836. This pin (PS/MFP5) does not alter the CMOS RAM or Clock/Calendar contents, but it does initialize

Figure 2 *Battery Backup Circuit*



the necessary RTC control register bits. De-assertion of PS disables the generation of RTC interrupts and sets a flag indicating that the contents of the RTC may not be valid. A recommended circuit for controlling the PS input is also shown in Figure 2 above.

Testability

As already mentioned, all unnecessary inputs are disabled while PWRGOOD is low. In addition, all outputs except OSC2 are held in a high-impedance state. OSC2, the 14.3 MHz output driver, is driven to a solid high logic level (rev. 1 82C836) or solid low (later revisions). For in-circuit manufacturing test, the Power Sense input should also be pulsed low momentarily while PWRGOOD is low to insure that the 82C836 is fully reset.

Operational Power Management

Average system power consumption can be reduced by slowing or stopping the processor clock during idle periods. If a non-static CPU is used, the processor clock can be slowed down. If a static CPU is used, the processor clock can be stopped completely.

In the SCATsx, a "sleep" mode is provided in which a HALT instruction executed by the CPU triggers the slowing or stopping of PROCCLK. The sleep mode is enabled by bit 7 in internal configuration register 46H, and bits 1 and 0 determine the frequency of PROCCLK during sleep mode. The sleep frequency is selectable between 0 (PROCCLK stopped), CXIN/2, CXIN/4 or CXIN/8.

Bits 3 and 2 in ICR 46H determine the normal or "run" mode frequency of PROCCLK, selectable between CXIN, CXIN/2, CXIN/4 or CXIN/8. The run mode is applicable whenever:

- The sleep mode is disabled; or
- An interrupt (or NMI) has occurred

In the case of an interrupt, run mode continues until a subsequent HALT cycle. In the case of a DMA, refresh or Master operation, sleep mode resumes upon completion of the operation. Reset also returns the system to run mode.

DMA, refresh and Master operations always occur with PROCCLK = CXIN, regardless of Run and Sleep frequency settings.

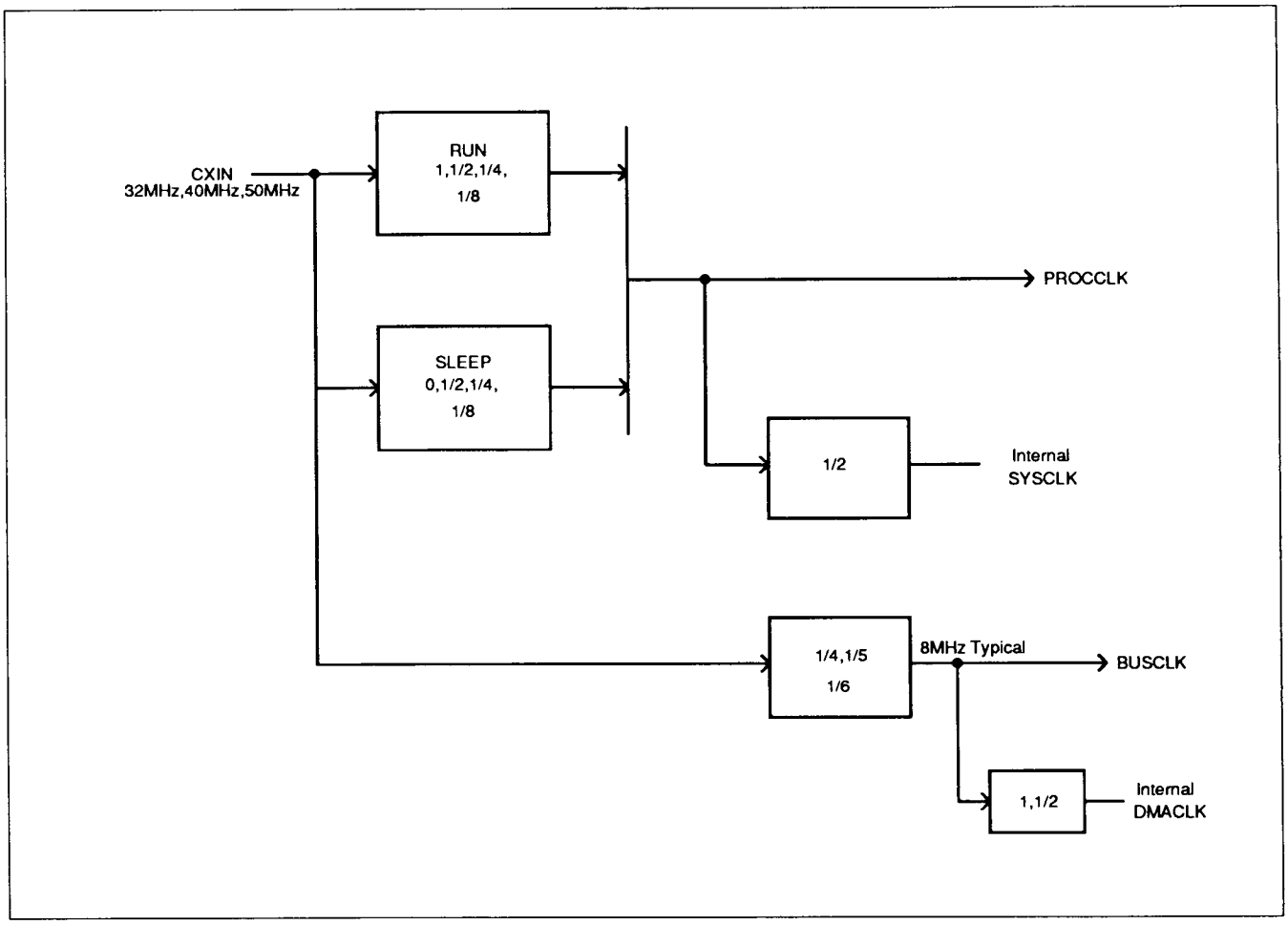
Clock Generation

The 82C836 clock generator produces the following signals (refer to Figure 3):

- PROCCLK (processor clock) is an output clock for the 80386sx processor. PROCCLK can be set (with internal configuration register 46H, bits 3-2) for the following values:
  - PROCCLK = CXIN
  - PROCCLK = CXIN/2
  - PROCCLK = CXIN/4
  - PROCCLK = CXIN/8

PROCCLK can also be halted in order to implement power management techniques (refer to *Operational Power Management* for further information).

Figure 3 *Clock Generation*



The effective speed of the 80386sx will always be half the PROCCLK frequency; e.g., for an 80386sx speed of 20 MHz, PROCCLK must be 40 MHz. PROCCLK is derived from an external crystal oscillator connected to CXIN.

- BUSCLK (bus clock) is an output clock for the I/O channel. BUSCLK can be set (with internal configuration register 41H, bits 3-2) for the following values:
  - BUSCLK = CXIN/4
  - BUSCLK = CXIN/5
  - BUSCLK = CXIN/6

The PROCCLK selection does not affect BUSCLK in either RUN or SLEEP modes.

- OSC2 (oscillator) is a 14.31818 MHz output signal used by the I/O channel. The oscillator can be derived either from an external crystal connected to pins OSC1 and OSC2, or from an external oscillator connected to OSC1 (leaving OSC2 unconnected).
- DMACLK (DMA clock) is an internal clock used by DMA controllers to time DMA operations. DMACLK can be set (with internal configuration register 01H, bit 0) to the following values:

  - DMACLK = BUSCLK
  - DMACLK = BUSCLK/2
- SYSCLK (system clock) is an internal clock used by 82C836 logic. The frequency of SYSCLK is always PROCCLK/2.

System Reset and Clock Synchronization

SCATsx supports several different reset signals:

- PWRGOOD is the main hardware reset input to the 82C836.
- XRST is the main hardware reset output from the 82C836. It resets everything except the CPU.
- CPURST is the CPU reset signal generated by the 82C836. It resets only the CPU.
- RESET2- (MFP3) comes from the 8042 keyboard controller and triggers resetting of the CPU only (CPURST).

After PWRGOOD goes high, indicating that all voltages are within specification, the 82C836 generates reset pulses on XRST and CPURST to reset the system and CPU. During the time that XRST is active, the DACK lines are sampled and latched for the strap options. The 82C836 also samples and latches NPERR- from the coprocessor to determine whether or not a coprocessor is present.

The timing of the falling edges of XRST and CPURST is critical for proper clock synchronization. System timing generated by the CPU consists of "T-states," each of which consists of two cycles of PROCCLK. The 80386sx and 80387sx contain internal "phase" clocks that track the first and second PROCCLK cycle of each T-state. For proper system operation, it is essential to synchronize both of these phase clocks and a similar phase clock inside the 82C836. An external cache controller, if implemented, also needs a phase clock if it relies on ADS- and READY- for cycle tracking.

The master phase clock for the entire system is the 82C836 phase clock, which is a free running divide-by-two from PROCCLK. The 82C836, in turn, precisely controls the timing of XRST and CPURST to synchronize all other phase clocks throughout the system. There is no provision for synchronizing the 82C836 master phase clock to an external source; rather, all external phase clocks must be synchronized to the 82C836 by means of XRST or CPURST. See *System Timing Relationships* for additional timing information.

Bus Control, Arbitration and Basic Timing

The 82C836 supports an IBM PC AT-compatible I/O channel, also known as the AT Bus. An internal bus controller, which is functionally similar to an 82288 bus controller, provides command generation and timing control for the AT-compatible I/O channel. To allow the processor to run faster than the I/O channel, DMA commands, timing, and accesses to the I/O channel can be programmed to run slower than local bus cycles.

Although the local bus is normally controlled by the CPU, the 82C836's internal DMA controller or refresh controller can request control by issuing a HOLD request to the CPU. When this occurs, the 80386sx relinquishes control and issues HLDA (hold acknowledge) to the 82C836.

The conditions that can trigger HOLD are as follows:

- System initiated refresh — 82C836 issues HOLD without any external prompting, based on the AT-compatible refresh timer.
- DMA request — 82C836 issues HOLD in response to a DREQ input. Upon receiving HLDA, the 82C836 asserts the appropriate DACK signal. Next, depending on whether or not the DMA Channel has been programmed for "cascade mode" (see *DMA Controller—8237 Compatible*), the 82C836 either generates one or more DMA cycles or waits for MASTER- to be asserted by an add-in card Bus Master.
- Master request — 82C836 follows the same DREQ/DACK protocol as for DMA, but then waits for the add-on Master to perform Master cycles as needed and eventually release DREQ. If the Master initiates a refresh (by driving REFRESH- low), the 82C836 performs the refresh and then returns control to the Master following de-assertion of XMEMR-. HOLD and HLDA remain asserted continuously throughout all Master cycles, including Master initiated refresh.

The 82C836 incorporates a state machine that generates I/O channel bus cycles for all CPU cycles not claimed by the internal memory address decode logic. The state machine synthesizes the address strobe signal (ALE), the bus command signals (XIOR-, XIOW-, XMEMR-, and XMEMW-), and MODA0 and MODA20. It monitors the state of the IOCS16- and MEMCS16- signals to determine if the device on the bus is capable of 16-bit operations for I/O and memory respectively. If a 16-bit operation is attempted with an 8-bit device, the 82C836 performs conversions of 16-bit CPU operations to paired 8-bit AT bus cycles.

The 82C836 supports XD-bus peripherals. Internal configuration register 44H controls the SDIRH and SDIRL signals for different peripherals. 16-bit X-Bus resources (other than ROM) must generate MEMCS16- and/or IOCS16- just as any 16-bit add-on card would.

The 82C836 itself asserts MEMCS16- and/or IOCS16- as follows:

- The 82C836 asserts MEMCS16- during CPU, DMA or Master accesses to ROM if it is 16 bits wide, and during CPU, DMA or Master accesses to local DRAM.
- The 82C836 asserts IOCS16- during accesses to EMS I/O ports 2x8H and 2x9H ("x" = 0 or 1, programmable), which operate as a 16-bit I/O resource.
- Although the coprocessor operates as a 16-bit I/O resource in most respects, the 82C836 does not assert IOCS16- during coprocessor accesses.

The CPU local bus and the AT Bus are "tightly coupled." Activity on either bus directly corresponds to similar activity on the other bus (except that the AT Bus remains idle during certain CPU local bus operations). The general start/end protocol for each bus cycle is as follows:

- On the CPU local bus, the start of a CPU controlled bus cycle is indicated by the low-to-high transition of ADS-, and the end of the cycle is indicated by READY- being active at the end of a subsequent T-state. The T-state in which ADS- goes high is equivalent to an 80286 TS state. This is true for either pipelined or non-pipelined 80386sx cycles. An 80386sx T1-T2-T2 sequence is equivalent to an 80286 TI-TS-TC sequence (TI = idle state). Virtually no useful work can be performed by the 82C836 during a T1 state because the CPU address and status signals are not guaranteed to be valid

until after the middle of T1. An 80386sx T1P-T2P sequence is equivalent to an 80286 TS-TC sequence, and useful work can be started during T1P.

- On the AT Bus, the start of a CPU generated bus cycle is indicated by an ALE pulse, followed by an I/O or memory command signal going active (low). The end of the cycle is indicated by the command signal going inactive. The unlatched address (UA17-23) must be valid before the end of the ALE pulse and remain valid for a short hold time after the start of command. The latched address (SA0-19) must be valid before the start of command and remain valid for a short hold time after the end of command. Read data must be valid before the end of the command pulse and remain valid for a short hold time after the end of command. Write data must be valid near the beginning of command and remain valid for a short hold time after the end of command. The command active time can be lengthened by de-asserting IOCHRDY, or (in most cases) shortened by asserting 0WS.
- Master cycles on the AT Bus follow the same protocol as CPU controlled cycles, except that ALE remains continuously active (high) and unlatched address timing follows latched address timing. Address and command signals are controlled by the add-on card Bus Master during Master cycles.
- DMA cycles on the AT Bus follow the same basic protocol as Master cycles except that the 82C836 controls the address and command signals, and commands always occur in pairs (I/O read and memory write, or memory read and I/O write).

The minimum CPU local bus cycle in pipelined mode is two T-states, which occurs most often on CAS-only accesses to local memory (see *Memory Interface*). In non-pipelined mode, the corresponding minimum is three T-states, which the 82C836 assures by appropriately controlling the timing of READY. If READY is controlled externally, two T-state cycles in non-pipelined mode are possible as follows:

- During coprocessor accesses in which the coprocessor generates READYO soon enough to end the cycle after only two T-states. This can occur only if the 82C836 has been programmed to let the coprocessor control ready during coprocessor accesses, and if the coprocessor READYO- output has been properly interfaced to CPU READY- as described in *Numeric Coprocessor Interface*.
- During external cache "hit" memory reads, as discussed in *Support for External Cache*. Detailed timing diagrams are included in *System Timing Relationships*.

The 80386sx, unlike the 80286, has the ability to keep the address valid until the Next Address signal (NA-) is asserted. During AT Bus accesses, the 82C836 takes advantage of this feature to eliminate the need for external address latches between CPU address and AT Bus address. Less expensive transparent buffers (74F245 or equivalent) may be used instead of latching buffers.

Alternatively, 74F543 latching buffers can be used if desired, with the enable pin directly controlled by NA-. The timing and decoding for the 82C836 NA- output is directly compatible with the address latch control signal needed by 74F543s. (See also *Support for External Cache*.) With latching buffers, the NA- input to the CPU can be tied low, allowing the CPU to run in pipeline mode during AT Bus accesses as well as during local memory accesses. This, in turn, may eliminate an unnecessary T1 state at the start of the next cycle, thereby producing a slight performance improvement over non-pipelined CPU operation.

Address bits A20 and A0 receive special treatment in AT-compatible architectures such as SCATsx, for the following reasons:

- For compatibility with the 8088 and 8086 at address FFFF:10H and above, it is necessary to force A20 low during CPU accesses to memory in an 80386sx or 80286 based system. The original AT-compatible approach used the GATE A20 signal from the keyboard controller to accomplish this, but there is considerable time delay in this approach. With the advent of the PS/2 architectures, a "Fast Gate A20" function was implemented in Port 92H. SCATsx supports both approaches. Setting either 8042 GATE A20 high or Fast Gate A20 to one allows the modified A20 to track CPU A20 during CPU memory accesses. Even if the modified A20 is being forced low during CPU accesses, it still tracks the CPU A20 during DMA and Master cycles.
- During conversion of 16-bit CPU accesses to paired 8-bit AT Bus cycles, A0 must be forced high during the second 8-bit cycle even though CPU A0 is still low.

Memory Interface

This section discusses four types of memory interface: ROM/shadow RAM, DRAM , extended/expanded memory and external cache.

ROM/Shadow RAM Interface

Memory accesses in the range 0C0000H-0FFFFFH can be programmed via internal configuration registers 48H through 4CH to map to ROM, local RAM, or external RAM, as follows:

- ROM can be enabled or disabled in eight independent blocks of 32 KB in the range 0C0000H-0FFFFFH.
- Local RAM can be enabled or disabled in 24 independent blocks of 16 KB throughout the range 0A0000H-0FFFFFH.
- Local RAM the range 0C0000H-0FFFFFH can be write protected in eight independent blocks of 32 KB.
- Memory accesses in the range FC0000H-FFFFFFH always go to ROM, regardless of other ROM or RAM enabling. Any local DRAM in that range is accessible only through EMS.
- If neither ROM nor RAM is enabled in a particular address block, memory accesses to that block go to the AT Bus.

The 82C836 provides one ROM chip select (ROMCS-), which is active for all ROM accesses.

On-board ROM can be either 8-bit wide or 16-bit wide and must be connected to the XD-bus. To select 16-bit width, the DACK7 line must be pulled low with a 4.7K ohm resistor; to select 8-bit mode, the line must be pulled high with a 4.7K ohm resistor. The 82C836 asserts MEMCS16- during all ROM cycles if 16-bit mode is selected.

RAM that can be accessed instead of ROM in the 0C0000H-0FFFFFH range is called shadow RAM. (RAM in the 0A0000H-0BFFFFH range is also referred to as shadow RAM.) This feature is invoked by copying an image of the BIOS (which is in ROM) into an area of RAM, thus allowing operating systems and

software applications to make faster accesses to the shadowed BIOS (rather than ROM which is much slower).

The degree of performance improvement derived from the use of shadow RAM depends primarily on the difference in access times between ROM and DRAM cycles. At higher system speeds, the difference can be significant. Additionally, shadow RAM maintains maximum BIOS performance when using either a single 8-bit ROM (such as a 27512) or two smaller 8-bit ROMs (for the same total ROM capacity). Using only one ROM reduces component count and circuit board size.

The following procedure enables shadow RAM. The program that performs this procedure must reside in RAM while it is executing, because ROM must be disabled while writing to the shadow RAM area:

1. Disable interrupts.
2. Copy the ROM BIOS into RAM, below the start of ROM.
3. Disable ROMCS- using internal configuration register 48H.
4. Enable the shadow RAM using internal configuration registers 4AH-4CH.
5. Copy the BIOS from low DRAM into the area of memory allocated for shadow RAM.
6. If desired, the BIOS can be copied in several blocks by enabling ROMCS- or shadow RAM as needed and repeating steps 2 through 5 for each block.
7. Make the shadow RAM read-only, using internal configuration register 49H.
8. Re-enable interrupts.

DRAM Interface

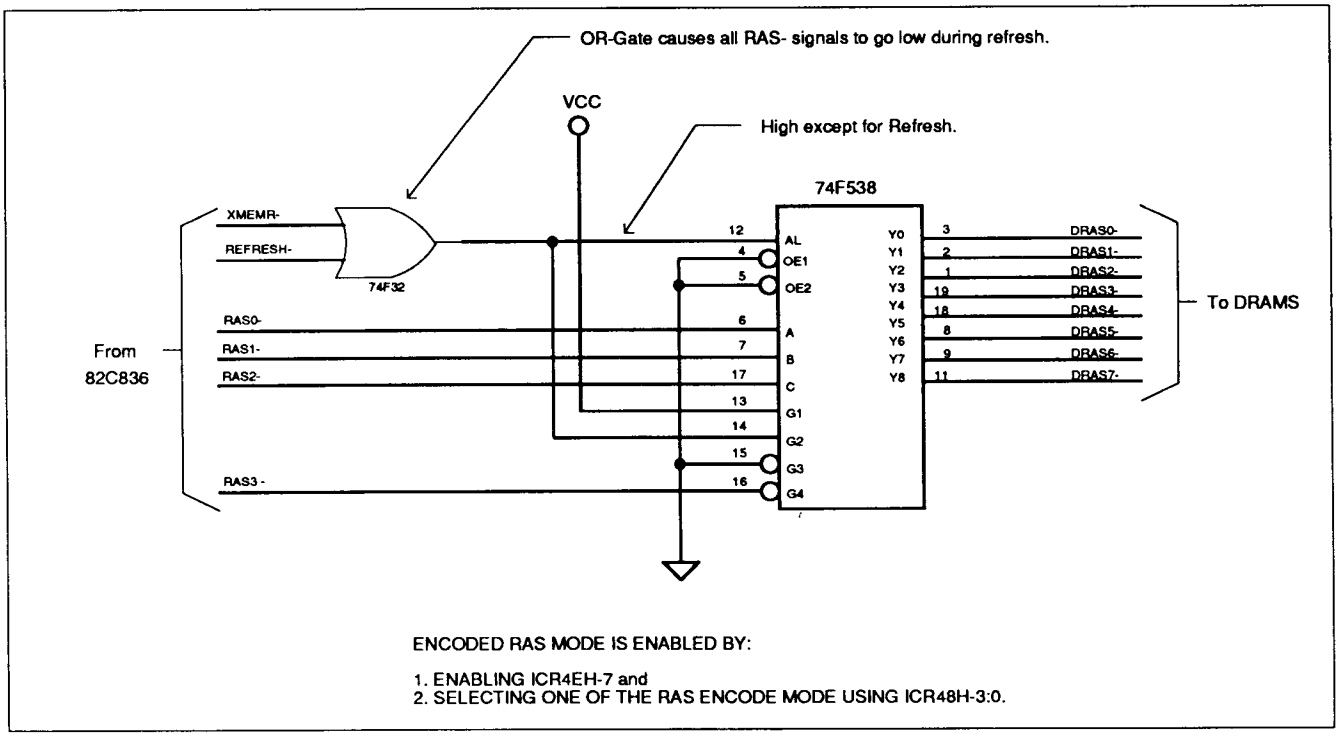
The 82C836 includes a DRAM controller that directly supports up to 8MB of memory in four banks. Up to eight banks (16 MB total) can be supported by using an external 74F538 decoder (or equivalent). In the normal mode (up to four banks), the 82C836 asserts one of the four RAS signals (RAS0- through RAS3-) that corresponds to a particular bank, plus CASL- and/or CASH- to specify the low and/or high byte of that bank. For write operations, the MWE- signal is asserted.

Support is provided for 256Kx1, 256Kx4, 1Mx1 and 1Mx4 DRAM chips. Corresponding SIMM modules are also supported, including 36-bit SIMMs utilizing either dedicated CAS or shared CAS between banks. The 82C836 uses one pair of CAS signals across up to eight 16-bit or 18-bit memory banks. Each bank must have a separate RAS- signal. Parity can be disabled, allowing the use of 16-bit wide memory banks instead of 18-bit. The 82C836 provides ten multiplexed address lines for row and column addressing, allowing direct support of banks up to 1Mx18 in size.

In the "encoded RAS" mode (up to eight banks), RAS0- through RAS2- specify the bank (one of eight), and RAS3- serves as a "RAS-enable" control for the 74F538. This mode is selected by programming an appropriate configuration using bits 4-0 of internal configuration register 4DH and enabling encoded RAS- mode with bit 7 of internal configuration register 4EH. This implementation is shown in Figure 4. In encoded RAS- mode, the multiplexed

address, CAS- signals, and MWE- operate exactly the same as in normal (8MB) mode.

Figure 4 *Encoded RAS Generation*



If parity checking is implemented, each bank is 18 bits wide: two 8-bit data bytes, each of which has one parity bit in addition to the data bits. Parity is odd, i.e., a data byte value of FFH will have a "one" for the associated parity bit. Whenever a byte or word is written, a parity bit is generated and written along with each byte. When a read occurs, the stored parity bit is compared to the parity calculated from the read byte. If a mismatch occurs during a read operation, a parity error is reported and an NMI is generated indicating a problem with memory. The NMI generation for parity errors can be disabled using bit 6 of internal configuration register 46H or bit 2 of I/O port 61H. If the system designer decides not to implement the parity bit (because of cost or other reasons), NMI generation due to parity errors should be disabled as described.

For minimum system parts count, the MA, RAS, CAS, MWE and PAR signals from the 82C836 may each drive up to 18 DRAM chips directly, without buffering. This corresponds to two SIMM modules having nine DRAM chips per module, or 6 SIMM modules having three DRAM chips (256Kx4 or 1Mx4 DRAMs) per module. There are two SIMM modules in each memory bank. Since each CAS line drives only one byte in each memory bank, the CAS lines effectively can drive twice as many memory banks as the MA and MWE lines. Also, since each memory bank is driven by a separate RAS line, RAS lines do not need buffering.

The parity lines (PARL, PARH) drive only one DRAM chip in each bank and should be able to drive up to eight banks without buffering. Similarly, the MD lines also drive only one DRAM chip in each bank, and the 80386sx and 80387sx are both rated for high capacitive loads, so no buffering should be

needed between MD lines and the CPU local data bus in the typical system implementation.

A four-bank memory and architecture consisting entirely of SIMMs having three DRAM chips each (24 DRAMs total) is a special exception to the 18-DRAM guideline. The load distribution in this case makes MA and MWE buffering unnecessary.

Memory address ranges for ROM and shadow RAM are discussed above. In addition, local RAM can be enabled or disabled in a single 384 KB block in the range 040000H-09FFFFH (ICR 4EH). Memory addresses in the range 100000H-FBFFFFH ("extended" memory) and EMS access ("expanded" memory) are discussed below. Memory accesses always go to the AT Bus unless local RAM or ROM has been enabled at the referenced memory address.

Bits 4-0 of internal configuration register 4DH must be set according to the physical DRAM configuration. Table 4 shows the valid configurations with either non-encoded RAS or encoded RAS. Table 5 shows additional configurations that are valid *only* with encoded RAS.

Table 4 *Valid Configurations - Non-encoded or Encoded RAS*

ICR 4DH Bits 4-0	Bank 0	Bank 1	Bank 2	Bank 3	Banks 4-7	Total Local Memory
00H	0	0	0	0	0	0
01H	256KW	0	0	0	0	512 KB
02H	256KW	256KW	0	0	0	1 MB
03H	256KW	256KW	0	0	0	640K + 384K
04H	256KW	256KW	256KW	0	0	1.5 MB
05H	256KW	256KW	256KW	256K	0	2 MB
06H	256KW	256KW	1MW	0	0	3 MB
07H	256KW	256KW	1MW	1MW	0	5 MB
08H	256KW	1MW	0	0	0	2.5 MB
09H	256KW	1MW	1MW	0	0	4.5 MB
0AH	256KW	1MW	1MW	1MW	0	6.5 MB
0BH	1MW	0	0	0	0	2 MB
0CH	1MW	1MW	0	0	0	4 MB
0DH	1MW	1MW	1MW	0	0	6 MB
0EH	1MW	1MW	1MW	1MW	0	8 MB

Table 5 *Valid Configurations - Encoded RAS Only*

ICR 4DH Bits 4-0	Banks 0 & 1	Banks 2 & 3	Bank 4	Bank 5	Bank 6	Bank 7	Total Local Memory
0FH	256KW	1MW	1MW	0	0	0	7 MB
10H	256KW	1MW	1MW	1MW	0	0	9 MB
11H	256KW	1MW	1MW	1MW	1MW	0	11 MB
12H	256KW	1MW	1MW	1MW	1MW	1MW	13 MB
13H	1MW	1MW	1MW	0	0	0	10 MB
14H	1MW	1MW	1MW	1MW	0	0	12 MB
15H	1MW	1MW	1MW	1MW	1MW	0	14 MB
16H	1MW	1MW	1MW	1MW	1MW	1MW	16 MB

Notes: The following notes apply to the above configurations as indicated:

1. "K" means 1024, "M" means 1048576. "B" means byte, "W" means word (two bytes).
2. In configuration 00H, all memory accesses go to the AT bus. This can be used to turn off on-board RAM even if on-board RAM exists.
3. In configuration 03H, the DRAM is mapped as 640KB conventional, 384KB extended. The 384KB block is addressed at 100000H-15FFFFH instead of 0A0000H-0FFFFFH. In configuration 02H, the DRAM is mapped as 1MB conventional (subject to enabling or disabling of the top 384K) and no extended memory.
4. In all configurations, all memory beyond the first 1MB (or beyond the first 640KB in configuration 03H) is available for use as extended memory (addressed linearly starting at 100000H). Extended memory can be accessed either directly (in 80386sx protected mode) or through expanded memory address translation (EMS). EMS is discussed further below.
5. In configuration 16H, RAM above FC0000H is accessible only through EMS. Direct accesses to FC0000H-FFFFFFH by the CPU go to ROM.
6. Banks normally are mapped in ascending physical address order, i.e., bank 1 has a higher starting address than bank 0, etc. However, in configurations 08H through 0AH, the mapping order is changed so that the 256KW bank has the highest starting address instead of the lowest.

The row address for the DRAMs comes from higher-order bits of the complete physical address, while the column address comes from bits 1 through 9 for 256K DRAMs, or 1 through 10 for 1M DRAMs. Thus, the memory addresses accessible by changing the column address only, without changing the row address, are contiguous and constitute a physical memory "page." Successive memory accesses to the same page do not require RAS- to be cycled, since it is already active and the row address is already valid. This saves considerable time in memory accessing because only CAS- needs to be cycled. The SCATsx architecture always uses CAS-only accessing (also known as page mode) whenever possible. In the SCATsx architecture, only one RAS- at a time can be active (except during refresh). Also, DMA and refresh leave RAS- inactive even if the preceding and following CPU memory accesses are in the same page.

Three different types of local memory cycles can occur, listed in order of increasing cycle time:

- Page hit — RAS- is already active and the row address is already valid, so only CAS- needs to be cycled.
- Bank switch — access to a different bank (or first access following DMA or refresh). RAS- for the target bank is always high initially, so there is no need to wait for RAS- precharge time for the previously active bank.
- Page miss — access to the same bank, but with a different row address, so that RAS- must be cycled high and low before CAS- timing can begin.

T-state counts for these cycles are as follows:

- Page hit read — 0 WS
- Page hit write — 1 WS
- Bank switch read or write, non-encoded RAS — 1 WS
- Bank switch read or write, encoded RAS — 2 WS
- Page miss, same bank, read or write — 3 WS

The total number of T-states for pipelined cycles is the WS amount plus 2; for non-pipelined cycles, WS plus 3. If EMS is enabled, one further T-state is added for all cycle types for accesses that require address translation. A minimum of 2.5 T-states are always allowed for read data access from RAS- (125 ns at 20 MHz CPU speed, 156 ns at 16 MHz CPU speed). This allows the use of 80 ns DRAMs at 20 MHz, or 100 ns DRAMs at 16 MHz.

The vast majority of all memory accesses generally are instruction fetches, which tend to cluster in short bursts of accesses in highly localized address ranges. Even jump operations frequently are very localized. Thus, paging will usually result in substantial performance improvement over non-paged memory timing, since a high percentage of memory cycles can be CAS-only.

In addition to paging, four-way page interleaving is automatically performed in banks 0-3 whenever they all contain the same size DRAM (configurations 05H, 0EH and 13H through 16H). Page interleaving means that the physical DRAM pages (CAS-only addressable blocks) are interleaved in sequence across the four banks, i.e., page n in bank 0, n+1 in bank 1, n+2 in bank 2, n+3 in bank 3, n+4 in bank 0 again, and so on. This has the effect of increasing the relative probability of bank switch cycles over page misses, resulting in significant performance improvement over simple paging without interleaving.

Similarly, if banks 4-7 are enabled and contain the same size DRAMs, four-way page interleaving is automatically performed in those four banks. Refer to configurations 12H and 16H.

If any two banks in a pair are of the same size, but the other pair is empty or has only one bank populated, then two-way page interleaving is automatically performed in the two same-size banks. The banks are paired as follows: 0 and 1, 2 and 3, 4 and 5, 6 and 7. In addition, banks 1 and 2 are two-way page interleaved in configurations 09H and 0AH, with non-interleaved page mode in the remaining banks. Thus, the only configurations in which no interleaving occurs are 00H, 01H, 08H and 0BH.

A RAS timeout feature is provided to support DRAMs that require a 10 microsecond maximum RAS-active time. If the timeout is enabled, RAS is not allowed to remain low continuously for more than about 9.5 microseconds. If the timeout is disabled, periodic refresh cycles limit the maximum possible RAS active time to about 15 microseconds.

Support for External Cache

Further system performance improvement can be achieved by implementing an external cache. Typical caches operate by "remembering" a group of addresses and the data that was last written at each address. When the same address is subsequently read back, the cache provides the read data at very high speed. The "remembered" addresses are generally referred to as "tag" addresses. A "cache read hit" refers to a read cycle in which the read address matches a

previously stored tag address, so that valid read data can be provided by the cache instead of the addressed system resource. A "cache write hit" refers to a write cycle in which the write address matches a stored tag address, so that the associated data for that tag address must either be updated or marked as invalid. A "cache miss" refers to a read or write cycle in which the read or write address does not match any stored tag addresses. The cache can either ignore the "miss" cycle or "remember" the new tag address and associated data.

A fundamental assumption of cache systems is that the read data remains the same on successive reads until subsequently re-written. This will always be true for memory read operations unless address remapping is occurring and particular addresses don't necessarily remain mapped to the same physical memory locations. It is also possible to cache I/O cycles as long as the basic "data constancy" assumption is satisfied. Typical cache schemes may differ radically from each other in the exact strategy they use for deciding when to update the cache and how to organize the stored tags.

The 82C836 provides the following features for external cache support:

- Three configuration bits in ICR 41H allow optional enabling of "Early READY" mode, "Local Bus Access" (LBA) mode, and/or "Force Bus Convert" mode. The "Early READY" mode allows an external cache to "claim" a non-pipelined, CPU-generated cycle by asserting READY- during the first T2 state. The 82C836 does not generate a cycle in this case and relies on the external cache to provide the read data or accept the write data. The result is a two T-state cycle (T1-T2).
- "LBA" mode is essentially the same as "Early READY" except that the 0WS- input is used instead of READY- as the "cache hit" signal, and READY- generation is left entirely to the external cache controller or other external source. This allows cache implementations in which READY- needs to be delayed. With an external gate or multiplexer, the 0WS- signal can still be used as a normal AT Bus 0WS- input even when LBA mode is enabled. The LBA- and AT Bus 0WS- signals can simply be ANDed together (i.e., low-true OR) to provide the LBA-/0WS- input to the 82C836.
- To allow cache implementations to cache 16 bits at a time and also cache AT Bus accesses, the "Force Bus Convert" mode can be used to force all AT Bus reads to be 16 bits, including reads from 8-bit memory or I/O resources. Reads from local DRAM are always 16 bits automatically, regardless of "Force Bus Convert" mode. (To avoid a performance penalty when using 8-bit AT Bus memory resources such as 8-bit video memory, the "Force Bus Convert" feature does not apply to 8-bit AT Bus memory resources residing in the first 1 MB of address space.)
- External devices such as a cache controller may need a "CPU Cycle Start" signal if they do not monitor ADS- and READY- directly. To provide such a signal, the NA- pin function can be changed by the DACK3- strap option to operate as a "Start Cycle" (STCYC-) signal instead of a "Next Address" (NA-) signal. The signal can still be used as a 74F543 latch enable, if desired, in either case.

The following types of CPU generated cycles can be "claimed" by using the "Early READY" or "LBA" mode:

- Local DRAM reads (but not local DRAM writes).
- AT Bus memory or I/O reads or writes, including accesses to on-board resources other than local DRAM.

When using the "Early READY" or "LBA" modes, the CPU NA- input must be tied high, causing the CPU to operate in non-pipelined mode only. This is necessary because of CPU address timing. The 82C836 may need to generate a local DRAM cycle if the cycle is not "claimed" by an external controller, and the 82C836 requires a valid CPU address beyond the point at which it could change in a pipelined cycle.

If either LBA or Early READY mode is enabled (or both), the 82C836 automatically generates an "early wait state" at the beginning of every local DRAM read cycle that is not claimed by an external controller. This allows time for the external controller to signal a cache hit before the 82C836 starts a DRAM cycle. The added wait state affects the total cycle time only in the event of a cache read miss. Cache read hits can be zero wait state if the external cache controller uses Early READY mode. There is no added wait state for local DRAM write cycles.

To avoid the need for parity bits in the cache SRAMs, parity checking on local DRAM read cycles is suspended in the event of a cache hit. Parity checking of local DRAM memory operates normally on cache misses.

When using both local memory caching and internal EMS (see next section), the EMS page windows and EMS target areas ("expanded" memory) must be excluded from the cacheable address ranges. Since EMS address translation is entirely internal to the 82C836, a local memory cache only has access to CPU addresses, not translated EMS addresses.

Expanded/Extended Memory

The 82C836 fully supports the LIM EMS 4.0 and 3.2 specifications. EMS allows operating systems and software applications to access memory above the 1MB DOS limit through a page mapping scheme managed by an EMS driver. The SCATsx architecture allows up to 64KB of address space in the first 1MB to be remapped, in 16KB pages, to anywhere in the 16MB address space, described as follows:

- **Remappable area.** The remappable address range is either 0D0000-0DFFFFH or 0E0000-0EFFFFH, determined by a bit in one of the EMS I/O ports (see below). In LIM EMS terminology, this 64KB remappable address range is referred to as the "Page Frame."
- **Page windows and target pages.** The remappable address range is divided into four page windows of 16KB each. Each page window can be mapped to any 16KB target page anywhere in the 16MB address space. In LIM EMS terminology, the page windows are referred to as "physical pages," and the target pages are referred to as "logical pages."
- **Page registers.** Associated with each page window is a page register. The 82C836 contains four page registers — one for each 16KB page window. Each page register specifies the absolute physical location of the 16KB target page to be mapped into the associated page window. Each page register also specifies whether or not the associated page window is enabled or disabled, i.e., whether or not to perform EMS address remapping in that window. If a page window is disabled, accesses to that window are treated as ordinary non-EMS memory accesses.
- **EMS I/O ports.** The page registers are accessed using three I/O ports located at either I/O addresses 208H-20AH or 218H-21AH (selectable via ICR 4F bit 0). In I/O port 20AH (or 21AH), bits 1-0 specify which page

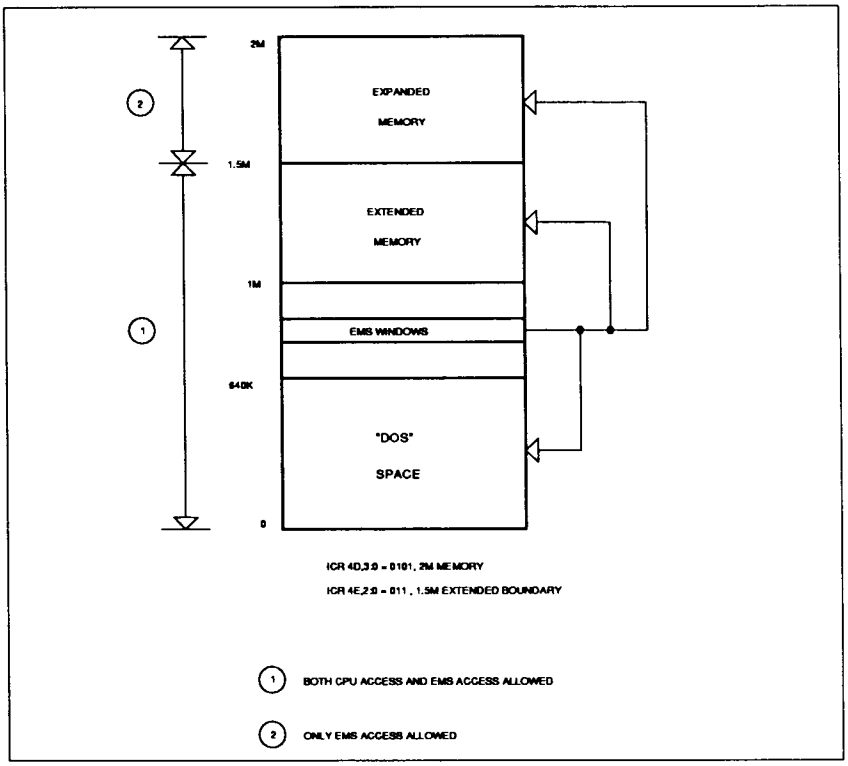
register is currently accessible. The selected page register is then accessed at I/O ports 208H and 209H (or 218H and 219H).

- **EMS enable/disable.** EMS translation can be enabled or disabled by ICR 4F bit 7. In addition, the EMS I/O ports can be enabled or disabled by ICR 4F bit 6.

The target pages can come from anywhere in system memory, including the first 1MB. No hardware checking is performed for conflicts with DOS memory or other memory areas. It is the responsibility of the EMS driver to select appropriate values for the page register contents.

To avoid potential conflicts between extended memory and EMS memory, i.e., software erroneously altering EMS memory through extended memory access, internal configuration register 4E, bits 3-0, can be used to set a limit on the top of extended memory. Memory above that limit will then be accessible only through the EMS mechanism. Direct CPU memory accesses at addresses above the limit will go to the AT bus. See Figure 5.

Figure 5 *EMS/Extended Memory*



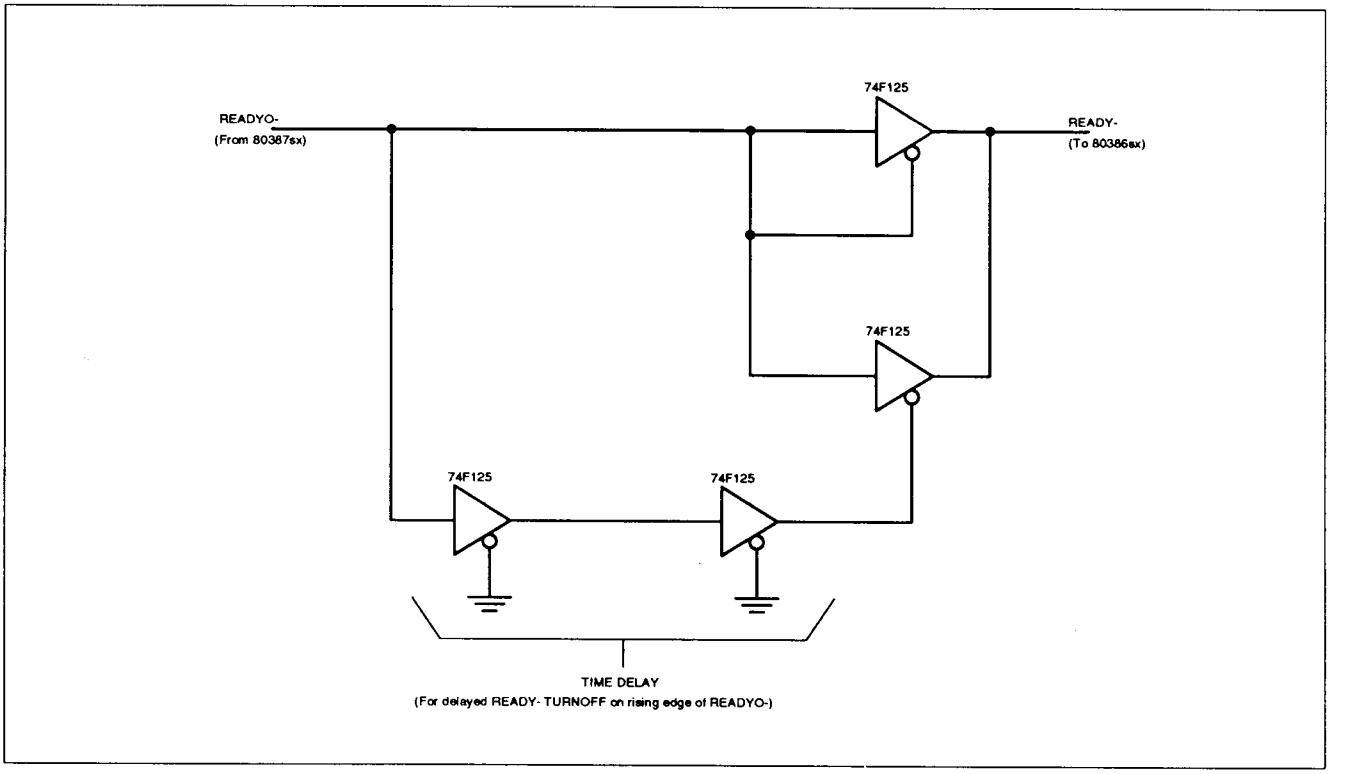
EMS paging and the EMS I/O ports are normally managed by application software or an EMS driver that allows applications to access more RAM than is normally allowed by DOS. More information on software implementation can be obtained from various Intel and Microsoft publications.

Numeric Co-Processor Interface

The 82C836 contains interface logic that supports use of an 80387sx numeric coprocessor. The logic provides the following features:

- Precisely times the system reset signal (XRST) to properly synchronize the 80387sx internal phase clock. Due to pin limitations, the 82C836 does not provide a separate coprocessor reset signal and does not support coprocessor reset via output to port F1H. An output to F1H is invariably followed by an FINIT instruction, which is sufficient to insure a full reset of the coprocessor. (FINIT alone is not sufficient for an 80287 because of Protected Mode, but the 80387sx doesn't differentiate between Protected and Real Modes.)
- Automatically detects the presence or absence of the 30387sx by sensing the coprocessor ERROR- signal during system reset (XRST).
- Detects and latches coprocessor error status.
- Generates an interrupt (IRQ13) to the system when an error occurs, and keeps BUSY- to the CPU active until error processing has begun, as indicated by an I/O write to port F0H. (As in all AT-compatible architectures, the ERROR- input to the CPU is not used.)
- Generates READY- to the CPU if desired, or, optionally, relies on the 80387sx to generate READY-. In order for the 80387sx to generate READY-, an external three-state interface such as that shown in Figure 6 is needed between 80387sx READYO- and 80386sx READY-. If the 82C836 is programmed to generate READY- during coprocessor accesses, the 80387sx READYO- output should be left open and the external READYO- interface is unnecessary.

Figure 6 *READYO- Interface*



- Generates a dummy BUSY- signal to the CPU when no coprocessor is present (for compatibility with software packages using certain critical

coprocessor instructions to test for coprocessor presence). The dummy BUSY- signal is just a buffered, inverted version of REFRESH-.

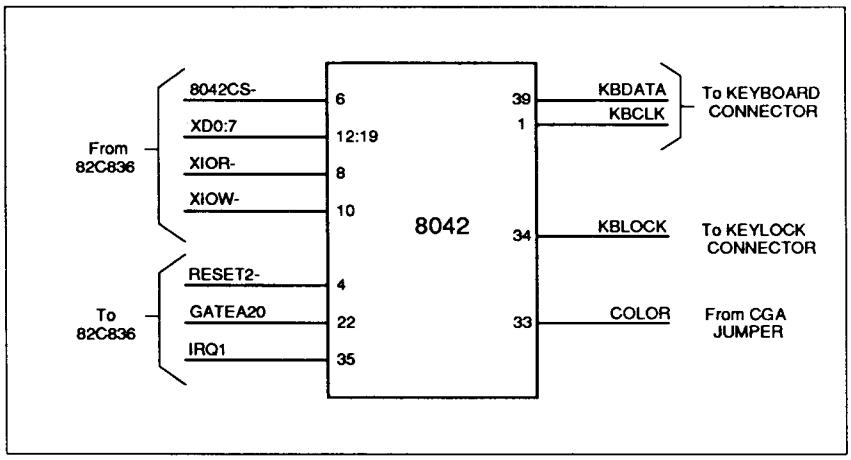
Additional considerations are as follows:

- Whenever a coprocessor error has occurred and the coprocessor is not busy, external logic should force PEREQ to the 80386sx active (high) to allow the 80386sx to finish any coprocessor I/O cycles remaining in the current (error causing) coprocessor instruction. This mode, forcing PEREQ when the coprocessor is not busy, should end as soon as the 80386sx acknowledges the coprocessor error condition, as indicated by the 82C836 BUSY- output going inactive. A circuit for accomplishing this is shown in the system schematics. Due to pin limitations, it was not possible to incorporate this logic inside the 82C836.
- Certain software packages are known to rely on reading coprocessor status to determine coprocessor presence or absence. To guarantee that coprocessor status will be invalid in the event that no coprocessor is present, a low-order data bit, such as D0, should be pulled down instead of pulled up. (See *System Schematics*.)

Keyboard Controller Interface — External 8042

The 82C836 relies on an external 8042 to handle keyboard operations (refer to Figure 7). The clock for the 8042 may be derived from BUSCLK or OSC and should have a frequency that is between 6 MHz and 10 MHz when used with standard keyboard controllers. The 8042 interfaces with the 82C836 through IRQ1 and a chip select line. The 8042 also provides two output signals: GATEA20 and RESET2-. These signals are brought into the 82C836 and combined internally with the FAST GATEA20 and FAST CPU RESET functions available via Port 92H. The keyboard controller also supports the keylock and CGA strap functions.

Figure 7 *Keyboard Controller*



RTC Interface — MC146818 Compatible

The 82C836 contains an internal, MC146818-compatible real-time clock (RTC) with a total of 128 bytes of CMOS RAM (including the dedicated bytes used for clock functions). The internal RTC may be disabled so that an external RTC

can be used. Internal/external RTC mode is selected by the state of the EXRTC- signal (DACK5-) during power-up. If this pin is pulled low with a 4.7K ohm resistor, external RTC mode is selected. A 4.7K ohm pull-up resistor puts the 82C836 into internal RTC mode.

When the internal RTC is used:

- 82C836 V_{CC} pins should connect to battery-backed power.
- MFP4 is used as the 32 KHz clock input.
- MFP5 is used as the power sense input.

When an external RTC is used:

- 82C836 V_{CC} pins should be connected to the normal power source.
- MFP4 is used as the IRQ8 input from the RTC.
- MFP5 is used as the RTCCS- chip select output for the RTC.

The internal RTC combines a complete time-of-day clock with alarm, one hundred year calendar, a programmable periodic interrupt, and 114 bytes of low-power static RAM. Provisions are made to enable the device to operate in a low-power (battery powered) mode and protect the contents of both the RAM and clock during system power-down.

Register Access

Table 6 *Address Map for the Real Time Clock*

Column	Function
00	Seconds
01	Seconds Alarm
02	Minutes
03	Minutes Alarm
04	Hours
05	Hours Alarm
06	Day of Week
07	Day of Month
08	Month
09	Year
0A	Register A
0B	Register B
0C	Register C
0D	Register D
0E	User RAM
0F	User RAM
.	
.	
.	
7E	User RAM
7F	User RAM

I/O ports 70H and 71H are used for accessing the 128 locations in the Real Time Clock. First the index address (0 to 7FH) is output to port 70H. Then the data is read or written at port 71H. The entire port 70H/71H sequence should be completed while interrupts are inhibited or during an interrupt service routine before re-enabling interrupts. Otherwise, an interrupt service routine potentially could intervene between the output to port 70H and the subsequent I/O to port 71H, over-writing the port 70H value.

RTC Address Map

Table 6 shows the internal register/RAM organization of the Real Time Clock portion of the 82C836. The 128 addressable locations in the Real Time Clock are divided into ten bytes that normally contain the time, calendar and alarm data, four control and status bytes, and 114 general purpose RAM bytes. All 128 bytes are readable by the CPU. The CPU may also write to all locations except Registers C, D, Bit 7 of Register A, and Bit 7 of the Seconds Byte, which is always zero.

Time Calendar and Alarm Bytes

The CPU can obtain the time and calendar information by reading the appropriate locations in the Real Time Clock. Initialization of the time, calendar, and alarm information is accomplished by writing to these locations. Information is stored in these locations in binary-coded decimal (BCD) format.

Before initialization of the internal registers can be performed, the SET bit in Register B should be set to a one to prevent Real Time Clock updates from occurring. The CPU then initializes the first ten locations in BCD format. The SET bit should then be cleared to allow updates. Once initialized and enabled, the Real Time Clock performs clock/calendar updates at a 1Hz rate.

Table 7 *Format for Clock, Calendar, and Alarm data*

Index Register Address	Function	BCD Range
0	Seconds	00-59
1	Seconds Alarm	00-59
2	Minutes	00-59
3	Minutes Alarm	00-59
4	Hours (12 hour mode) Hours (24 hour mode)	01-12 (AM) 81-92 (PM) 00-23
5	Hours Alarm (12 hour mode) Hours Alarm (24 hour mode)	01-12 (AM) 81-92 (PM) 00-23
6	Day of Week	01-07
7	Day of Month	01-31
8	Month	01-12
9	Year	00-99

Table 7 shows the format for the ten clock, calendar, and alarm data. The 24/12 bit in Register B determines whether the hour locations are updated using a 1-12 or 0-23 format. In 12 hour format, the high order bit of the hours byte in both the time and alarm bytes indicates PM when it is a one.

During updates, which occur once per second, the ten bytes of time, calendar, and alarm information are unavailable to be read or written by the CPU for a period of 2ms. These ten locations cannot be written to during this time. Information read while the Real Time Clock is performing update is undefined. The Update Cycle section describes how Update Cycle/PCU contention problems can be avoided.

The alarm bytes can be programmed to generate an interrupt at a specific time or they can be programmed to generate a periodic interrupt.

Static RAM

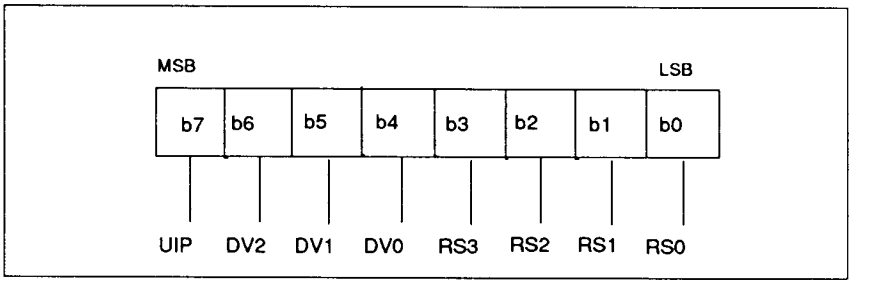
The 114 bytes of RAM from Index Address 0EH to 7FH are not affected by the Real Time Clock. These bytes are accessible during the update cycle and may be used for whatever the designer wishes. Typical applications use this as non-volatile storage for configuration and calibration parameters since this device is normally battery powered when the system is turned off.

Control and Status Registers

The 82C836 contains four registers used to control the operation and monitor the status of the Real Time Clock. These registers are located at Index Address 0AH-0DH and are accessible by the CPU at all times. Figures 8 through 11 show the contents of these registers.

Figure 8 shows the format of Register A.

Figure 8 *Register A (address 0AH). All bits except UIP are read/write.*



b7 is UIP—Update in progress flag is a status bit used to indicate when an update cycle is about to take place. A one indicates that an update cycle is taking place or is imminent. UIP goes active (HIGH) 244 µs prior to the start of an update cycle and remains active for an additional 2 ms while the update is taking place. The UIP bit is read only and is not affected by RESET. Writing a one to the SET bit in Register B inhibits any update cycle and then clears the UIP status bit

b6—b4 is DV<0:2> which controls the Divider Prescaler on the Real-Time Clock. While the 82C836 RTC can operate at frequencies higher than 32.768 kHz, this is not recommended for battery-powered

operation due to the increased power consumption at these frequencies. OSCI frequencies and modes are:

4.194304 MHz in operate mode when DV2 = 0 and DV1 = 0 and DV0 = 0
1.048576 MHz in operate mode when DV2 = 0 and DV1 = 0 and DV0 = 1
32.768 kHz in operate mode when DV2 = 0 and DV1 = 1 and DV0 = 0
The divider is reset when DV2 = 1 and DV1 = 1 regardless of DV0

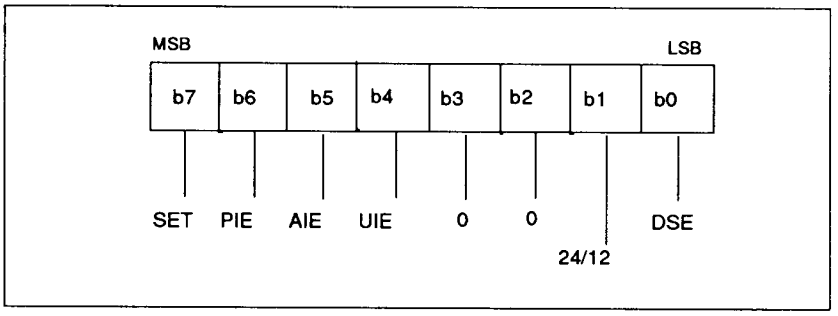
b3—b0 is RS<0:3> which controls the periodic interrupt rate. The Periodic Interrupt is derived from the Divider/Prescaler in the Real-Time Clock and is separate from the alarm Interrupt. Both the alarm and periodic interrupts use the same interrupt channel in the Interrupt Controller. Use of the Periodic Interrupt allows the generation of interrupt rates higher than one per second. Table 8 shows the interrupt rates for which the Real-Time Clock may be programmed.

Table 8 *Programmable Interrupt Rates for Real-Time Clock*

Rate Selection				Time Base	
RS3	**RS2**	**RS1**	**RS0**	**4.194304 MHz 1.048576 MHz**	**32.768 kHz**
0	0	0	0	None	None
0	0	0	1	30.517 µs	3.90526 ms
0	0	1	0	61.035 µs	7.8125 ms
0	0	1	1	122.070 µs	122.070 µs
0	1	0	0	244.141 µs	244.141 µs
0	1	0	1	488.281 µs	488.281 µs
0	1	1	0	976.562 µs	976.562 ms
0	1	1	1	1.953125 ms	1.953125 ms
1	0	0	0	3.90625 ms	3.90625 ms
1	0	0	1	7.8125 ms	7.8125 ms
1	0	1	0	15.625 ms	15.625 ms
1	0	1	1	31.25 ms	31.25 ms
1	1	0	0	62.5 ms	62.5 ms
1	1	0	1	125 ms	125 ms
1	1	1	0	250 ms	250 ms
1	1	1	1	500 ms	500 ms

Figure 9 shows the format of Register B.

Figure 9 *Register B (address 0BH), Read Only.*



b7 is SET. Writing a zero to this bit enables the update cycle and allows the real time clock to function normally. When set to one, the Update Cycle is inhibited and any cycle in progress is aborted. The SET bit is not affected by the RESET input pin.

b6 is PIE. The Periodic Interrupt Enable Bit controls the generation of interrupts based on the value programmed into bits b3-b0 of Register A. This allows the user to disable this function without affecting the programmed rate. Writing a one to this bit enables the generation of periodic interrupts. This bit is cleared to a zero by RESET.

b5 is AIE. The generation of alarm interrupts is enabled by setting this bit to a one. Once this bit is enabled, the Real-Time Clock generates an alarm whenever a match occurs between the programmed alarm and clock information. If the "don't care" condition is programmed into one or more of the alarm registers, this enables the generation of periodic interrupts at rates of one second or greater. This bit is cleared to a zero by RESET.

b4 is UIE. The Update-ended Interrupt Enable Bit enables the UF (update end flag) bit in Register C to assert interrupt request. This bit is cleared to a zero by RESET.

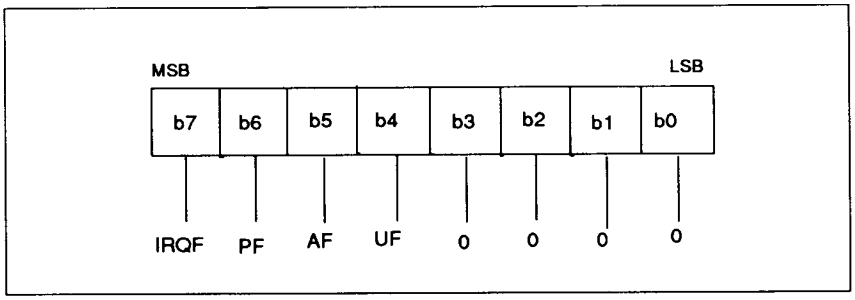
b3, b2 are read as zero.

b1 is 24/12. The 24/12 control bit is used to establish the format of both the Hours and Hours Alarm bytes. If this bit is a one, the Real-Time Clock interprets and updates the information in these two bytes using the 24-hour mode. This bit can be read or written to by the CPU and is not affected by RESET.

b0 is DSE. The Real-Time Clock can be instructed to handle daylight savings time changes by setting this bit to a one. This enables two exceptions to the normal time-keeping sequence to occur. Setting this bit to zero disables the execution of these two exceptions.

Figure 10 shows the format of Register C.

Figure 10 *Register C (address 0CH), Read Only.*



b7 is IRQF. The Interrupt Request Flag is set to one when any of the conditions that can cause an interrupt is true and the interrupt enable for that condition is true. The condition that causes this bit to be set also generates an interrupt. The logical expression for this flag is:

IRQF = PF & PIE + AF & AIE + UF & UIE

This bit and all other active bits in this register are cleared by reading this register or by activating the PS- input pin. Writing to this register has no effect on the contents.

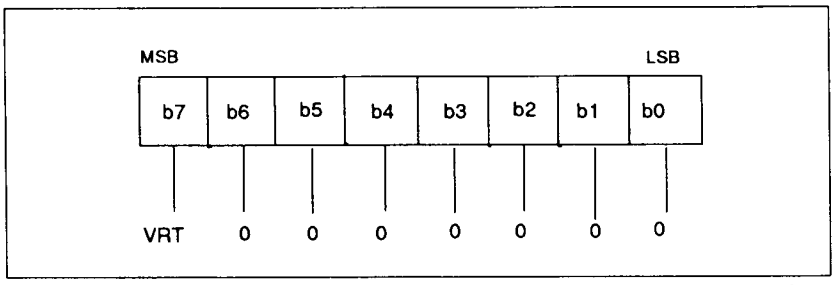
b6 is PF. The Period Interrupt Flag is set to one when a transition, selected by RS3-RS0, occurs in the divider chain. This bit becomes active independent of the condition of the PIE control bit. The PF bit then generates an interrupt and sets IRQF if PIE is one.

b5 is AF. A one appears in the AF bit whenever a match has occurred between the time register and alarm register during an update cycle. This flag is also independent of its enable (AIE) and generates an interrupt if AIE is true.

b4 is UF. The update-ended flag bit is set after each cycle. When the UIE bit is a one, the one in UF causes the IRQF bit to be a one asserting IRQ. UF is cleared by a Register C read or by a RESET.

Figure 11 shows the format for Register D

Figure 11 *Register D (Address 0DH), Read Only.*



b7 is VRT. The Valid RAM and Time Bit indicates the condition of the contents of the Real-Time Clock. This bit is cleared to a zero whenever the PS input pin is low. This pin is normally derived from the power supply, which supplies V_{CC} to the device and allows the user to determine whether the registers have been initialized since

power was applied to the device. POWERGOOD has no effect on this bit and it can only be set by reading Register D. All unused register bits will be zero when read and are not writeable.

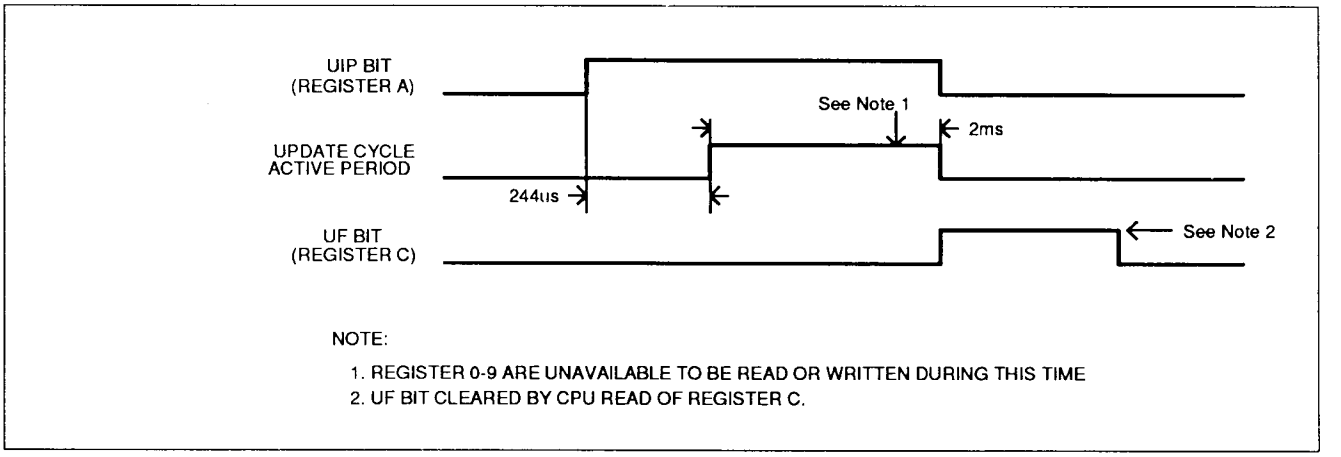
b6 to b0 are read as zeros.

Update Cycle

During normal operation, the Real Time Clock performs an update cycle once every second. The performance of an update cycle is contingent upon the divider bits DV<0:2> not being cleared, and the SET bit in Register B being cleared. The function of the update cycle is to increment the clock/calendar registers and compare them to the Alarm Registers. If a match occurs between the two sets of registers, an alarm is issued and an interrupt is issued if the alarm and interrupt control bits are enabled.

During the time that an update is taking place, the lower ten registers are unavailable to the CPU. This is done to prevent the possible corruption of data in the registers or the reading of incorrect data. To avoid contention problems between the Real Time Clock and the CPU, a flag is provided in Register A to alert the user of an impending update cycle. This Update In Process Bit (UIP) is asserted 244 µs before the actual start of the cycle and is maintained until the cycle is complete. Once the cycle is complete, the UIP bit is cleared and the Update Flag (UF) in Register C is set. Figure 12 illustrates the update cycle.

Figure 12 *Update Cycle*



CPU access is always allowed to Register A through D during update cycles.

Two methods for reading and writing to the Real Time Clock are recommended. Both of these methods allow the user to avoid contention between the CPU and the Real Time Clock for access to the time and date information.

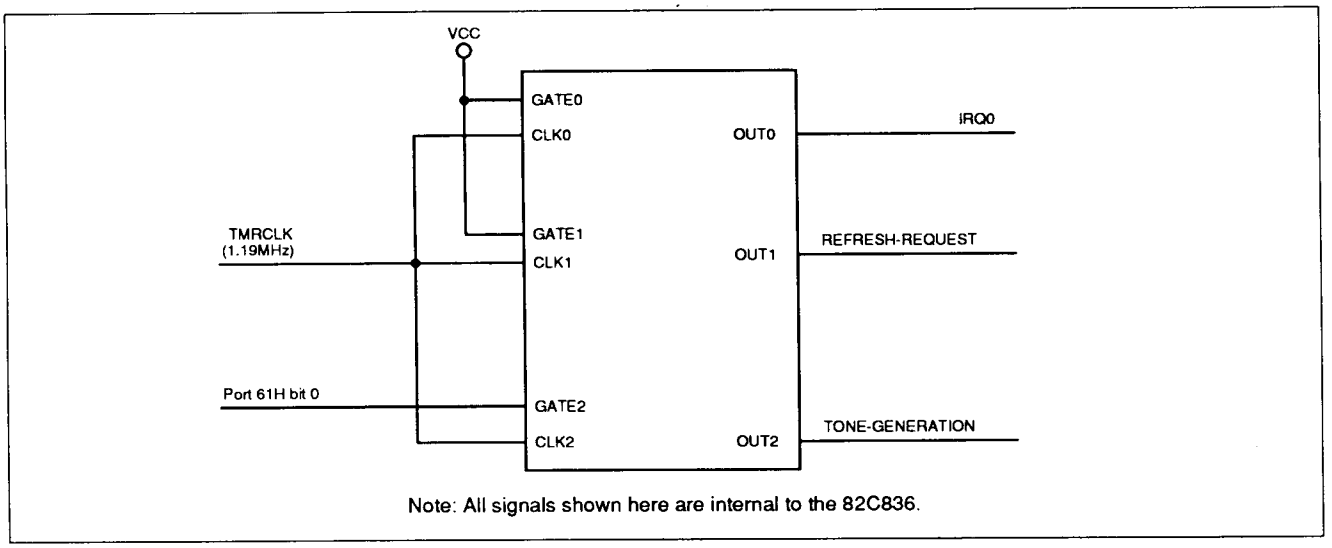
The first method is to read Register A, determine the state of the UIP bit. If the UIP is zero, perform the read or write operation. For this method to work successfully, the entire read or write operation (including any interrupt service routines which might occur) must not require more that 244us to complete from the beginning of the read of Register A to the completion of the last read or write operation to the Clock Calendar Registers.

The second method of accessing the lower ten registers is to read Register C once and disregard the contents. Continue reading this register until the UF bit is a one. This bit becomes true immediately after an update is completed. The user then has until the start of the next update cycle to complete a read or write operation.

Programmable Interval Timer — 8254 Compatible

The programmable interval timer, which is equivalent to the Intel 8254 Programmable Interval Timer/Counter, is programmable through external I/O ports 0040H through 0043H (refer to Figure 13).

Figure 13 *Programmable Timer Control*



Note: All signals shown here are internal to the 82C836.

The inputs of the three channels are connected to a 1.19MHz clock. The 1.19MHz clock is internally generated by dividing OSC1 (14.31818MHz) by 12. The output of the three channels are as follows:

- Channel 0 is a general purpose and software interrupt timer. The output of this channel is connected directly to the IRQ0 pin of the internal programmable interrupt controller.
- The output of Channel 1 is used internally by the 82C836 to generate refresh requests.
- The output of Channel 2 supports tone generation for the audio speaker.

The Counter/Timer (CTC) in the 82C836 is general purpose, and can be used to generate accurate time delays under software control. The CTC contains three 16-bit counters (Counter 0-3) that can be programmed to count in binary or binary coded decimal (BCD). Each counter operates independently of the other two and can be programmed for operation as a timer or a counter.

All three of the counters are controlled from a common set of control logic. The control logic decodes control information written to the CTC and provides the controls necessary to load, read, configure, and control each counter. Counter 0 and Counter 1 can be programmed for all six modes, but Mode 1 and Mode 5

have limited usefulness due to the lack of an external hardware trigger signal. Counter 2 can be operated in any of six modes listed below.

- Mode 0 - Interrupt on terminal count
- Mode 1 - Hardware re-triggerable one-shot
- Mode 2 - Rate generator
- Mode 3 - Square wave generator
- Mode 4 - Software triggered strobe
- Mode 5 - Hardware re-triggerable strobe

All three counters in the CTC are driven from a common clock TMRCLK (1.19MHz) that is derived by dividing OSC1 (14.31818 MHz) by 12. Counter zero output (OUT0) is connected to IRQ0 of INTC1 (see *Interrupt Controller Functional Description*) and may be used as an interrupt to the system for time keeping and task switching. Counter 1 is programmed to generate pulses for use by the refresh generator. The third counter (Counter 2) is a full function Counter/Timer. This channel can be used as an interval timer, a counter, or as a gated rate/pulse generator (which is normally used as a speaker tone generator).

Counter Description

Each counter in the CTC contains a Control Register, a Status Register, a 16-bit Counting Element (CE), a pair of 8-bit Counter Input Latches (CIL and CIH), and a pair of 8-bit Counter Output Latches (COL and COH). Each counter also has a clock input for loading and decrementing the CE, a mode defined GATE input for controlling the counter (only GATE2 is controlled by I/O port 61, bit 0), and an OUT signal. The OUT signal's state and function are controlled by the Counter Mode and condition of the CE (see Mode Definitions).

The Control Register stores the mode and command information used to control the counter. The Control Register may be loaded by writing a byte, containing a pointer to the desired counter, to the Write Control Word address (043H). The remaining bits in the byte contain the mode, the type of command, and count format information.

The Status register allows the software to monitor counter condition and read back the contents of the Control Register.

The Counting Element is a loadable 16-bit synchronous down-counter. The CE is loaded or decremented on the falling edge of TMRCLK. The CE contains the maximum count when a zero is loaded; this is equivalent to 655536 in binary operation or 10000 in BCD. The CE does not stop when it reaches zero. In Modes 2 and 3, the CE is reloaded and in all other modes it wraps around to FFFFH in binary operation or 9999 in BCD.

The CE is indirectly loaded by writing one or two bytes (optional) to the Counter Input Latches, which are in turn loaded into the CE. This allows the CE to be loaded or reloaded in one TMRCLK cycle.

The CE is also read indirectly by reading the contents of the Counter Output Latches. COL and COH are transparent latches that can be read while transparent or latched (see Latch Counter Command).

Programming the CTC

After power-up, the condition of CTC Control Registers, counter registers, CE, and the output of all counters is undefined. Each counter must be programmed before it can be used.

Counters are programmed by writing a Control Word and then an initial count. The Control Register of a counter is written to by writing to the Control Word address, as shown in Table 9. Figure 14 shows the Control Word Structure. The Control Word is a write-only location.

Table 9 *CTC I/O Addresses*

Address	Function
040H	Counter 0 Read/Write
041H	Counter 1 Read/Write
042H	Counter 2 Read/Write
043H	Control Word, write only

Figure 14 *Control Word (Address 043H)*

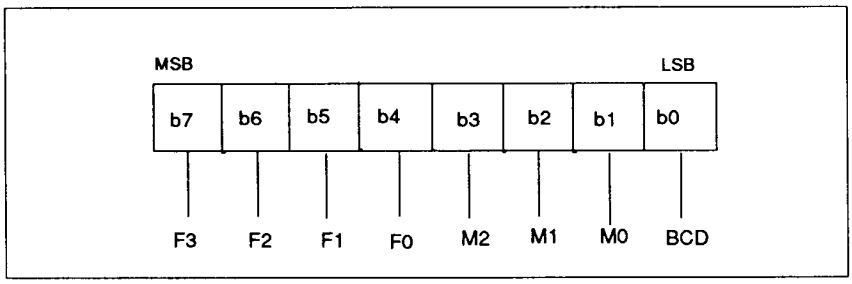


Table 10 *Command Table*

F3	F2	F1	F0	Command
0	0	0	0	Latch Counter 0 (See Counter Latch Command
0	0	0	1	Read/Write Counter 0 LSB Only
0	0	1	0	Read/Write Counter 0 MSB Only
0	0	1	1	Read/Write Counter 0 LSB then MSB
0	1	0	0	Latch Counter 1 (See Counter Latch Command
0	1	0	1	Read/Write Counter 1 LSB Only
0	1	1	0	Read/Write Counter 1 MSB Only
0	1	1	1	Read/Write Counter 1 LSB then MSB
1	0	0	0	Latch Counter 1 (See Counter Latch Command)
1	0	0	1	Read/Write Counter 2 LSB Only
1	0	1	0	Read/Write Counter 2 MSB Only
1	0	1	1	Read/Write Counter 2 LSB then MSB
1	1	X	X	Read-Back Command (see Counter Read-Back Command)

MSB = most significant byte
LSB = least significant byte
X = don't care

A new initial count can be written into the counter any time after programming without rewriting the control word, as long as the programmed format is observed.

b7 to b4 is F<3:0>. These four bits determine the command to be performed as shown in Table 10

b3 to b1 is M<2:0>. These three bits.determine the counter's mode during Read/Write Counter Commands (see Read/Write Counter Command) or select the counter during a Read-Back Command (see Read-Back command). Bits 1-3 become "don't care" during Latch Counter Commands.

b0 is BCD. BCD selects binary coded decimal counting format during Read/Write Counter Commands. When bit 0 is set to zero, the count is binary; when bit 0 is set to one, the count is BCD. Note that during Read-Back Command this bit must be zero.

Read/Write Counter Command

When writing to a counter, two conventions must be observed:

- Each counter's Control Word must be written before the initial count is written.
- Writing the initial count must follow the format specified in the Control Word (least significant byte only, most significant byte only, or least significant byte, then most significant byte).

A new initial count can be written into the counter any time after programming without rewriting the Control Word, as long as the programmed format is observed.

During Read/Write Counter Commands M<2:0> are defined as follows:

Select Mode 0	when M2 =0 and M1 =0 and M0 = 0
Select Mode 1	when M2 =0 and M1 =0 and M0 = 1
Select Mode 2	when M1 =1 and M0 = 0 (M2 is "don't care")
Select Mode 3	when M1 =1 and M0 = 1 (M2 is "don't care")
Select Mode 4	when M2 =1 and M1 =1 and M0 = 0
Select Mode 5	when M2 =1 and M1 =1 and M0 = 1

Latch Counter Command

When a Latch Counter Command is issued, the counter's output latches (COL and COH) latch the current state of the CE. COL and COH remain latched until read by the CPU, or the counter is reprogrammed. The output latches then return to a "transparent" condition. In this condition, the latches are enabled and the contents of the CE may be read directly.

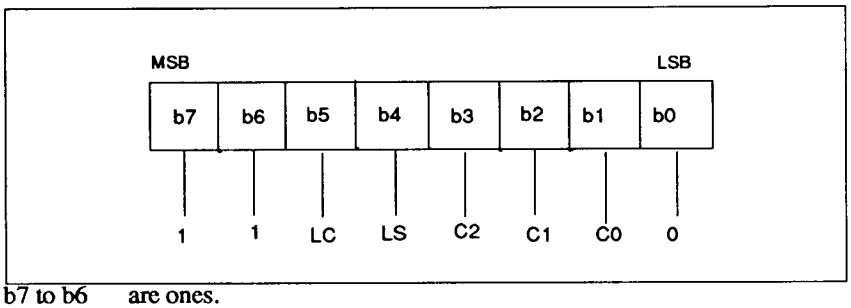
Latch Counter Commands may be issued to more than one counter before reading the first counter to which the command was issued. Also, multiple Latch Counter Commands issued to the same counter without reading the counter cause all but the first command to be ignored.

Read-Back Command

The Read-Back Command allows the user to check the count value, mode, and state of the OUT signal and Null Count Flag of the selected counter(s).

The format of the Read-Back Command is shown in Figure 15.

Figure 15 *Read-Back Command Format*



b7 to b6 are ones.

b5 is LC. Writing a zero in bit 5 causes the selected counter(s) to latch to the state of the CE in COL and COH.

b4 is LS. Writing a zero in bit 4 causes the selected counter(s) to latch the current condition of its Control Register, Null Count, and Output into the Status Register. The next read of the Counter results in the contents of the Status Register being read (see Status Read).

b3—b1 is C<2:0>. Writing a one in bit 3 causes Counter 3 to latch one or both of the registers specified by LC and LS. The same is true for bits 2 and 1 except that they enable Counters 1 and 0 respectively.

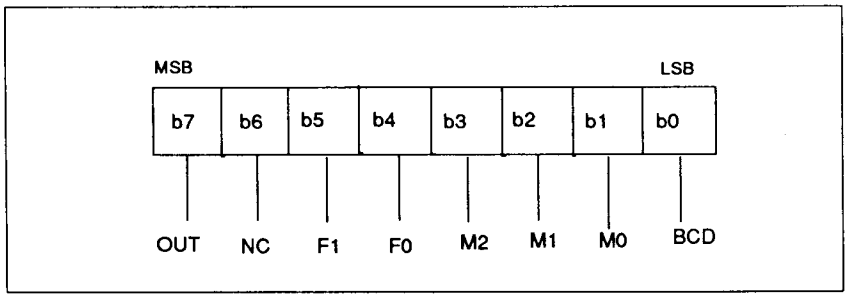
Each counter's latches remain latched until either the latch is read or the counter is reprogrammed.

If LS = LC = 0, status is returned on the next read from the counter. The next one or two reads (depending on whether the counter is programmed to transfer one or two bytes) from the counter result in the count being returned.

Status Byte

Figure 16 shows the format for the status byte command.

Figure 16 *Status Byte Command*



b7 is OUT. Bit 7 contains the state of the OUT signal of the counter.

b6 is NC. Bit 6 contains the condition of the Null count Flag. This flag is used to indicate that the contents of the CE are valid. NC is set to a one during a write to the Control Register or the counter. NC is cleared to a zero whenever the counter is loaded from the counter input registers.

b5-b4 is F<1:0>. Bits 4-5 contain the F0 and F1 Command bits, which were written to the Command Register of the counter during initialization. This information is useful when determining whether the high byte, the low byte, or both must be transferred during counter read/write operations.

b3-b1 is M<2:0>. These bits reflect the mode of the counter and are interpreted in the same manner as in Write Command operations.

b0 BCD. Bit 0 indicates the CE is operating in BCD format.

Counter Operation

Due to the previously stated restrictions in Counter 0 and Counter 1, Counter 2 is used as the example in describing counter operation, but the description of Mode 0, 2, 3, and 4 is relevant to all counters.

The following terms are defined for describing CTC operation.

- TMRCLK pulse—A clock equivalent to OSC1 (14.31818MHz) divided by 12.
- trigger—The rising edge of the GATE2 input.
- counter load—The transfer of the 16-bit value in CIL and CIH to the CE.
- initialized—A Control Word written and the Counter Input Latches loaded.

Counter 2 operates in one of the following modes.

- Mode 0 - Interrupt on Terminal Count

Writing the Control Word causes OUT2 to go low and remain low until the CE reaches zero, at which time it returns to high and remains high until a new count or Control Word is written. Counting is enabled when GATE2 = 1. Disabling the count has no effect on OUT2. The CE is loaded with the first TMRCLK pulse after the Control Word and initial count are loaded. When both CIL and CIH are written, the CE is loaded after CIH is written (see Write Operations). This TMRCLK pulse does not decrement the count (for an initial count of N, OUT2 does not go high until N+1 TMRCLK pulses after initialization). Writing a new initial count to the counter reloads the CE on the TMRCLK pulse and counting continues from the new count.

If an initial count is written with GATE2 = 0, it is still loaded on the next TMRCLK pulse but counting does not begin until GATE2 = 1. Therefore, Out2 goes high N TMRCLK pulses after GATE2 = 1.

- Mode 1 - Hardware Re-Triggerable One-Shot

Writing the Control Word causes OUT2 to go high initially. Once initialized, the counter is armed and a trigger causes OUT2 to go low on the next TMRCLK pulse. OUT2 then remains low until the counter reaches zero. An initial count of N results in a one-shot pulse N TMRCLK cycles long.

Any subsequent triggers while OUT2 is low causes the CE to be reloaded, extending the length of the pulse. Writing a new count to CIL and CIH does not affect the current one-shot unless the counter is re-triggered.

- Mode 2 - Rate Generator

Mode 2 functions as a divide-by-N counter, with OUT2 as the carry. Writing the Control Word during initialization sets OUT2 high.

When the initial count is decremented to one, OUT2 goes low on the next TMRCLK pulse. The following TMRCLK pulse returns OUT2 high, reloads the CE, and the process is repeated. In Mode 2, the counter continues counting (if GATE2 = 1) and generates an OUT2 pulse every N TMRCLK cycles. Note that a count of one is illegal in Mode 2.

GATE2 = 0 disables counting and forces OUT2 high immediately. A trigger reloads the CE on the TMRCLK pulse. Thus GATE2 can be used to synchronize the counter to external events.

Writing a new count while counting does not affect current operation unless a trigger is received. Otherwise, the new count is loaded at the end of the current counting cycle.

- Mode 3 - Square Wave Generator

Mode 3 is similar to Mode 2 in every respect except for the duty cycle of OUT2. OUT2 is set high initially and remains high for the first half of the count. When the first half of the initial count expires, OUT2 goes low for the remainder of the count. If the counter is loaded with an even count, the duty cycle of OUT2 is 50% (high = low = N/2). For odd count values, OUT2 is high one TMRCLK cycle longer than it is low. Therefore, high = (N+1)/2 and low = (N-1)/2.

- Mode 4 - Software Triggered Strobe

Writing the Control Word causes OUT2 to go initially. Expiration of the initial count causes OUT2 to go low for one TMRCLK cycle. GATE2 = 0 disables counting but has no effect on OUT2. Also, a trigger does not reload the CE.

The counting sequence is started by writing the initial count. The CE is loaded on the TMRCLK pulse after initialization. The CE begins decrementing one TMRCLK pulse later. OUT2 goes low for one TMRCLK cycle, (N+1) cycles after the initial count is written. If a new initial count is written during a counting sequence, it is loaded into the CE on the next TMRCLK pulse and the sequence continues from the new count. This allows the sequence to be re-triggerable by software.

- Mode 5 - Hardware Triggered Strobe

Writing the Control Word causes OUT2 to go high initially. Counting is started by trigger. The expiration of the initial count causes OUT2 to go low for one TMRCLK cycle. GATE 2 = 0 disables counting.

The CE is loaded during counting, the current counting sequence is not affected unless a trigger occurs. A trigger causes the counter to be reloaded from CIL and CIH, making the counter re-triggerable.

GATE2

In Modes 0, 2, 3, and 4 GATE2 is level sensitive and is sampled on the rising edge of TMRCLK. In Modes 1, 2, 3, and 5 the GATE2 input is rising-edge sensitive. This rising edge sets an internal flip-flop whose output is sampled on the next rising edge of TMRCLK. The flip-flop resets immediately after being sampled. Note that in Modes 2 and 3 the GATE2 input is both edge- and level-sensitive as shown in Table 11.

Table 11 *Gate Pin Function*

Mode	Condition		
	Low	Rising	High
0	Disables Counting		Enables Counting
1		a)Initiates Counting b)Resets Out Pin	
2	a) Disables Counting b) Forces high output pin	Initiates Counting	Enables Counting
3	a) Disables Counting b) Forces high output pin	Initiates Counting	Enables Counting
4	Disables Counting		Enables Counting
5		Initiates Counting	

Interrupt Controller — 8259 Compatible

The 82C836 incorporates two programmable interrupt controllers that are compatible with the Intel 8259A. The controllers accept requests from peripherals, resolve priority on pending interrupts and interrupts in service, issue an interrupt request to the CPU, and provide a vector that is used as an index by the CPU to determine which interrupt service routine to execute.

A variety of priority assignment modes are provided, which can be reconfigured at any time during system operation, allowing the complete interrupt subsystem to be restructured, based on the system requirements. The controllers are cascaded in a fashion compatible with the IBM PC AT.

Table 12 shows the interrupt levels used by the system board and I/O channel. The interrupts are shown by priority, starting with the highest level.

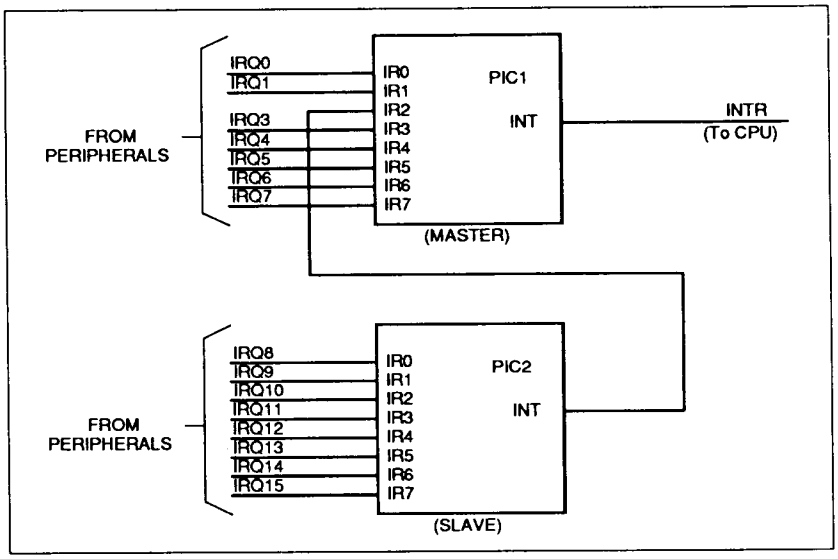
Table 12 *Interrupt Levels for System Board*

Interrupt	System Board	I/O Channel
NMI	Parity Check	IOCHCK
IRQ0	Timer	Not Available
IRQ1	Keyboard	Not Available
IRQ8*	Real-Time Clock	Not Available
IRQ9	Not Used	Available
IRQ10	Not Used	Available
IRQ11	Not Used	Available
IRQ12	Not Used	Available
IRQ13	Co-Processor	Not Available
IRQ14	Not Used	Available
IRQ15	Not Used	Available
IRQ3	Serial Port 2	Available
IRQ4	Serial Port 1	Available
IRQ5	Parallel Port 2	Available
IRQ6	Hard Disk/Diskette	Available
IRQ7	Parallel Port 1	Available

* In internal RTC mode, IRQ8 is an internal signal.

The two devices are interconnected and must be programmed to operate in Cascade Mode (see Figure 17) for proper operation of all 16 interrupt channels. INTC1 is located at addresses 020H-021H and is configured for Master operation (defined below) in Cascade Mode. INTC2 is a Slave device (defined below) and is located at 0A0h-0A1H. The Interrupt Request output signal from INTC2 (INT) is internally connected to the interrupt request input Channel 2 IR2) of INTC1. The address decoding and Cascade interconnection matches that of the IBM PC AT.

Figure 17 *Cascaded Interrupt Controllers*



Two additional interconnections are made to the interrupt request inputs of the interrupt controllers. The output of Timer 0 in the Counter/Timer subsystem is connected to Channel 0 (IR0) of INTC1. An Interrupt request from the Real Time Clock is connected to Channel 0 (IR0) of INTC2.

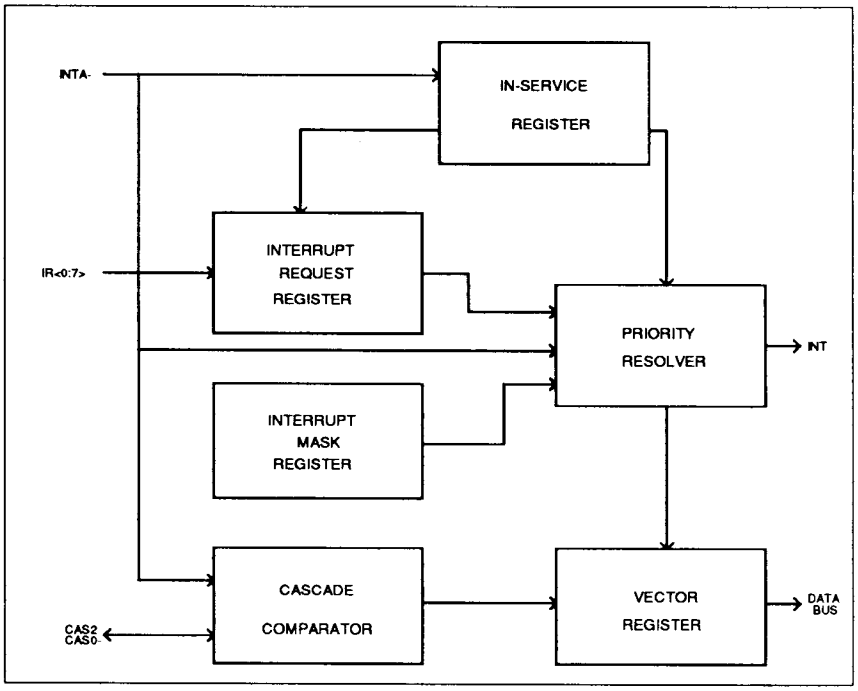
The following description of the Interrupt Subsystem pertains to both INTC1 and INTC2 unless otherwise noted. Wherever register addresses are used, the address for the INTC1 register is listed first, and the address for the INTC2 register follows in parenthesis, e.g., 020H (0A0H).

Controller Operation

Figure 18 is a block diagram of the major elements in the interrupt controller. The Interrupt Request Register (IRR) is used to store requests from all the channels that are requesting service. Interrupt Request Register bits are labeled using the Channel Name IR<0:7>. The In-Service register (ISR) contains all the channels that are currently being serviced (more than one channel can be in service at a time). In-Service Register bits are labeled IS<0:7>. The Interrupt Mask Register (IMR) allows the CPU to disable any or all of the interrupt channels. The Priority Resolver evaluates inputs from the above three registers, issues an interrupt request, and latches the corresponding bit into the In-Service Register. During interrupt acknowledge cycles, a master controller outputs a code to the slave device that is compared in the Cascade Buffer/Comparator with a three bit ID code previously written. If a match occurs in the slave controller, it generates an interrupt vector. The contents of the Vector Register

are used to provide the CPU with an interrupt vector during Interrupt Acknowledge (INTA) cycles.

Figure 18 *Interrupt Controller Block Diagram*



Interrupt Sequence

The 82C836 allows the CPU to perform an indirect jump to a service routine in response to a request for service from a peripheral device. The indirect jump is based on a vector that is provided by the 82C836 on the second of two CPU generated INTA cycles (the first INTA cycle is used for resolving priority; the second cycle is used for transferring the vector to the CPU). The events that occur during an interrupt sequence are as follows:

1. One or more of the interrupt requests (IR7-IR0) becomes active, setting the corresponding IRR bit(s).
2. The interrupt controller resolves priority based on the state of IRR, IMR, and ISR and asserts the INTR output if appropriate.
3. The CPU accepts the interrupt and responds with an INTA cycle.
4. During the first INTA cycle, the highest priority ISR bit is set and the corresponding IRR bit is reset. The internal Cascade address is generated and LD<0:7> outputs remain tri-stated.
5. The CPU executes a second INTA cycle, during which the 82C836 drives an 8-bit vector onto the data pins LD<0:7>, which is in turn latched by the CPU. The format of this vector is shown in Table 13. Note that V<3:7> in Table 12 is programmable by writing to Initialization Control Word 2. See *Initialization Command Words* and Figure 19.
6. At the end of the second INTA cycle, the ISR bit is cleared if the Automatic End-Of-Interrupt mode is selected (see End-Of-Interrupt section

below). Otherwise, the ISR bit must be cleared by an End-Of-Interrupt (EOI) command from the CPU at the end of the interrupt service routine.

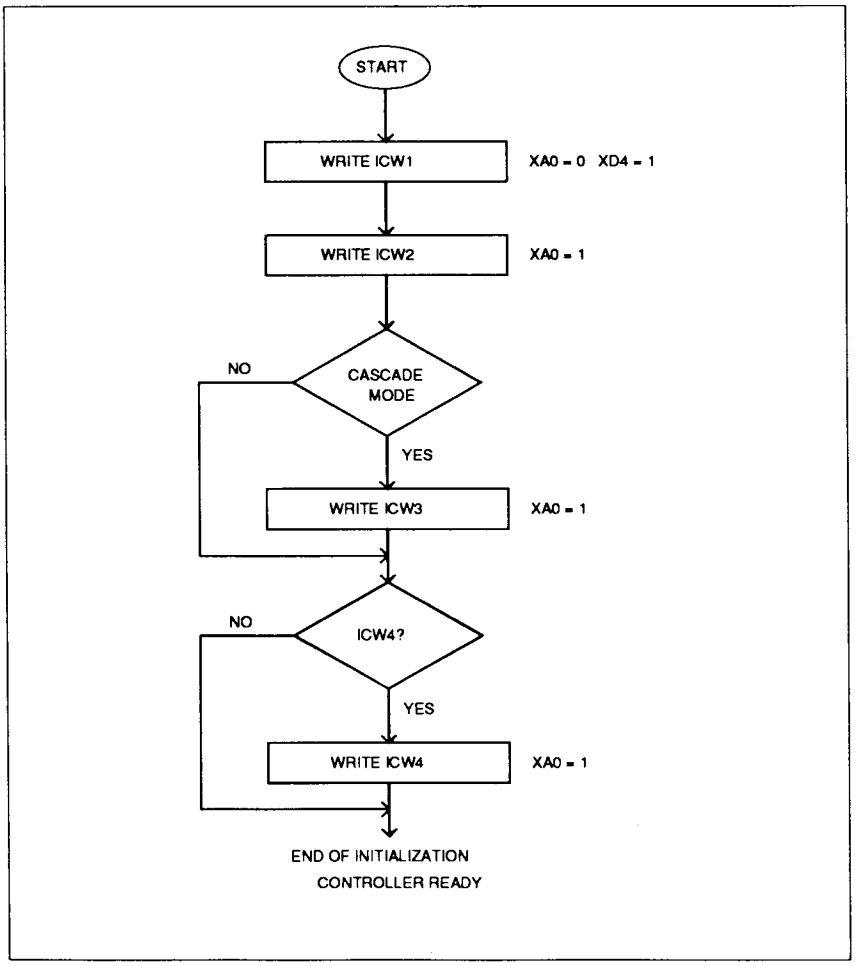
If no interrupt request is present at the beginning of the first INTA cycle (i.e., a spurious interrupt), INTC1 issues an interrupt level 7 vector during the second INTA cycle.

Table 13 *Interrupt Vector Format*

IR7	V7	V6	V5	V4	V3	1	1	1
IR6	V7	V6	V5	V4	V3	1	1	0
IR5	V7	V6	V5	V4	V3	1	0	1
IR4	V7	V6	V5	V4	V3	1	0	0
IR3	V7	V6	V5	V4	V3	0	1	1
IR2	V7	V6	V5	V4	V3	0	1	0
IR1	V7	V6	V5	V4	V3	0	0	1
IR0	V7	V6	V5	V4	V3	0	0	0

End-Of-Interrupt (EOI)

Figure 19 *Initialization Sequence*



EOI is defined as the condition that causes an ISR bit to be reset. Determination of which ISR bit is to be reset can be done by a CPU command (specific EOI) or the Priority Resolver can be instructed to clear the highest priority IST bit (non-specific EOI).

The 82C836 can determine the correct ISR bit to reset when operated in modes that do not alter the fully nested structure, since the current highest priority ISR bit is necessarily the last level acknowledged and serviced. In conditions where the fully nested structure is not preserved, a specific EOI must be generated at the end of the interrupt service routine. An ISR bit that is masked, in Special Mask Mode by a IMR bit, is not cleared by a non-specific EOI command. Optionally, the interrupt controller can generate an Automatic End-Of-Interrupt (AEOI) on the trailing edge of the second INTA cycle.

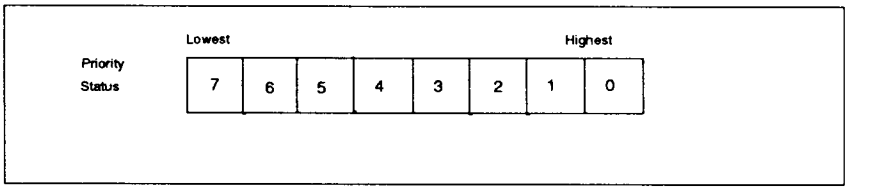
Priority Assignment

Assignment of priority is based on an interrupt channel's position relative to the other channels in the interrupt controller. After the initialization sequence, IR0 has the highest priority, IR7 has the lowest, and priority assignment is fixed (Fixed Priority Mode). Priority assignment can be rotated either manually (Specific Rotation Mode) or automatically (Automatic Rotation Mode) by programming Operational Command Word 2 (OCW2).

Fixed Priority Mode

This is the default condition that exists unless rotation (either manual or automatic) is enabled, or the controller is programmed for Polled Mode. In Fixed Priority Mode, interrupts are fully nested with priority assigned as shown in Table 14.

Table 14 *Fixed Priority Mode*

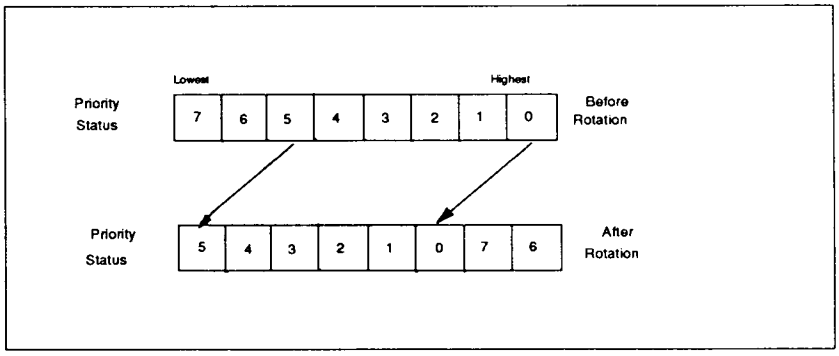


Nesting allows interrupts with higher priorities to generate interrupt requests prior to the completion of the interrupt in service. When an interrupt is acknowledged, priority is resolved, the highest priority request's vector is placed on the bus, and the ISR bit for that channel is set. This bit remains set until an EOI (automatic or CPU generated) is issued to that channel. While the ISR bit is set, all interrupts of equal or lower priority are inhibited. Note that a higher priority interrupt that occurs during an interrupt service routine, is acknowledged only if the CPU has internally re-enabled interrupts.

Specific Rotation Mode

Specific Rotation allows the system software to re-assign priority levels by issuing a command that redefines the highest priority channel. Table 15 illustrates the result of a specific rotation command, assigning highest priority to Channel 5.

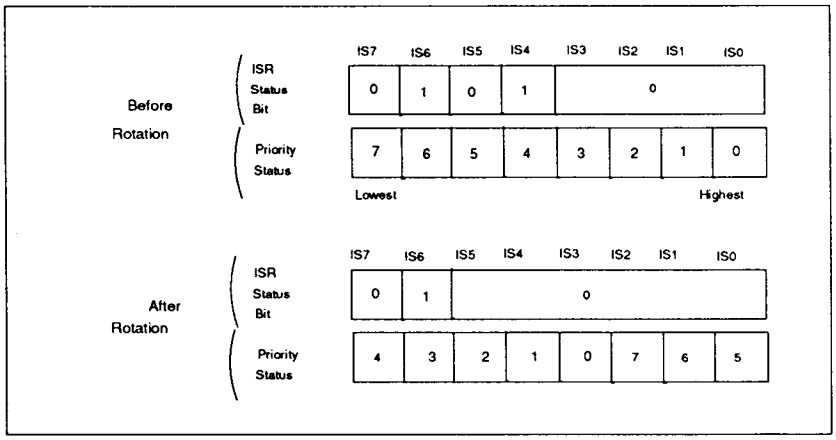
Table 15 *Specific Rotation Mode*



Automatic Rotation Mode

In applications in which a number of equal priority peripherals are requesting interrupts, Automatic Rotation may be used to equalize the priority assignment. In this mode a peripheral, after being serviced, is assigned the lowest priority. All peripherals connected to the controller are serviced at least once in eight interrupt requests to the CPU from the controller. Automatic rotation occurs, if enabled, due to the occurrence of EOI (automatic or CPU generated). This is illustrated inTable 16.

Table 16 *Automatic Rotation Mode*



Programming the Interrupt Controller

Two types of commands are used to control the 82C836 interrupt controllers: Initialization Command Words (ICWs) and Operational Command Words (OCWs).

Initialization Command Words

The initialization process consists of writing a sequence of four bytes to each interrupt controller. The initialization sequence is started by writing the first Initialization Command Word (ICW1) to address 020H (0A0H) with a one on bit 4 of the data byte. The interrupt controller interprets this as the start of the initialization sequence and does the following:

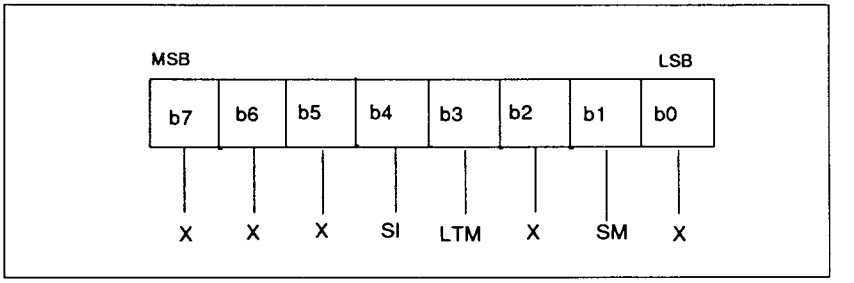
1. The Initialization Command Word Counter is reset to zero.

2. ICW1 is latched into the device.
3. Fixed Priority Mode is selected.
4. IR7 is assigned the highest priority.
5. The Interrupt Mask Register is cleared.
6. The Slave Mode Address is set to seven.
7. Special Mask Mode is disabled.
8. The IRR is selected for Status Read operations.

The next three I/Os write to address 021H (0A1H) will load ICW2-ICW4. See Figure 19 for a flow chart of the initialization sequence. The initialization sequence can be terminated at any point (all four bytes must be written for the controller to be properly initialized) by writing to address 020H (0A0H) with a zero in data bit 4. Note: this causes OCW2 or OCW3 to be written.

Figure 20 shows the structure of ICW1.

Figure 20 *ICW1 (Address 020H (0A0H)), Write Only*



b7 to b5 are ignored.

b4 is SI. Bit 4 indicates to the interrupt controller that an Initialization Sequence is starting and must be a one to write ICW1.

b3 is LTM. Bit 3 selects level or edge triggered inputs to the IRR. If a one is written to LTM, a high-level on the IRR input generates an interrupt request. The IR must be active until the first INTA cycle is started to generate the proper interrupt vector (an IR7 vector is generated if the IRR input is de-asserted early), and the IR must be removed prior to EOI to prevent a second interrupt from occurring.

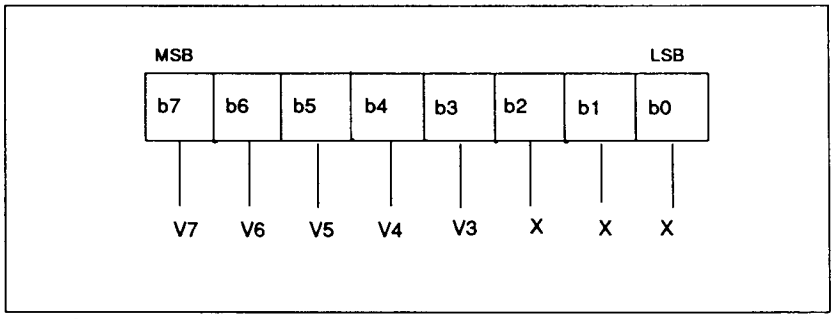
b2 is ignored

b1 is SM. Bit 1 selects between Single Mode and Cascade Mode. Single Mode is used whenever only one interrupt controller (INTC1) is used and is not recommended for this device. Cascade Mode allows the two interrupt controllers to be connected through IR2 of INTC1. INTC1 allows INTC2 to generate its own interrupt vectors if Cascade Mode is selected and the highest priority IR pending is from an INTC2 input. INTC1 andINTC2 must be programmed for Cascade Mode for both devices to operate

b0 is ignored

Figure 21 shows ICW2.

Figure 21 *ICW2 (Address 021H (0A1H)), Write Only.*

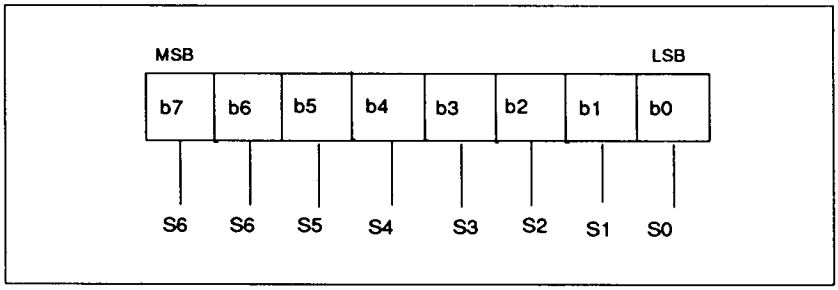


b7 to b3 are V<7:3>. These bits are the upper five bits of the interrupt vector and are programmable by the CPU. The lower three bits of the vector are generated by the Priority Resolver during INTA. INTC1 and INTC2 need not be programmed with the same value in ICW2.

b2 to b0 are ignored.

Figure 22 shows the ICW3 format for INTC1.

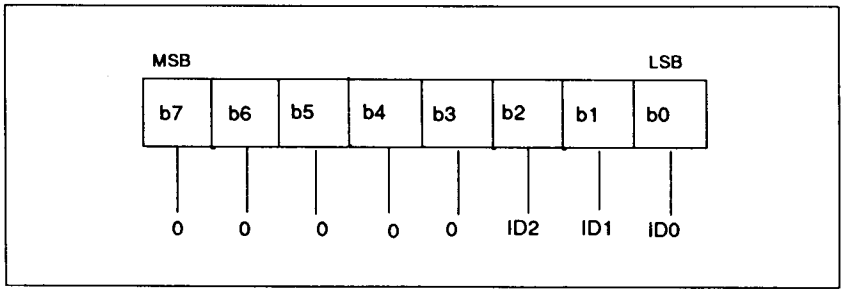
Figure 22 *ICW3 Format for INTC1 (Address 021H), Write On.ly.*



b7-b0 are S<7:0>. Select which IR inputs have Slave Mode controllers connected. ICW3 in INTC1 must be written with 04H for INTC2 to function.

Figure 23 shows the ICW3 format for INTC2.

Figure 23 *ICW3 Format for INTC2 (Address 0A1H), Write Only*



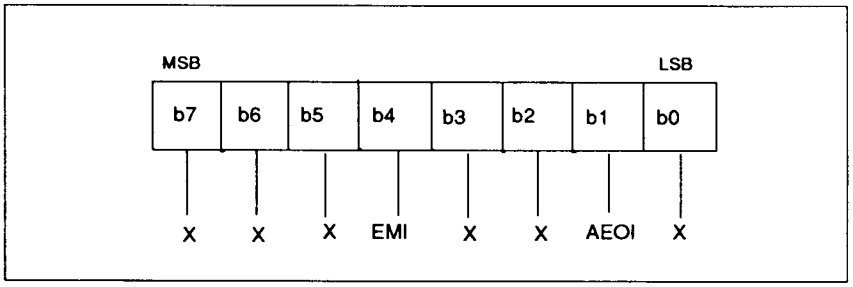
b7 to b3 are zeroes.

b2-b0 is ID2-ID0. Determine the Slave Mode address the controllers will respond to during the cascaded INTA sequence. ICW3 in INTC2

should be written with 02H for cascade Mode operation. Note that b7-b3 should be zero.

Figure 24 shows the ICW4 format.

Figure 24 *ICW4 (Address 021H (0A1H)), Write On.ly.*



b4 is EMI. Bit 4 enables multiple interrupts from the same channel in fixed Priority Mode. This allows INTC2 to fully nest interrupts, when Cascade Mode with Fixed Priority Mode are both selected, without being blocked by INTC1. Correct handling of this mode requires the CPU to issue a non-specific EOI command to INTC2 and to check its In-Service Register for zero when exiting an interrupt service routine. If zero, a non-specific EOI command should be sent to INTC1. If non-zero, no command is issued.

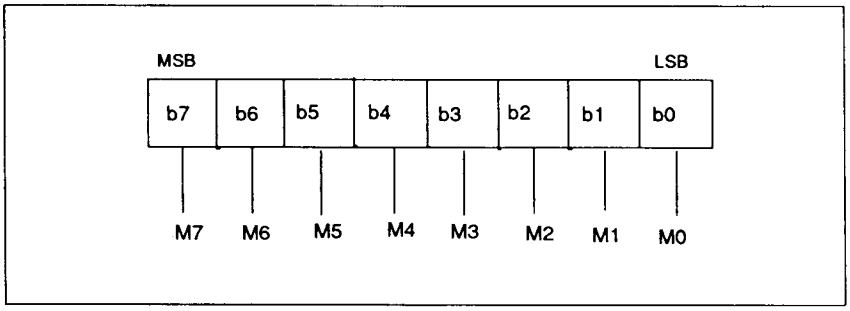
b1 is AEOI—Auto End-Of-Interrupt is enabled when ICW4 is written with a zero in both. The interrupt controller performs a non-specific EOI on the trailing edge of the second INTA cycle. Note that this function should not be used in a device with fully nested interrupts unless the device is a cascade Master.

b0 does not matter.

Operational Command Words

Operational Command Word One (OCW1) is located at address 021H (0A1H) and may be written any time the controller is in Initialization Mode. Operational Command Words Two and Three (OCW2 and OCW3) are located at address 020H (0A0H). Writing to address 020H (0A0H) with a zero in bit 4 places the controller in operational mode and loads OCW2 (if data bit 3 = 0) or OCW3 (if data bit 3 = 1). Refer to Figure 25.

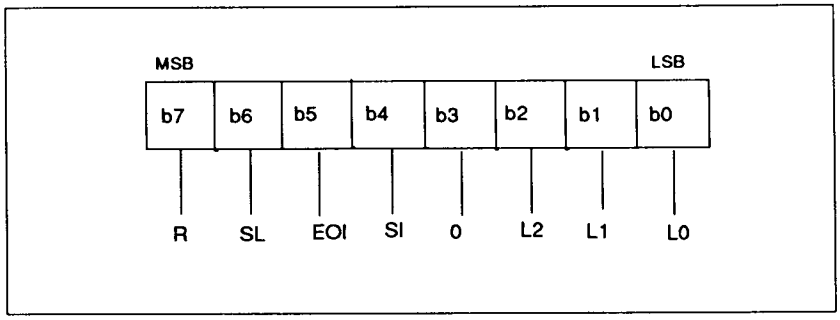
Figure 25 *OCW1 (Address 021H (0A1H)), Read/Write*



b7-b0 is M<7:0>. These bits control the state of the Interrupt Mask Register. Each Interrupt Request can be masked by writing a one in the appropriate bit position (M0 controls IR0 etc.). Setting an IMR bit has no effect on lower priority requests. All IMR bits are cleared by writing ICW1.

Figure 26 shows the structure for OCW2.

Figure 26 *OCW2 (Address 020H (0A0H)), Write Only*



b7 is R. This bit together with SL and EOI selects operational function. Writing a one in bit 7 causes one of the rotate functions to be selected.
Rotate on auto EOI enable.* when R = 1 and SL = 0 and EOI = 0.
Rotate on non-specific EOI when R = 1 and SL = 0 and EOI = 1
Specific Roate Command when R = 1 and SL = 1 and EOI = 0
Rotate on specific EOI when R = 1 and SL = 1 and EOI = 1

*This function is disabled by writing a zero to all three bit positions.

b6 is SL. This bit in conjunction with R and EOI selects operational function. Writing a one in this bit position causes a specific or immediate function to occur. All specific commands require L2-L0 to be valid except no operation.

No Operation when R =0 and SL = 1 and EOI = 0
Specific EOI Command when R = 0 and SL = 1 and EOI = 1
Specific Rotate Command when R = 1 and SL = 1 and EOI = 0
Rotate on Specific EOI when R = 1 and SL = 1 and EOI = 1

b5 is EOI. This bit in conjunction with R and SL selects operational function. Writing a one in this bit position causes a function related to EOI to occur.

Non-specific EOI Command when R = 0 and SL = 1 and EOI = 0
Specific EOI Command when R = 0 and SL = 1 and EOI = 1
Rotate on Non-specific EOI when R = 1 and SL = 0 and EOI = 1
Rotate on Specific EOI when R = 1 and SL = 1 and EOI = 1

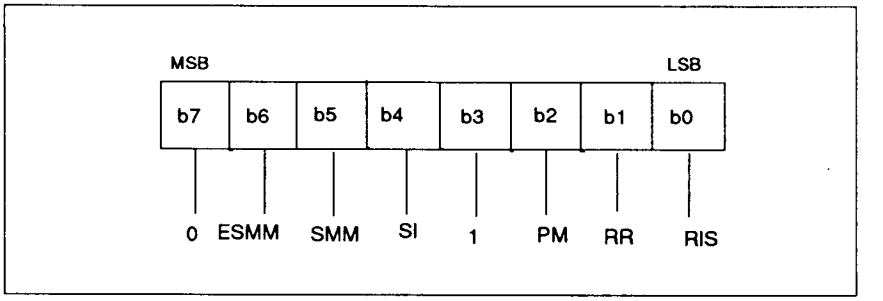
b4 is SI. Writing a zero in this bit position takes the interrupt controller out of initialize mode and writes OCW2 or OCW3.

b3 is zero

b2 to b0 is L<2:0>. These three bits are internally decoded to select which interrupt channel is to be affected by the Specific command. L<0:2> must be valid during three of the four specific cycles (see SL above).

Figure 27 shows the structure of OCW3.

Figure 27 *OCW3 (Address 020H (0A0H)), Write Only*



b6 is ESMM. Writing a one in this bit position enables the Set/Reset Special Mask Mode function controlled by bit 5 (SMM). ESMM allows the other functions in OCW3 to be accessed and manipulated without affecting the Special Mask Mode state.

b5 is SMM. If ESMM and SMM both are written with a one, the Special Mask Mode is enabled. Writing a one to ESMM and a zero to SMM disables Special Mask Mode. During Special Mask Mode, writing a one to any bit position inhibits interrupts and a zero enables interrupts on the associated channel by causing the Priority Resolver to ignore the condition for the ISR.

b4 is SI. See SI above (OCW2).

b2 is PM—Polled Mode is enabled by writing a one to bit 2 of OCW3 causing the 82C836 to perform the equivalent of an INTA cycle during the next I/O read operation to the controller. The byte read during this cycle sets bit 7 if an interrupt is pending. If bit 7 of the byte is set, the level of the highest pending request is encoded on bits 2-0. The IRR remains frozen until the read cycle is completed, at which time the PM bit is reset.

b1 is RR. When the RR bit is one, reading the Status Port at address 020H (0A0H) causes the contents of IRR or ISR (determined by RIS) to be placed on XD7-XD0. Asserting PM forces RR reset.

b0 is RIS. This bit selects between the IRR and the ISR during Status Read operations if RR = 1. IRR is selected if this bit is set to one. ISR is selected if this bit is set to zero.

DMA Controller — 8237 Compatible

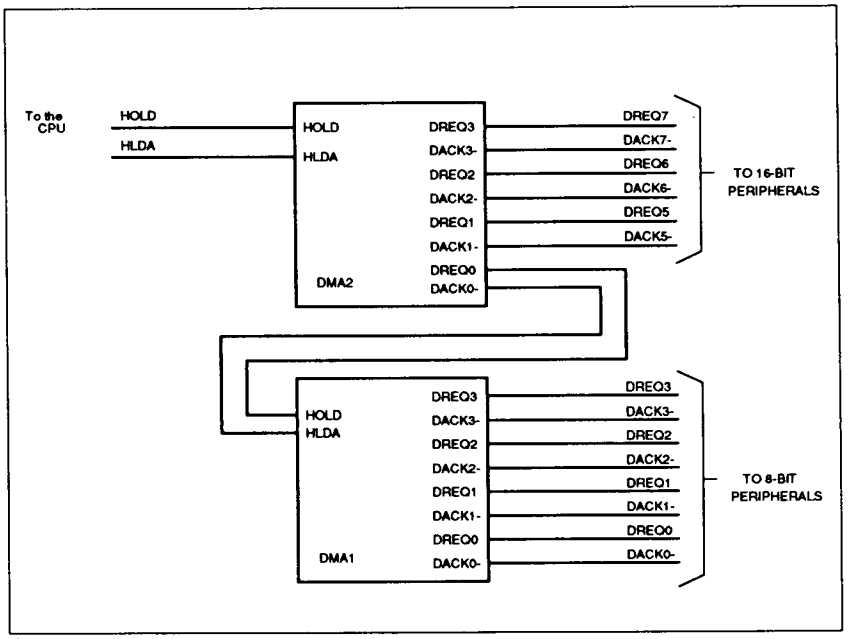
Memory refresh and DMA functions are implemented within the 82C836 as independent bus masters, i.e., the refresh controller is separate from the 8237-compatible DMA controllers. The 82C836 arbitrates refresh and DMA functions with internal logic, and gains control of the local bus via the HOLD/HLDA protocol of the 80386sx.

The 82C836 contains two DMA controllers that are compatible with the Intel 8237. Each controller is a four-channel DMA device that can generate the memory addresses and control signals necessary to transfer information directly between a peripheral device and memory. This allows efficient information transfer with little CPU intervention and a minimum of bus overhead.

The two DMA controllers are internally cascaded to provide four DMA channels for transfers to 8-bit peripherals (DMA1) and three channels for transfers to 16-bit peripherals (DMA2). DMA2 Channel 0 provides the cascade interconnection for the two DMA devices, thereby maintaining IBM PC AT compatibility.

Figure 28 shows how the two controllers are cascaded. DMA channels 0-3 are used for 8-bit transfers, while channels 5-7 are used for 16-bit transfers. DMA operations are allowed within the full range of 16MB memory through the use of DMA page registers.

Figure 28 *Cascaded DMA Controllers*



The DMA clock, which is internal to the 82C836, controls DMA transfer rate and timing. It is derived from BUSCLK and is selectable as either BUSCLK/2 (default, AT-compatible) or BUSCLK (see *Clock Generation*). Wait states during DMA operations are programmed with internal configuration register 01H, bits 5-2.

Refresh, which occurs at 15 µs intervals (nominally), is determined by counting down the 1.19 MHz (OSC1/12) clock. The refresh period is set by programming timer channel 1 of the 8254 timer/counter.

Table 17 shows the usage of DMA levels on the I/O channel. The DMA requests are shown by priority, starting with the highest level.

DMA cycle length control is provided internally in the 82C836 allowing independent control for both 8-bit and 16-bit cycles. This is done through the programmable registers, which can extend command signals or insert wait states.

Each DMA channel has a pair of 16-bit counters and a reload register for each counter. The 16-bit counters allow the DMA to transfer blocks as large as 65,536 words. The register associated with each counter allows the channel to re-initialize without reprogramming. The following description of the DMA subsystem pertains to both DMA1 and DMA2 unless otherwise noted.

Table 17 *DMA Request Levels for each I/O Channel*

DMA Level	System Board	I/O Channel
DRQ0	Not Used	Available
DRQ1	Not Used	Available
DRQ2	Not Used	Diskette Drive
DRQ3	Not Used	Available
DRQ4	Not Used	Available
DRQ5	Not Used	Available
DRQ6	Not Used	Available
DRQ7	Not Used	Available

Although the 8237 provides a "memory-to-memory" mode of transfer, this 8237-compatible mode is not usable in PC AT-compatible architectures, including SCATsx. The memory-to-memory mode requires the use of the temporary data holding register in the 8237, which is not accessible during DMA because of the way the various bus buffers are controlled. Furthermore, AT-compatible architectures such as SCATsx do not contain steering logic to allow memory write data to come from the 8237 instead of an I/O resource. It is possible that 8-bit memory-to-memory transfers might work successfully anyway (accidentally) if there is sufficient bus capacitance to preserve the data between the memory read cycle and memory write cycle, but this approach cannot be considered reliable. Information concerning memory-to-memory transfer is included in this data sheet only because the 82C836 implements the same 8237-compatible functionality found in the PC AT.

DMA Operation

During normal operation of the 82C836, the DMA subsystem is in either an Idle condition, a Program condition, or an Active condition. In the Idle condition the DMA controller executes cycles consisting of only one state. The Idle state, SI, is the default condition, and the DMA remains in this condition unless the device has been initialized and one of the DMA requests is active or the CPU attempts to access one of the internal registers.

When a DMA request becomes active, the device enters the Active condition and issues a hold request to the system. Once in the Active condition, the 82C836 generates the necessary memory addresses and command signals to accomplish a memory-to-I/O or I/O-to-memory transfer. Timing for Memory-to-memory transfer can also be generated (but without access to the data holding register). Memory-to-I/O and I/O-to-memory transfers take place in one bus cycle, while memory-to-memory transfer timing requires two bus cycles. During transfers between memory and I/O, data is presented on the system bus by either memory or the requesting device, and the transfer is completed in one bus cycle. Memory-to-memory transfers, however, require that the DMA store data from the read operation in an internal register, which is not accessible in AT-compatible architectures.

During transfers between memory and I/O, two commands are activated during the same bus cycle. In the case of a memory-to-I/O transfer, the 82C836 asserts both XMEMR- and IOW-, allowing data to be transferred directly to the requesting device from memory. Note that the 82C836 neither latches data from, nor drives data out, on this type of cycle.

The number of clock cycles required to transfer a word of data may be varied by programming the DMA, or may be optionally extended by the peripheral device. During an Active cycle the DMA goes through a series of states. Each state is one DMA clock cycle in length, and the number of states in a cycle varies depending on how the device is programmed and the type of cycle being performed. The states are labeled S0-S4. They are explained in detail in the section titled *Active Condition*.

Idle Condition

When no device is requesting service, the DMA is in an Idle condition that keeps the state machine in the SI state. During this time, the 82C836 samples the DREQ input pins every clock cycle. The internal select from the top level decoder and HLDA are also sampled at the same time to determine if the CPU is attempting to access the internal registers. When either of these situations occurs, the DMA exits the Idle condition. Note that the Program condition has priority over the Active condition since a CPU cycle has already started.

Program Condition

The Program condition is entered whenever HLDA is inactive and internal select is active. The internal select is derived from the top-level decode described earlier. During this time, address lines A0-A3 become inputs if DMA1 is selected, or A1-A4 become inputs if DMA2 is selected. NOTE: When DMA2 is selected, A0 is ignored. These address inputs are used to select the DMA controller registers that are to be read or written to. Due to the large number of internal registers in the DMA subsystem, an internal flip-flop is used to supplement the addressing of the count and address registers. This bit is used to select between the high and low bytes of these registers. The flop-flop toggles each time a read or write occurs to any of the word count or address registers in the DMA. The internal flop-flop is cleared by a hardware RESET or a Master Clear command and may be set or cleared by the CPU issuing the appropriate command.

Special commands are supported by the DMA subsystem in the Program condition to control the device. These commands do not make use of the data bus but are derived from a set of addresses, the internal select, and IOW- or IOR-. These commands are Master Clear, Clear Register, Clear Mode Register Counter, Set, and Clear Byte Pointer Flip-Flop.

The 82C836 enables programming whenever HLDA has been inactive for one DMA clock cycle. It is the responsibility of the system to ensure that programming and HLDA are mutually exclusive. Erratic operation of the 82C836 can occur if a request for service occurs on an unmasked channel that is being programmed. The channel should be masked or the DMA disabled to prevent the 82C836 from attempting to service a device with a channel which is partially programmed.

Active Condition

The 82C836 DMA subsystem enters the Active condition whenever a software request occurs or a DMA request on an unmasked channel occurs, and the device is not in the Program condition. The 82C836 then begins a DMA transfer cycle.

In an I/O-to-memory cycle, for example, after receiving a DREQ, the 82C836 asserts HOLD to the CPU. On the next clock cycle, the DMA exits Idle and enters state S0, where it remains until HLDA is returned. After detecting HLDA, the DMA enters state S1, during which the DMA controller generates the memory address, resolves priority and issues DACK on the highest priority channel requesting service. The DMA then proceeds to state S2, at which time the 82C836 asserts XIOR-. Next, the device transitions into S3, where the XMEMW- command is asserted. This is followed by a minimum of one DMA Wait State, SW, where the 82C836 remains until the wait-state counter has decremented to zero *and* IOCHRDY is true. At least one SW always occurs, but the S3 state can be deleted if Compressed Timing is selected (see *Compressed Timing*). Once a ready condition is detected, the DMA enters S4, where both commands are de-asserted. In Burst Mode and Demand Mode (discussed below), subsequent cycles begin in S2 unless the intermediate address bits require updating. In these subsequent cycles, the lower address bits are changed in S2.

See *System Timing Relationships* for more detailed timing information and diagrams.

The DMA can be programmed on a channel-by-channel basis to operate in one of four modes. The four modes are described below:

- **Single transfer Mode** — This mode directs the DMA to execute only one transfer cycle at a time. DREQ must be held active until DACK becomes active. If DREQ is held active throughout the cycle, the 82C836 de-asserts HRQ and releases the bus after the transfer is complete. After HLDA is inactive, the 82C836 again asserts HRQ and executes another cycle on the same channel unless a request from a higher priority channel is received. In this mode, the CPU is allowed to execute at least one bus cycle between transfers. Following each transfer, the word count is decremented and the address is incremented or decremented. When the word count decrements from 0000H to FFFFH, the terminal count bit in the status register is set and a T/C pulse is generated. If the auto-initialization option is enabled, the channel re-initializes itself. If Auto-initialize is not selected, the DMA sets the DMA request bit mask and suspends transferring on the channel.
- **Block Transfer Mode** — When Block Transfer Mode is selected, the 82C836 begins transfers in response to either a DREQ or a software request. This continues until a terminal count (FFFFH) is reached, at which time TC is pulsed and the status register terminal count bit is set. In this mode DREQ need only be held active until DACK is asserted. Auto-initialization is operational in this mode also.
- **Demand Transfer Mode** — In Demand Transfer mode, the DMA begins transfers in response to the assertion of DREQ and continues until either terminal count is reached or DREQ becomes inactive. This mode is normally used for peripherals that have limited buffering availability. The peripheral can initiate a transfer and continue until its buffer capacity is exhausted. The peripheral may re-establish service by again asserting DREQ. During idle periods between transfers, the CPU is released to operate and can monitor the operation by reading intermediate values from the address and word count registers. Once DREQ is de-asserted, higher priority channels are allowed to intervene. Reaching terminal count results in the generation of a TC pulse, the setting of the terminal count bit in the status register, and auto-initialization (if enabled.)
- **Cascade Mode** — This mode is used to interconnect more than one DMA controller, to extend the number of DMA channels while preserving the

priority chain. In Cascade mode, the master DMA controller does not generate address or control signals. The DREQ and DACK signals of the master are used to interface the HRQ and HLDA signals of the slave DMA devices. Once the master has received an HLDA from the CPU in response to a DREQ caused by the HRQ from a slave DMA Controller, the master DMA controller ignores all inputs except HLDA from the CPU and DREQ on the active channel. This prevents conflicts between the DMA devices.

Figure 28, shown previously, portrays the cascade interconnection between the two levels of DMA devices. Note that Channel 0 of DMA2 is internally connected for Cascade mode to DMA1. Additional devices can be cascaded to the available channels in either DMA1 or DMA2 since cascade is not limited to two levels for DMA controllers.

When programming cascaded controllers, begin with the device which is actually generating HRQ to the system (first level device) and then proceed to the second level devices. RESET causes the DACK outputs to become active low and are placed into an inactive state. To allow the internal cascade between DMA1 and DMA2 to function correctly, the active low state of DACK should not be modified. This is because the 82C836 has an inverter between DACK0 of DMA2 and HLDA of DMA1. The first level device's DMA request mask bit prevents second level cascaded devices from generating unwanted hold requests during the initialization process.

DMA Transfers

Four types of transfer modes are provided in the 82C836 DMA subsystem. These transfer types are:

- **Read Transfer**—Read transfers move data from memory to an I/O device by generating the memory address and asserting XMEMR- and XIOW- during the same cycle.
- **Write Transfer**—Write transfers move data from an I/O device to memory by generating the memory address and asserting XIOR- and XMEMW-.
- **Memory-to-Memory Transfer**—(not usable in AT-compatible architectures). The memory-to-memory transfer in an 8237 is designed to move a block of data from one location in memory to another. DMA channels 0 and 1 may be programmed to operate as memory-to-memory channels by setting a bit in the Command Register. Once programmed to perform a memory-to-memory transfer, the process can be started by generating either a software request or an external request to channel 0. Once the transfer is initiated, Channel 0 provides the address for the source block during the memory read portion of the cycle. Channel 1 generates the address for the memory write cycle. During the read cycle, the 8237-compatible subsection attempts to latch a byte of data in the internal Temporary Register (not accessible in AT-compatible architectures). The 8237-compatible subsection then attempts to output the contents of the Temporary Register on the XD0-7 data lines during the write portion of the cycle, so that the data can subsequently be written to memory. However, AT-compatible architectures, including SCATsx, do not contain the necessary data bus steering logic to implement this data path. Channel 0 may be programmed to maintain the same source address on every cycle, so that the CPU can initialize large blocks of memory with the same value. The 82C836 continues to perform transfer cycles until Channel 1 reaches terminal count.

- **Verify Transfer**—The verify transfer is a pseudo-transfer that is useful for diagnostics. In this type of transfer, the DMA operates as if it is performing a Read or Write Transfer by generating HRQ, addresses, and DACK, but does so without asserting a command signal. Since no transfer actually takes place, IOCHRDY is ignored during Verify transfer cycles.

Auto-Initialization

Each of the four DMA channel Mode Registers contains a bit that causes the channel to re-initialize after reaching terminal count. During this process, referred to as Auto-Initialization, the Base Address and Base Word Count Registers, which were originally written by the CPU, are reloaded into the Current Address and Current Word Count Registers (both the base and current registers are loaded during a CPU write cycle). The base register remains unchanged during DMA Active cycles and can only be changed by the CPU. If the channel is programmed to auto-initialize, the request mask bit is not set upon reaching terminal count. This allows the DMA to continue operation without CPU intervention.

During memory-to-memory transfers, the Word Count Registers of both Channel 0 and Channel 1 must be programmed with the same starting value for full auto-initialization. If Channel 0 reaches terminal count before Channel 1, then Channel 0 and Channel 1 must be programmed with the same starting value for full auto-initialization. If Channel 0 reaches terminal count before Channel 1, then Channel 0 reloads the starting address and word count, and continues transferring data from the beginning of the source block. Should Channel 1 reach terminal count first, it reloads the current registers and Channel 0 remains un-initialized.

DREQ Priority

The 82C836 supports two schemes for establishing DREQ priority. The first is fixed priority, which assigns priority based on channel position. In this method Channel 0 is assigned the highest priority. Priority assignment then progresses in order down through the channels, with Channel 3 receiving the lowest priority.

The second type of priority assignment is rotating priority. In this scheme the ordering of priority from Channel 0 to Channel 3 is maintained but the actual assignment of priority changes. The channel most recently serviced is assigned the lowest priority and, since the order of priority assignment remains fixed, the remaining three channels rotate accordingly.

In instances where multiple requests occur at the same time, the 82C836 issues an HRQ but does not freeze the priority logic until HLDA is returned. Once HLDA becomes active, the priority logic is frozen and DACK is asserted on the highest requesting channel. Priority is not reevaluated until HLDA is deactivated.

Address Generation

DMA addresses consist of three separate parts:

- The low-order bits (0-7 or 1-8) are automatically incremented after each DMA transfer (during state S1 or S2).

- The middle bits (8-15 or 9-16) are updated only when there is a carry from the low-order bits. An S1 state is required whenever the middle bits need to be updated.
- The high-order bits (16-23 or 17-23) come from the DMA Page Registers, separate from the 8237-compatible DMA subsections. The high-order bits cannot be updated without software intervention.

During Demand and Block Transfers, the 82C836 generates multiple sequential transfers. For most of these transfers only the low-order address bits need to change. The 82C836 updates the middle bits only when necessary. This results in an overall throughput improvement, since S1 states are not necessary when only the low-order bits are updated.

The DMA Page Register file is a set of 16 8-bit registers in the 82C836 that are used to generate the high order address bits during DMA cycles. Only eight of the registers are actually used, but all sixteen are included to maintain IBM PC/AT compatibility. Each DMA channel has a register associated with it, except for Channel 0 of DMA2, which is used for internal cascading to DMA1. I/O address assignments for these registers are shown in Table 18.

Table 18 *DMA Page Register Function I/O Ports*

I/O Port	Register Function
080H	Unused
081H	8-bit DMA Channel 2 (DACK2)
082H	8-bit DMA Channel 3 (DACK3)
083H	8-bit DMA Channel 1 (DACK1)
084H	Unused
085H	Unused
086H	Unused
087H	8-bit DMA Channel 0 (DACK0)
088H	Unused
089H	16-bit DMA Channel 2 (DACK6)
08AH	16-bit DMA Channel 3 (DACK7)
08BH	16-bit DMA Channel 1 (DACK5)
08CH	Unused
08DH	Unused
08EH	Unused
08FH	Refresh Cycle

I/O port 80H is normally used externally for diagnostic LEDs, updated by the AT-compatible BIOS during power-on self-test (POST). The LEDs are optional; they can either be designed onto the motherboard (usually for evaluation and testing purposes only), or they can be provided on a removable AT bus add-on card.

Compressed Timing

The DMA subsystem in the 82C836 can be programmed to transfer a word in as few as three DMA clock cycles (states). The normal AT-compatible DMA cycle consists of four states: S2, S3, SW and S4 (this assumes Demand or Block Transfer Mode). Additional DMA wait states (SW) can be programmed. In systems capable of supporting high throughput, the 82C836 can be

programmed to omit the S3 state and assert both commands in S2. This reduces the minimum cycle to just S2, SW and S4. If Compressed Timing is selected, TC is output in S2, and S1 cycles are executed as necessary to update the middle address bits. Compressed Timing is not allowed in the memory-to-memory transfer mode.

DMA Register Descriptions

The next several sections describe the registers used during DMA functions.

Current Address Register

Each DMA channel has a 16-bit Current Address Register that holds the address used during transfers. Each channel can be programmed to increment or decrement this register whenever a transfer is completed. This register can be read or written by the CPU in consecutive 8-bit bytes. If Auto-Initialization is selected, this register is reloaded from the Base Address Register upon reaching terminal count in the Current Word Count Register. Channel 0 can be prevented from incrementing or decrementing by setting the Address Hold Bit in the Command Register.

Current Word Count Register

Each channel has a Current Word Count Register that determines the number of transfers to perform. The actual number of transfers performed is one greater than the value programmed into the register. The register is decremented after each transfer until it goes from zero to FFFFH. When this roll-over occurs, the 82C836 generates T/C, suspends operation on that channel, sets the appropriate Request Mask Bit or Auto-Initialize, and continues.

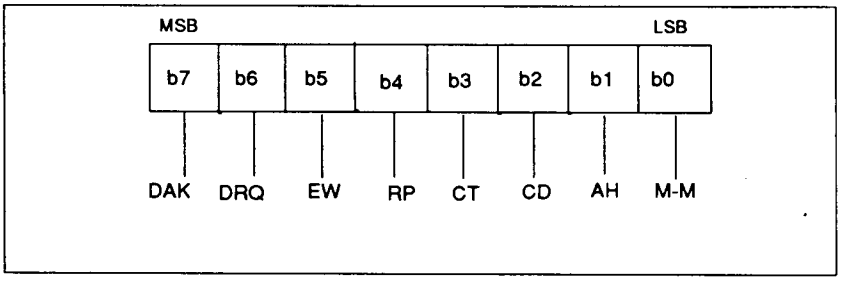
Base Word Count Register

This register preserves the initial value of the Current Word Count Register. It is also a write-only register that is loaded by writing to the Current Word Count Register. This register is loaded in the Current Word Count Register during Auto-Initialization.

Command Register

This register controls the overall operation of a DMA subsystem. The register can be read or written by the CPU and is cleared by either a RESET or a Master Clear command. Refer to Figure 29.

Figure 29 *Command Register*



b7 is DAK. DACK active level is determined by bit 7. Programming a one in this bit position makes DACK an active high signal.

b6 is DRQ. DREQ active level is determined by bit 6. Writing a one in this bit position causes DREQ to become active low.

b5 is EW. Extended Write is enabled by writing a one to bit 5, causing the write commands to be asserted one DMA cycle earlier during a transfer. The read and write commands both begin in state S2 when enabled.

b4 is RP. Writing a one to bit 4 causes the 82C836 to utilize a rotating priority scheme for honoring DMA requests. The default condition is fixed priority.

b3 is CT. Compressed timing is enabled by writing a one to bit 3 of this register. The default 0 condition causes the DMA to operate with normal timing.

b2 is CD. Bit 2 is the master disable for the DMA controller. Writing a one to this location disables the DMA subsystem (DMA1 or DMA2). This function is normally used whenever the CPU needs to reprogram one of the channels to prevent DMA cycles from occurring

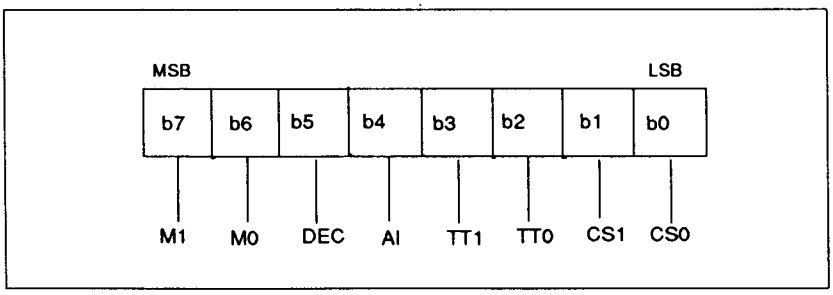
b1 is AH. Writing a one to bit 1 enables the address hold feature in Channel 0 when performing memory-to-memory transfers.

b0 is M-M. A one in the bit 0 position enables Channel 0 and Channel 1 to be used for memory-to-memory transfers.

Mode Register

Each DMA channel has a Mode Register associated with it. All four Mode Registers reside at the same I/O address. Bits 0 and 1 of the Write Mode Register command determine which channel's Mode Register is written to. The remaining six bits control the mode of the selected channel. Each channel's Mode Register can be read by sequentially reading the Mode Register location. A Clear Mode Register Counter command is provided to allow the CPU to restart the mode read process at a known point. During mode read operation, bits 0 and 1 are one. Refer to Figure 30.

Figure 30 *Mode Register*



b7-b6 is M1-M0—Mode selection for each channel is accomplished by bits 6 and 7. Modes are as follows:

Demand Mode when M1 = 0 and M0 = 0
Single Cycle Mode when M1 = 0 and M0 = 1

Block Mode	when M1 = 1 and M0 = 0
Cascade Mode	when M1 = 1 and M0 = 0

b5 is DEC. Determines direction of the address counter. A one in bit 5 decrements the address after each transfer.

b4 is AI. The Auto-Initialization function is enabled by writing a one in bit 4 of the Mode Register.

b3-b2 is TT1-TT0. Bits 2 and 3 control the type of transfer that is to be performed. The type of transfer is as follows:

Verify Transfer	when TT1 = 0 and TT0 = 0
Write Transfer	when TT1 = 0 and TT0 = 1
Read Transfer	when TT1 = 1 and TT0 = 0
Illegal	when TT1 = 1 and TT0 = 1

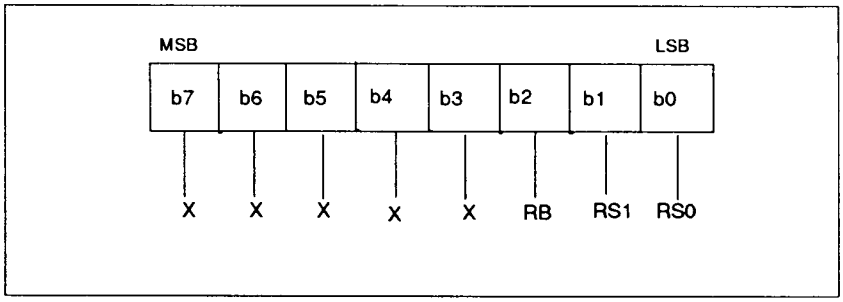
b1-b0 is CS1-CS0—Channel Select bits 1 and 0 determine the channel for which the Mode Register is written. Read back of a mode register results in bits 1 and 0 both being ones. Channel Select is as follows:

Channel 0 Select	when CS1 = 0 and CS0 = 0
Channel 1 Select	when CS1 = 0 and CS0 = 1
Channel 2 Select	when CS1 = 1 and CS0 = 0
Channel 3 Select	when CS1 = 1 and CS0 = 1

Request Register

This is a four-bit register used to generate software requests. (DMA service can be requested either externally or under software control.) Request Register bits can be set or reset independently by the CPU. The Request Mask has no effect on software generated requests. All four bits are read in one operation and appear in the lower four bits of the byte. Bits 4 through 7 are read as ones. All four request bits are cleared to zero by a RESET. Refer to Figure 31.

Figure 31 *Request Register (Write Operation)*



b7 to b3 are ignored

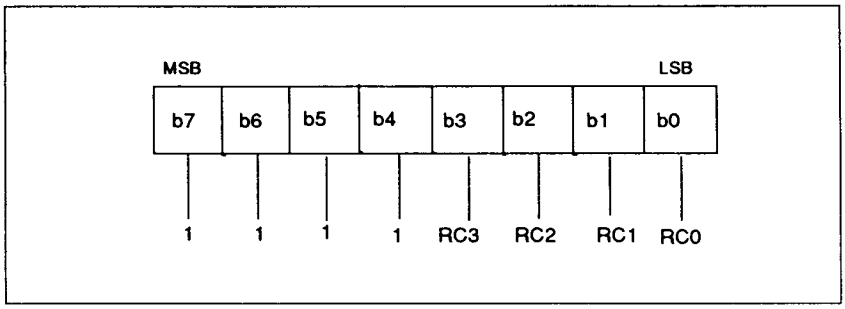
b2 is RB. The request bit is set by writing a one to bit 2. RS1-RS0 select which bit (channel) is to be manipulated.

b1 to b0 is RS<1:0>. Channel Select 0 and 1 determine which channel's Mode Register is written to. Read back for the mode register results in bits 0 and 1 both being ones.

Channel 0 Select	when RS1 = 0 and RS0 = 0
Channel 1 Select	when RS1 = 0 and RS0 = 1
Channel 2 Select	when RS1 = 1 and RCS0 = 0
Channel 3 Select	when RS1 = 1 and RS0 = 1

The format for the Request Register read operation is shown in Figure 32.

Figure 32 *Request RegisterRead Format*



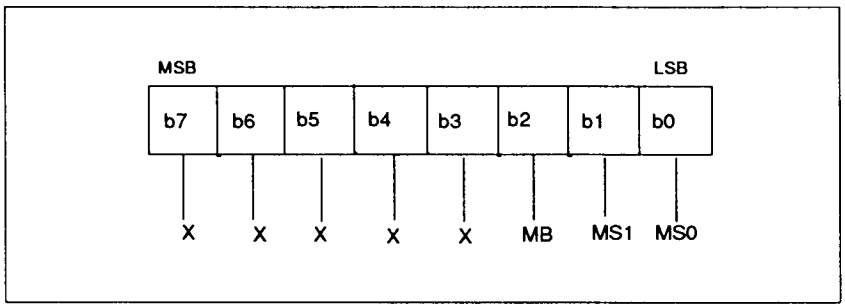
b7 to b4 are ones.

b3 is RC<3:0>. During a Request Register read, the state of the request bit associated with each channel is returned in bits 0 through 3 of the byte. The bit position corresponds to the channel number.

Request Mask Register

The Request mask register is a set of four bits that are used to inhibit external DMA requests from generating transfer cycles. This register can be programmed in two ways. Each channel can be independently masked by writing to the Write Single Mask Bit location. The data format for this operation is shown in Figure 33.

Figure 33 *Request Mask Register—Write Single Mask Bit*



b7 to b3 are ignored

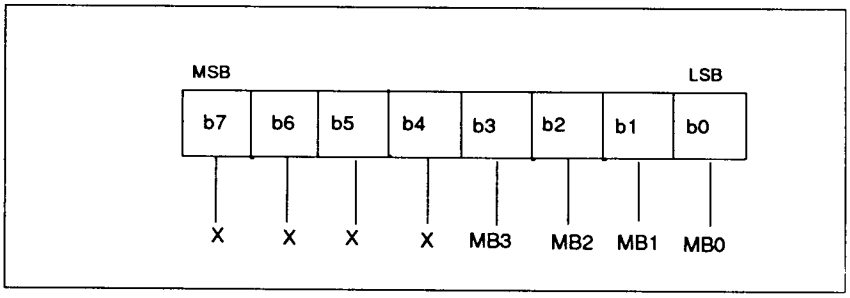
b2 is MB. Bit 2 sets or resets the request mask bit for the channel selected by MS1 and MS0. Writing a one in this bit position sets the mask, inhibiting external requests.

b1-b0 is MS<1:0>. These two bits select the specific mask bit that is to be set or reset.

Channel 0 Select	when MCS1 = 0 and MCS0 = 0
Channel 1 Select	when MCS1 = 0 and MCS0 = 1
Channel 2 Select	when MCS1 = 1 and MCS0 = 0
Channel 3 Select	when MCS1 = 1 and MCS0 = 1

Alternatively, all four mask bits can be programmed in one operation by writing to the Write All Mask Bits address. The data format for this function and the Read All Mask Bits function is shown in Figure 34.

Figure 34 *Request Mask Register—Write All Mask Bits*



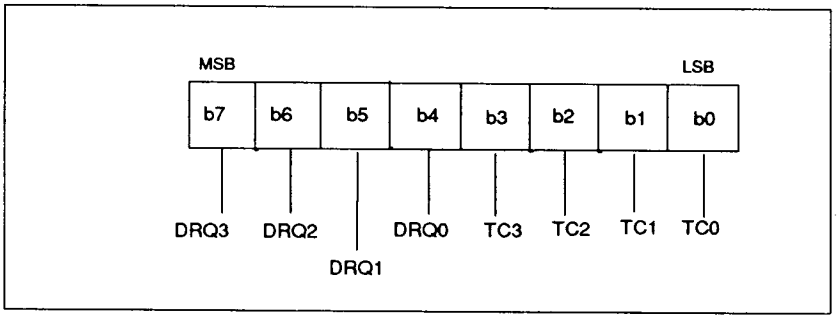
b7 to b4 are ignored.

b3 is MB<3:0>. Each bit position in the field represents the mask bit of a channel. The mask bit number corresponds to the channel number associated with the mask bit. All four mask bits are set following a RESET or a Master Clear command. Individual channel mask bits are set as a result of terminal count being reached, if Auto-Initialize is disabled. The entire register can be cleared, enabling all four channels, by performing a Clear Mask Register operation.

Status Register

The status of all four channels can be determined by reading the Status Register. Information is available to determine if a channel has reached terminal count and whether an external service request is pending. Bit 0-3 of this register are cleared by a RESET, a Master Clear, or each time a Status Read takes place. Bits 4-7 are cleared by a RESET, a Master Clear, or the pending request being de-asserted. Bits 4-7 are not affected by the state of the Mask Register Bits. The Channel number corresponds to the bit position. Refer to Figure 35

Figure 35 *Status Register (Read Only)*



b7-b4 is DRQ<7:4>

b3-b0 is TC<3:0>

Temporary Register

The Temporary Register (not accessible in AT-compatible architectures, including SCATsx) is used as a temporary holding register for data during memory-to-memory transfers. The register is loaded during the first cycle of a memory-to-memory transfer from XDO-7. During the second cycle of the transfer, the 8237-compatible subsection attempts to output the data on the XD0-7 pins, but there is no bus steering logic implementing this data path.

Data from the last memory-to-memory transfer remains in the register unless a RESET or a Master Clear occurs.

Special Commands

Five special commands are provided to make the task of programming the device easier. These commands are activated as a result of a specific address and assertion of either an IOR- or IOW-. Information on the data lines is ignored by the 82C836 whenever an IOW- activated command is issued. Thus data returned on IOR- activated commands is invalid. Descriptions of the five special commands follow:

- **Clear Byte Pointer Flip-Flop**—This command is normally executed prior to reading or writing to the address or word count register. This initializes the flop-flop to point to the low byte of the register and allows the CPU to read or write the register bytes in correct sequence.
- **Set Byte Pointer Flip-Flop**—Setting the Byte Pointer Flip-Flop allows the CPU to adjust the pointer to the high byte of an address or word count register.
- **Master Clear**—This command has the same effect as a hardware RESET. The Command Register, Status Register, Request Register, Temporary Register, Mode Register Counter, and Byte Pointer Flip-Flop are cleared and the Request Mask Register is set. Immediately following Master Clear or RESET, the DMA is in the Idle Condition.
- **Clear Request Mask Register**—This command enables all four DMA channels to accept requests by clearing the mask bits in the register.
- **Clear Mode Register Counter**—In order to allow access to four Mode Registers while only using one address, an additional counter is used. After clearing the counter, all four Mode Registers may be read by doing successive reads to the Read Mode Register address. The order in which the register is read is Channel 0 first and Channel 3 last.

Address Maps

This section describes 82C386 I/O and memory maps

I/O Address Maps

Tables 19 through 29 list the I/O ports used in the 82C836. (Refer to an IBM PC AT technical reference manual for reserved locations.)

Table 19 *DMA Controller 1*

I/O Address	R/W	Description
0000H	R/W	DMA Channel 0 current address
0001H	R/W	DMA Channel 0 current word count
0002H	R/W	DMA Channel 1 current address
0003H	R/W	DMA Channel 1 current word count
0004H	R/W	DMA Channel 2 current address
0005H	R/W	DMA Channel 2 current word count
0006H	R/W	DMA Channel 3 current address
0007H	R/W	DMA Channel 3 current word count
0008H	R/W	Command/Status Register
0009H	R/W	DMA Request Register
000AH	R/W	DMA Single Bit Mask Register
000BH	R/W	DMA Mode Register
000CH	R/W	DMA Clear Byte Pointer
000DH	R/W	DMA Master Clear
000EH	R/W	Clear Mask Register
000FH	R/W	DMA Write All Mask Register Bit

Table 20 *Interrupt Controller 1*

I/O Address	R/W	Description
0020H	W	INTC ICW1
	W	INTC OCW2, OCW3
	R	INTC Interrupt Request Register (IIR)
	R	INTC In-Service Register (ISR)
	R	INTC Polling Data Byte
0021H	W	INTC ICW2
	W	INTC ICW3
	W	INTC ICW4
	W	INTC OCW1
	R	INTC Interrupt Mask Register (IMR)

Table 21 *Internal Configuration Registers*

I/O Address	R/W	Description
0022H	W/O	Internal Configuration Register Index (See Table 22)
0023H	R/W	Internal Configuration Register Data

Table 22 *Index Registers*

ICR Index	R/W	Description
01H	R/W	DMA Wait-State Control Register
40H	R/O	Version Register
41H	R/W	Channel Environment Register
42H		(Reserved) Do Not Write to this Register.
43H		(Reserved) Do Not Write to this Register.
44H	R/W	Peripheral Control Register
45H	R/O	Miscellaneous Status Register
46H	R/W	Power Management Register
47H		(Reserved) Do Not Write to this Register.
48H	R/W	ROM Enable Register
49H	R/W	RAM Write Protect Register
4AH	R/W	Shadow RAM Enable Register 1
4BH	R/W	Shadow RAM Enable Register 2
4CH	R/W	Shadow RAM Enable Register 3
4DH	R/W	DRAM Configuration Register
4EH	R/W	Extended Boundary Register
4FH	R/W	EMS Control Register

Table 23 *Timer Counter Registers*

I/O Address	R/W	Description
0040H	R/W	Timer 0 Count Load/Read
0041H	R/W	Timer 1 Count Load/Read
0042H	R/W	Timer 2 Count Load/Read
0043H	W	Timer Control Word

Table 24 *Miscellaneous I/O Registers*

I/O Address	R/W	Description
0061H	R/W	Control/Status Port
0070H	W	Real-Time Clock Index and NMI Mask
0071H	R/W	Real-Time Clock Data Port
0092H	R/W	System Control Port

Table 25 *DMA Page Registers*

I/O Address	R/W	Description
0081H	W	DMA Channel 2 Page Register
0082H	W	DMA Channel 3 Page Register
0083H	W	DMA Channel 1 Page Register
0087H	W	DMA Channel 0 Page Register
0089H	W	DMA Channel 6 Page Register
008AH	W	DMA Channel 7 Page Register
008BH	W	DMA Channel 5 Page Register
008FH	W	Refresh Address Page Register

Table 26 *Interrupt Controller 2*

I/O Address	R/W	Description
00A0H	W	INTC ICW1
	W	INTC OCW2
	W	INTC OCW3
	R	INTC Interrupt Request Register (IIR)
	R	INTC In-Service Register (ISR)
	R	INTC Polling Data Byte
00A1H	W	INTC ICW2
	W	INTC ICW3
	W	INTC ICW4
	W	INTC OCW1
	R	INTC Interrupt Mask Register (IMR)

Table 27 *DMA Controller 2*

I/O Address	R/W	Description
00C0H	R/W	Channel 0 base and current address
00C2H	R/W	Channel 0 base and current word count
00C4H	R/W	Channel 1 base and current address
00C6H	R/W	Channel 1 base and current word count
00C8H	R/W	Channel 2 base and current address
00CAH	R/W	Channel 2 base and current word count
00CCH	R/W	Channel 3 base and current address
00CEH	R/W	Channel 3 base and current word count
00D0H	R/W	Read Status Register/Write Command Register 0
00D2H	R/W	Write Request Register
00D4H	R/W	Write Single Mask Register Bit
00D6H	R/W	Write Mode Register
00D8H	R/W	Clear Byte Pointer Flip-Flop
00DAH	R/W	Read Temporary Register/Write Master Clear 0
00DCH	R/W	Clear Mask Register
00DEH	R/W	Write All Mask Register Bits
00DFH	R/W	DMA Write All Mask Register Bit

Table 28 *Coprocessor Registers*

I/O Address	R/W	Description
00F0H	W	Clear Coprocessor Busy
00F1H	W	Not used in 82C836

Table 29 *EMS Page Registers*

I/O Address	R/W	Description
02x8H	R/W	EMS Page Register, A21-A14
02x9H	R/W	EMS Page Register, Enable & A23-A22
02xAH	R/W	EMS Page Register Select
		Note: x = 0 or 1 depending on ICR 4F, bit 0

Memory Address Map

Table 30 lists the memory addresses used in the 82C836. For more detailed information about memory addressing, refer to the *IBM PC AT Technical Reference Manual.*

Table 30 *Memory Address Map*

I/O Address	R/W	Description
000000-09FFFFH	R/W	"Conventional" System RAM
0A0000-0BFFFFH	R/W	Video Memory or Shadow RAM
0C0000-0EFFFFH	R/W	BIOS Extension or Shadow RAM
0F0000-0FFFFFH	R/W	ROM or Shadow RAM
100000-FBFFFFH	R/W	Extended or Expanded Memory
FC0000-FFFFFFH	R/W	Expanded Memory (EMS)
FC0000-FEFFFFH	R/O	BIOS Extension Image
FF0000- FFFFFFH	R/O	ROM Image

Internal Configuration Registers (ICRs)

Internal configuration registers (ICRs) control the enhanced options of the 82C836. ICRs are accessed by addressing an index register at port 22H and then immediately writing (or reading) data to (or from) port 23H. The 82C836 contains ICRs at index values 01H and 40H through 4FH. In the definitions that follow, power-on default states of the registers are given. All ICRs are read/write except as noted. To avoid unpredictable results, all "reserved" bits must be written as zero unless otherwise noted. The value read from a reserved bit is not defined and may vary unpredictably from one read to the next.

Index 01H — DMA Wait State Control

Bit	Description
7-6	Reserved.
5-4	16-bit DMA WAIT-STATES. These bits control the number of wait-states inserted during 16-bit DMA transfers, as follows: 00 = One wait-state (default) 01 = Two wait-states 10 = Three wait-states 11 = Four wait-states
3-2	8-bit DMA WAIT-STATES. These bits control the number of wait-states inserted during 8-bit DMA transfers, as follows: 00 = One wait-state (default) 01 = Two wait-states 10 = Three wait-states 11 = Four wait-states
1	DMA XMEMR EXTENSION. In the IBM PC AT, the assertion of XMEMR- is delayed by one DMA clock cycle compared to XIOR-. This may not be desirable in some systems. 0 = Enables delayed XMEMR- function (default). 1 = Starts XMEMR- at the same time as XIOR-.
0	DMA CLOCK SEL. This bit allows the user to program the DMA clock to operate at either BUSCLK or BUSCLK/2. The same DMA clock drives both 8-bit and 16-bit operations. 0 = BUSCLK/2 (default). 1 = BUSCLK.

Note: If bit 0 is changed during operation, an internal synchronizer controls the actual switching of the clock to prevent a short clock pulse from causing a DMA malfunction.

Index 40H — 82C836 Version

Bit	Description
All bits in this register are read only.	
7-4	Family type. Identifies the specific part within a pin-compatible family. SCATsx is denoted by 0001.
3-0	Device revision code. Identifies the revision level of the 82C836 silicon, starting with 0000.

Index 41H — Channel Environment

Bit	Description
7	Reserved.
6	Early READY Enable. Allows external devices to assert READY during the first T2 after T1 to terminate the memory cycle after only two T-states. Enables "Early Wait State" for local memory reads not "claimed" by early READY. 1 = enable 0 = disable (default).
5	LBA Enable. Allows external devices to assert LBA during the first T2 after T1 to prevent the 82C836 from starting a memory cycle. Enables "Early Wait State" for local memory reads not claimed by LBA. 1 = enable 0 = disable (default). If both Early READY and LBA are enabled, there will still be only one early wait state for unclaimed local memory reads.
4	Bus Convert Enable. Enables conversion of 8-bit AT bus reads into 16-bit reads so that a local cache, if present, can cache 16 bits at a time. 1 = enable 0 = disable (default). This feature does not apply to 8-bit AT Bus memory resources residing in the first 1 MB. Also, the usual AT-compatible conversion of "even word" accesses into paired byte operations is not affected by this bit. This bit has no effect unless bit 6 is set.
3-2	BUSCLK SEL. Selects AT Bus clock. 00 = selects BUSCLK as CXIN/4 01 = selects BUSCLK as CXIN/5 (default) 10 = selects BUSCLK as CXIN/6 11 = (reserved)

Bit	Description
1-0	REFRESH WIDTH. These bits specify XMEMR- pulse width during a refresh cycle. These pulse widths are derived from OSC. The width of XMEMR- directly affects the overall bus cycle time for refresh operations. 00 = 140 ns 01 = 210 ns 10 = 280 ns (default) 11 = 350 ns

Note: Bits 5 and 6 must remain zxero if CPU pipeline mode is enabled.

Index 42H — Reserved

Index 43H — Reserved

Index 44H — Peripheral Control

Bit	Description
	This register controls whether peripherals are accessed on the XD-bus or the SD-bus. The SDIRH and SDIRL SD-bus controls are affected by bits in this register.
7	Reserved.
6	VIDEO ON THE XD-BUS. This bit indicates whether or not a video controller is resident on the XD-bus. 0 = Video controller is on the SD-bus. 1 = Video controller is on the XD-bus. The I/O address ranges covered for the video controller are 0102H-0104H, 03B0H-03DFH, and 46E8H. Video memory address range is 0A0000H through 0BFFFFH. The power-on default value of this bit is determined by the state of DACK2- (inverted) during power-on reset. DACK2- high causes this bit to default to zero. If the DACK2 strap option is used to denote something other than "Video on X-Bus," software (BIOS) must write this ICR bit to the correct value before accessing the video controller.

Bit	Description
5	GAME PORT ON THE XD-BUS. This bit specifies whether or not the game port is resident on the XD-bus. The I/O address range covered 0200H-0207H. 0 = Game port on the SD-bus (default). 1 = Game port on the XD-bus.
4	SERIAL PORT CHANNEL 2 ON THE XD-BUS. Specifies whether or not serial port 2 is resident on the XD-bus. The I/O address range covered is 02F8H-02FFH. 0 = serial port 2 on the SD-bus (default). 1 = serial port 2 on the XD-bus.
3	SERIAL PORT CHANNEL 1 ON THE XD-BUS. Specifies whether or not serial port 1 is resident on the XD-bus. The I/O address range covered 03F8H-03FFH. 0 = serial port 1 on the SD-bus (default). 1 = serial port 1 on the XD-bus.
2	PARALLEL PORT ON THE XD-BUS. Specifies whether or not a parallel port is resident on the XD-bus. The address range covered 0378H-037FH. 0 = parallel port on the SD-bus (default). 1 = parallel port on the XD-bus.
1	HDC/FDC ON THE XD-BUS. Specifies whether or not the hard drive and diskette drive controllers are resident on the XD-bus. The I/O address ranges covered are 01F0H-01F7H and 03F0H-03F7H, as well as DMA Channel 2 (DACK2-). 0 = HDC/FDC on the SD-bus (default). 1 = HDC/FDC on the XD-bus.
0	COPROCESSOR READY. Determines whether READY is controlled by the 82C836 or 80387sx during coprocessor accesses. If no coprocessor is present, the 82C836 always controls READY during coprocessor I/O cycles, regardless of the state of this bit. If a coprocessor is present and the READYO logic is not implemented, bios should set this bit to one. Otherwise, the system will hang due to lack of READY as soon as POST or other software tries to access the coprocessor. 0 = 80387sx generates READY (default) 1 = 82C836 generates READY

Note: Memory resources on the XD bus can be ROM or video RAM. The only DMA device allowed on the XD bus is the HDC/FDC. The BIOSes associated with video and/or HDC should be included in the ROMCS- decoding (ICR 48H); they are not included in SDIR decoding (ICR 44H) for XD- bus peripherals.

Index 45H — Miscellaneous Status

Bit	Description
All bits in this register are read-only.	
7	NMI INHIBIT. This bit indicates the current state of the NMI inhibit bit of I/O port 70H, bit 7. Bit 7 of I/O port 70H is write only. 0 = NMI enabled. 1 = NMI disabled (port 70 default).
6	GATEA20 FROM 8042. This bit indicates the current state of the GATEA20 input from the 8042 keyboard controller. 0 = GATEA20 is low. 1 = GATEA20 is high (A20 enabled).
5	BUSY TO CPU. This bit indicates the current state of the latched busy signal to the 80386sx. 0 = BUSY- signal low. 1 = BUSY- signal high.
4	INTERNAL RTC ENABLED. This bit indicates the state of the EXRTC- (DACK5-) pin during reset. 0 = external RTC used (XD bus). 1 = internal RTC used.
3	NA-/STCYC- SELECT. Normally the NA- pin is used as an NA- and/or ADRL- signal. For external cache support, it can be changed to operate as a CPU cycle start (STCYC-) and/or address latch (ADRL-) signal. 0 = STCYC-/ADRL- function 1 = NA-/ADRL- function The state of this bit is determined by the state of DACK3- during reset. DACK3- should be pulled up if NA- goes to the CPU.

Bit	Description
2-0	SENSE LINES 2-0. These lines (2-0) sense the power-up state of DACK lines (2-0), respectively (non-inverted). These lines can be used for power-on setup parameters. DACK2- (inverted) also determines the power-up state of ICR 44H bit 6. 0 = corresponding DACK line pulled low. 1 = corresponding DACK line pulled high. A 4.7K ohm resistor is recommended for the pull-up or pull-down.

Index 46H Power Management

Bit	Description
This register controls sleep mode and PROCCLK frequency for power conservation.	
7	SLEEP ENABLE. This bit enables sleep mode. If this bit is set, sleep mode will be entered upon execution of a HALT instruction. An interrupt (INTR or NMI) will terminate sleep mode until a subsequent HALT; a DMA or refresh will terminate sleep mode temporarily, then return to sleep mode upon completion of the DMA or refresh. 0 = sleep mode disabled (default). 1 = sleep mode enabled.
6	AUX PARITY DISABLE. This bit, when set, is an alternate parity error clear and disable, in addition to Port 61H bit 2. Either this bit (if set) or Port 61H bit 2 (if set) will clear the internal parity error latch and prevent further parity error detection. 0 = Enable parity checking (default). Port 61H bit 2 must also be zero to enable parity checking. 1 = Parity error clear and disable, independent of Port 61H bit 2.
5-4	Reserved.
3-2	RUN FREQ. These bits select the frequency of the PROCCLK signal to the CPU when in normal run mode. 00 = CXIN (default) 01 = CXIN/2 10 = CXIN/4 11 = CXIN/8

Bit	Description
1-0	SLEEP FREQ. These bits select the frequency of the PROCCLK signal to the CPU when in sleep mode. 00 = stopped, solid high (default) 01 = CXIN/2 10 = CXIN/4 11 = CXIN/8

Note: Refresh, DMA and Master cycles always run at full speed, PROCCLK = CXIN, regardless of bits 3-0.

Index 47H—Reserved

Index 48H — ROM Enable

Bit	Description

This register determines the address ranges to be included in the ROMCS- output. The power-on default address range is 0F0000H-0FFFFFH. The range from FC0000H to FFFFFFH is always enabled. The DACK7- pin is used during reset to specify the width of the ROM accessed with the ROMCS- signal. If a 4.7K pull-down is placed on DACK7-, execution accesses to ROM will always be 16-bits wide (two ROMs required), and MEMCS16- will be asserted for all ROM cycles. A 4.7K pull-up is needed on DACK7- for 8-bit wide ROM.

Each bit enables or disables ROMCS- for the address range specified below. 1=enable ROM, 0=disable ROM.

7	0F8000H-0FFFFFH ROM ENABLE. Default = 1.
6	0F0000H-0F7FFFH ROM ENABLE. Default = 1.
5	0E8000H-0EFFFFH ROM ENABLE. Default = 0.
4	0E0000H-0E7FFFH ROM ENABLE. Default = 0.
3	0D8000H-0D8FFFH ROM ENABLE. Default = 0.
2	0D0000H-0D7FFFH ROM ENABLE. Default = 0.
1	0C8000H-0CFFFFH ROM ENABLE. Default = 0.
0	0C0000H-0C7FFFH ROM ENABLE. Default = 0.

Index 49H — RAM Write Protect

Bit	Description

This register is used to write protect RAM in the address ranges specified below.

1=read only

0=read/write

Default for all bits is 0.

7	0F8000H-0FFFFFH RAM READ ONLY.
6	0F0000H-0F7FFFH RAM READ ONLY.
5	0E8000H-0EFFFFH RAM READ ONLY.
4	0E0000H-0E7FFFH RAM READ ONLY.
3	0D8000H-0DFFFFH RAM READ ONLY.
2	0D0000H-0D7FFFH RAM READ ONLY.
1	0C8000H-0CFFFFH RAM READ ONLY.
0	0C0000H-0C7FFFH RAM READ ONLY.

Index 4AH — Shadow RAM Enable 1

Bit	Description

This register controls whether shadow RAM is enabled or disabled in the 16KB address ranges specified below. 1=enable RAM, 0=disable RAM. Default for all bits is 0.

7	0BC000H-0BFFFFH RAM ENABLE.
6	0B8000H-0BBFFFH RAM ENABLE.
5	0B4000H-0B7FFFH RAM ENABLE.
4	0B0000H-0B3FFFH RAM ENABLE.
3	0AC000H-0AFFFFH RAM ENABLE.
2	0A8000H-0ABFFFH RAM ENABLE.
1	0A4000H-0A7FFFH RAM ENABLE.
0	0A0000H-0A3FFFH RAM ENABLE.

Index 4BH — Shadow RAM Enable 2

Bit	Description

This register controls whether shadow RAM is enabled or disabled in the 16KB address ranges specified below.

1=enable RAM

0=disable RAM.

Default for all bits is 0.

7	0DC000H-0DFFFFH RAM ENABLE.
6	0D8000H-0DBFFFH RAM ENABLE.
5	0D4000H-0D7FFFH RAM ENABLE.
4	0D0000H-0D3FFFH RAM ENABLE.
3	0CC000H-0CFFFFH RAM ENABLE.
2	0C8000H-0CBFFFH RAM ENABLE.
1	0C4000H-0C7FFFH RAM ENABLE.
0	0C0000H-0C3FFFH RAM ENABLE.

Index 4CH — Shadow RAM Enable 3

Bit	Description

This register controls whether shadow RAM is enabled or disabled in the 16KB address ranges specified below.

1=enable RAM

0=disable RAM

Default for all bits is 0.

7	0FC000H-0FFFFFH RAM ENABLE.
6	0F8000H-0FBFFFH RAM ENABLE.
5	0F4000H-0F7FFFH RAM ENABLE.
4	0F0000H-0F3FFFH RAM ENABLE.
3	0EC000H-0EFFFFH RAM ENABLE.
2	0E8000H-0EBFFFH RAM ENABLE.
1	0E4000H-0E7FFFH RAM ENABLE.
0	0E0000H-0E3FFFH RAM ENABLE.

Note: Do not enable ROM and shadow RAM in the same address range at the same time; do not enable shadow RAM when using memory configuration #03H (640K + 384K).

Index 4DH — DRAM Configuration

Bit	Description
7	CAS WAIT STATE. Allows an additional T-state (two PROCCLK cycles) to be inserted on all local memory accesses. The added T-state is inserted during the CAS active interval, extending the width of the CAS pulse. 0 = no added Wait State (default) 1 = Wait State enabled
6	RAS TIMEOUT. Disables the RAS Timeout feature. 0 = RAS timeout enabled (default) 1 = RAS timeout disabled
5	Reserved
4-0	CFG4-0. These bits specify the DRAM configuration as described in "DRAM Interface". The default value after reset is 0001 (512KB total DRAM).

Index 4EH — Extended Memory Boundary

Bit	Description
7	RAS ENCODE ENABLE. This bit changes the four RAS- lines into three encoded RAS- lines and one RAS- timing line. These four lines are used with an external 3-8 decoder to support eight banks of DRAM. 0 = RAS- encode disabled (default). 1 = RAS- encode enabled.
6	Reserved
5	RAM DISABLE 040000H-09FFFFH. This bit disables the 82C836's internal DRAM controller for accesses in the range 040000H-09FFFFH. These accesses are directed to the I/O channel. 0 = enables 040000H-09FFFFH DRAM range (default). 1 = disables 040000H-09FFFFH DRAM range

Bit	Description
4	Reserved.
3-0	EXTENDED BOUNDARY. These bits specify the upper boundary of memory within the total memory defined by internal configuration register 4DH. Memory residing above this boundary is accessible only via EMS.

Bits 3210	Extended Memory Boundary
0000	No Limit (default); entire DRAM accessible
0001	1 MB (i.e., no Extended Memory)
0010	1.25 MB
0011	1.5 MB
0100	2 MB
0101	3 MB
0110	4 MB
0111	5 MB
1000	7 MB
1001	8 MB
1010	9 MB
1011	10 MB
1100	11 MB
1101	12 MB
1110	13 MB
1111	15 MB

Index 4FH — EMS Control Register

Bit	Description
7	EMS ADDRESS TRANSLATION ENABLE. This bit enables EMS memory access. 0 = EMS is disabled (default). 1 = EMS is enabled.
6	EMS I/O ENABLE. This bit enables access to the EMS I/O ports. 0 = EMS I/O port access is disabled (default). 1 = EMS I/O port access is enabled.

Bit	Description
5-1	Reserved.
0	I/O BASE. This bit specifies which I/O ports are used to read or write EMS page registers. 0 = I/O ports 0208H, 0209H, and 020AH (default). 1 = I/O ports 0218H, 0219H, and 021AH.

EMS Page Registers, I/O Ports 2x8H - 2xAH

This section lists the bit assignments for the three I/O ports used to access the four EMS page registers. The EMS mechanism is described in an earlier section of this Data Book.

EMS I/O Port 208H (or 218H) — Address bits 21-14

Bit	Description
	This port accesses bits 21 through 14 of the address of the 16 KB target page to be mapped into the corresponding page window in the 64 KB remappable address range. The page register to be accessed by this port is selected by port 20AH (or 21AH).
7	A21 of target page address
6	A20 of target page address
5	A19 of target page address
4	A18 of target page address
3	A17 of target page address
2	A16 of target page address
1	A15 of target page address
0	A14 of target page address

EMS I/O Port 209H (or 219H) — Page Enable & Address 23-22

Bit	Description
	This port accesses bits 23 and 22 of the address of the 16 KB target page to be mapped into the selected page window of the 64 KB remappable address range. The page register to be accessed by this port is selected by port 20AH (or 21AH).
7	Page Enable—1=page enable, 0=page disable. "Page disable" means that the page window in the 64 KB remappable address range is not remapped, but is treated as ordinary non-EMS memory.
6-2	Reserved.

1	A23 of target page address
0	A22 of target page address

EMS I/O Port 20AH (or 21AH) — EMS Page Select

Bit	Description
7	AUTO-INCREMENT. Enables auto-incrementing of the page register selection index (bits 1-0 in this register) whenever port 208H (or 218H) is read or written. Incrementing does not occur on access to port 209H (or 219H). 16-bit I/O may be used to read or write both ports 208H and 209H (or 218H/219H) simultaneously, with or without auto-increment. 1 = enable auto-incrementing. 0 = disable auto-incrementing.
6	PAGE FRAME BASE ADDRESS. Determines whether the 64KB remappable address range (page frame) begins at 0D0000H or 0E0000H. 0 = 0D0000H (default) 1 = 0E0000H
5-2	Reserved.
1-0	INDEX. Selects one of the four EMS Page Registers for access via ports 208H and 209H (or 218H and 219H). Can be programmed for automatic incrementing via bit 7 above. Page registers are numbered in order from 00 through 11 (binary), and the associated page window addresses in the 64 KB remappable address range follow in ascending order.

IBM PC/AT Compatible Registers

I/O Port 61H — Control Port/Status

This port controls several system-level functions. The port can be accessed through any *odd* I/O port address from 61H through 6FH.

Bit	Description
7	PARITY ERROR (read only). This bit indicates that a parity error has been detected during a local memory read. Causes NMI if Port 70H bit 7 is 0. 0 = no error detected 1 = error detected

6	IOCHCK (read only). This bit indicates an I/O channel check has occurred (usually a parity error) on the system I/O channel. Causes NMI if Port 70H bit 7 is 0. 0 = no error detected 1 = error detected
5	TMR 2 OUT (read only). This bit returns the condition of the timer 2 output (speaker tone).
4	REFRESH DETECT (read only). This bit toggles on each refresh cycle.
3	CHCK DIS (read/write). This bit clears and disables the internal I/O Channel Check detection latch. 1 = latch clear and disable. Bit 6 still responds to IOCHCK (unlatched) but does not cause NMI. 0 = enable I/O Channel Check detection.
2	PARITY DISABLE (read/write). This bit, if set, clears and disables the internal parity error detection latch. 1 = Parity error latch clear and disable. 0 = Parity checking enabled (default), if ICR 46H bit 6 is also zero.
1	SPKR DATA (read/write). This bit gates the output of channel 2 of the timer/counter (speaker tone). 0 = output is forced low (default). 1 = output is enabled, i.e., speaker tone on.
0	TMR 2 GATE (read/write). This bit gates the clock input for timer channel 2 (speaker tone). 0 = channel 2 timer clock disabled (default). 1 = channel 2 timer clock enabled.

I/O Port 70H — RTC/CMOS Index and NMI Mask

This register is used to access the Real-Time Clock (RTC) and its CMOS RAM. It can also block NMIs from going to the CPU. Bits 6-0 are not used when the external RTC option is selected.

Bit	Description
7	NMI MASK (write only). This bit masks the generation of NMIs. If an NMI-causing condition (see Port 61H) exists at the time this bit is cleared, the NMI will then be sent to the CPU. 0 = enable NMI 1 = inhibit NMI (default).

6-0	RTC INDEX bits 6:0. These bits specify the index (address) for accessing the 128 locations in the RTC and CMOS RAM (114 user RAM, 14 RTC registers). Port 71H accesses the indexed location. If an external real-time clock is used, ports 70H and 71H (except Port 70H bit 7) are inoperative in the 82C836 and should be implemented externally.

I/O Port 71H — RTC/CMOS Data

This register is used to transfer data to and from the internal real-time clock and CMOS RAM. The RTC register is selected by bits 6:0 of I/O port 70H as described above. If an external RTC is used, this register must be implemented externally.

Bit	Description
7-0	RTC DATA bits 7:0.

I/O Port 92H — System Control

This register is used as a fast alternative to gating A20 and resetting the CPU, rather than using the 8042 keyboard controller. This register is compatible with IBM PS/2 architecture.

Bit	Description
7-2	Reserved.
1	ALT GATEA20. Allows address bit 20 to be compatible with the 8088 and 8086 at address FFFF:10H and above. Either a 1 in this bit or a high logic level on the GATEA20 signal from the 8042 causes MODA20 to follow CPU A20. To force MODA20 low during CPU cycles, this bit must be zero and the GATEA20 input signal must also be low. 0 = MODA20 is forced low (8088 compatible). 1 = MODA20 follows address bit A20 from CPU.

Bit	Description
0	ALT CPU RESET. A 0-to-1 transition causes a reset pulse on the CPURST pin to reset the CPU. There is a minimum delay of 6.72 microseconds before the reset pulse begins. This bit retains its state following CPU reset so the BIOS can determine why the CPU reset occurred. The main reason for resetting the CPU alone is to return from protected mode in an 80286-compatible manner.

Pin Assignments, Descriptions and Characteristics

This section contains information about the input and output signals for the 82C836. The signals are grouped by functional category. The signal groups are followed by a table listing all the pins in numerical order, along with an illustration of the 160-pin plastic quad flat package (PQFP) showing how the pins are numbered.

Signal Descriptions

A signal name appended by a hyphen, such as RAS0-, specifies an active-low signal. The following abbreviations are used to denote the signal type:

Abbreviation	Meaning
I	Input to 82C836
O	Output from 82C836
B	Bi-directional

Clock Input and Output Signals

Pin	Type	Name	Description
122	I	OSC1	14.318 MHz Crystal Oscillator Input, either from an external oscillator module or from a crystal that is also tied to OSC2. The 82C836 circuit is an inverting amplifier and relies on low crystal impedance at resonance (i.e. series resonance).
121	O	OSC2	14.318 MHz Crystal Oscillator Output from the internal crystal oscillator. If an external oscillator module is used, leave OSC2 unconnected.
3	I	CXIN	CPU Clock Oscillator Input from an external oscillator. Serves as the source for PROCCLK.
2	O	PROCCLK	Processor Clock output to the 80386sx processor. The frequency of PROCCLK is twice the processor's internal clock frequency.
56	O	BUSCLK	Bus Clock output to the 8042 and AT Bus.

Local Bus Interface Signals

Pin	Type	Name	Description
5-18	B	A00-A13	
20-21	B	A14-A15	

Pin	Type	Name	Description
23-30	B	A16-A23	CPU Address, bits 0-23. A0 is also known as BLE-. These signals are driven by the CPU during CPU generated bus cycles, by the 82C836 during Refresh and DMA cycles, and by the add-in card Bus Master (via address buffers) during Master cycles.
58	B	MODA0	Modified A0 is the internally latched state of CPU address bit A0 during CPU generated bus cycles. The 82C836 toggles this bit during conversion cycles. The 82C836 also controls MODA0 during Refresh and DMA cycles, and the add-in card Bus Master controls it (via address buffers) during Master cycles.
98	B	MODA20	Modified A20 is gated A20 from the 82C836's gate A20 logic and should be used instead of CPU A20. MODA20 is an input during Master cycles.
33	B	BHE-	Byte High Enable is an input from the 80386sx during CPU and Master cycles, and an output during DMA cycles. BHE- and A0 indicate the type of bus transfer. BHE- is pulled high internally.
160	B	D00	
158	B	D01	
156	B	D02	
154	B	D03	
152	B	D04	
150	B	D05	
148	B	D06	
146	B	D07	
159	B	D08	
157	B	D09	

BHE-	A0	Function
0	0	Word Transfer
0	1	Odd Byte Transfer
1	0	Even Byte Transfer
1	1	(Reserved)

Pin	Type	Name	Description
155	B	D10	
153	B	D11	
151	B	D12	
149	B	D13	
147	B	D14	
145	B	D15	CPU Data Bus, bits 0 to 15.
4	I	ADS-	Processor Address Strobe from the 80386sx. Provides timing markers for new address and start of cycle. Tri-stated during a hold acknowledge state.
31	I	W/R-	Processor Write/Read status. Indicates whether the CPU cycle is a write or read.
32	I	D/C-	Processor Data/Control status. Indicates whether the CPU cycle is a data access or code/control cycle.
34	I	M/IO-	Memory or I/O select from CPU. Indicates whether the CPU cycle is a memory access or an I/O.
37	B	READY-	Active-low signal that indicates the end of a cycle. READY- is normally controlled by the 82C836 and should be connected to the 80386sx READY- input. For external cache and coprocessor support, this signal can be programmed to operate as an input.
137	O	NA-	Next Address Request to the 80386sx. Requests the CPU to enter pipeline mode if possible. Can be programmed to operate as a CPU Cycle Start indicator instead of NA- if desired. Usable as an address latch (ADRL-) control in either case.
36	O	HOLD	Hold Request is an active-high output to the processor that requests bus access for Refresh, DMA or Master cycles. HOLD should be connected to the processor's HOLD input.
35	I	HLDA	Hold Acknowledge is an active-high input from the processor that indicates when the CPU local bus has been given up by the processor. HLDA should be connected to the 80386sx processor's HLDA pin.
143	O	NMI	Non-Maskable Interrupt is generated as a result of a parity error or an I/O channel error (IOCHCK-). NMI should be connected to the processor's NMI input. The 80386sx responds to a low-to-high transition on NMI.

Pin	Type	Name	Description
144	O	INTR	Interrupt Request is an active-high request to the CPU to suspend the current process and acknowledge the request. INTR should be connected to the INTR input to the CPU.

Numeric Coprocessor Interface Signals

Pin	Type	Name	Description
133	O	BUSY-	Numeric Coprocessor Busy Status output to the 80386sx. This signal normally echoes the state of NPBUSY- from the 80387sx. During coprocessor error conditions, it is held low. Also, it is pulsed repetitively by the 82C836 when no coprocessor is present.
127	I	NPBUSY-	Busy Status from 80387sx. This signal indicates that the 80387sx is currently executing a command.
128	I	NPERR	Error signal from the coprocessor.

Memory Interface Signals

Pin	Type	Name	Description
108-117	O	MA0-MA9	Multiplexed DRAM Address Bits MA 0 to 9 are outputs to the DRAMs, which must be appropriately buffered and terminated depending on the number of DRAM banks.
107-104	O	RAS0-3-	Row Address Strobes 0 to 3 are active-low strobes used as RAS controls for the banks of DRAM. Each bank is 18 bits wide (including 2 bits for parity). Each byte is addressed with a low or high CAS signal. RAS0-3- perform different functions in an encoded RAS mode. For further details, refer to "DRAM Interface."
101	O	CASH-	Column Address Strobe High is an active-low output to all high (odd) byte DRAMs.
100	O	CASL-	Column Address Strobe Low is an active-low output to all low (even) byte DRAMs.
119	B	PARH	Parity High is the parity bit from the high-order bytes of the DRAMs.
118	B	PARL	Parity Low is the parity bit from the low-order bytes of the DRAMs.

Pin	Type	Name	Description
103	O	MWE-	Memory Write Enable is an active-low output connected to all DRAMs. MWE- is normally low, but is high for read cycles. This signal can also be used directly to control the direction of the transceivers (if present) that buffer the DRAM data to or from the CPU local data bus.
38	O	ROMCS-	ROM Chip Select is an active-low output to the EPROM(s). ROMCS- becomes active for the programmed address range. For further information, refer to "Memory Interface".

I/O Channel Interface Signals

Pin	Type	Name	Description
60-61	B	XD00-XD01	
63-76	B	XD02-XD15	16-Bit system data bus for on-board I/O, on-board ROM and (via buffers) AT Bus data.
136	O	SDIRH	
135	O	SDIRL	Channel Data Bus Controls are outputs that control the direction of the data buffer between XD-bus and SD-bus. When the signals are high (default), data flows from XD to SD; when the signals are low, data flows to XD from SD.
97	I	MASTER-	Master is an active-low input from the I/O channel's 16-bit extension. MASTER- allows a microprocessor or DMA controller residing on the I/O channel to control the system address, data, and control lines.
124	B	REFRESH-	Memory Refresh Control is an active-low output to the I/O channel that indicates a refresh cycle. During Master cycles, REFRESH- is an input to the 82C836.
57	O	ALE	Address Latch Enable is an active-high output used to latch valid addresses on the I/O channel. ALE is held continuously high during Refresh, DMA and Master cycles.
79	B	XIOR-	I/O Read Command is an active-low output used by I/O devices to put their data on the bus. This signal is used by on-board peripherals as well as the I/O channel. During Master cycles, XIOR- is an input command to the peripherals and the 82C836.

Pin	Type	Name	Description
80	B	XIOW-	I/O Write Command is an active-low output used by I/O devices to capture data from the bus. This signal is used by XD-bus peripherals as well as the I/O channel. During Master cycles, XIOW- is an input command to the peripherals and the 82C836.
77	B	XMEMR-	Memory Read Command is an active-low output used by XD-bus, video memory and I/O channel memory. During Master cycles, XMEMR- is an input command to the peripherals and the 82C836. During memory refresh and DMA cycles, XMEMR- is always an output.
78	B	XMEMW-	Memory Write Command is an active-low output used by XD-bus video memory and I/O channel memory. During Master cycles, XMEMW- is an input command to the peripherals and the 82C836. During DMA cycles, it is an output.
131	O	LOMEGCS-	Low Meg Chip Select is an active-low output that is a decode of memory accesses below 1MB. This output is used to gate SMEMR- and SMEMW- onto the AT Bus (8-bit section) from XMEMR- and XMEMW-, respectively. LOMEGCS- operates in this manner for DMA and Master cycles as well as CPU cycles.
41	I	IOCHRDY	I/O Channel Ready is used by I/O channel or XD-bus devices to lengthen their R/W cycles. Normally, IOCHRDY is high; it is pulled low for extending the cycle time. This input should be driven by an open collector driver.
40	I	0WS-	Zero Wait-state is an active-low input from the I/O channel. This signal allows the present bus cycle to terminate without inserting any additional wait-states. 0WS should be driven with an open collector or a tri-state driver. For external cache support, this signal can also be programmed to act as a Local Bus Access (LBA-) input as well as the AT Bus 0WS- input.
83	B	IOCS16-	I/O 16-bit Chip Select is an active-low signal. IOCS16- is an input from the I/O channel and XD-bus peripherals, indicating that the present cycle is a 16-bit I/O cycle. IOCS16- is an output for I/O accesses to the EMS I/O ports. IOCS16- should be driven with an open collector or tri-state driver.

Pin	Type	Name	Description
82	B	MEMCS16-	Memory 16-bit Chip Select is an active-low signal. MEMCS16-is an input from the I/O channel and XD-bus peripherals, indicating that the accessed resource is a 16-bit memory resource. MEMCS16- is an output for CPU or Master accesses to on-board DRAM or to 16-bit on-board ROM.
42	I	IOCHCK-	I/O Channel Check is an active-low signal from the I/O channel used to trigger an NMI in the processor in the event of an unrecoverable I/O channel error.
54-50	I	IRQ03-07	
49	I	IRQ09	
84-86	I	IRQ10-12	
88-87	I	IRQ14-15	Interrupt Requests 3-7, 9, 10-12, 14-15 are asynchronous inputs to the 82C836 interrupt controllers. These requests are prioritized with IRQ03 having the highest priority and IRQ15 the lowest. The request line is held active until acknowledged by the processor with an interrupt acknowledge cycle.
90	I	DRQ0	
48	I	DRQ1	
46	I	DRQ2	
44	I	DRQ3	
92	I	DRQ5	
94	I	DRQ6	
96	I	DRQ7	DMA Requests 0-3, 5-7 are asynchronous requests used by peripherals to request DMA services. These requests are prioritized with DRQ0 having the highest priority and DRQ7 the lowest. DRQ must be held active until the corresponding DMA acknowledge (DACK) line goes active.
89	O	DACK0-	
47	O	DACK1-	
45	O	DACK2-	
43	O	DACK3-	
91	O	DACK5-	
93	O	DACK6-	

Pin	Type	Name	Description
95	O	DACK7-	DMA Acknowledge 0-3, 5-7 are active-low acknowledge signals issued by the 82C836 after a DMA service request (via a DRQ line) and successful arbitration. During power-on reset, these signals are used for Reset Strap Options described in an earlier section of this Data Book.
55	O	TC	Terminal Count is an active-high output pulse to the I/O channel that indicates the end of a DMA transfer.

Miscellaneous Signals

Pin	Type	Name	Description
39	I	PWRGOOD	Power Good is an active-high input from the power supply. When this signal is high, it indicates that all power supply voltages have reached their working levels. CPURST (pin 138) and XRST (pin 134) remain high for at least 200 µs after PWRGOOD goes high. While PWRGOOD is low, all pins except OSC2 float.
138	O	CPURST	CPU Reset is an active-high output that resets the 80386sx processor. CPURST goes active at power-up or during a software-initiated CPU reset.
134	O	XRST	Peripheral Reset is an active-high output that resets external peripherals and the coprocessor (if present) during power-up.
132	O	SPKOUT	Speaker Data is a waveform (the output of channel 2 of the timer/counter gated by bit 1 of port 61H) that is routed to a driver circuit, a low pass filter, and then to the speaker.
141	I/O	MFP5	Multi Function Pin 5 is a bi-directional pin that has two functions (see DACK5-): In internal RTC mode, MFP5 is the Power Sense (PS) input, active high, indicating the state of the battery. The PS signal should be powered from the battery back-up circuit. In external RTC mode, MFP5 is the Real Time Clock Chip Select (RTCCS-), active low. It is a decode of the I/O address range 070H-071H.

Pin	Type	Name	Description
140	I	MFP4	Multi Function Pin 4 is an input that has two functions (see DACK5-). In internal RTC mode, MFP4 is the 32KHz Oscillator Input, which should be connected to a square wave source with a frequency of 32.768 KHz. This input signal is the timing reference for the internal RTC of the 82C836. In external RTC mode, MFP4 is Interrupt Request 8 (IRQ08) from the external RTC.
126	I	MFP3	Multi Function Pin 3 is an input pin that is driven by the RESET2- signal from the 8042 keyboard controller. In response to this input (active low), the 82C836 generates CPURST to reset the CPU. The signal is under software control via the 8042. This pin may have an alternate function in a future version of the 82C836, but only the RESET2- function is available at present.
130	O	MFP2	Multi Function Pin 2 is an output pin that is used for 8042 Chip Select (8042CS-). This signal, an active-low output to the 8042 keyboard controller, is a decode of the I/O address range 060H-06FH. This pin may have an alternate function in a future version of the 82C836, but only the 8042CS- function is available at present.
129	I	MFP1	Multi Function Pin 1 is an input pin that is driven by the GATEA20 signal from the 8042 keyboard controller. In response to a low level on this pin, the 82C836 keeps MODA20 low during CPU cycles (depending on Fast Gate A20, Port 92H). A high level on GATEA20 allows MODA20 to follow A20. This pin may have an alternate function in a future version of the 82C836, but only the GATEA20 function is available at present.
125	I	MFP0	Multi Function Pin 0 is an input pin that acts as Interrupt Request 1 (IRQ1). This signal, an active-high input from the 8042 keyboard controller, is used by the 82C836's internal interrupt controller. This pin may have an alternate function in a future version of the 82C836, but only the IRQ1 function is available at present.
19, 62		V_{CC}	+5 Volts
102, 139		V_{CC}	
1, 22		VSS	Ground (substrate)

59, 81	VSS
99, 120	VSS
123, 142	VSS

Numerical Listing of Pin Assignments

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	VSS	21	A15	41	IOCHRDY	61	XD01
2	PROCCLK	22	VSS	42	IOCHCK-	62	VCC
3	CXIN	23	A16	43	DACK3-	63	XD02
4	ADS-	24	A17	44	DRQ3	64	XD03
5	A00	25	A18	45	DACK2-	65	XD04
6	A01	26	A19	46	DRQ2	66	XD05
7	A02	27	A20	47	DACK1-	67	XD06
8	A03	28	A21	48	DRQ1	68	XD07
9	A04	29	A22	49	IRQ09	69	XD08
10	A05	30	A23	50	IRQ07	70	XD09
11	A06	31	W/R-	51	IRQ06	71	XD10
12	A07	32	D/C-	52	IRQ05	72	XD11
13	A08	33	BHE-	53	IRQ04	73	XD12
14	A09	34	M/IO-	54	IRQ03	74	XD13
15	A10	35	HLDA	55	TC	75	XD14
16	A11	36	HOLD	56	BUSCLK	76	XD15
17	A12	37	READY-	57	ALE	77	XMEMR-
18	A13	38	ROMCS-	58	MODA0	78	XMEMW-
19	VCC	39	PWRGOOD	59	VSS	79	XIOR-
20	A14	40	0WS-	60	XD00	80	XIOW-

Numerical Listing of Pin Assignments (continued)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
81	VSS	101	CASH-	121	OSC2	141	MFP5
82	MEMCS16-	102	VCC	122	OSC1	142	VSS
83	IOCS16-	103	MWE-	123	VSS	143	NMI
84	IRQ10	104	RAS3-	124	REFRESH-	144	INTR
85	IRQ11	105	RAS2-	125	MFP0	145	D15
86	IRQ12	106	RAS1-	126	MFP3	146	D07
87	IRQ15	107	RAS0-	127	NPBZ-	147	D14
88	IRQ14	108	MA0	128	ERROR-	148	D06
89	DACK0-	109	MA1	129	MFP1	149	D13
90	DRQ0	110	MA2	130	MFP2	150	D05
91	DACK5-	111	MA3	131	LOMEGCS-	151	D12
92	DRQ5	112	MA4	132	SPKOUT	152	D04
93	DACK6-	113	MA5	133	BUSY-	153	D11
94	DRQ6	114	MA6	134	XRST	154	D03
95	DACK7-	115	MA7	135	SDIRL	155	D10
96	DRQ7	116	MA8	136	SDIRH	156	D02
97	MASTER-	117	MA9	137	NA-	157	D09
98	MODA20	118	PARL	138	CPURST	158	D01
99	VSS	119	PARH	139	VCC	159	D08
100	CASL-	120	VSS	140	MFP4	160	D00

System Timing Relationships

This section contains timing diagrams for all the major timing sequences in an AT-compatible SCATsx system. The diagrams emphasize the functional timing relationships between signals, i.e., what conditions cause the various timing events. For clarity, propagation delays and rise/fall times are either not shown or not drawn to scale.

Each diagram is accompanied by explanatory notes describing important aspects of the timing sequence.

CPU Access to AT Bus

Figure 36 shows a bus convert read or write. CMD- refers to the following possibilities:

- XMEMR- for a memory read cycle
- XMEMW- for a memory write cycle
- XIOR- for an I/O read cycle
- XIOW- for an I/O write cycle

LOMEGCS- is generated during memory cycles in which the memory address is in the first 1 MB of address space, i.e., 0-0FFFFFH. The only intended use of LOMEGCS- outside the 82C836 is to enable the SMEMR- and SMEMW- drivers, allowing SMEMR- to be asserted if XMEMR- is asserted, or SMEMW- if XMEMW- is asserted.

If the cycle is a memory read in an address range allocated for on-board ROM, ROMCS- is asserted at the same time as XMEMR-. ("Same time" here means on the same clock edges, neglecting propagation delays.)

The delay from ALE to the start of command, and the width of the command, depend on the type of cycle, as shown in Table 31.

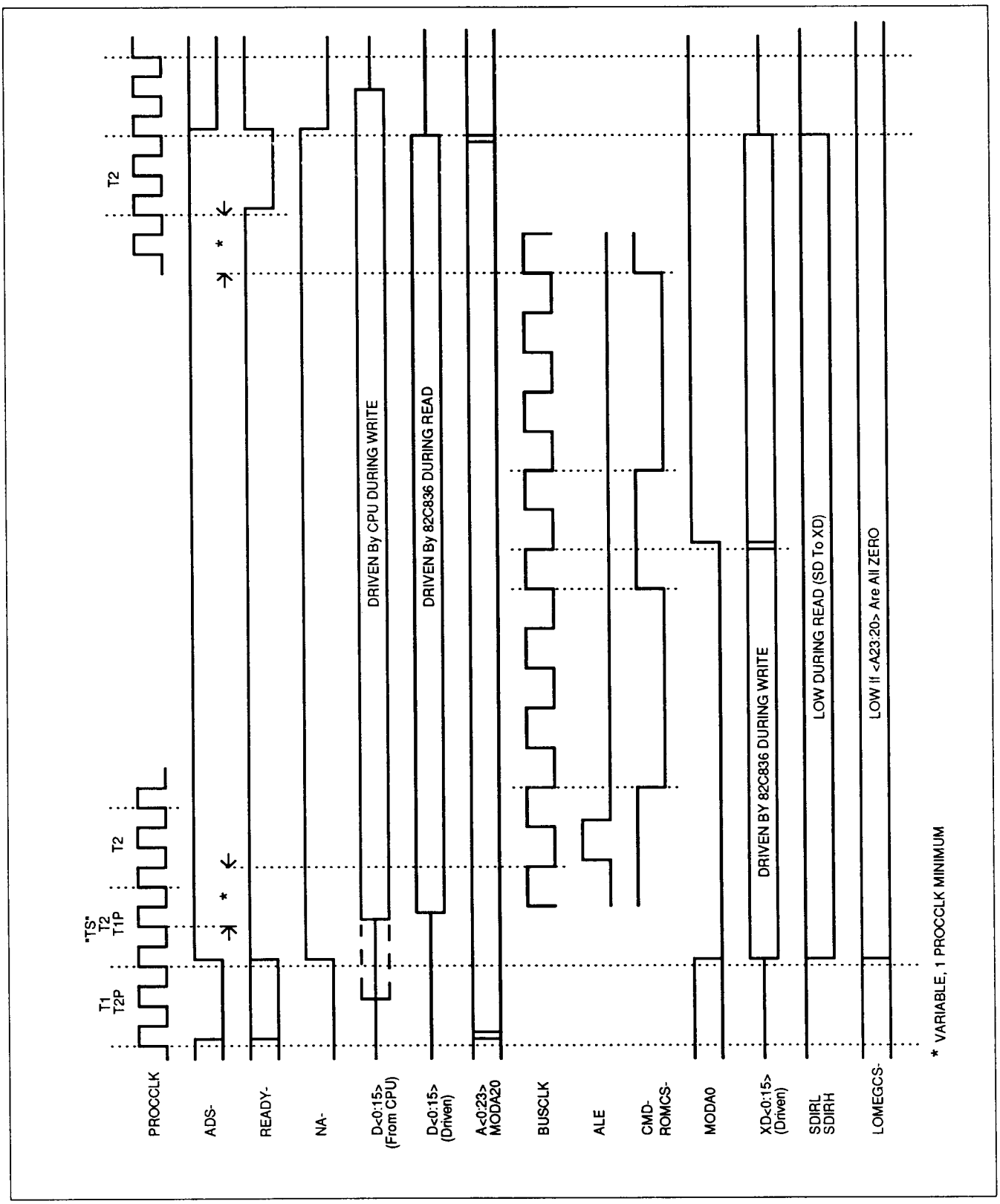
Table 31 *Delay from ALE to Start of Command (in BUSCLK cycles)*

Type of Cycle	ALE:CMD	CMD Width
Memory Access to 16-bit resource	0	2.0
I/O Access to 16-bit resource	0.5	1.5
Memory or I/O to 8-bit resource	0.5	4.5

Figure 36 shows a command width of 2.5 cycles of BUSCLK and a CMD delay of 0.5 cycle, a combination that never actually occurs unless 0WS- is asserted. If the CMD delay is 0.5 cycle of BUSCLK, the default command width will always be either 1.5 or 4.5 cycles of BUSCLK. During bus conversions (which can occur only when accessing an 8-bit resource), the command inactive delay between the two bus cycles is always 1.5 cycles of BUSCLK, as shown in the diagram. It is the resource type (8-bit or 16-bit), not data transfer size, that determines cycle timing. For example, an 8-bit access to a 16-bit resource (MEMCS16- and/or IOCS16- asserted) follows the 16-bit timing shown above. MEMCS16- and IOCS16- timing is shown in Figure 38.

For the 82C836, the first useful T-state of each CPU cycle is either T1P or the first T2 after T1, i.e., the T-state in which ADS- changes from low to high. This T-state is referred to as "TS" (T-start) and is directly equivalent to the TS state in 80286-based systems. The delay from the middle of "TS" to the rising edge

Figure 36 *CPU Access to AT Bus*



of ALE is variable, depending on the phase and frequency relationship of BUSCLK to PROCCLK. Although BUSCLK is derived from (and therefore synchronous with) PROCCLK, the exact phase can still vary from one bus cycle to the next, especially if BUSCLK = PROCCLK/5 or PROCCLK/6. The minimum delay from mid-TS to ALE rise is one cycle of PROCCLK. Likewise, the minimum delay from command rise to READY- fall is also one cycle of PROCCLK. READY-, in turn, is always synchronized to the last T2 or T2P of the bus cycle.

Bus conversions always begin with MODA0 forced low for the first AT bus cycle, then forced high for the second cycle. This is true for the "Force Bus Convert" mode as well as normal bus conversions.

Normal bus conversion occurs on any CPU-generated memory or I/O read or write in which the CPU is requesting 16 bits of data to or from an 8-bit resource at an even memory or I/O address. (If the address is odd, the CPU itself automatically performs two separate byte transfers for a word operand.)

If no bus conversion is needed, i.e., only a single AT bus cycle is needed, there will be only one command pulse. End-of-cycle timing then begins at the end of the first command pulse instead of the second one. MODA0 tracks CPU A0 (BLE-), but is still latched during each TS (transparent during the first half of TS).

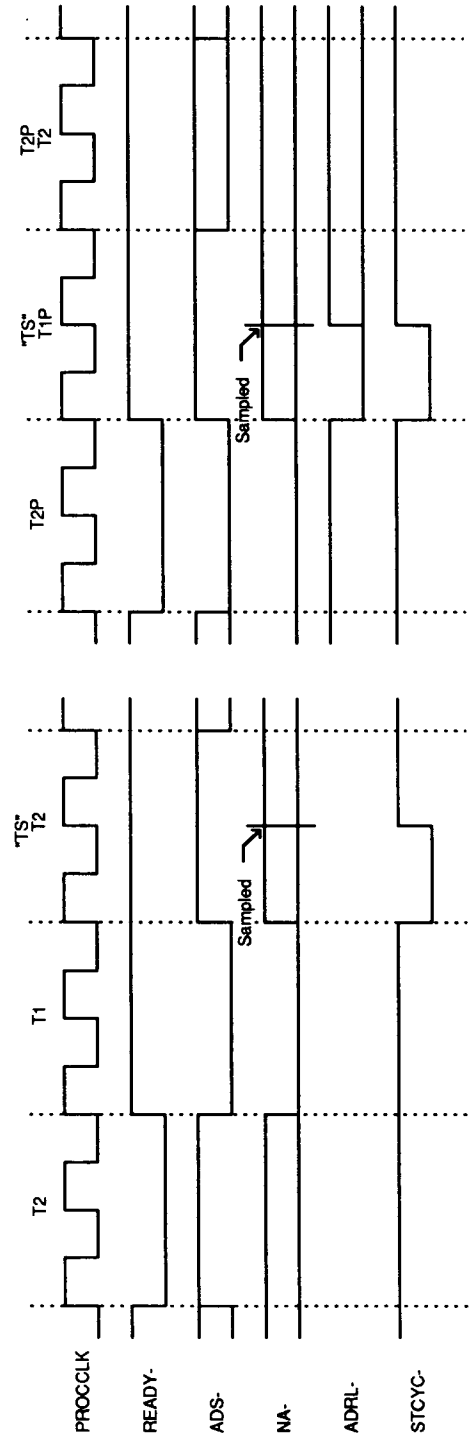
CPU operation during AT Bus cycles is always non-pipelined, as shown in Figure 37, if the NA- output from the 82C836 is connected to the NA- input on the CPU. However, the 82C836 NA- signal can be used for other purposes, and the CPU NA- input can be tied low or high or be controlled by other external logic. In such cases, it is possible for the CPU to operate in pipelined mode during an AT Bus cycle, resulting in the following changes in the timing relationships:

- The last state of the cycle can be a T2P instead of T2, in which case the next T-state will be T1P, with a new CPU cycle starting immediately. (If the final state is T2, then the earliest that a new cycle can start is the first T2 after T1.)
- NA- timing is affected as shown in Figure 37.
- CPU address no longer remains valid throughout the cycle, so it must be latched externally in order to meet SA-bus requirements. If 74F543 latches are used, NA- from the 82C836 can be used as a latch enable. (A 74F244 and 74F373 can be used instead of a 74F543 if desired.)

CPU accesses to the AT Bus, whether memory or I/O, read or write, can be "claimed" by external logic using the Early READY or LBA modes (see "Early Ready and LBA Modes"). If either of these modes is enabled, but an AT bus access is not "claimed" by external logic, the 82C836 will perform a normal AT Bus cycle exactly as shown in the diagram, except for one minor timing difference: one additional PROCCLK cycle is inserted in the delay from mid-TS to ALE. In effect, the 82C836 waits until the end of TS instead of the middle to decide whether or not it should perform an AT Bus cycle.

SDIRL and SDIRH typically are either both low or both high. During DMA or Master cycles, however, byte swapping may require one of these signals to be low while the other remains high. The default state during bus idle periods is both signals high.

Figure 37 *NA-/STCYC- Timing*



8042CS- and RTCCS- (MFP5 with external RTC), if asserted, follow LOMEGCS- timing, namely, they are updated at the start of each TS and remain latched until the next TS state or HLDA assertion.

Interrupt Acknowledge cycles are treated as I/O reads from an 8-bit resource, except that ALE and command are not generated, and the read data is an interrupt vector number originating from the interrupt controller section of the 82C836.

Halt cycles result in assertion of READY-, but no command is generated on the AT Bus.

Shutdown cycles result in the assertion of both READY- and CPURST. Shutdown is used intentionally by some software programs, especially on 80286-based systems, to trigger a CPU reset and thereby bring the CPU out of protected mode. CPU reset doesn't necessarily result in system reboot; in AT-compatible architectures, certain designated locations in CMOS RAM are used by the BIOS to determine how the BIOS will respond to the CPU reset. The BIOS can be instructed to return control to a resident Real-Mode program rather than reboot the operating system.

Other than shutdown, the conditions that can cause the 82C836 to reset the CPU are:

- Fast CPU reset via port 92H (PS/2 compatible).
- CPU reset via the 8042 keyboard controller (AT-compatible).
- Hardware system reset via PWRGOOD (also causes XRST).

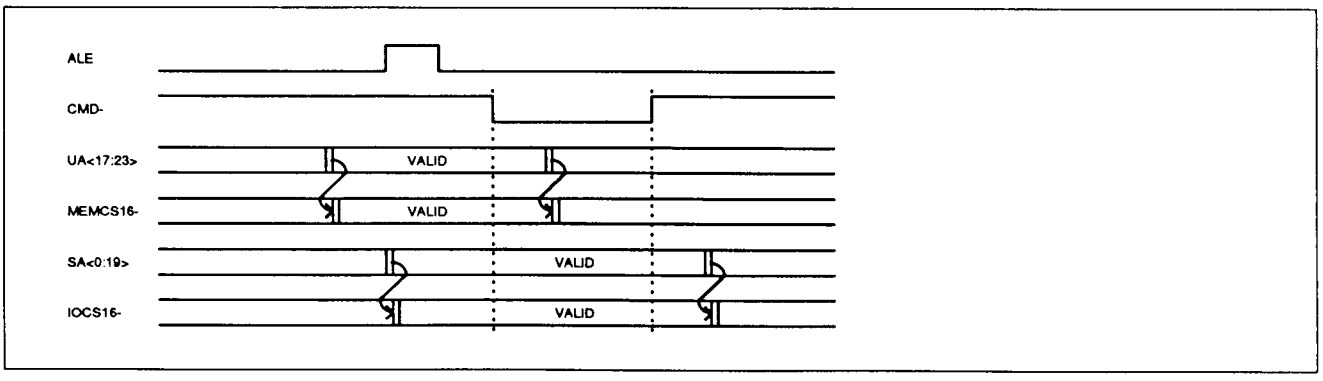
Memory writes to an area that has been programmed as EPROM (ROMCS-) result in write cycles on the AT bus. XMEMW- is generated.

Memory writes to write-protected shadow RAM go nowhere, i.e., normal DRAM write timing is followed except that CAS- is suppressed. No cycle is generated on the AT bus.

MEMCS16- and IOCS16- Timing

Figure 38 shows the timing relationships for MEMCS16- and IOCS16-. In most AT-compatible architectures, MEMCS16- will be an unlatched decode of the high-order unlatched address bits, UA17-23. Similarly, IOCS16- will be an unlatched decode of the latched address bits, SA0-19.

Figure 38 *MEMCS16 -& IOCS16- Timing*



When an attached memory resource detects an address in its assigned range, it should assert MEMCS16-. When an attached I/O resource detects an address in its assigned range, it should assert IOCS16-. Since there is no way for a device on the AT Bus to know whether the cycle is a memory access or I/O access prior to a command signal going active, it is possible for IOCS16- and MEMCS16- both to be asserted simultaneously, potentially by different AT Bus add-on cards.

As was shown in Figure 36, the 82C836 (via NA-) latches address bits 1-23 on the AT Bus before the start of ALE, and keeps the address latched until after the end of the command pulse. MODA0 also follows the same timing except for the low to high transition after the first cycle of a bus conversion.

The 82C836 requires MEMCS16- to be valid at the end of ALE, i.e., from slightly before the falling edge of ALE until slightly after ALE falls.

The 82C836 requires IOCS16- to be valid from slightly before the start of command until slightly after the end of command.

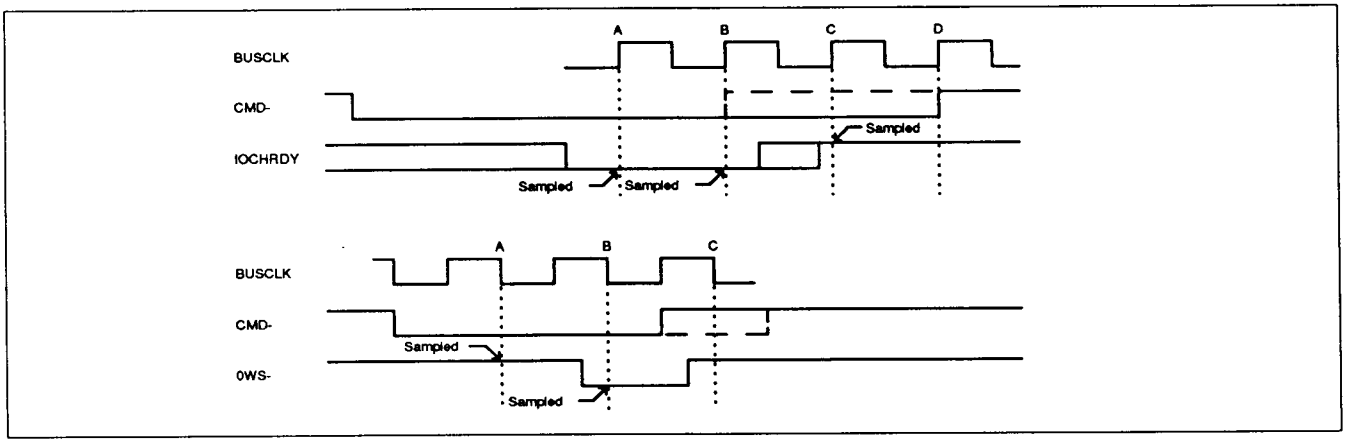
Although MEMCS16- and IOCS16- normally are asserted by 16-bit resources on the AT Bus, the 82C836 can assert either or both of these signals in the following cases:

- During on-board ROM read, if on-board ROM is 16 bits wide.
- During CPU, DMA or Master access to on-board DRAM.
- During access to EMS I/O ports 2x8H and/or 2x9H (where "x" is programmable as ether 0 or 1). A byte access to either port, or a word access at I/O address 2x8H, results in assertion of IOCS16. These two EMS I/O ports operate as a 16-bit I/O resource; data transfer to or from 2x9H always uses D8-15 and/or XD8-15.

IOCHRDY and 0WS

Figure 39 shows the effects of IOCHRDY and 0WS- during CPU accesses to the AT bus. If IOCHRDY and 0WS- both remain high, the command width is determined by the defaults listed earlier. The default command timing is represented by the dashed lines. IOCHRDY can increase the command width, while 0WS can reduce it.

Figure 39 *IOCHRDY & 0WS-*



IOCHRDY is first checked at time "A," one BUSCLK before the default end of command (B). Detection of IOCHRDY low at time (A) causes the command to remain active for an additional BUSCLK. IOCHRDY is sampled again on each successive rising edge of BUSCLK until it is detected high. The command ends exactly one BUSCLK after IOCHRDY is detected high.

0WS- is checked on each falling edge of BUSCLK after command is active. Detection of 0WS- low causes the command to end on the next rising edge of BUSCLK. 0WS- is operative in this manner on all CPU accesses to the AT Bus, including I/O as well as memory, 8-bit as well as 16-bit.

It is not valid for IOCHRDY and 0WS- both to be low during the same bus cycle.

Neither IOCHRDY nor 0WS- is ever driven by the 82C836. These signals are 82C836 inputs only.

NA-/STCYC- Timing

Figure 37 above shows all possible timing relationships for the NA-/STCYC- output from the 82C836. If NA- mode has been selected (via the DACK3- strap option), the default state of NA- during idle periods is low. NA- is sampled by the CPU at the middle of T2 or T1P.

If the 82C836 NA- output is connected to the CPU NA- input and NA- is high during a bus cycle, the CPU will operate in non-pipelined mode during the cycle, and the cycle ends with a T2 state, not T2P. NA- always goes low at the end of the final T2 of each cycle. It goes high again if needed at the start of the next T2 after T1.

If NA- stays low during a cycle, it will go high if needed at the start of the next "TS" (T1P or first T2 after T1).

If the 82C836 NA- output is not connected to the CPU NA- input, then it is possible for the CPU to remain in pipelined mode even though the 82C836 NA- output is high. In this case, NA- still goes low at the end of the final T2P, but it may go high again at the middle of the following T1P. Thus, there is a guaranteed low time of one PROCCLK on the 82C836 NA- output, which allows it to remain usable as an address latch enable signal (ADRL-).

If the STCYC- mode is selected instead of NA-, then the CPU NA- input must be tied high or low or controlled externally, and external latches (74F543 or equivalent) must be used between the CPU address bus and SA-bus. As shown in the diagram, STCYC- goes low during the first half of each TS, thereby signalling external devices that a bus cycle is starting. This timing also allows STCYC- to function as an address latch enable for the A0-23 to SA-bus latches.

The NA-/STCYC- output is forced low when HLDA is high. This is redundant in the NA- mode, since NA- is low between CPU bus cycles in any case. In STCYC- mode, forcing STCYC- low causes the SA-bus address latches (if used) to become transparent while HLDA is asserted, thereby allowing DMA and Refresh addresses to pass through. (Master addresses flow through in the opposite direction and are not latched by ADRL-.)

In NA- mode, the NA- output operates as a "local memory hit" indicator. Low on NA- at mid-TS indicates that the 82C836 has detected a CPU read or write to local DRAM. In all other cases, NA- is high at mid-TS. Depending on CPU

address delay and the exact address decoding path taken through the 82C836, it is possible for NA- to "glitch" momentarily at the start of TS if NA- was low during the preceding bus cycle. As long as CPU address delay is within CPU specifications, NA- will be stable and valid by the middle of TS, where the CPU samples it.

The duty cycle of ADS- provides a quick indication of whether or not the CPU is operating in pipeline mode. If ADS- goes back low at any time during a bus cycle before the end of READY-, the CPU has entered the pipeline state (T2P) and the next T-state after READY will always be T1P, never TI or TH. On the other hand, if ADS- is still high when READY- is asserted, the next T-state after READY will be T1, TI or TH, but never T1P. If a bus cycle ends in a non-pipelined state (T2), the only way the CPU can get back into the pipeline state (T2P) is to go through T1 and at least one T2 first.

NA-/STCYC- operation and timing are unaffected by "cycle claiming" in the Early READY or LBA modes. The state of NA- at mid-TS still indicates whether or not the 82C836 detected a CPU memory access at an address that resides in local DRAM.

CAS-Only DRAM Access by CPU

Figure 40 shows the fastest possible CPU accesses to local DRAM. RAS- and row address are already valid from a previous cycle, so only CAS- needs to be cycled. The basic protocol is:

- Column address becomes valid at start of TS.
- CAS- goes low one PROCCLK later (for read), four PROCCLKs for write.
- CAS- stays low until end of cycle.
- Read data becomes valid by end of cycle.
- Write data and parity are valid at mid-TS.

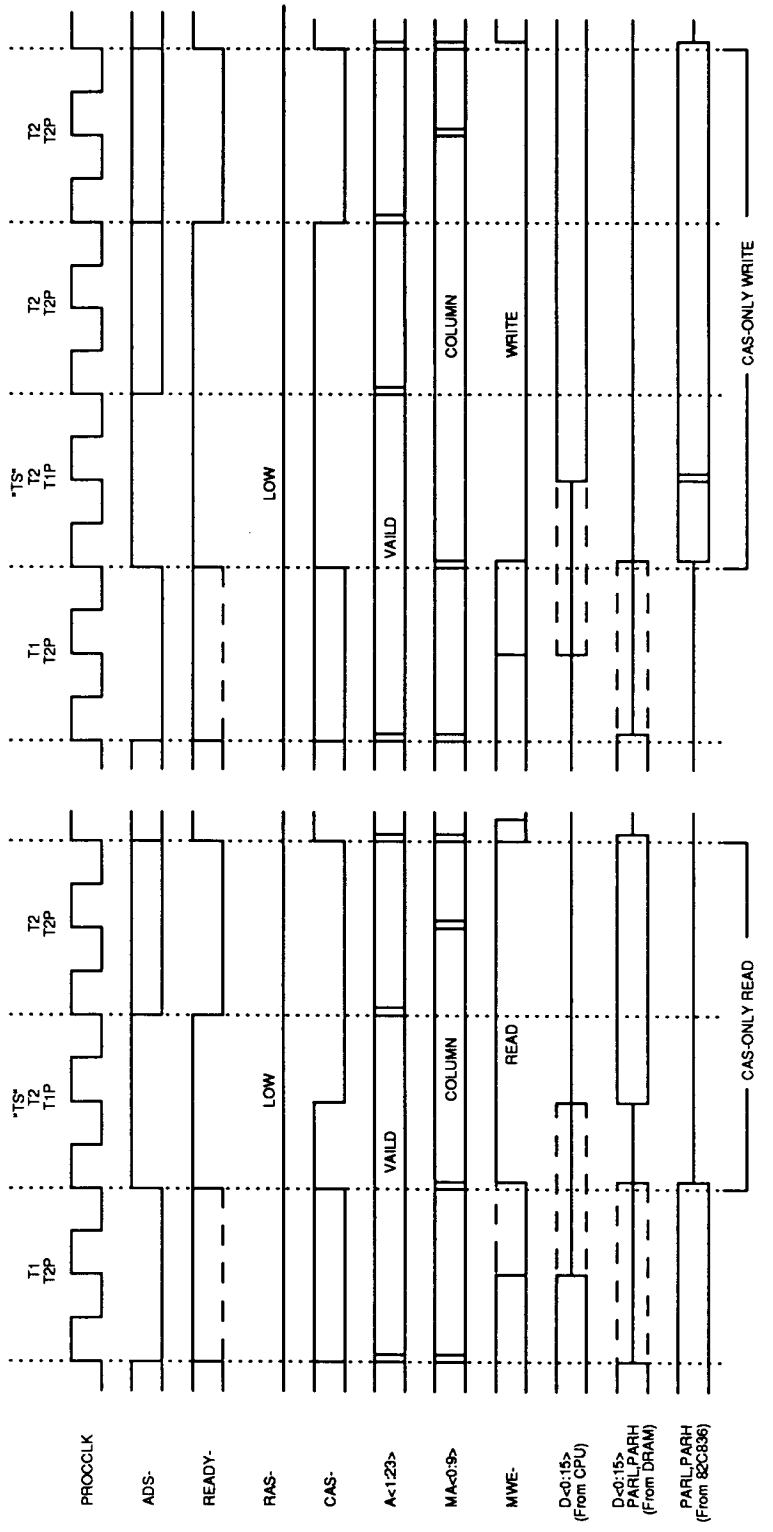
In pipeline mode, the shortest possible local memory read cycle is two T-states: T1P, T2P. The shortest possible pipelined memory write is T1P, T2P, T2P (three T-states). The extra T2P during writes allows time for parity generation. Non-pipelined cycles follow the same timing except for an added T1 state at the beginning.

The CAS- active time is three PROCCLK cycles for read, two PROCCLK cycles for write. The minimum CAS- high (precharge) time is one cycle of PROCCLK. Minimum column address setup time before CAS- goes active is one PROCCLK. Minimum column address hold time after CAS- goes active is also one PROCCLK.

For external cache support, the end of the CAS- pulse has a small additional propagation delay built-in so that the read data will remain valid long enough for the cache data RAM to capture it (cache read miss cycle). See "Early READY and LBA Modes" for other cache related timing.

MWE- is normally low so that it can be used externally as a memory data buffer direction control if desired. A small extra MWE- delay is provided at the end of the CAS- pulse to insure that CAS- always goes inactive before MWE- goes back low. If the cycle is non-pipelined, the falling edge of MWE- is delayed until the middle of the next T-state.

Figure 40 *CAS-Only DRAM Access by CPU*



If the "CAS-Extend" wait state is enabled, one additional T-state is inserted in the CAS- active time.

If internal EMS, "Early READY" mode or LBA mode is enabled (individually or in any combination), an extra T-state may be inserted at the beginning of the cycle before CAS- goes active. The extra T-state allows time for EMS address translation and/or "cycle claiming" by external logic such as a cache controller. In the case of Early READY or LBA, the early wait state occurs only on bus cycles that can be "claimed" by external logic, and only if the cycle is not externally "claimed." In the case of internal EMS, the early wait state occurs only if access is made to one of the four page windows and the window has been enabled for EMS, i.e., only if an EMS address translation actually needs to occur.

ALE and command are not generated on the AT Bus during local memory accesses by the CPU.

Local Dram Bank Switch

Figure 41 shows CPU accesses to local DRAM in "zero wait state" mode with RAS- initially high. These timing relationships apply when the DRAM bank that needs to be accessed is different from the bank for which RAS- was already active. Since RAS- for the desired bank was already high, no further RAS- precharge time is required for the desired bank. The basic protocol is the same as for CAS-only cycles, except as follows:

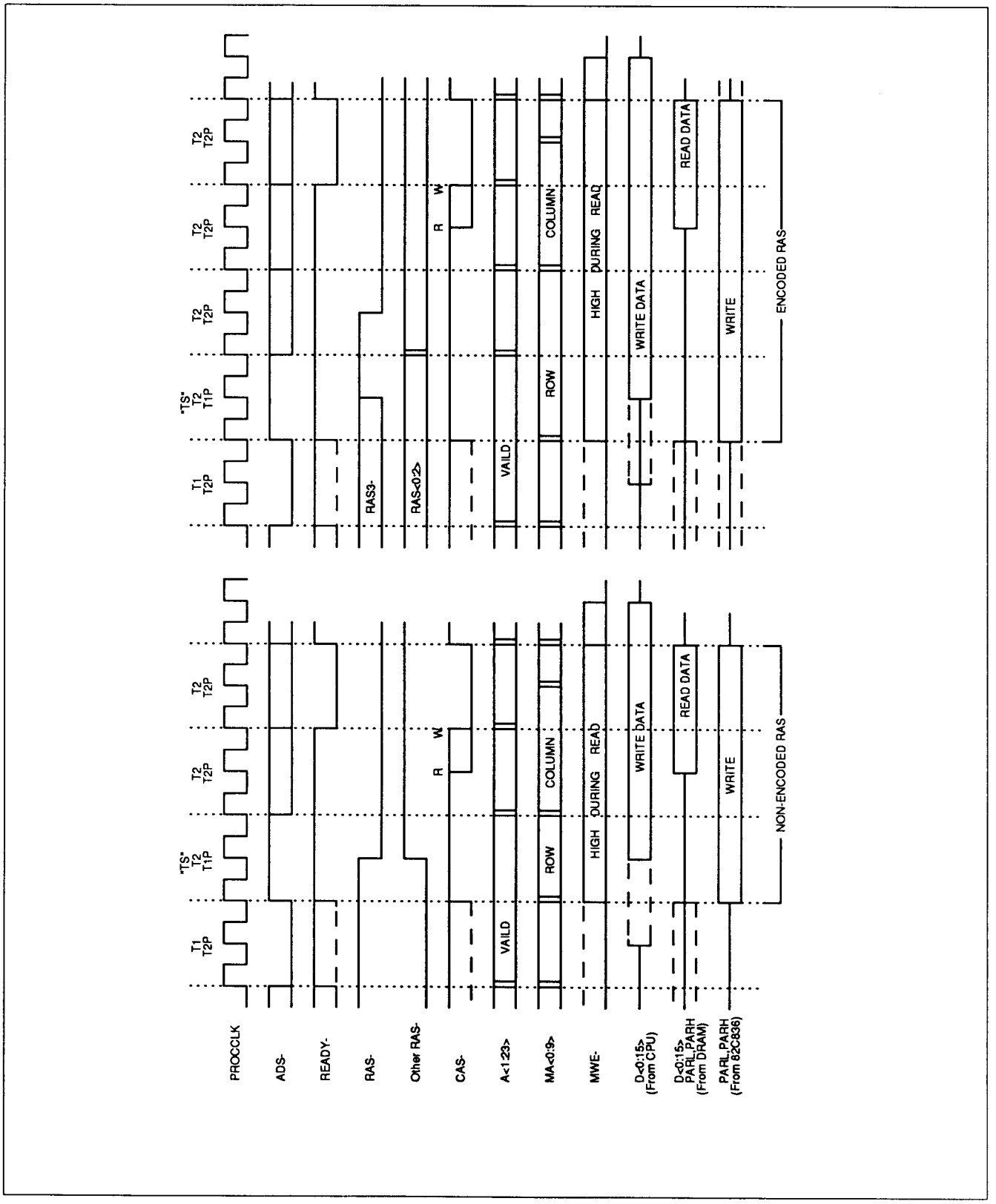
- RAS- for the target bank goes active at the middle of TS, and RAS- for the previously active bank simultaneously goes inactive.
- Row address is valid for one PROCCLK before and after RAS- goes active.
- If encoded RAS is being used, RAS3- goes inactive at the middle of TS; one PROCCLK later, RAS0-2 change to the new value; after one more PROCCLK, RAS3- again goes active. The net penalty for encoded RAS is one T-state (only for bank switch cycles).

These diagrams also apply if no RAS- was previously active. This can happen during the first local memory access following a refresh or DMA cycle, or following a RAS timeout.

The minimum bank switch cycle for non-encoded RAS consists of T1P, T2P, T2P (three T-states). For encoded RAS, an extra T2P is needed. For non-pipelined operation, an extra T1 occurs at the beginning of the cycle. As with CAS-only cycles, a "CAS Extend" wait state can also be enabled. An "Early wait state" can also be enabled for EMS or external cache support. Minimum CAS- active time is two PROCCLK cycles for write, or three PROCCLK cycles for read.

The minimum time allowed for read data access from RAS- active is five PROCCLK cycles.

Figure 41 *Local DRAM Bank Switch*



Maximum Wait State Page Miss

Figure 42 shows the fully-burdened local DRAM cycle, i.e., the longest possible cycle (eight T-states). It is a page miss cycle, meaning that the target bank is the same as the previously selected bank, but the row address must be changed. Thus, the same RAS- signal must be cycled high and then low again.

- State T1 occurs only if the CPU ended the previous cycle in non-pipeline mode.
- Interval "A" is the "Early Wait State," applicable only when internal EMS, "Early READY" or LBA modes are enabled.
- Intervals "B" and "C" are needed for RAS- precharge time. For a bank switch, only "B" is needed. For CAS-only cycles, neither "B" nor "C" is needed.
- Intervals "D" and "E" are the normal RAS, Row/Column and CAS sequence. Only "E" is needed with CAS-only cycles.
- Interval "F" is the "CAS Extend" wait state (optional).

With the "Early Wait State" enabled, the minimum CAS- precharge time increases from one PROCCLK cycle to three. With the CAS Extend wait state, the minimum CAS- active time increases from two PROCCLK cycles to four. Neither type of wait state affects the delay from RAS- to row address hold, or from column address valid to CAS- active.

RAS timeout, if enabled, operates as follows:

- If RAS- remains low long enough to cause a RAS timeout (nominally 9.5 microseconds), the current memory cycle (if any) is allowed to finish, and then RAS- is forced high at the middle of the next T-state after the end of the cycle.
- The RAS- that had been active remains high for a minimum of four PROCCLKs. If an attempt is made to access the same bank again, the timing will follow either a page miss or a bank switch sequence as needed to insure four PROCCLK cycles minimum high time on RAS-.
- If access is made to a different bank, normal bank switch timing is followed.

Early READY and LBA Modes

Figure 43 shows how external logic can "claim" a CPU generated cycle by asserting LBA- or READY- during the first T2. In both cases, the 82C836 is prevented from generating a local memory or AT bus cycle. For a read, the external logic (typically a local cache) must provide the read data and generate READY-. The 82C836 is responsive to early READY- or LBA- only if one or both of these modes has been enabled by Internal Configuration Register 41H.

Either a local memory read or an AT Bus memory or I/O read or write can be "claimed" in this fashion. Local memory writes cannot be claimed by external logic and do not cause an "early wait state."

When these "cycle claiming" capabilities are used, the CPU must run in non-pipelined mode only; the CPU NA- input must be tied high. This is necessary because of CPU address timing. If the cycle is not "claimed," then the 82C836 may need to generate a local memory cycle. The 82C836 needs a valid CPU address in order to do this, but in pipeline mode the CPU address may become invalid too soon for the 82C836.

Figure 42 *Maximum Wait State Page Miss*

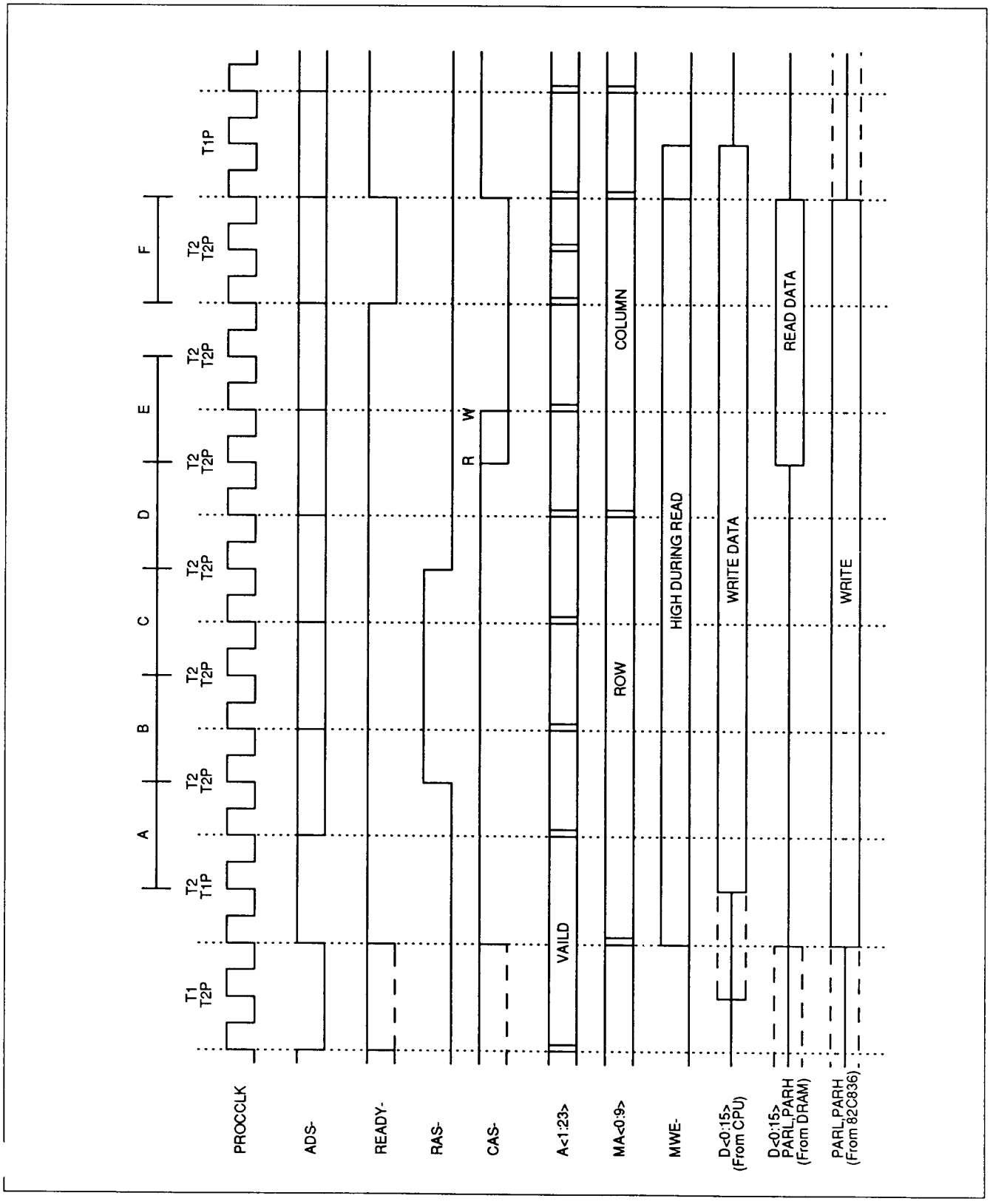
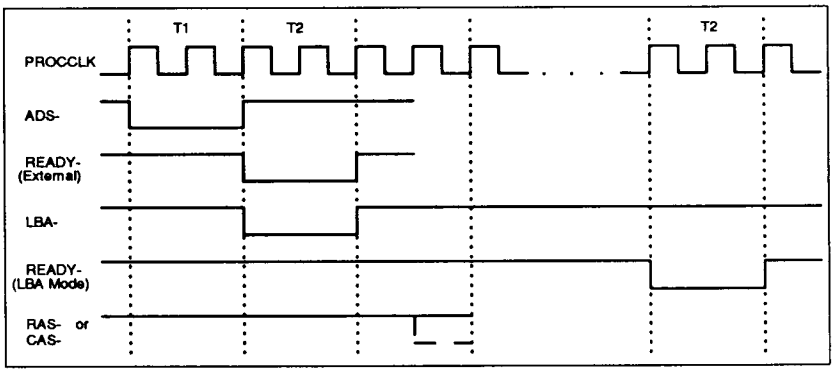


Figure 43 *Early READY and LBA-*



Whenever the 82C836 finishes driving READY- during any cycle, it continues to drive READY- actively high for one PROCCLK cycle. Then READY- is tri-stated, and an external pull-up resistor keeps it high. The tri-stating of READY- cannot occur later than mid-T1. Since the earliest that the 82C836 could assert READY- again is during the third T2 after T1 ("early wait state" enabled), the first T2 after T1 is available for external logic to assert READY- without conflicting with the 82C836.

The 82C836 samples LBA- and READY- at the end of the first T2 after T1. READY- active at that time terminates the cycle. LBA- active at that time causes the 82C836 to look for READY- at the end of each subsequent T2 state. The cycle terminates when the external logic asserts READY-.

If the external logic asserts both LBA- and READY- at the end of the first T2 (not a valid combination), the cycle terminates just as if READY- alone had been asserted.

The net performance effect of local cache is as follows:

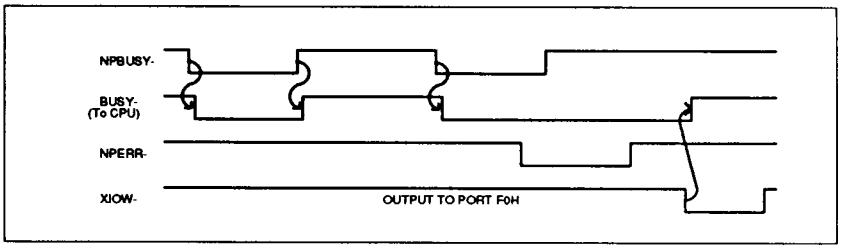
- Greatly increased percentage of two T-state memory reads due to high cache hit ratio. (Without cache, two T-state memory reads would still be possible in pipelined page mode, but the hit ratio would be considerably lower than with a cache.)
- Cache read misses increase from two T-states minimum to four T-states for CAS-only memory reads. Since cache read misses occur far less often with cache than without it, this penalty for cache read misses should have only a minor impact on overall system performance.
- Memory write cycles have a maximum of one extra T-state (the leading T1) compared to a non-cache architecture. (Any architecture will have some percentage of T1 states, even if the CPU pipeline mode is always enabled.) Since the vast majority of memory accesses normally will be reads rather than writes, the added T1 state for writes should have only a minor impact on overall system performance.

Coprocessor Timing

Figure 44 shows the relationship between the coprocessor busy and error signals and the busy signal sent to the CPU. Normally, BUSY- to the CPU simply follows NPBUSY- from the coprocessor. When a coprocessor exception occurs (NPERR- asserted), BUSY- to the CPU is latched (active) until the CPU acknowledges it by performing an I/O write to port F0H. This protocol is

AT-compatible and differs from the "generic" coprocessor interface internal to the 80386sx CPU. In particular, the ERROR- input to the CPU should be tied high, and AT-compatible software will rely on interrupt level 13 for reporting coprocessor exceptions.

Figure 44 *Coprocessor Timing*



The 80387sx coprocessor monitors ADS-, READY-, and CPU address bit 23 directly to detect I/O operations addressed to it. The coprocessor operates roughly as a 16-bit data resource except as noted follows:

1. In all cases, coprocessor data is exchanged on the CPU local data bus, not on the AT bus. IOCS16- is not asserted.
2. If no coprocessor is present, attempted coprocessor I/O accesses result in bus convert cycles, including ALE and command on the AT bus, but the data bus drivers in the 82C836 remain tri-stated. This allows plenty of time for the floating CPU data bus to stabilize to a valid state. (A stable data bus is important for compatibility with certain older software products that rely on attempted coprocessor I/O operations to detect coprocessor presence or absence.
3. If a coprocessor is present, but the 82C836 has been programmed to generate READY- during coprocessor accesses, ALE and command are generated and the cycle follows normal 16-bit I/O timing except as mentioned in (1) above.
4. If a coprocessor is present and the 82C836 has been programmed to rely on the coprocessor to generate READY-, ALE and command are not generated, and the cycle ends as soon as the coprocessor issues READY-, which may be considerably sooner than in (3) above.

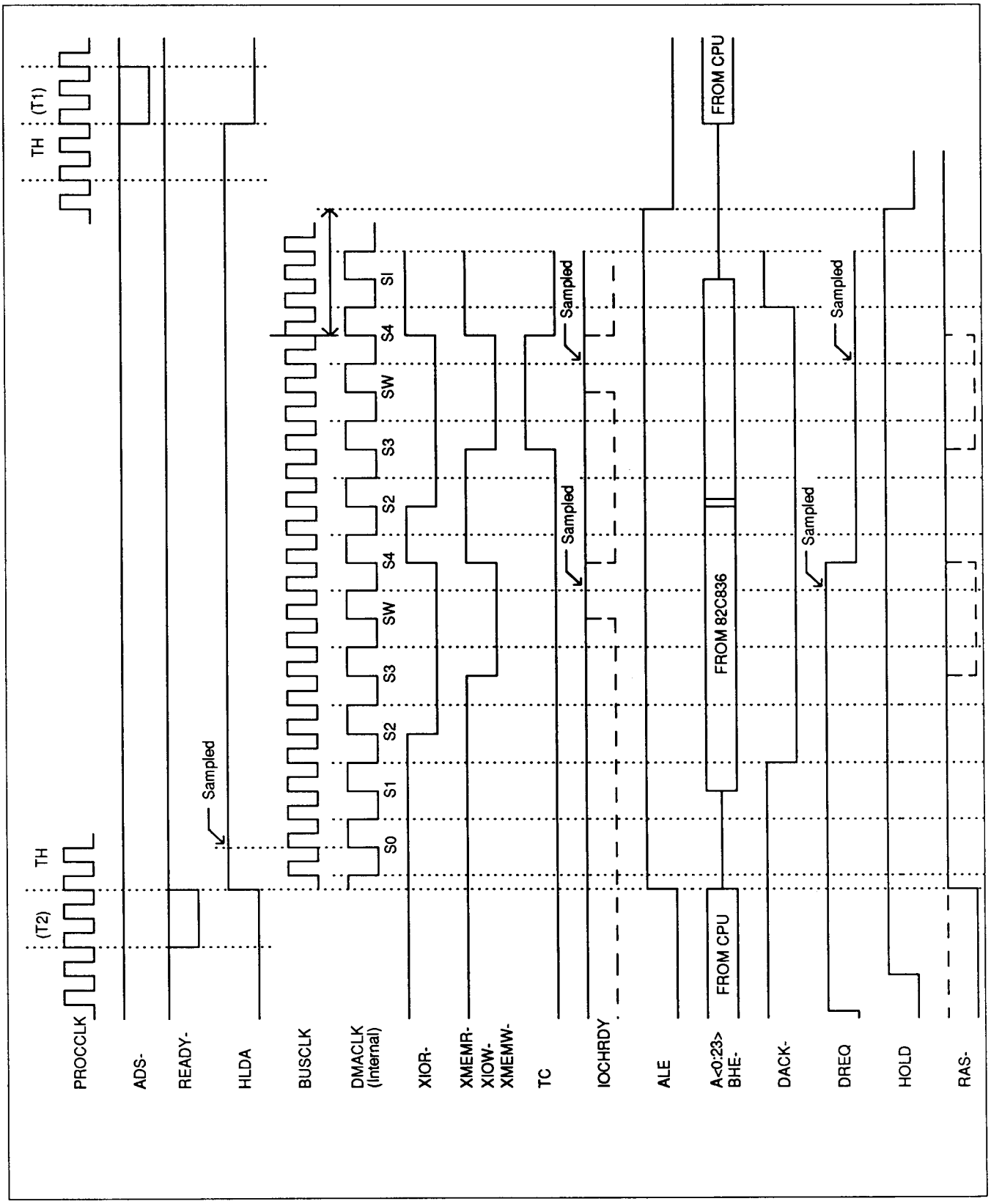
Since the coprocessor interface is tightly coupled to the CPU (rather than using an XIOR-/XIOW- interface), the coprocessor is not accessible by add-on card bus masters.

DMA Timing

Figure 45 shows a DMA operation consisting of two back-to-back DMA transfers between a memory resource and an I/O resource. The general protocol is as follows:

- A DMA requestor asserts an assigned DREQ signal. The 82C836 then asserts HOLD to the CPU.
- Eventually the CPU responds with HLDA, causing AEN to be asserted on the AT Bus (externally gated with MASTER-). HLDA also causes ALE to be asserted. ALE remains continuously asserted until HOLD is subsequently de-asserted.

Figure 45 *DMA Timing*



- Some time after detecting HLDA, the 82C836 generates the DMA memory address and issues the appropriate DACK- signal, followed by either XIOR- and XMEMW-, or XMEMR- and XIOW-.
- The address bus contains the memory address and is used by the memory resource. The I/O resource that should respond to the DMA cycle is determined by the DACK- signal. Other I/O resources not involved in the DMA cycle generally rely on AEN to signify that the address on the bus is not an I/O address.

DMA Channels 0-3 are designed for 8-bit transfers involving 8-bit I/O resources. The memory resources can be either 8-bit or 16-bit. IOCS16- is ignored, but MEMCS16- is used in conjunction with SA0 and SBHE- to control byte swapping between SD0-7 and SD8-15. Address bits 0-15 increment or decrement by one after each DMA cycle. Software intervention is needed to change the contents of address bits 16-23.

DMA Channels 5-7 are designed for 16-bit transfers involving 16-bit I/O resources. The memory resources must be 16-bit, also. IOCS16- and MEMCS16- are ignored, and BHE- and A0 are both driven low. Address bits 1-16 increment or decrement by one (i.e., address 0-16 increments or decrements by two, always even) after each DMA cycle. Software intervention is needed to change the contents of address bits 17-23.

DMA timing is driven by the internal DMA clock, which is either the same as BUSCLK or reduced to BUSCLK/2 (programmable). Figure 45 shows hardware default timing, which is AT-compatible:

- The default DMA clock is BUSCLK/2. Timing is divided into "S" states, each consisting of one DMA clock.
- The middle address bits (8-15 or 9-16) are updated at the middle of S1. Low-order address bits (0-7 or 1-8) are incremented at the middle of each S2.
- XIOR- is asserted (if needed) at mid-S2, but XMEMR- is asserted (if needed) at mid-S3. Assertion of XMEMR- can be changed to mid-S2 using ICR 01H.
- XIOW- and XMEMW- are asserted as needed at mid-S3. This can be changed to mid-S2 using "extended write" mode.
- There is one DMA wait state, SW (programmable up to four), and all commands are de-asserted at mid-S4.

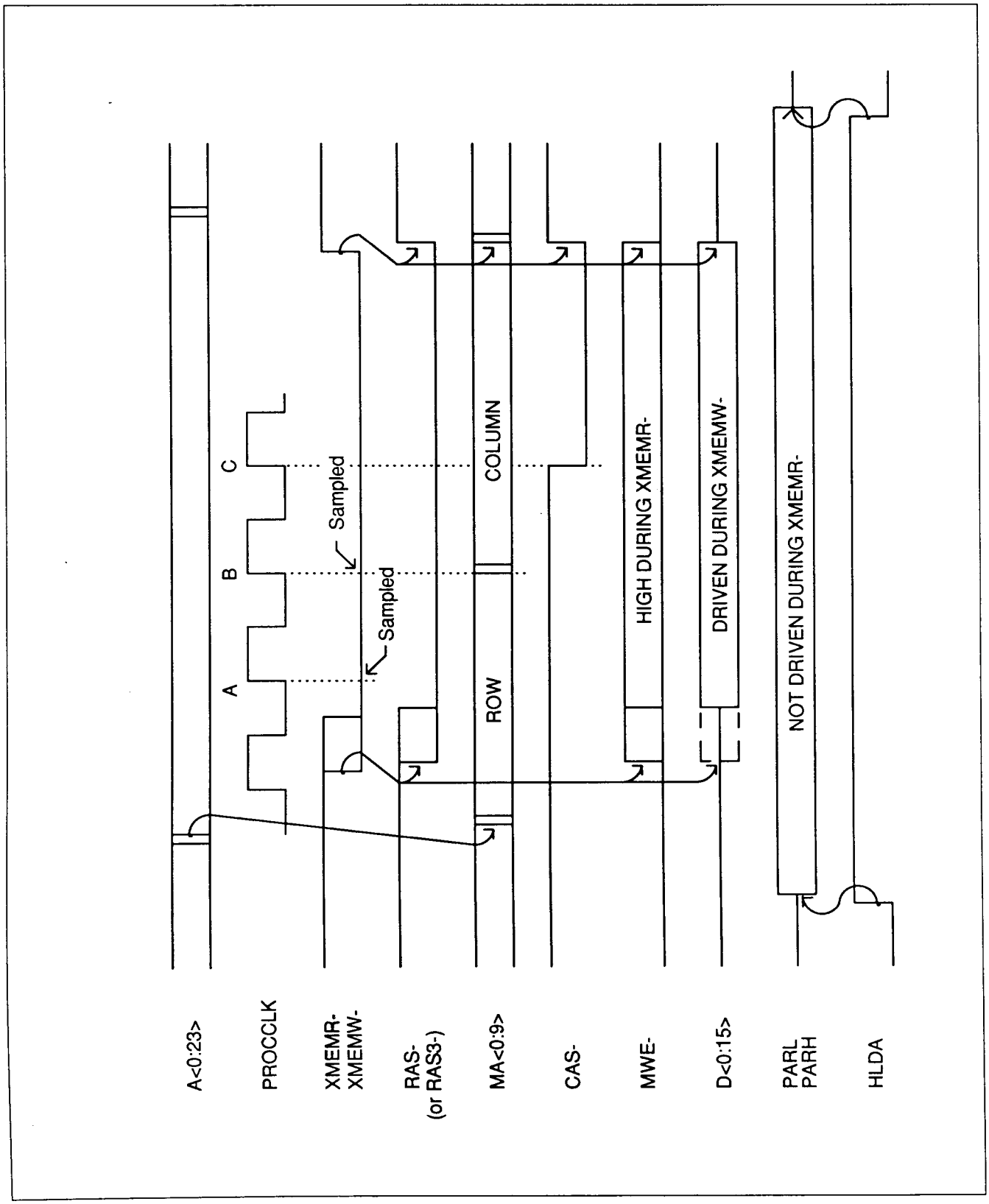
Using "compressed mode," the S3 states can be eliminated. S3 events in that case occur in S2 instead of S3. SW states are not affected.

TC, if asserted, coincides with XMEMW- or XIOW-.

Detection of IOCHRDY low at the end of SW causes additional SW states to be inserted until IOCHRDY is detected high at the end of an added SW. Although IOCHRDY during DMA is normally controlled only by memory resources, not I/O resources, the 82C836 has no means to distinguish the source of an IOCHRDY de-assertion and will respond in the same way regardless of which resource is controlling IOCHRDY.

DREQ is also sensed at the end of each SW. If DREQ is detected low, the DMA operation terminates after the end of the current DMA cycle. DMA and master access to local memory is shown in Figure 46.

Figure 46 *DMA & Master Access to Local Memory*



If RAS- was active before HLDA was asserted, RAS- is forced high immediately (rising edge of HLDA). Additional memory timing during DMA and Master cycles is shown in Figure 47.

To meet CPU timing requirements, HOLD is synchronized to PROCCLK. From the middle of the final S4 state to the falling edge of HOLD, there is a minimum delay of two complete cycles of the 14.3 MHz OSC clock, and an additional minimum delay of one complete cycle of PROCCLK. The involvement of the 14.3 MHz clock is for arbitration with refresh.

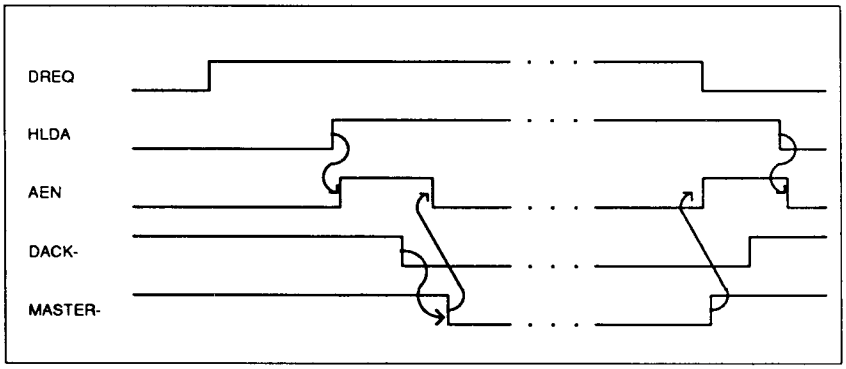
S0 is entered after assertion of HOLD. S0 is then repeated as needed until HLDA is detected. When HLDA is detected high at mid-S0, S1 will be entered, either immediately (DMA Channels 5-7) or after three more S0 states (Channels 0-3). The additional S0 states for Channels 0-3 occur because of the cascading between those Channels and Channels 5-7.

If the DMA clock is set to BUSCLK rather than BUSCLK/2, each "S" state is still one cycle of the DMA clock, and, except for the frequency of the DMA clock, S-state timing remains the same as described previously.

Master Arbitration

Figure 47 shows how an add-on card bus master obtains and later releases control of the buses. The general AT-compatible protocol is as follows:

Figure 47 *Master Arbitration*



- The Master asserts an assigned DREQ and waits for the corresponding DACK-, just as in a DMA sequence. However, before a DMA channel can be used for Master operation, software must program the channel for "cascade mode." Then, when DREQ is received for that channel, the 82C836 refrains from starting a DMA cycle after issuing DACK-.
- Upon receiving DACK-, the requesting Master asserts the MASTER- signal. This reverses the directions of various bus drivers, forces AEN to be de-asserted, and signifies that the Master is now in control.
- The Master can remain in control indefinitely, performing I/O or memory reads and writes as needed. The Master releases control by de-asserting MASTER- and DREQ.

Master read and write cycles on the AT Bus follow the same basic protocol as CPU reads and writes to the AT Bus, except as follows:

- Address and command are controlled by the add-on card bus master and are not necessarily synchronized to BUSCLK.

- ALE remains continuously high, and timing for UA17-23 is the same as the timing for SA0-19.
- Data transfer is between the Master and an I/O resource, or between the Master and a memory resource, using the same command signals as CPU controlled cycles.
- Byte swapping remains the responsibility of the 82C836, but any bus conversion that might be needed must be performed by the add-on card bus master. For byte swapping purposes, the Master is treated as a 16-bit resource and must not drive the SD0-7 and SD8-15 buses in a way that would conflict with byte swapping being performed by the 82C836.
- SMEMR- and SMEMW- remain under control of LOMEGCS-, which is driven by the 82C836 in response to the AT Bus address generated by the Master.

While an add-on card bus Master is in control, it can request a refresh cycle by driving the REFRESH- signal low. The 82C836 will then perform a refresh cycle in the normal manner, controlled by the 82C836.

By holding REFRESH- low longer than needed for a single refresh, the Master can request multiple back-to-back refresh cycles.

DMA and MASTER Access to Local Memory

During DMA and Master cycles, (see Figure 46) local memory timing follows a somewhat different protocol than during CPU cycles:

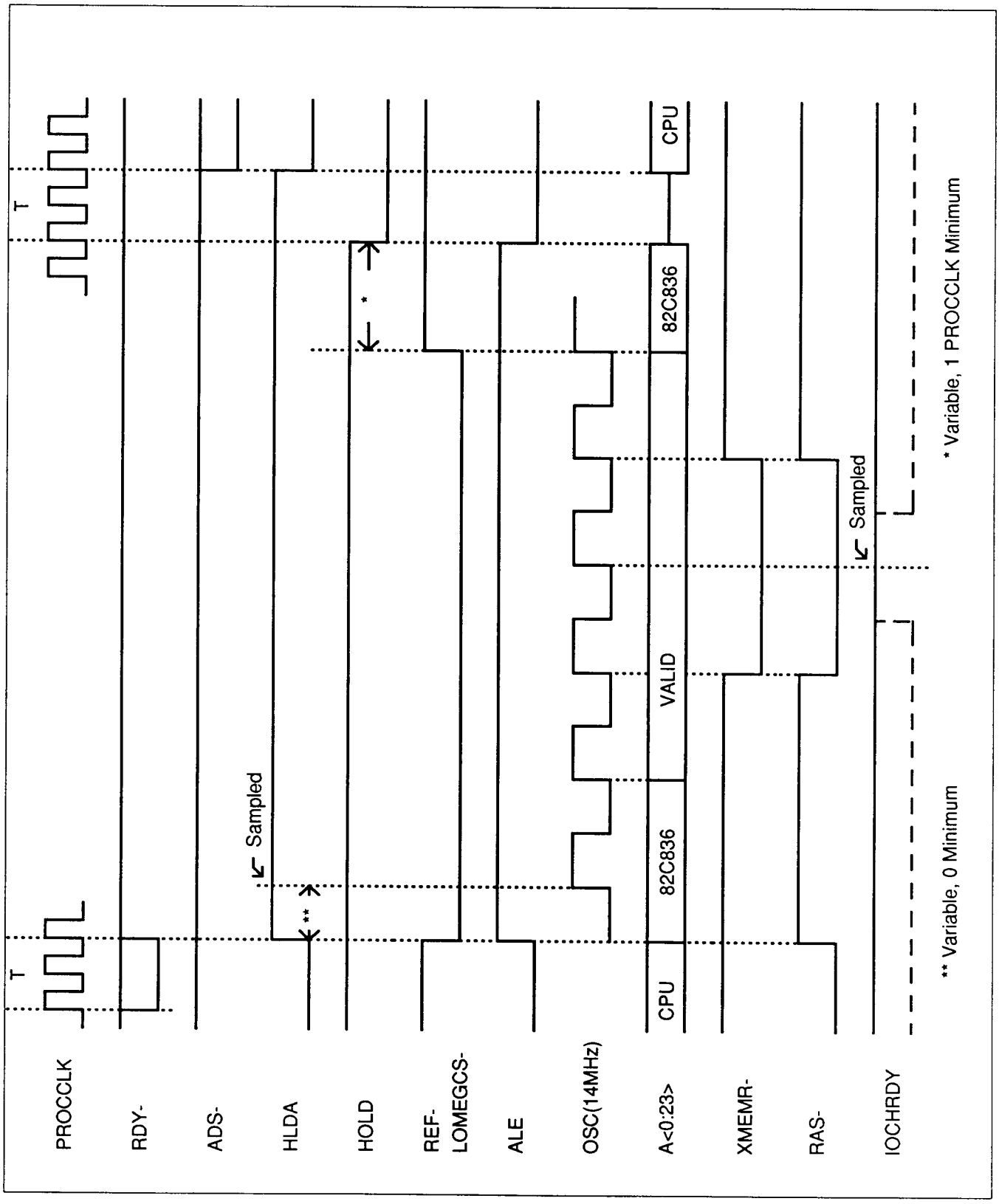
- If any RAS- signal was low, it is driven high when HLDA goes high.
- A0-23 flow through the 82C836 to determine the row address.
- The 82C836 then waits for XMEMR- or XMEMW- to be asserted.
- The memory command causes RAS- to go low immediately.
- Two rising edges of PROCCLK must then occur (A,B) before row/column changeover takes place.
- One PROCCLK after that, CAS- is asserted (C).
- CAS-, RAS- and column address remain valid until the end of command.
- If the cycle is a memory read, MWE- is driven high at the same time that RAS- is driven low.

Memory Refresh

Figure 48 shows RAS-only refresh. It applies to either system-initiated refresh or Master-initiated refresh while power is good (PWRGOOD high). Refreshing ceases during power down; laptop applications based on the 82C836 must allow for standby memory refresh external to the 82C836 to preserve DRAM data during power down. The basic 82C836 refresh protocol is as follows:

- For system initiated refresh, HLDA causes assertion of REFRESH. For Master initiated refresh, the add-on card bus master asserts REFRESH-. ALE follows HLDA and HOLD as in DMA cycles.
- LOMEGCS- follows REFRESH-
- Refresh address, XMEMR- and RAS- are synchronized to the 14.3 MHz clock.

Figure 48 *Refresh Timing*



- The command interval can be extended, one 14.3 MHz clock cycle at a time, by driving IOCHRDY low.

MWE- remains continuously low during refresh.

When using non-encoded RAS, RAS1- and RAS2- are delayed slightly relative to RAS0- and RAS3- during refresh. This "staggered refresh" is intended to reduce the net instantaneous DRAM power surge resulting from RAS- activation.

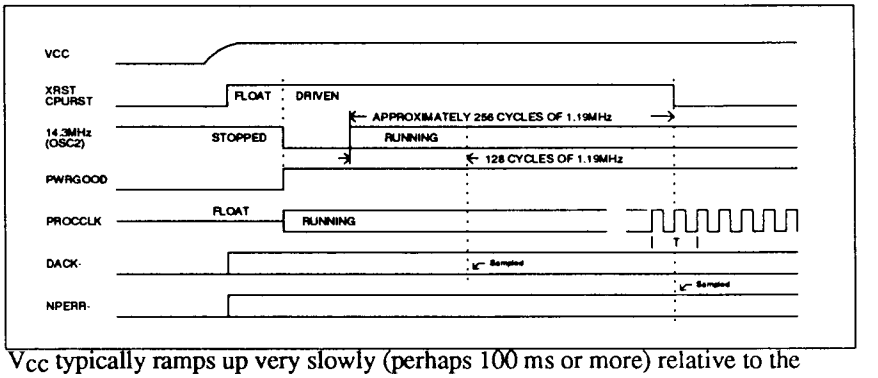
A0-9 contain the refresh address, incremented at the end of REF- (rising edge). A16-23 contain the Refresh Page Register value, normally zero (programmable). A10-15 are undefined.

As in DMA cycles, HOLD is synchronized to PROCCLK. The falling edge always occurs at the start of a T-state.

Power Turn-On and System Reset

Figure 49 shows what happens immediately following power turn-on. This sequence also applies to the cycling of PWRGOOD by means of a hardware reset switch.

Figure 49 *Power On*



V_{CC} typically ramps up very slowly (perhaps 100 ms or more) relative to the high speed of the 82C836 and CPU. At around 2 or 3 VDC, logic that wasn't already functioning on battery power begins operating, but PWRGOOD from the power supply remains low. PWRGOOD typically stays low until several hundred microseconds after V_{CC} reaches its valid operational value (4.75 VDC minimum).

While PWRGOOD is still low:

- All outputs except the 14.3 MHz driver are held in the high impedance state. In particular, XRST and CPURST are floating and should be externally pulled up. PROCCLK and BUSCLK also float.
- The 14.3 MHz driver is held at a solid high (Rev. 1) or low (Rev. 2 and beyond) level. It does not float.

The DACK- lines should be pulled up or down depending on the desired strap options. NPERR- should be pulled up, but will be driven low if a coprocessor is present.

Eventually PWRGOOD goes high. At this point, PROCCLK and BUSCLK start running and the 14.3 MHz oscillator is allowed to start running. There will typically be a significant start-up time for the 14.3 MHz oscillator, potentially 300 microseconds or longer.

An internal 1.19 MHz clock (14.3 MHz divided by 12) starts running when the 14.3 MHz clock starts up. The DACK- lines (strap options) are sampled and latched approximately 128 cycles of 1.19 MHz after the 14.3 oscillator starts running. After an additional 128 cycles of 1.19 MHz, NPERR- is sampled and XRST and CPURST go low. The sampling of NPERR- is for automatic coprocessor-present detection.

There is therefore an interval of around 200 microseconds during which all clocks are running and both resets are still high. Allowing BUSCLK and 14.3 MHz to run during reset has been found to be necessary for reliably resetting the 8042/8742 keyboard controller and certain add-on cards on the AT bus.

CPURST can occur alone, without XRST (see "CPU Access to AT Bus). The pulse width of CPURST in such cases is 22 cycles of PROCCLK, and the falling edge occurs at the middle of a T-state.

There is no mechanism for synchronizing the 82C836 internal phase clock to any external source; rather, all phase clocks outside the 82C836 must be synchronized to the 82C836 via CPURST or XRST.

Physical Characteristics

The absolute maximum ratings, operating conditions, and capacitive charactersistics that define the physical characteristics of the 82C836 chip are listed in Table 32.

Table 32 *Physical Characteristics of the 82C836*

Symbol	Parameter	Min.	Max.	Units	Test Conditions
Absolute Maximum Ratings*					
V_{CC}	Supply Voltage		5.5		
V_{IN}	Input Voltage	-0.5	5.5	V	
V_O	Output Voltage	-0.5	5.5	V	
T_{op}	Operating Temperature	-25	85	° C	
T_{stg}	Storage Temperature	-40	125	° C	
Operating Conditions					
V_{CC}	Supply Voltage	4.75	5.25	V	Operating
V_{CC}	Supply Voltage	3.0	5.25	V	Standby
T_A	Ambient Temperature	0	70	° C	
Capacitive Characteristics (TA = 25 C, VCC = 4.75 to 5.25 V)					
C_{IN}	Input Capacitance		10	pF	f_C = 1 MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins grounded
C_{OUT}	Output Capacitance		20	pF	Unmeasured pins grounded

*Permanent device damage may occur if Absolute Maximum Ratings are exceeded. Functional operation should be restricted to the conditions described under Operating conditions.

DC Characteristics

Table 33 describes the DC characteristics of the 82836 chip.

Table 33 *DC Characteristics of the 82C836*

Symbol	Parameter	Min.	Max.	Units	Conditions
DC Characteristics (Ta = 25 C, VCC = 4.75 to 5.25 V)					
V_{IL}	Input Low Voltage		0.8	V	
V_{IH}	Input High Voltage	2.0		V	
V_{OL}	Output Low Voltage		0.4	V	See Footnote*
V_{OH}	Output High Voltage	2.4		V	See Footnote*
V_{CL}	PROCCLK Output Low		0.4	V	I_{OL} = 4.0 mA
V_{CH}	PROCCLK Output High	3.9		V	I_{OH} = -4.0 mA
I_{IL}	Input Leakage Current	-10	20	µA	V_I = 0 to V_{CC}
I_{OL}	Output Leakage Current	-10	20	µA	V_O = 0.45V to V_{CC}
I_{CC}	V_{CC} Supply Current		100	mA	C_{XIN} = 40 MHz
I_{CCSB}	V_{CC} Standby Supply Current		50	µA	PWRGOOD low

*I_{OL} = 8mA and I_{OH} = -8mA for MWE-, XIOR-, XIOW-, XMEMR-, XMEMW-, ALE, REFRESH-, PROCCLK
I_{OL} = 4mA and I_{OH} = -4mA for A0-23, MODA0, MODA20, MA0-9, RAS0-3, CASH-, CASL-, PARH, PARL, BUSCLK, D0-15, XD0-15, DACKn-, IOCS16-, MEMCS16-, NA-, BHE-, READY-, XRST
IOL = 2mA and IOH = -2mA for all outputs not otherwise specified.

82C836 AC Characteristics

All timing parameters are specified under capacitive load of 50 pf and temperature of 70 degree C, and will allow operation at 16 MHz at 100 ns DRAM or 20 MHz at 80 ns DRAM without a "CAS-extended" wait state. Also, all the units discussed in the following timing tables are in nanoseconds unless otherwise specified. All AC specifications mentioned in this document are subject to change.

Notes on AC Timing Specifications

Examine these timing specifications to assist you in interpreting the following tables.

- Timing parameters are grouped into subsections according to the most common modes and cycle types to which they apply, but parameters are not necessarily limited only to those cases. Unless otherwise noted, all min/max timing limits are valid for all modes and cycle types in which the referenced signals follow the stated functional relationship.
- Terminology definitions:
  - "Command" refers to XIOR-, XIOW-, XMEMR- or XMEMW-
  - "Rise" refers to a low-to-high transition
  - "Fall" refers to a high-to-low transition
  - "Float" refers to a signal becoming high-impedance
  - "Turn-on" refers to a signal coming out of "float" and being driven high or low.
  - "Active" state means high for high-true signals, or low for low-true.
  - "Inactive" state means high for low-true signals, or low for high-true
- Unless otherwise noted, the memory timing limits listed above will support the following combinations of buffered or unbuffered signals:
  - 1 to 8 banks- no buffers needed on MD lines.

- 1 bank only - no buffers needed on RAS, CAS, WE, or MA lines.
- 2 banks only - no buffers needed on RAS or CAS.
- 3 or 4 banks - no buffers needed on RAS lines.
- 5 to 8 banks - buffers needed on CAS, WE and MA; 74F538 on RAS.

- To assure reliable local memory operation under extreme worst-case conditions, the DRAMS need to meet the specifications shown in Table 34. Most DRAMS utilizing CMOS technology can readily meet these specifications:

Table 34 *Dram Parameter Specifications*

DRAM PARAMETER	16	20	MHz
Read data valid from RAS active (t_{RAC})	100	80	ns max
Read data valid from CAS active (t_{CAC})	50	35	ns max
Read data valid from column address (t_{AA})	60	45	ns max
CAS active time (t_{CAS})	35	25	ns min
CAS precharge time in page mode (t_{CP})	15	15	ns min
CAS cycle time in page mode (t_{PC})	90	75	ns min
CAS fall to RAS rise (t_{RSH})	40	25	ns min

Table 35 describes the input requirements, CPU to local memory; Table 36 shows the corresponding input requirements.

Table 35 *Output Responses—CPU to Local Memory*

	Parameter	16 MHz		20 MHz	
		Min.	Max.	Min.	Max
t100	RAS active from PROCCLK rise		30		21
t101	RAS inactive from PROCCLK rise		33		32
t102	CAS active from PROCCLK rise		23		21
t103	CAS inactive from PROCCLK rise		28		24
t104	Row address valid from PROCCLK rise		40		32
t105	Row address hold from PROCCLK rise		35		17
t106	Row address valid from A0-23 in		50		31
t107	Column address valid from PROCCLK rise		38		30
t108	Column address hold from PROCCLK rise	3		3	
t109	Column address valid from A0-23 in		50		33
t110	MWE- fall from CAS- rise	10	60	12	60
t111	MWE- rise from PROCCLK rise		33		32
t112	PARL, PARH valid from D0-15 write data in		50		35
t113	READY- delay from PROCCLK rise	5	40	5	30

Table 36 *Input Requirements—CPU to Local Memory*

	Parameter	16 and 20 MHz	
		Min.	Max
t120	PARL, PARH setup before PROCCLK rise during read	10	
t121	PARL, PARH hold after PROCCLK rise during read	6	
t122	D0-15 setup before PROCCLK rise during read	10	
t123	D0-15 hold after PROCCLK rise during read	6	

CPU Access to AT-Bus, On-board I/O and ROM.

Table 37 shows the 82C836 Output Responses. The input requirements are shown in Table 38.

Table 37 *Output Responses—CPU to AT-Bus, On-board I/O and ROM*

	Parameter	16 MHz		20 MHz	
		Min.	Max.	Min.	Max
t130	LOMEGCS delay from PROCCLK rise		40		35
t131	NA- delay from PROCCLK rise		25		20
t132	READY- delay from PROCCLK rise	5	40	5	30
t133	ALE rise from BUSCLK low		20		15
t134	ALE fall from BUSCLK high		20		15
t135	Command & ROMCS- active from BUSCLK		20		15
t136	Command & ROMCS- inactive from BUSCLK rise		20		15
t137	MODA0 delay from PROCCLK rise		35		30
t138	MODA0 rise from BUSCLK fall during bus convert		25		20
t139	MODA20 delay from A20 (if MODA20 enabled		25		20
t140	MODA20 delay from HOLD fall		75		70
t141	SDIRL, H delay from PROCCLK rise		55		50
t142	XD0-15 delay from PROCCLK rise during write		55		50
t143	XD0-15 turn-on delay from PROCCLK rise.	5		5	
t144	XD0-15 turn-off delay from PROCCLK rise		55		50
t145	XD0-15 delay from D0-15 during write		45		40
t146	XD0-7 delay from D8-15 during write		45		40
t147	D0-15 delay from XD0-15 during read		45		40
t148	D0-15 delay from XD0-7 during read		45		40

Table 38 *Input Requirements—CPU to AT-Bus, On-board I/O and ROM*

	Parameter	16 MHz		20 MHz	
		Min.	Max.	Min.	Max
t160	ADS- setup before PROCCLK rise	24		15	
t161	ADS- hold after PROCCLK rise	6		6	
t162	A0-23 setup before PROCCLK rise	24		15	
t163	A0-23 hold after PROCCLK rise	4		4	
t164	BHE- setup before PROCCLK rise	24		15	
t165	BHE- hold after PROCCLK rise	4		4	
t166	DC-, W/R- and M/IO- setup before PROCCLK rise	24		15	
t167	DC-, W/R- and M/IO- hold after PROCCLK rise	4		4	
t168	IOCHRDY setup before BUSCLK rise*	20		15	
t169	IOCHRDY hold after BUSCLK rise*	20		15	
t170	0WS setup before BUSCLK fall*	15		10	
t171	0WS hold after BUSCLK fall*)	25		20	
t172	IOCS16- setup before Command active	20		15	
t173	IOCS16- hold after Command inactive	0		0	
t174	MEMCS16- setup before ALE fall	15		10	
t175	MEMCS16- hold after ALE fall	1025		10	
t176	XD0-15 valid before read command rise			20	

*Certain input parameters, as noted, are "non-restrictive." Violating these parameters causes no adverse effect except that the signal may not be recognized until the next subsequent clocking period. The parameterspecifies only the condition needed to guarantee recognition of the signal on a particular clock edge.

DMA Access to AT-Bus, On-board I/O and ROM

Table 39 shows the 82C836 Output Responses. The input requirements are shown in Table 40.

Table 39 *Output Responses—DMA to AT-Bus, On-board I/O and ROM*

	Parameter	16 MHz		20 MHz	
		Min.	Max.	Min.	Max
t190	DACK- delay from BUSCLK fall	5		5	
t191	ALE rise from HLDA rise		75		70
t192	DMA Address delay from BUSCLK rise. Note: DMA address refers to A0-23, MODA0, MODA20, and BHE-		40		35
t194	Command fall from BUSCLK rise		45		40
t195	Command rise from BUSCLK rise		45		40
t197	TC rise from BUSCLK rise		35		30
t198	TC fall from BUSCLK rise		40		35
t199	LOMEGCS delay from A0-23		30		25

Table 40 *Input Requirements—DMA to AT-Bus, On-board I/O and ROM*

	Parameter	16 MHz.		20 MHz	
		Min.	Max.	Min.	Max
t210	HLDA setup before BUSCLK rise*	20		15	
t212	DREQ setup before BUSCLK fall*	20		15	
t213	DREQ hold after BUSCLK fall*	20		15	
t214	IOCHRDY setup before BUSCLK fall*	15		10	
t215	IOCHRDY hold after BUSCLK fall*	7		5	

*Certain input parameters, as noted, are "non-restrictive." Violating these parameters causes no adverse effect except that the signal may not be recognized until the next subsequent clocking period. The parameterspecifies only the condition needed to guarantee recognition of the signal on a particular clock edge.

DMA & AT-Bus Master Access to Local Memory

Table 41 shows the 82C836 Output Responses. The Input Requirements are shown in Table 42.

Table 41 *Output Responses—DMA & AT-Bus Master Access to Local Memory*

	Parameter	16 MHz		20 MHz	
		Min.	Max.	Min.	Max
Note: For XD, D, and PAR timing, use t145-t148 and t112. For row/column changeover and CAS active, use t102, t105 and t107.					
t230	RAS- high from HLDA rise		60		50
t231	A0-23 and BHE- float from MASTER- active		70		60
t232	Command float from MASTER- active		35		30
t233	MWE- rise from XMEMR- fall		45		40
t234	MWE- fall from CAS- rise		60	12	60
t235	RAS- inactive from XMEMR- or XMEMW- high		50		40
t236	CAS- inactive from XMEMR- or XMEMW- high		55		50
t239	RAS- active from XMEMR or XMEMW		48		42
t240	A20 valid from MODA20 during MASTER access		26		20
t241	A0 valid from MODA0 during MASTER access		35		30

Table 42 *Input Requirements—DMA & AT-Bus Master Access to Local Memory*

	Parameter	16 & 20 MHz	
		Min.	Max
t250	PARL, PARH setup before XMEMR- rise during mem read	20	
t251	PARL, PARH hold after XMEMR- rise during mem read	0	

Refresh

Table 43 shows the 82C836 Output Responses. The Input Requirements are shown in Table 44.

Table 43 *Refresh—Output Responses*

	Parameter	16 MHz		20 MHz	
		Min.	Max.	Min.	Max
t260	REFRESH- active from HLDA		30		25
t261	REFRESH- float from OSC2 rise		50		45
t262	Refresh address valid from REFRESH- active. NOTE: Refresh address refers to MODA0, A0-9 and MA0-9		30		25
t263	XMEMR- active from OSC2 rise		60		55
t264	XMEMR- inactive from OSC2 rise		55		50
t265	RAS0-, RAS3- active from XMEMR- fall		35		30
t266	RAS0-, RAS3- inactive from XMEMR- rise		30		25
t267	RAS1-, RAS2- active from XMEMR- fall		75		70
t268	RAS1-, RAS2- inactive from XMEMR- rise		75		70
t269	LOMEGCS delay from REFRESH-		30		25

Table 44 *Refresh—Input Requirements*

	Parameter	16 MHz		20 MHz	
		Min.	Max.	Min.	Max
t280	IOCHRDY setup before OSC2 rise*	20		15	
t281	IOCHRDY hold after OSC2 rise*	7		5	
t282	REFRESH- setup before OSC2 rise*	25		20	

*Certain input parameters, as noted, are "non-restrictive." Violating these parameters causes no adverse effect except that the signal may not be recognized until the next subsequent clocking period. The parameterspecifies only the condition needed to guarantee recognition of the signal on a particular clock edge.

Miscellaneous Parameters

Table 45 shows the 82C836 Output Responses. The Input Requirements are shown in Table 46.

Table 45 *Miscellaneous Parameters—Output Responses*

	Parameter	16 MHz		20 MHz	
		Min.	Max.	Min.	Max
t290	XRST inactive from PROCCLK rise		18		13
t291	CPURST inactive from PROCCLK rise		18		13
t292	BUSCLK delay from PROCCLK (all ratios)		25		20
t293	BUSY- delay from NPBUSY-		25		20
t294	BUSY- inactive from XIOW- (output to port F0H)		35		30
t295	8042CS-, RTTCS- delay from PROCCLK rise		75		70
t296	PROCCLK rise from CXIN fall		25		20
t297	PROCCLK fall from CXIN rise		25		20

Table 46 *Miscellaneous Parameters—Input Requirements*

	Parameter	16 MHz		20 MHz	
		Min.	Max.	Min.	Max
t310	CXIN cycle time	31		25	
t311	CXIN low time	12		10	
t312	CXIN high time	12		10	
t313	NPBUSY- active pulse width	30		30	
t314	NPERR- fall after BUSY- fall	0		0	
t315	NPERR- fall before NPBUSY- rise	30		25	
t316	NPERR- active pulse width	35		30	
t317	READY- setup before PROCCLK rise	12		10	
t318	READY- hold after PROCCLK rise	3		2	

Local Bus Access and Cache

Table 47 shows the 82C836 Output Responses. The Input Requirements are shown in Table 48.

Table 47 *Local Bus Access and Cache—Output Responses*

	Parameter	16 MHz		20 MHz	
		Max.	Min.	Min.	Max
t340	STCYC- active from PROCCLK rise		25		20
t341	STCYC- inactive from PROCCLK rise		25		20
t342	CAS- inactive from READY- rise	3		3	

Table 48 *Local Bus Access and Cache—Input Requirements*

	Parameter	16 MHz		20 MHz	
		Max.	Min.	Min.	Max
t350	LBA- setup before PROCCLK rise	12		10	
t351	LBA- hold after PROCCLK rise	3		2	
Note: t317 and t318 apply to LBA/Cache mode as well as to coprocessor accesses					

Timing Diagrams

Figure 50 shows the timing waveforms. The load circuit is shown in Figure 51.

Figure 50 *Timing Waveforms*

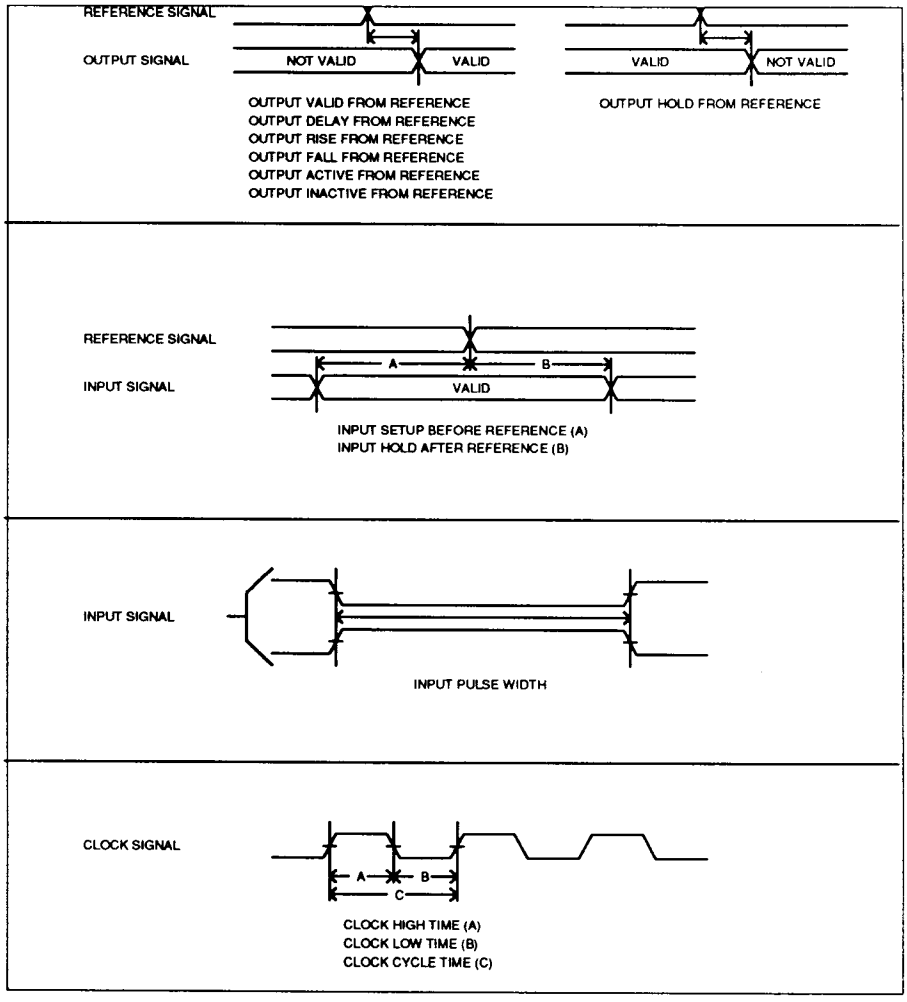


Figure 51 *Load Circuit*

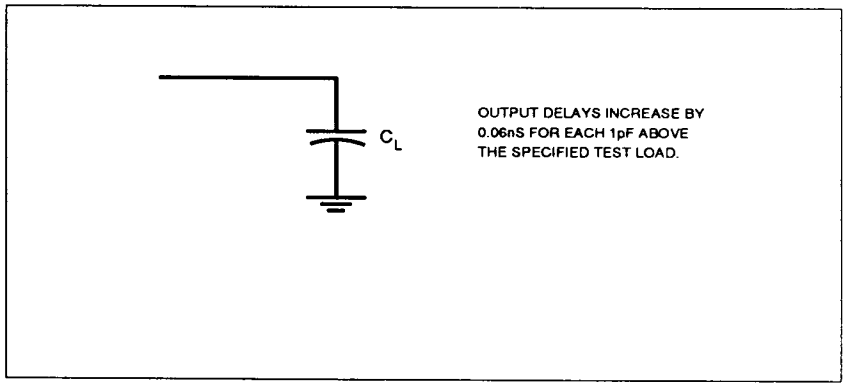


Figure 52 *160-PIN PFP Pinout*

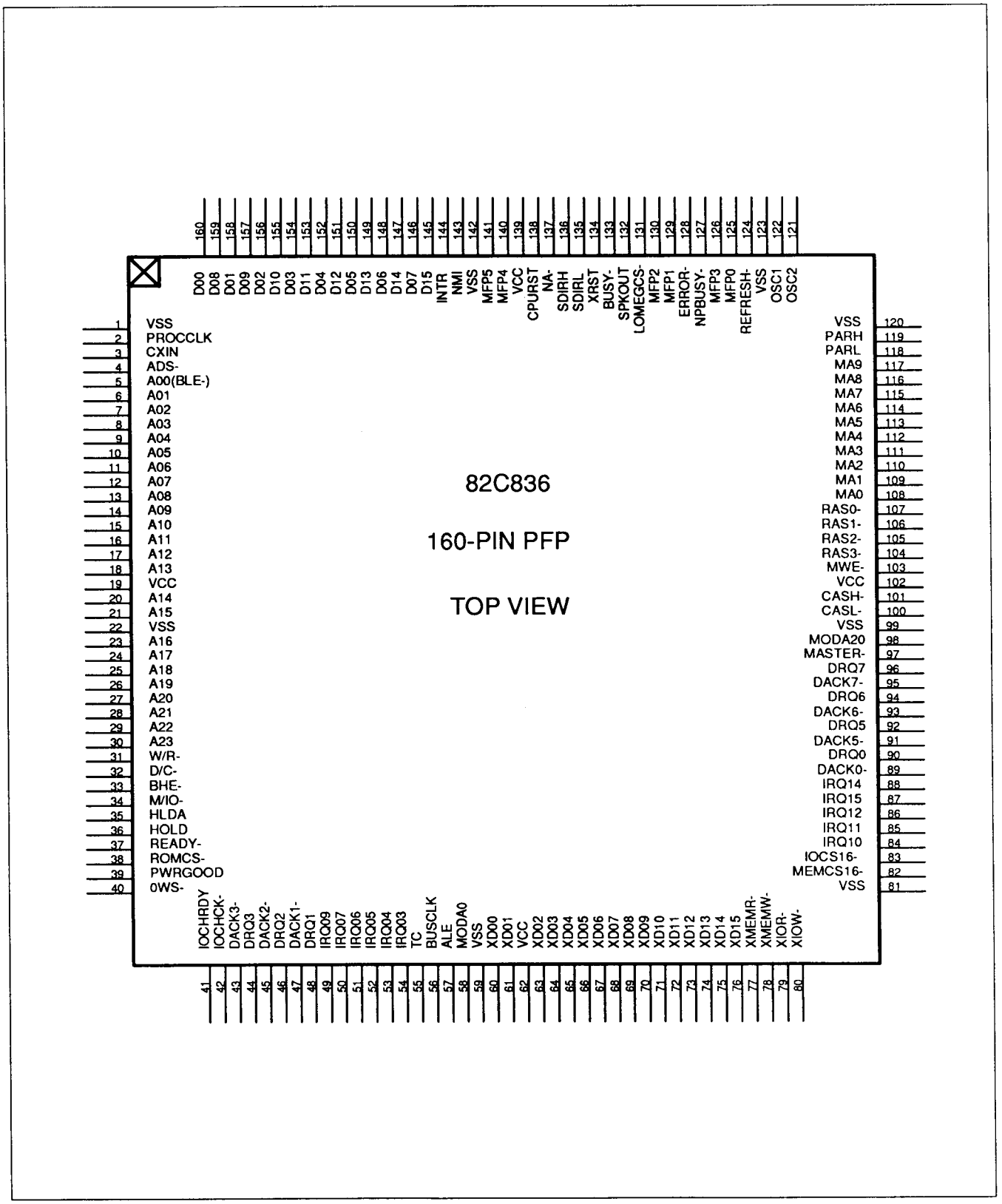
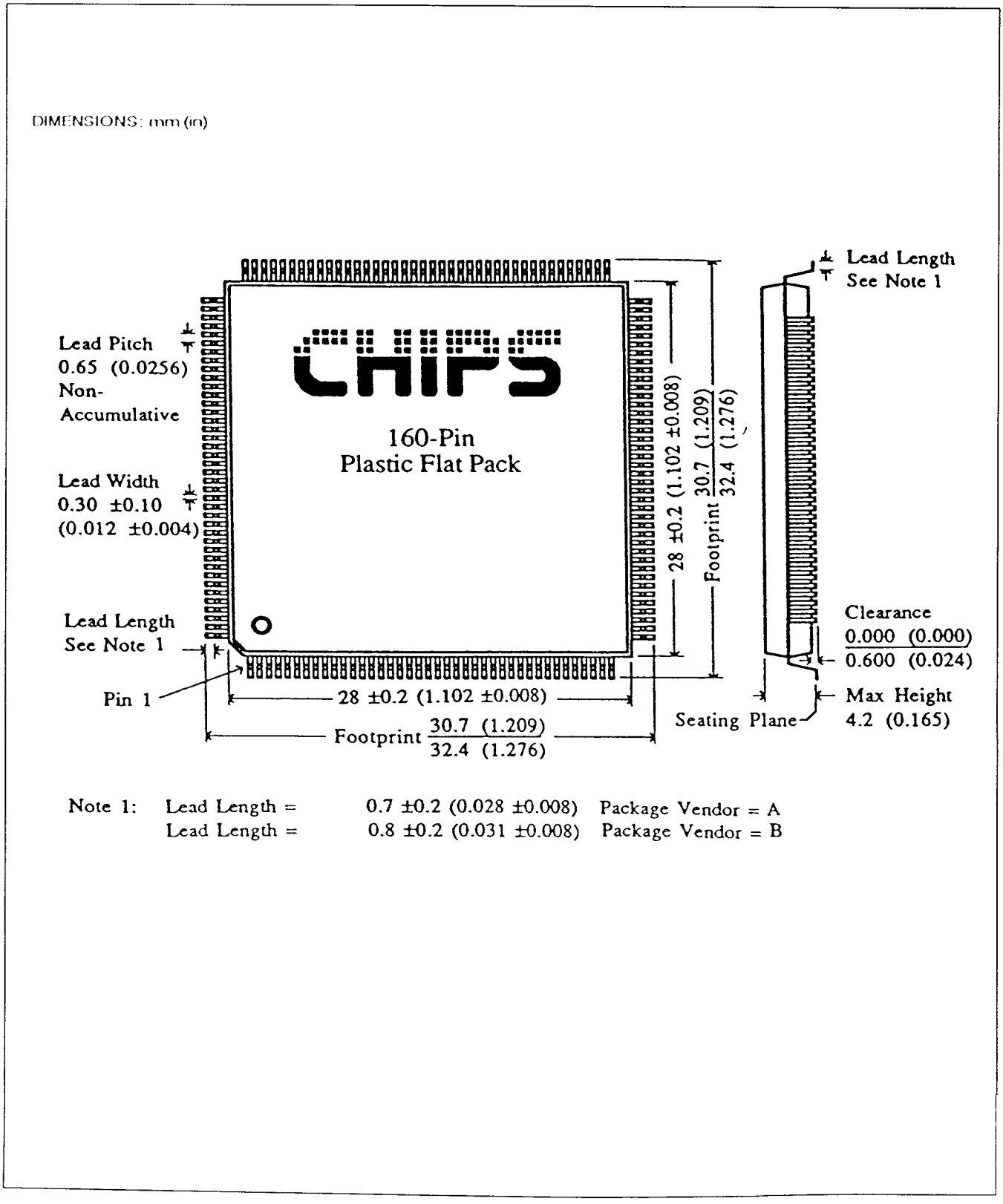


Figure 53 *Physical (mechanical) Dimensions*



Scat/sx Schematic Diagrams