82C836
CHIPSet

Single-Chip 386sx AT

Data Sheet

March 1993

P R E L I M I N A R Y

Trademark Acknowledgment

CHIPS® and CHIPS logotype are registered trademarks of Chips and Technologies, Inc.

NEAT™, NEATsx™, SCAT™, and SCATsx™ are trademarks of Chips and Technologies, Inc.

IBM®, IBM PC®, IBM PC/AT®, PS2™, and Microchannel™ are trademarks of International Business Machines Corporation.

Intel®, 386™, 387™, are trademarks of Intel Corporation.

Lotus® is a registered trademark of Lotus Development Corporation.

MS-DOS® is a registered trademark of Microsoft Corporation.

Motorola® is a registered trademark of Motorola.

Chips and Technologies, Inc.
2950 Zanker Road
San Jose, California 95134
Phone: 408-434-0600

Contents

Section 1

Section 2

Section 3

Section 4

Section 5

Section 6

Section 7

Section 8

Section 9

Section 10

Section 11

Section 12

Section 13

Section 14

Appendix A

List of Figures

List of Tables

Section 1

82C836 CHIPSet Introduction

The 82C836 (also known as SCATsx) is a VLSI device incorporating the motherboard logic required to build a low-cost, highly-integrated, IBM PC/AT-compatible computer. It is designed to be used in conjunction with other Chips and Technologies controllers such as the 82C45X VGA 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 significantly reducing motherboard size, component count, and the need for many I/O channel slots.

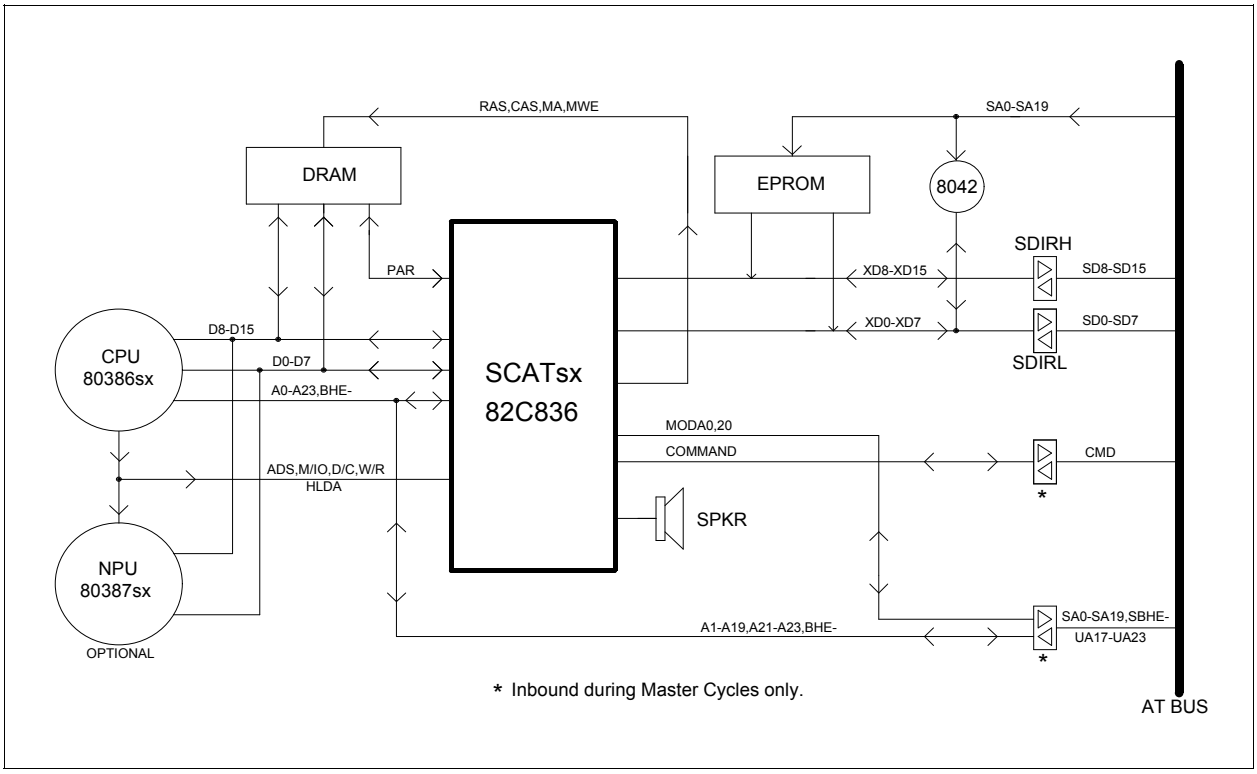
Features

The 82C836 provides the following features:

- 80386sx control logic and clocks to support CPU speeds of up to 25MHz 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 refresh controller
- Power management features
- A DRAM controller that supports up to 16 MB of DRAM using 256kB, 1MB or 4MB DRAMs
- A memory controller that provides shadow RAM and support of either 8-bit or 16-bit BIOS ROM
- Support for fast local cache RAM via external cache controller
- 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 GATEA20 and fast CPU reset logic
- Compact packaging in a single 160-pin plastic flat pack (160 PFP).

The SCATsx basic system architecture is shown in Figure 1-1.

Figure 1-1. SCATsx Basic System Architecture



Architectural Overview

The major address and data buses are described in Table 1-1. 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 1-1. Address and Data Buses

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. It should be connected to the D-bus though series resistors to minimize undershoot and overshoot.
XD15-XD0	X-Bus	A 16-bit bidirectional bus for data transfer to or from on-board peripherals
SD15-SD0	AT Bus Data	The main 16-bit bidirectional bus for transferring data to or from add-on cards.
A23-A0	CPU Addr	A 24-bit 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 unlatch address bus providing the high-order address bits for memory resources on the AT bus.
MA10-MA0	Row/Col Addr	A multiplexed address bus driven by the 82C836 for DRAM row and column addressing.

Typical on-board I/O resources external to the 82C836 include: keyboard controller, optional numeric coprocessor, and optional Real Time Clock (RTC). 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 and 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-address 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 following summarizes all possible cycles 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 memory read and I/O write)
  - I/O to memory (simultaneous I/O read and memory write)
- Master
  - Memory read or write (may access on-board DRAM and 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 1MB or not
  - Delay or speed-up via IOCHRDY or 0WS

The major signal groups in a SCATsx AT-compatible architecture are as follows:

- AT bus interface signals, 8-bit section
  - XD7:0, SDIRL (SD0-7)
  - A19:0, 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
  - XD15:8, SDIRH (SD8:15)
  - A23:17, 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

Note: Signals listed 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 1-2 lists the key signals that signify bus ownership. Additional information on signal functions and timing relationships is found in Section 11, System Timing Relationships and Section 12, System Characteristics, subsection titled AC Characteristics.

Table 1-2. 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

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

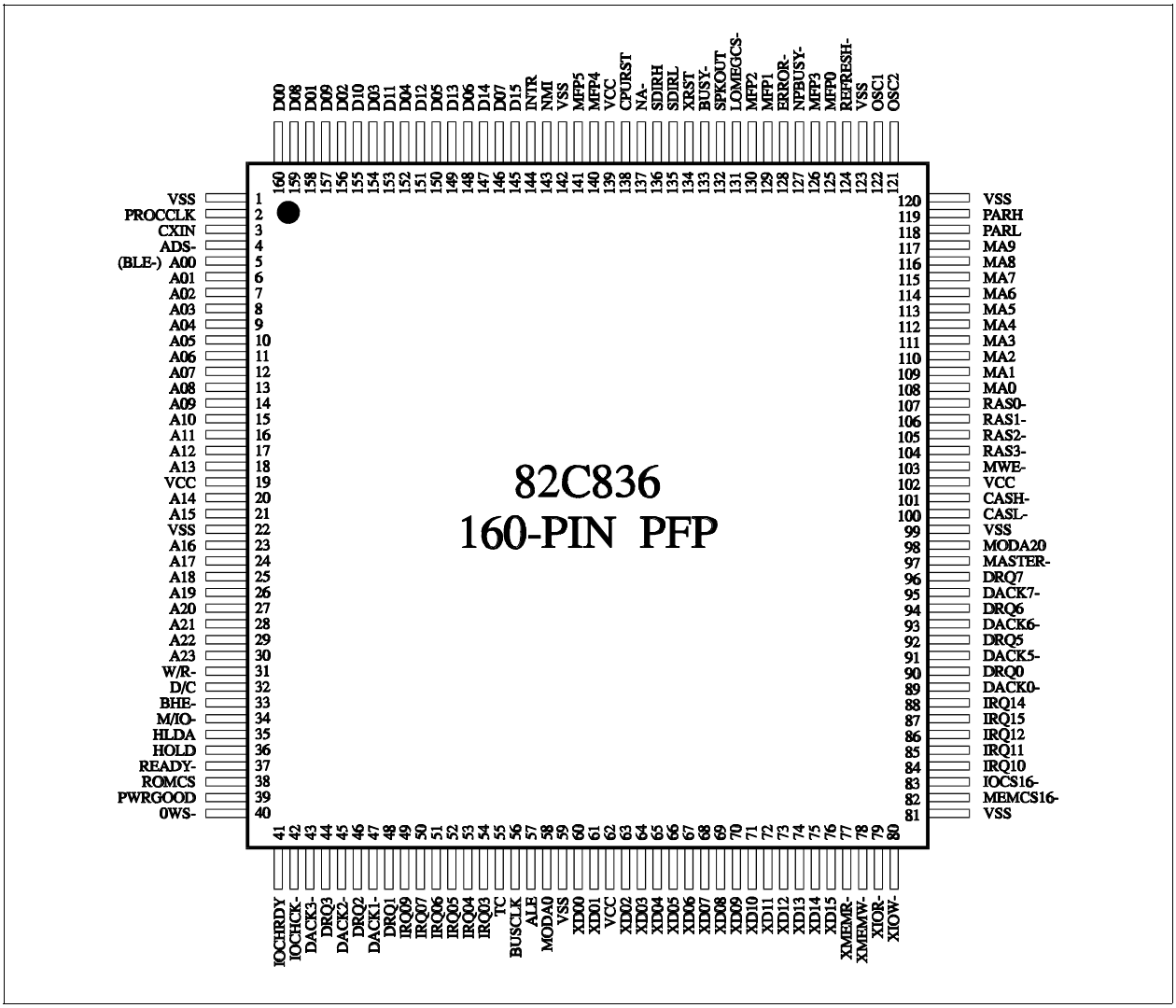
Section 2

Pin Assignments

Pin Assignments

The 82C836 CHIPSet single-chip 386sx AT is packaged in a 160-pin plastic flat pack package. Figure 2-1. shows the top view of the chip layout.

Figure 2-1. 160-Pin PFP Pinout (Top View)



Signal Descriptions

A signal name prefixed by a hyphen, such as -RAS0, specifies an active-low signal. Tables 2-1 through 2-6 describes the functions and characteristics of the signals on each pin or group of pins for the 82C836 chipset.

Table 2-1. Clock Input and Output Signals

Pin	Type	Name	Description
122	Input	OSC1	14.318MHz 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	Output	OSC2	14.318MHz crystal oscillator output from the internal crystal oscillator. If an external oscillator module is used, leave OSC2 unconnected.
3	Input	CXIN	CPU clock oscillator input from an external oscillator. Serves as the source for PROCCLK.
2	Output	PROCCLK	Processor Clock output to the 80386sx processor. The frequency of PROCCLK is twice the processor's internal clock frequency.
56	Output	BUSCLK	Bus Clock output to the 8042 and AT bus.

Table 2-2. Local Bus Interface Signals

Pin	Type	Name	Description
5-18	Bidirectional	A<0:13>	—
20-21	Bidirectional	A<14:15>	—
23-30	Bidirectional	A<16:23>	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	Bidirectional	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	Bidirectional	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	Bidirectional	-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. BHE A0 Function 0 0 Word transfer 0 1 Odd byte transfer 1 0 Even byte transfer 1 1 Reserved
160	Bidirectional	D00	—
158	Bidirectional	D01	—
156	Bidirectional	D02	—
154	Bidirectional	D03	—
152	Bidirectional	D04	—
150	Bidirectional	D05	—
148	Bidirectional	D06	—
146	Bidirectional	D07	—
159	Bidirectional	D08	—
157	Bidirectional	D09	—
155	Bidirectional	D10	—
153	Bidirectional	D11	—
151	Bidirectional	D12	—
149	Bidirectional	D13	—
147	Bidirectional	D14	—
145	Bidirectional	D15	CPU data bus, bits 0 to 15.

Table 2-2. Local Bus Interface Signals (continued)

Pin	Type	Name	Description
4	Input	-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	Input	W/-R	Processor Write/Read status. Indicates whether the CPU cycle is a write or read.
32	Input	D/-C	Processor Data/Control status. Indicates whether the CPU cycle is a data access or code/control cycle.
34	Input	M/-IO	Memory or I/O select from CPU. Indicates whether the CPU cycle is a memory access or an I/O.
37	Bidirectional	-READY	Active-low signal 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	Output	-NA (-STCYC)	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	Output	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	Input	HLDA	Hold Acknowledge is an active-high input from the processor indicating when the CPU local bus has been given up by the processor. HLDA should be connected to the 80386sx processor's HLDA pin.
143	Output	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.
144	Output	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.

Table 2-3. Numeric Coprocessor Interface Signals

Pin	Type	Name	Description
133	Output	-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	Input	-NPBUSY	Busy status from 80387sx. This signal indicates the 80387sx is currently executing a command.
128	Input	-NPERR	Error signal from the coprocessor.

Table 2-4. Memory Interface Signals

Pin	Type	Name	Description
108-117	Output	MA0-MA9	Multiplexed DRAM address bits MA 0 to 9 are outputs to the DRAMs
107-104	Output	-RAS<0:3>, (MA10)	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 an even or odd CAS signal. -RAS<0:3>, perform different functions in an encoded RAS mode. When using a 4MB DRAM configuration, -RAS3 becomes MA10. For further details, refer to Section 5, System Interface, subsection titled DRAM Interface.
101	Output	-CASH	Column Address Strobe High is an active-low output to all high (odd) byte DRAMs.
100	Output	-CASL	Column Address Strobe Low is an active-low output to all low (even) byte DRAMs.
119	Bidirectional	PARH	Parity High is the parity bit from the high-order bytes of the DRAMs.
118	Bidirectional	PARL	Parity Low is the parity bit from the low-order bytes of the DRAMs.
103	Output	-MWE	Memory Write Enable is an active-low output connected to all DRAMs. -MWE is normally low, but is high for read cycle. 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	Output	-ROMCS	ROM Chip Select is an active-low output to the EPROM(s). -ROMCS becomes active for the progammed address range.

Table 2-5. I/O Channel Interface Signals

Pin	Type	Name	Description
60-61	Bidirectional	XD0<0:1>	—
63-76	Bidirectional	XD0<2:15>	16-bit system data bus for on-board I/O, on-board ROM and (via buffers) AT bus data.
136	Output	SDIRH	—
135	Output	SDIRL	Channel data bus controls are outputs that control the direction of the data buffer between the XD-bus and the SD-bus. When the signals are high (default), data flows from XD to SD; when the signals are low, data flows XD from SD.
97	Input	-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	Bidirectional	-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	Output	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	Bidirectional	-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 perpherals and the 82C836.
80	Bidirectional	-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 on-board peripherals as well as the I/O channel. During Master cycles, -XIOW is an input command to the peripherals and the 82C836.
77	Bidirectional	-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	Bidirectional	-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 memory refresh and DMA cycles, -XMEMW is always an output.
131	Output	-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.

Table 2-5. I/O Channel Interface Signals (continued)

Pin	Type	Name	Description
41	Input	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 to extend the cycle time. This input should be driven by an open collector driver.
40	Input	-0WS (-LBA)	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	Bidirectional	-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 accessed resource can support 16-bit data transfers. -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.
82	Bidirectional	-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 can support 16-bit data transfers. -MEMCS16 is an output for CPU or Master accesses to on-board DRAM or to 16-bit on-board ROM.
42	Input	-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	Input	IRQ0<3:7>	—
49	Input	IRQ09	—
84-86	Input	IRQ<10:12>	—
88-87	Input	IRQ<14:15>	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	Input	DRQ0/DSELB	—
48	Input	DRQ1/DSELA	—
46	Input	DRQ2/DRQA	—
44	Input	DRQ3/DRQB	—
92	Input	DRQ5/-CAS3H	—
94	Input	DRQ6/-CAS2H	—

Table 2-5. I/O Channel Interface Signals (continued)

Pin	Type	Name	Description
96	Input	DRQ7/-CAS1H	DMA Request 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. In MRA mode, the functions of these pins change as indicated.
89	Output	-DACK0/-DACKEN	—
47	Output	-DACK1/DACKA	—
45	Output	-DACK2/DACKB	—
43	Output	-DACK3/DACKC	—
91	Output	-DACK5/-CAS3L	—
93	Output	-DACK6/-CAS2L	—
95	Output	-DACK7/-CAS1L	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. In MRA mode, the functions of these pins change as indicated.
55	Output	TC	Terminal Count is an active-high output pulse to the I/O channel that indicates the end of a DMA transfer.

Table 2-6. Miscellaneous Signals

Pin	Type	Name	Description
39	Input	PWRGOOD	Power Good is an active-high input from the power supply. When this signal is high, it indicates all power supply voltages have reached their working levels. CPURST (pin 138) and XRST (pin 134) remain high for at least 200µs after PWRGOOD goes high.
138	Output	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. CPURST without XRST is often used to return the CPU to real mode from protected mode.
134	Output	XRST	Peripheral Reset is an active-high output that resets external peripherals and the coprocessor (if present) during power-up.
132	Output	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 an external driver circuit, a low pass filter, and then to the speaker.

Table 2-6. Miscellaneous Signals (continued)

Pin	Type	Name	Description
141	Bidirectional	MFP5	Multi Function Pin 5 is a bidirectional pin that has two functions (see -DACK5 in "Reset Strap Options"): 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.
140	Input	MFP4	Multi Function Pin 4 is an input that has two functions (see -DACK5 in "Reset Strap Options"). In internal RTC mode, MFP4 is the 32KHz Oscillator Input, which should be connected to a square wave source with a frequency of 32.768KHz. 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	Input	MFP3	Multi Function Pin 3 is an input pin 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	Output	MFP2	Multi Function Pin 2 is an output pin 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 verison of the 82C836, but only the -8042CS function is available at present.
129	Input	MFP1	Multi Function Pin 1 is an input pin 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 GATEA20, 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	Input	MFP0	Mutli Function Pin 0 is an input pin acting 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.

Signal Pinouts

The signal pinouts for the 82C836 CHIPSet are summarized in Tables 2-7 and 2-8.

Table 2-7. Alphabetical Pin Definitions

Signal Name	Pin	Signal Name	Pin	Signal Name	Pin	Signal Name	Pin
-0WS/-LBA	40	D06	148	IRQ12	86	-REFRESH	124
A00	5	D07	146	IRQ14	88	-ROMCS	38
A01	6	D08	159	IRQ15	87	SDIRH	136
A02	7	D09	157	-LOMEGCS	131	SDIRL	135
A03	8	D10	155	MA0	108	SPKOUT	132
A04	9	D11	153	MA1	MA1	109	TC
A05	10	D12	151	MA2	110	Vcc	19
A06	11	D13	149	MA3	111	Vcc	62
A07	12	D14	147	MA4	112	Vcc	102
A08	13	D15	145	MA5	113	Vcc	139
A09	14	-DACK0	89	MA6	114	Vss	1
A10	15	-DACK1	47	MA7	115	Vss	22
A11	16	-DACK2	45	MA8	116	Vss	59
A12	17	-DACK3	43	MA9	117	Vss	81
A13	18	-DACK5	91	-MASTER	97	Vss	99
A14	20	-DACK6	93	-MEMCS16	82	Vss	120
A15	21	-DACK7	95	MFP0	125	Vss	123
A16	23	D/-C	32	MFP1	129	Vss	142
A17	24	DRQ0	90	MFP2	130	W/-R	31
A18	25	DRQ1	48	MFP3	126	XD00	60
A19	26	DRQ2	46	MFP4	140	XD01	61
A20	27	DRQ3	44	MFP5	141	XD02	63
A21	28	DRQ5	92	M/IO	34	XD03	64
A22	29	DRQ6	94	MODA0	58	XD04	65
A23	30	DRQ7	96	MODA20	98	XD05	66
-ADS	4	-ERROR	128	-MWE	103	XD06	67
ALE	57	HLDA	35	-NA/-STCYC	137	XD07	68
-BHE	33	HOLD	36	NMI	143	XD08	69
BUSCLK	56	INTR	144	-NPBZ	127	XD09	70
-BUSY	133	-IOCHCK	42	OSC1	122	XD10	71
-CASH	101	IOCHRDY	41	OSC2	121	XD11	72
-CASL	100	-IOCS16	83	PARH	119	XD12	73
CPURST	138	IRQ03	54	PARL	118	XD13	74
CXIN	3	IRQ04	53	PROCCLK	2	XD14	75
D00	160	IRQ05	52	PWRGOOD	39	XD15	76
D01	158	IRQ06	51	-RAS0	107	-XIOR	79
D02	156	IRQ07	50	-RAS1	106	-XIOW	80
D03	154	IRQ09	49	-RAS2	105	-XMEMR	77
D04	152	IRQ10	84	-RAS3/MA10	104	-XMEMW	78
D05	150	IRQ11	85	-READY	37	XRST	134

Table 2-8. Numerical Pin Definitions

Pin	Signal Name	Pin	Signal Name	Pin	Signal Name	Pin	Signal Name
1	Vss	41	IOCHRDY	81	Vss	121	OSC2
2	PROCCLK	42	-IOCHCK	82	-MEMCS16	122	OSC1
3	CXIN	43	-DACK3	83	-IOCS16	123	Vss
4	-ADS	44	DRQ3	84	IRQ10	124	-REFRESH
5	A00	45	-DACK2	85	IRQ11	125	MFP0
6	A01	46	DRQ2	86	IRQ12	126	MFP3
7	A02	47	-DACK1	87	IRQ15	127	-NPBZ
8	A03	48	DRQ1	88	IRQ14	128	-ERROR
9	A04	49	IRQ09	89	-DACK0	129	MFP1
10	A05	50	IRQ07	90	DRQ0	130	MFP2
11	A06	51	IRQ06	91	-DACK5	131	-LOMEGCS
12	A07	52	IRQ05	92	DRQ5	132	SPKOUT
13	A08	53	IRQ04	93	-DACK6	133	-BUSY
14	A09	54	IRQ03	94	DRQ6	134	XRST
15	A10	55	TC	95	-DACK7	135	SDIRL
16	A11	56	BUSCLK	96	DRQ7	136	SDIRH
17	A12	57	ALE	97	-MASTER	137	-NA
18	A13	58	MODA0	98	MODA20	138	CPURST
19	Vcc	59	Vss	99	Vss	139	Vcc
20	A14	60	XD00	100	-CASL	140	MFP4
21	A15	61	XD01	101	-CASH	141	MFP5
22	Vss	62	Vcc	102	Vcc	142	Vss
23	A16	63	XD02	103	-MWE	143	NMI
24	A17	64	XD02	104	-RAS3	144	INTR
25	A18	65	XD04	105	-RAS2	145	D15
26	A19	66	XD05	106	-RAS1	146	D07
27	A20	67	XD06	107	-RAS0	147	D14
28	A21	68	XD07	108	MA0	148	D06
29	A22	69	XD08	109	MA1	149	D13
30	A23	70	XD09	90	DRQ0	150	D05
31	W/-R	71	XD10	91	-DACK5	151	D12
32	D/-C	72	XD11	92	DRQ5	152	D04
33	-BHE	73	XD12	93	-DACK6	153	D11
34	M/-IO	74	XD13	94	DRQ6	154	D03
35	HLDA	75	XD14	95	-DACK7	155	D10
36	HOLD	76	XD15	96	DRQ7	156	D02
37	-READY	77	-XMEMR	97	-MASTER	157	D09
38	-ROMCS	78	-XMEMW	98	MODA20	158	D01
39	PWRGOOD	79	-XIOR	99	Vss	159	D08
40	-0WS	80	-XIOW	100	-CASL	160	D00

Section 3

Functional Description

Configuration Registers

The 82C836 contains an extensive and versatile set of internal configuration registers (ICRs) 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.

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 or high for strap option sampling. Table 3-1 describes the pins used for strap options. Figure 3-1 shows an alternate way to implement a pull-low function in cases where a simple pull-down resistor is not suitable because of signal loading.

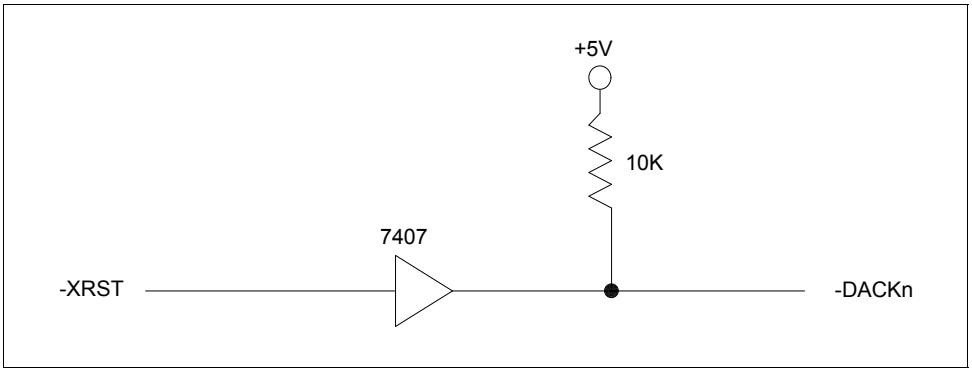
Note: The circuit shown in Figure 3-1 should not be used in systems using Standby Refresh in MRA mode.

Table 3-1. Pins for Strap Options

Signal	Strap Function	Description
-DACK0	SENSE0	Determines ICR 45H bit 0
-DACK1	SENSE1	Determines ICR 45 bit 1
-DACK2	VIDEO	Video on SD ↑ or XD bus ↓; also determines ICR 45H bit 2
-DACK3	NA/STCYC	-NA↑ or -STCYC ↓ Function
-DACK5	-EXRTC	Internal RTC ↑ or external ↓
-DACK6	Reserved	(Should be pulled up)
-DACK7	-ROM16	8-bit ROM ↑ or 16-bit ROM ↓

↑ = Pull-up ↓ = Pull-down

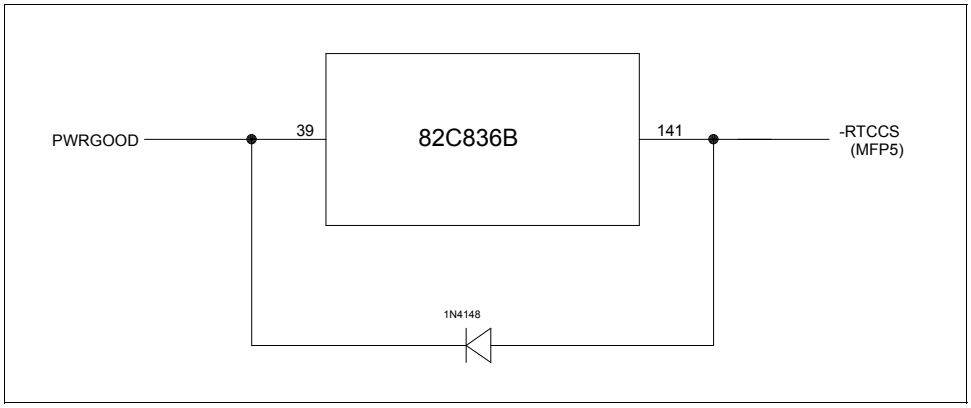
Figure 3-1. Pull-Low Function—Alternate Implementation



SENSE0 and SENSE1 are general purpose and can be used according to specific system requirements. -DACK2 can also be used as a general purpose SENSE2 if ICR44H is appropriately programmed to specify where the video controller resides.

If external Real Time Clock is used (-DACK5 pulled low), the -RTCCS pin (MFP5) should be connected to PWRGOOD through a diode as shown in Figure 3-2. This insures that the standby enable bit (ICR 60H) will always be cleared on power-up. Otherwise, the 82C836 could power-up with standby mode enabled, which would prevent -DACK5 from being sampled properly. See the ICR 60H description in Section 4, Configuration Registers for further explanation of standby initialization.

Figure 3-2. PS Diode for External Real Time Clock Systems

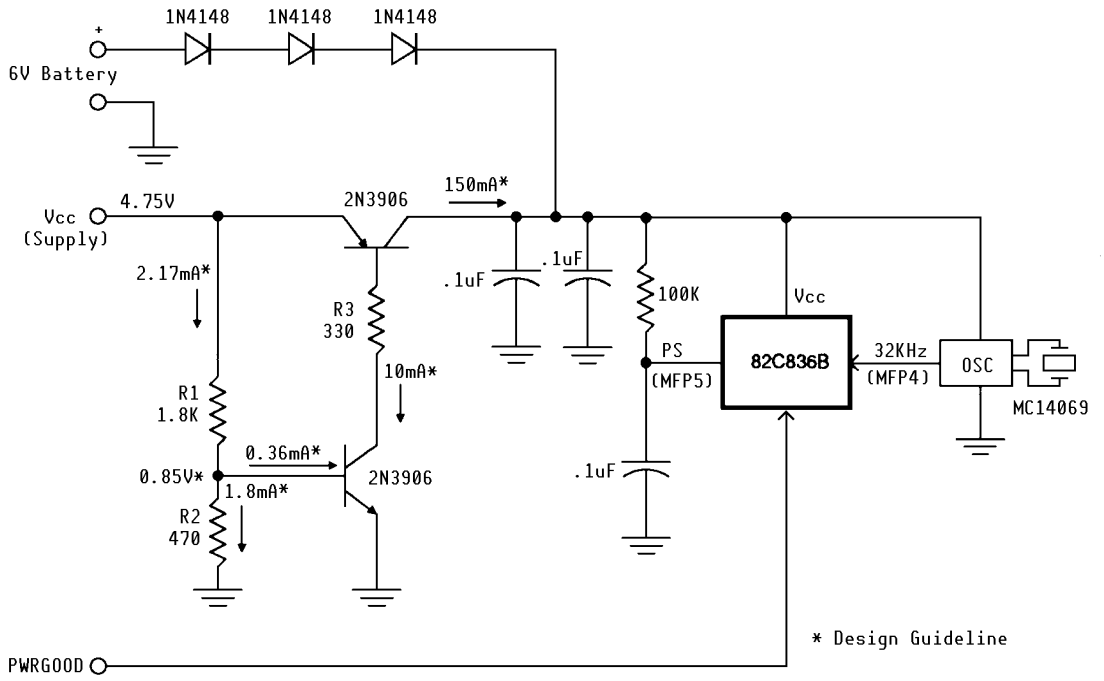


Battery Backup and Power-Up/Down

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 3-3 may be used to eliminate power drain on the battery when system power is available.

The circuit 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 32KHz 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).

Figure 3-3. Battery Backup Circuit



A similar precaution is needed during Vcc ramp up or ramp down. The Vcc "changeover" threshold, determined by R1 and R2 should be high enough so that the 2N3906 cannot remain on when Vcc is less than 4VDC. Otherwise, depending on battery voltage, the 2N3906 may again reverse conduct and load down the battery momentarily as Vcc 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 1.8VDC 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 one diode and a series resistor.)

A pin is provided in the 82C836 to protect the contents of the Real Time Clock and reduce power consumption whenever the system is powered down. The PWRGOOD pin 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 most AT-compatible architectures, the reset and power good functions normally available separately on an MC146818 are combined in the 82C836. The PWRGOOD 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. The PS/MFP5 pin does not alter the CMOS RAM or Clock/Calendar contents, but it does initialize the necessary RTC control register bits. Deassertion of Power Sense (PS) disables the generation of RTC interrupts and sets a flag indicating that the contents of the real time clock may not be valid. A recommended circuit for controlling the PS input is also shown in Figure 3-3 above (refer to applicable Product Alerts for the 82C836A Circuit).

Testability

All unnecessary inputs are disabled while PWRGOOD is low. In addition, all outputs except OSC2 are held in a high-impedance state. OSC2, the 14.3MHz output driver, is driven to a continuously low logic level. For in-circuit manufacturing test, the PS input should also be pulsed low momentarily while PWRGOOD is low to insure that the 82C836 is fully reset.

Standby Power Management and Laptop Support

The 82C836B includes the following features for stand-by power management in laptop applications:

- When the system has been powered down except for SCATsx and DRAM, SCATsx can automatically switch from normal 14.318MHz based DRAM refresh to 32KHz based DRAM refresh. This allows the contents of DRAM to be preserved during power-down. ICR 60H is used to enable or disable the stand-by refresh capability. If the stand-by refresh capability is enabled, SCATsx automatically switches to stand-by refresh in response to a logic low level on the PWRGOOD input.
- Stand-by refresh is designed to utilize the CAS-before-RAS mode. In this mode of refresh, the refresh address is generated internally by the DRAM and is not provided by the 82C836B. ICR 60H is used to enable or disable CAS-before-RAS refresh. If the CAS-before-RAS refresh mode is enabled, it applies during normal powered-on operation as well as during power down intervals.

- The refresh interval during stand-by refresh is programmable: 15 µs, 122µs, 244µs, or 488µs between refreshes (.5, 2, 4, or 8 cycles of the 32.768KHz input frequency).
- The Power Sense input (PS/MFP5) must remain high during power down in order for stand-by refresh to work. A logic low level on PS while PWRGOOD is low will clear the stand-by enable bit in ICR 60H. Since PWRGOOD is low, SCATsx will then become fully reset.
- During power-down stand-by refresh, the following output signals remain active instead of becoming high impedance:
  - -CAS3, -CAS2, -CAS1 (if multiple RAS active mode is enabled). Refer to Section 5, System Interface, subsection titled Memory Interface for a detailed discussion of the multiple RAS active mode and 4MB DRAM Configurations.
  - -CAS0, -MWE, -RAS0, -RAS1, -RAS2.
  - -RAS3 (if not using a 4MB DRAM configuration). Refer to Section 5, System Interface, subsection titled Memory Interface for a detailed discussion of the multiple RAS active mode and 4MB DRAM Configurations.
  - If a 4MB DRAM configuration is used, MA10 is actively driven to a continuous logic high level.
  - MA0-9 are actively driven to a continuous logic high level.
  - OSC2 remains actively driven to a continuous logic low level; this is always the case during power-down even if the Standby Refresh is disabled.

During normal powered-on operation, the 82C836B can optionally generate fast 8-bit timing or true 16-bit cycles for video I/O and/or video memory accesses in selected programmable address ranges (see ICR 61H and 62H). This capability is designed to improve video performance in products that use on-board video subsystems, including very compact laptop and notebook PC's with 8-bit video interfaces.

Operational Power Management

Average system power consumption can be reduced by slowing or stopping the processor clock during idle periods. If a nonstatic 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 82C836, "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 CXIN8. 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.

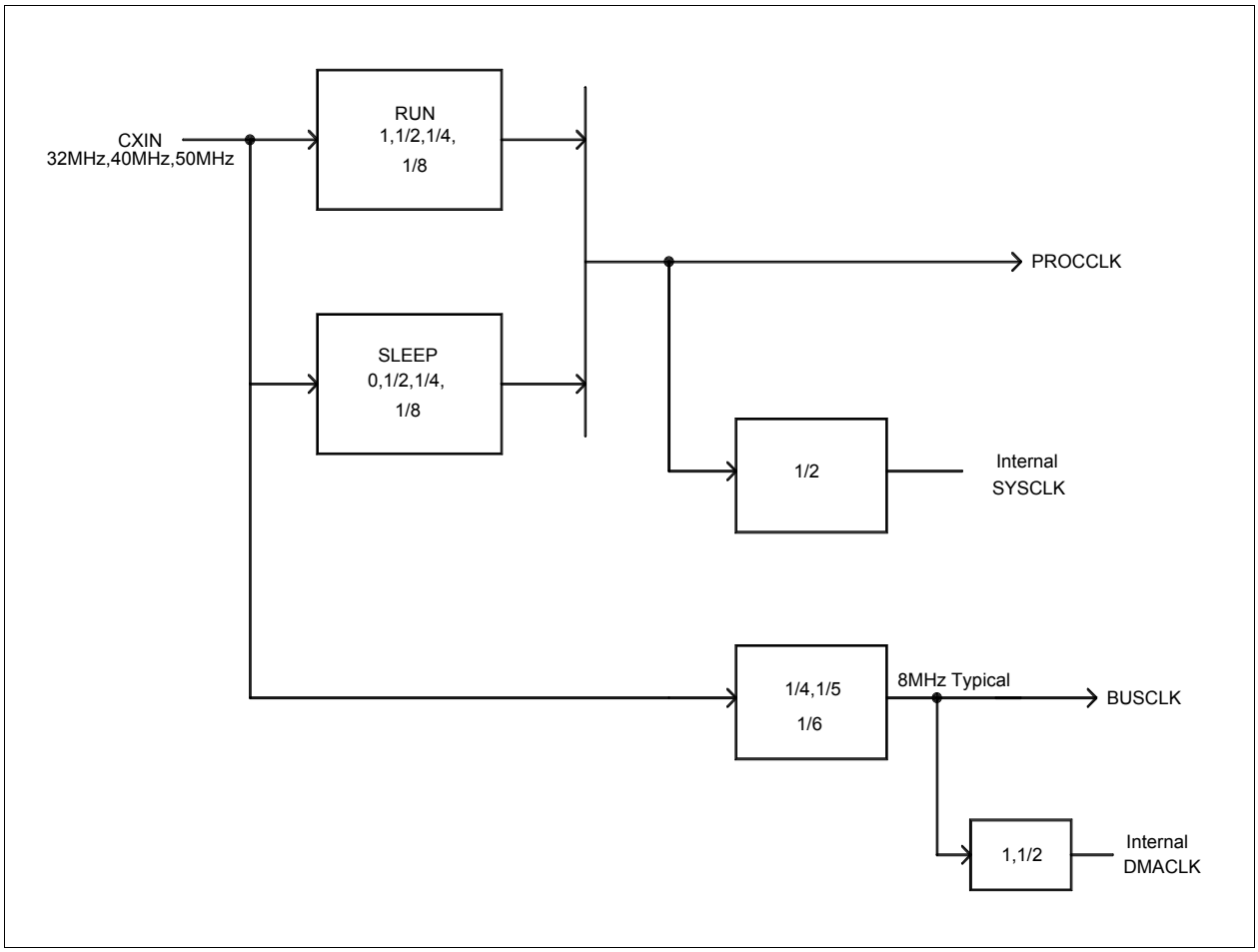
Section 4

Clock/Bus Control

Clock Generation

The 82C836 clock generator produces the signals shown in Figure 4-1.

Figure 4-1. Clock Generation



- 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.

The effective speed of the 80386sx is always half the PROCCLK frequency; e.g. for an 80386sx speed of 20MHz, PROCCLK must be 40MHz. 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 14.31818MHz output signal used by the I/O channel. The oscillator can be derived 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 (AT-compatible)
- SYSCLK (system clock) is an internal T-state 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. While 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 Section 11, 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. This allows the processor to run faster than the I/O channel, DMA commands, and timing. Accesses to the I/O channel can be programmed to run slower than the 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 capable of triggering 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, the 82C836 generates one or more DMA cycles or waits for -MASTER to be asserted by an add-in card bus master. (See Section 8, DMA Controller, subsection titled DMA Controller—8237 Compatible),
- Master request —82C836 follows the same DREQ/DACK protocol as for DMA, 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 then returns control to the Master following deassertion 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 (the AT bus, however, 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 nonpipelined 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 remains 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 must 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 must remain valid for a short hold time after the end of command. Write data

must be valid near the beginning of command; and must remain valid for a short hold time after the end of command. The command active time can be lengthened by deasserting IOCHRDY, or (in most cases) shortened by asserting 0WS.

- Master cycles on the AT bus follow the same protocol as CPU controlled cycles, except 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 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 Section 5, System Interface, subsection titled ROM/Shadow RAM Interface). In nonpipelined 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 nonpipelined 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 Section 5, System Interface, subsection titled Numeric Coprocessor Interface.
- During external cache "hit" memory reads, as discussed in Section 5, System Interface, subsection titled DRAM Interface, 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.

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 port92H. SCATsx supports both approaches. Setting 8042 GATE A20 high or Fast Gate A20 to one allows the modified A20 to track CPU A20 during CPU accesses. During DMA and Master cycles, the modified A20 always tracks CPU A20, even if modified A20 is being forced low during CPU accesses.
- 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.

Table 4-1 summarizes the possible types of bus cycles based on bus owner, target resource type and bus, and operand type. For CPU cycles alone, there are 78 different cases; DMA and Master cycles account for an additional 46 and 56 cases, respectively. SCATsx is designed to handle all these possible cases properly (AT-compatible).

Table 4-1. SCATsx Cycle Types

CPU Accesses						
Target Resource			Operand			
Size	Type	Possible Buses	Size	Direction	Address	Total Cases
16-bit	Memory	LD, XD, SD	8, 16	Read/Write	Even, Odd	18†
16-bit	Input/Output	Stx‡, LD†, XD, SD	8, 16	Read/Write	Even, Odd	24†
8-bit	Memory	XD, SD	8, 16	Read/Write	Even, Odd	12†
8-bit	Input/Output	Stx, LD†, XD, SD	8, 16	Read/Write	Even, Odd	24†
					Total Cases	78

† Coprocessor accesses are 16-bit if coprocessor is present, 8-bit if absent. For 16-bit operand at odd address, CPU itself performs two 8-bit transfers; so 16-bit operand at odd address is not counted in the Total Cases column. CPU can also perform INTA, HALT, and Shutdown cycles not listed.

‡ Stx denotes a SCATsx internal resource.

DMA Transfer						
		Memory Resources				
DMA Channel and I/O Size	I/O Bus	Size	Bus	Direction	Address	Total Cases
16-bit	SD	16	LD, XD, SD	Read/Write	Even	6
8-bit	XD†, SD	16	LD, XD, SD	Read/Write	Even, Odd	24
8-bit	XD†, SD	8	XD, SD	Read/Write	Even, Odd	16
					Total Cases	46

† DMA I/O resource on XD can be FDD only (Channel 2). Memory resource on XD can be EPROM or video RAM only. SCATsx can also perform Refresh, not listed above.

Master Cycles						
Target Resource			Operand†			
Size	Type	Bus	Size	Direction	Address	Total Cases
16-bit	Memory	LD, XD, SD	16	Read/Write	Even	6
16-bit	Memory	LD, XD, SD	8	Read/Write	Even, Odd	12
16-bit	Input/Output	Stx, XD, SD	16	Read/Write	Even	6
16-bit	Input/Output	Stx, XD, SD	8	Read/Write	Even, Odd	12
8-bit	Memory	XD, SD	8	Read/Write	Even, Odd	8
8-bit	Input/Output	Stx, XD, SD	8	Read/Write	Even, Odd	12
					Total Cases	56

† Master can reside on SD bus only. Local bus masters are also possible using HOLD/HLDA to get control, then generating CPU equivalent protocol. These cases are included in the CPU Accesses section.

Section 5

System Interface

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 32KB in the range 0C0000H-0FFFFFH.
- Local RAM can be enabled or disabled in 24 independent blocks of 16KB throughout the range 0A0000H-0FFFFFH.
- Local RAM the range 0C0000H-0FFFFFH can be write protected in eight independent blocks of 32 KB.
- If neither ROM nor RAM is enabled in a particular address block, memory accesses to that block go to the AT bus.

Memory accesses in the range FC0000H-FFFFFFH are treated as follows:

- The default is for all memory accesses in this range go to ROM in the range of 0C0000H-0FFFFFH, regardless of whether or not shadow RAM has been enabled in the target area.
- There is a programmable option to reduce the size of this special area to the top 128KB (FE0000H-FFFFFFH) instead of the top 256KB (see ICR 4E bit 4).
- There is another programmable option to map the affected high memory area (starting at FC0000H or FE0000H) to shadow RAM instead of ROM (see ICR 46H bit 5). This option is intended for laptop architectures in which the initial CPU code fetches following CPU reset need to map into shadow RAM instead of ROM.

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-wide 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 82C836B includes a DRAM controller that directly supports up to 16MB of memory in four banks. The DRAM controller in the 82C836B is capable of operating in three possible modes:

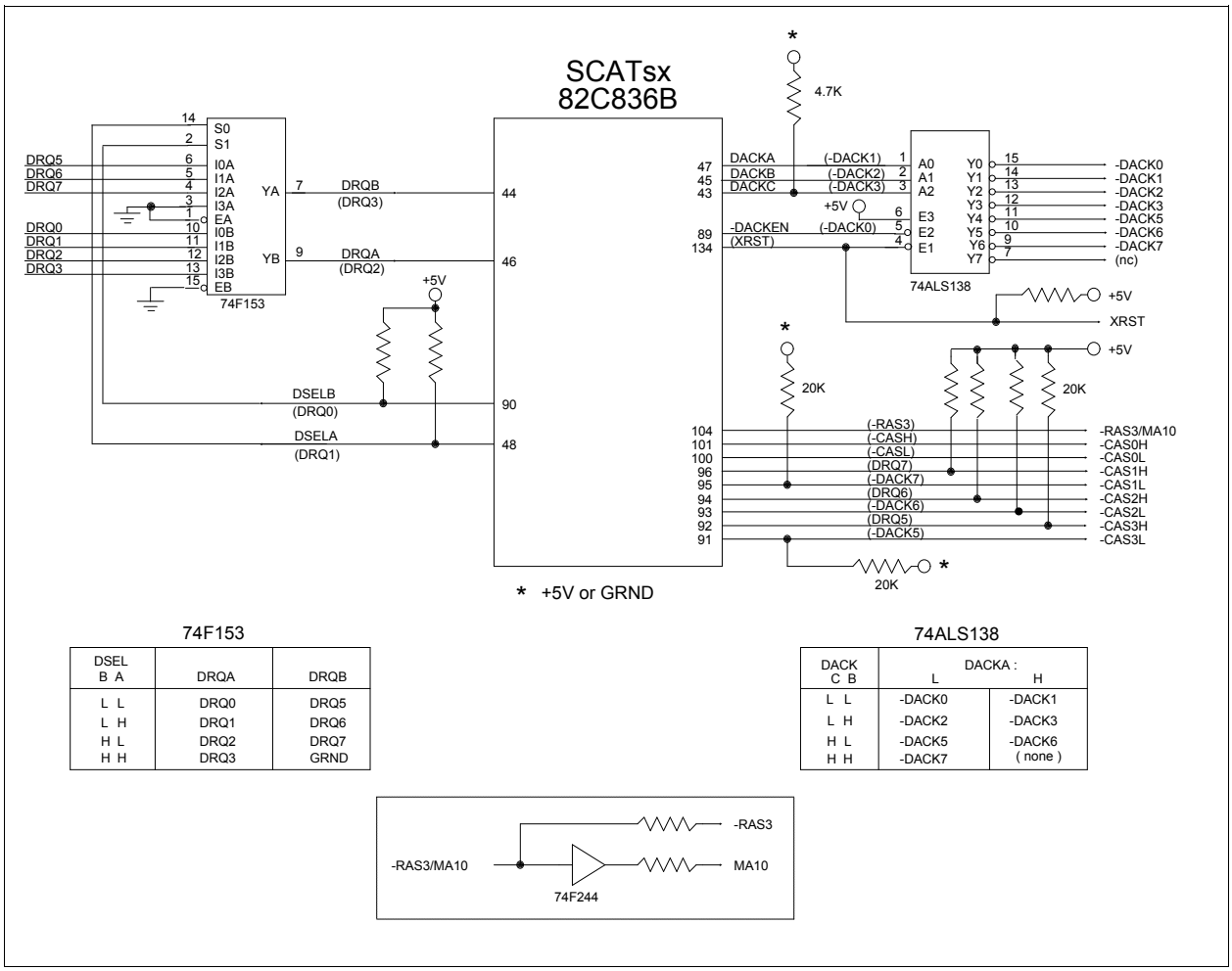
- Multiple RAS Active (MRA), also known as dedicated CAS

  In this mode, SCATsx provides four RAS signals and eight CAS signals for controlling up to four DRAM banks, each bank 16-bits wide (plus two parity bits, if desired). Each CAS signal is dedicated to controlling one eight-bit half of one bank. Page interleaved access is used whenever possible, with up to four RAS signals allowed to be active at the same time. The result is very high performance in four-bank systems, since a high percentage of DRAM accesses can be CAS-only zero wait-state accesses. MRA mode is recommended for all new designs, both for performance reasons and to improve worst-case timing margins, particularly at 25MHz. The only disadvantage is that an external 74F153 and 74ALS138 are needed to provide seven DREQ and DACK signals, see Figure 5-1.

- Single RAS Active (SRA), also known as shared CAS

  In this mode, SCATsx provides four RAS signals and only two CAS signals for controlling up to four DRAM banks. Each CAS drives four half-banks, so the CAS signals should be buffered in systems having more than two banks. The buffering adds delay which diminishes worst-case timing margins, particularly in 25MHz systems. In addition, the sharing of CAS signals between banks means that only one RAS signal can be allowed to remain active at one time. This adds an extra T-state when switching between banks (as compared to MRA mode, which can switch banks in zero wait states if the new bank already has its RAS signal active). The added T-state represents a significant performance penalty. The only advantage of SRA mode is that the external 74F153 and 74ALS138 aren't needed, since the six CAS signals become DREQs and DACKs, see Figure 5-1. Note: Signal names in parentheses ( ) refer to SRA mode. Signal names without parentheses refer to MRA mode.

Figure 5-1. MRA Mode Implementation



74F153

DSEL B A	DRQA	DRQB
L L	DRQ0	DRQ5
L H	DRQ1	DRQ6
H L	DRQ2	DRQ7
H H	DRQ3	GRND

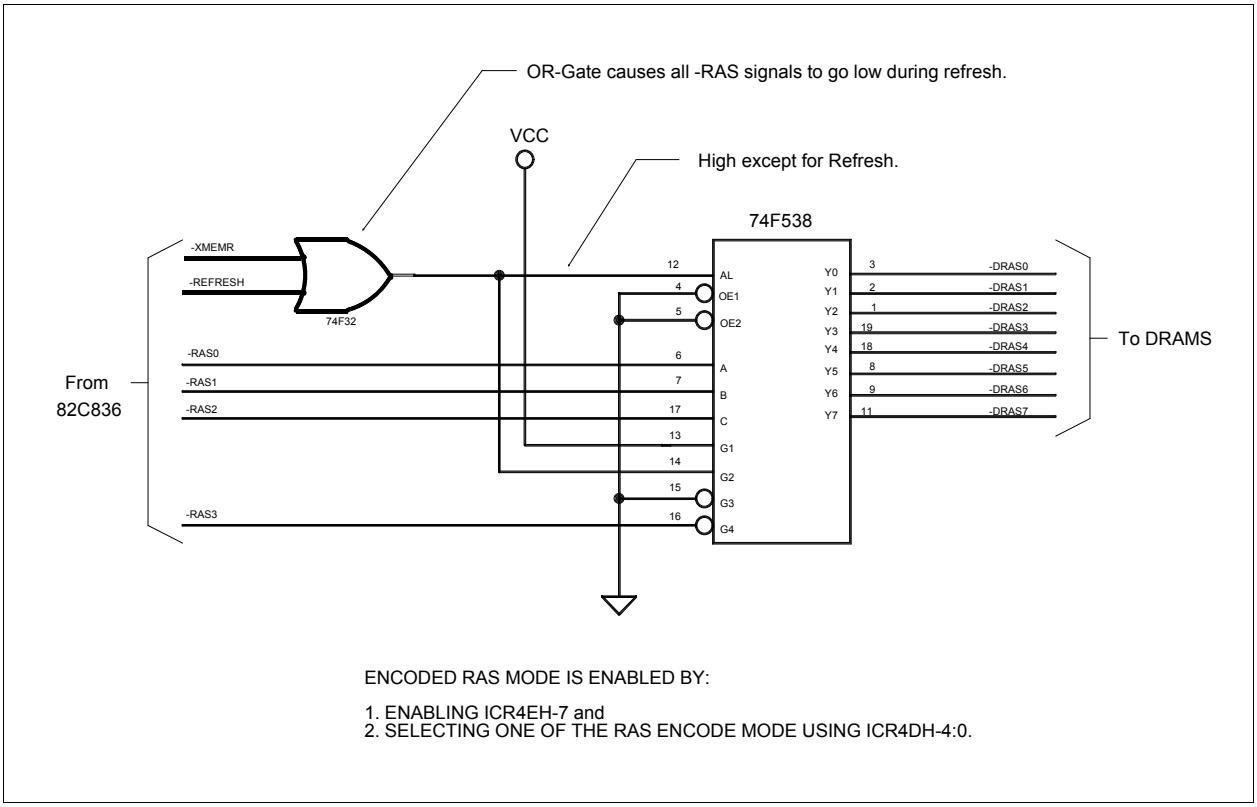
74ALS138

DACK C B	DACKA : L	H
L L	-DACK0	-DACK1
L H	-DACK2	-DACK3
H L	-DACK5	-DACK6
H H	-DACK7	(none)

- Encoded RAS (shared CAS)

  This mode is the same as SRA mode, except the four RAS signals are encoded. An external 74F538 can than be used to decode the four signals into eight RAS lines for controlling up to eight banks of DRAM (see Figure 5-2). With the advent of 4MB DRAMs and the 4MB modes in SCATsx, there is probably no further reason to use encoded RAS in new designs. In addition, the added delay on RAS and CAS signals (as compared to MRA mode) makes 25MHz zero wait-state operation unachievable if worst-case DRAM specifications are to be satisfied. Even at reduced speeds, the need to change the RAS code when switching between memory banks adds a performance degrading T-state as compared to SRA mode (two T-states as compared to MRA).

Figure 5-2. Encoded RAS Generation



In all three modes, support is provided for 256Kx1, 256Kx4, 1Mx1, and 1Mx4 DRAMs. In addition, MRA and SRA modes support 4MBx1 and 4MBx4 DRAMs. Corresponding SIMM modules are also supported; including 36-bit SIMMs utilizing either dedicated CAS signals or shared CAS between banks (the shared-CAS type cannot be used in MRA mode).

With 4MBx1 or 4MBx4 DRAMs, a maximum of three banks can be supported. The RAS signal for the third bank (-RAS3) becomes MA10. In non-4MB configurations, the multiplexed memory address is limited to MA0-9.

The usage of DREQ and DACK signals in SRA and MRA modes is summarized as follows (see Figure 5-1 shown earlier):

- In SRA mode, there are seven DREQs and seven DACKs.
- In MRA mode, two DREQs become DSEL signals to control the 74F153 multiplexer, and three DREQs become CAS signals. Similarly, four DACKs become encoded to drive the 74ALS138 decoder, and three DACKs become CAS signals. The net result is six additional CAS signals as compared to SRA mode.

If parity checking is implemented, each bank is 18 bits wide: two 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 error 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, 1Mx4, or 4Mx4 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 (in SRA or encoded RAS modes) 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.

In addition to the memory address ranges for ROM and shadow RAM discussed earlier, local RAM can be enabled or disabled in a single 384KB block in the range 040000H-09FFFFH (ICR 4EH). Memory addresses in the range 100000H-FFFFFFH (extended memory) and EMS access (expanded memory) are discussed in a later section titled SCATsx Memory Bank Utilization. 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 5-1 shows the valid configurations with either nonencoded RAS or encoded RAS. Table 5-2 shows additional configurations that are valid only with encoded RAS. Table 5-3 shows the 4MB configurations, valid only with nonencoded RAS.

Table 5-1. Valid Configurations—Nonencoded or Encoded RAS

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

K = 1024 M = 1048576 W = word (two bytes) B = byte

① All memory accesses go to the AT bus. This can be used to turn off on-board RAM even if on-board RA M exists.

② The DRAM is mapped as 1MB conventional (subject to enabling or disabling of the top 384K) and no ext ended memory.

③ The DRAM is mapped as 640KB conventional, 384KB extended. The 384KB block is addressed at 100000H-1 5FFFFH. There is no shadow RAM.

④ In all configurations, all memory beyond the first 1MB (or beyond the first 640KB in configuration 0 3H) is available for use as extended memory (addressed linearly starting at 100000H). Extended memory can be accessed either directly (in 80386sx protecte d mode) or through expanded memory address translation (EMS).

⑤ Banks are normally mapped in ascending physical address order i.e., Bank 1 has a higher starting add ress than Bank 0, etc. However, in configuration 08-0AH, the mapping order is changed so that the 256KW bank has the highest starting address instead of the lowest.

Table 5-2. Valid Configurations—Encoded RAS Only

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

K = 1024 M = 1048576 W = word (two bytes) B = byte

① In all configurations, all memory beyond the first 1MB is available for use as extended memory (addr essed linearly starting at 100000H). Extended memory can be accessed either directly (in 80386sx protected mode) or through expanded memory address trans lation (EMS).

② RAM above FC0000H is accessible only through EMS. Direct accesses to FC0000H-FFFFFFH by the CPU go to ROM.

Table 5-3. 4MB DRAM configurations—Nonencoded RAS Only

ICR 4DH Bits 4-0 ①	Banks					Total Local Memory
	0	1	2	3	4-7	
17H	256KW	256KW	4MW	0	0	9 MB
18H ②	1MW (or 0)	4MW	0	0	0	10 MB (or 8MB)
19H	4MW	4MW	0	0	0	16 MB

K = 1024 M = 1048576 W = word (two bytes) B = byte

① The function of the -RAS3 pin automatically changes to MA10 function whenever a 4MB DRAM configurati on (17H, 18H, 19H) is selected.

② Bank 1 covers address 0-7FFFFFH and Bank 0 begins at 800000H. Consequently, Bank 0 can remain empty if no internal EMS accesses are performed in the Bank 0 range (800000H-9FFFFFH), and if ICR 4EH is set to 800000H top-of-RAM address.

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, 1 through 10 for 1MB DRAMs, or 1 through 11 for 4MB 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. DMA, Master cycles, 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.
- RAS high: -RAS for the target bank is high initially, so there is no need to wait for -RAS precharge time.
- Page miss: Access to the same bank, but with a different row address, so -RAS must be cycled high and low before -CAS timing can begin.

The major difference between MRA and SRA modes, and the reason for implementing MRA mode, is the performance improvement in bank switch cycles. "Bank switch" means accessing a different DRAM bank than the preceding DRAM access. With MRA mode, RAS for the new bank frequently will be already active from an earlier access, so the bank switch can be performed without any delay for RAS cycling. The result is a cycle equal in speed to a page hit. In contrast, SRA mode forces all bank switch cycles to be RAS high cycles, since RAS for the new bank will always be high initially.

T-state counts for these cycles are as follows:

- Page hit read is 0WS
- Page hit write is 1WS
- RAS high read or write, nonencoded RAS is 1WS
- RAS high read or write, encoded RAS is 2WS
- Page miss, same bank, read or write is 3WS

The total number of T-states for pipelined cycles is the WS amount plus 2; for nonpipelined 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 (100ns at 25MHz CPU speed, 125ns at 20MHz CPU speed, 156ns at 16MHz CPU speed). This allows the use of 60ns DRAMs at 25MHz, 80ns DRAMs at 20MHz, or 100ns DRAMs at 16MHz.

The majority of all memory accesses are instruction fetches, which tend to cluster in short bursts of accesses in highly localized address ranges. Even jump operations frequently are localized. Thus, paging usually results in substantial performance improvement over nonpaged 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 adressable 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 in Table 5-2 (shown earlier).

If any two bank pairs 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 noninterleaved page mode in the remaining banks. Thus, the only configurations in which no interleaving occurs are 00H, 01H, 08H, 0BH, and 18H.

A RAS timeout feature is provided to support DRAMs requiring 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 microsecond. If the timeout is disabled, periodic refresh cycles limit the maximum possible RAS active time to about 15 microseconds.

SCATsx Memory Address Mapping Modes

The following table summarizes how CPU address bits are mapped into DRAM row and column address bits and which bits determine bank selection. The "R" bits do not necessarily map in an exact one-to-one bit order, but the R bits as a group do correspond to the CPU address bits indicated. "Page Size" refers to the block size accessible by changing column address bits only.

Table 5-4. Memory Address Mapping Modes

CPU Address Bit	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Page Size
4MW, 2-way	R	R	R	R	R	R	R	R	R	R	R	*	C	C	C	C	C	C	C	C	C	C	C	—	4KB
4MW, Page only	—	R	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	C	C	—	4KB
1MW, 4-way	—	R	R	R	R	R	R	R	R	R	R	*	*	C	C	C	C	C	C	C	C	C	C	—	2KB
1MW, 2-way	—	—	R	R	R	R	R	R	R	R	R	R	*	C	C	C	C	C	C	C	C	C	C	—	2KB
1MW, Page only	—	—	—	R	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	C	—	2KB
256KW, 4-way	—	—	—	R	R	R	R	R	R	R	R	R	*	*	C	C	C	C	C	C	C	C	C	—	1KB
256KW, 2-way	—	—	—	—	R	R	R	R	R	R	R	R	R	*	C	C	C	C	C	C	C	C	C	—	1KB
256KW, Page Only	—	—	—	—	—	R	R	R	R	R	R	R	R	R	C	C	C	C	C	C	C	C	C	—	1KB

R = Row address bit
C = Column address bit
* = Bank select bit for interleaving
— = High level decoding and byte Selection
W = Word (even and odd bytes)

SCATsx Memory Bank Utilization

The following tables summarize the exact address ranges and interleaving sequences for each supported memory configuration. It is assumed that shadow RAM is enabled, internal EMS is disabled, and the top of RAM (ICR 4E) is set to the entire on-board memory.

Table 5-5. Memory Configuration Address Ranges and Interleaving Sequences

Physical Configuration	Address Ranges	Map Mode	Banks	Page Interleave Size
00 = no on-board DRAM	None	—	—	—
01 = 2x256KB	000000-07FFFFH	256KW/P	0	—
02 = 4x256KB (1M/0)	000000-0FFFFFH	256KW/2WI	0, 1	400H
03 = 4x256KB (640/384)	000000-09FFFFH	256KW/2WI	0, 1	400H
03 = 4x256KB (640/384)	100000-15FFFFH	256KW/2WI	0, 1	400H
04 = 6x256KB	000000-0FFFFFH	256KW/2WI	0, 1	400H
04 = 6x256KB	100000-17FFFFH	256KW/P	2	—
05 = 8x256KB	000000-1FFFFFH	256KW/4WI	0-3	400H
06 = 4x256KB, 2x1MB	000000-0FFFFFH	256KW/2WI	0, 1	400H
06 = 4x256KB, 2x1MB	100000-2FFFFFH	1MW/P	2	—
07 = 4x256KB, 4x1MB	000000-0FFFFFH	256KW/2WI	0, 1	400H
07 = 4x256KB, 4x1MB	100000-4FFFFFH	1MW/2WI	2, 3	800H
08 = 2x256KB, 2x1MB	000000-1FFFFFH	1MW/P	1	—
08 = 2x256KB, 2x1MB	200000-27FFFFH	256KW/P	0	—
09 = 2x256KB, 4x1MB	000000-3FFFFFH	1MW/2WI	1, 2	800H
09 = 2x256KB, 4x1MB	400000-47FFFFH	256KW/P	0	—
0A = 2x256KB, 6x1MB	000000-3FFFFFH	1MW/2WI	1, 2	800H
0A = 2x256KB, 6x1MB	400000-5FFFFFH	1MW/P	3	—
0A = 2x256KB, 6x1MB	600000-67FFFFH	256KW/P	0	—
0B = 2x1MB	000000-1FFFFFH	1MW/P	0	—
0C = 4x1MB	000000-3FFFFFH	1MW/2WI	0, 1	800H
0D = 6x1MB	000000-3FFFFFH	1MW/2WI	0, 1	800H
0D = 6x1MB	400000-5FFFFFH	1MW/P	2	—
0E = 8x1MB	000000-7FFFFFH	1MW/4WI	0-3	800H

Table 5-6. Memory Configuration Address Ranges and Interleaving Sequences Encoded RAS Only

Physical Configuration	Address Ranges	Map Mode	Banks	Page Interleave Size
0F = 4x256KB, 6x1MB	000000-0FFFFFH	256KW/2WI	0, 1	400H
0F = 4x256KB, 6x1MB	100000-4FFFFFH	1MW/2WI	2, 3	800H
0F = 4x256KB, 6x1MB	500000-6FFFFFH	1MW/P	4	—
10 = 4x256KB, 8x1MB	000000-0FFFFFH	256KW/2WI	0, 1	400H
10 = 4x256KB, 8x1MB	100000-4FFFFFH	1MW/2WI	2, 3	800H
10 = 4x256KB, 8x1MB	500000-8FFFFFH	1MW/2WI	4, 5	800H
11 = 4x256KB, 10x1MB	000000-0FFFFFH	256KW/2WI	0, 1	400H
11 = 4x256KB, 10x1MB	100000-4FFFFFH	1MW/2WI	2, 3	800H
11 = 4x256KB, 10x1MB	500000-8FFFFFH	1MW/2WI	4, 5	800H
11 = 4x256KB, 10x1MB	900000-AFFFFFH	1MW/P	6	—
12 = 4x256KB, 12x1MB	000000-0FFFFFH	256KW/2WI	0, 1	400H
12 = 4x256KB, 12x1MB	100000-4FFFFFH	1MW/2WI	2, 3	800H
12 = 4x256KB, 12x1MB	500000-CFFFFFH	1MW/4WI	4-7	800H
13 = 10x1MB	000000-7FFFFFH	1MW/4WI	0-3	800H
13 = 10x1MB	800000-9FFFFFH	1MW/P	4	—
14 = 12x1MB	000000-7FFFFFH	1MW/4WI	0-3	800H
14 = 12x1MB	800000-BFFFFFH	1MW/2WI	4, 5	800H
15 = 14x1MB	000000-7FFFFFH	1MW/4WI	0-3	800H
15 = 14x1MB	800000-BFFFFFH	1MW/2WI	4, 5	800H
15 = 14x1MB	C00000-DFFFFFH	1MW/P	6	—
16 = 16x1MB	000000-7FFFFFH	1MW/4WI	0-3	800H
16 = 16x1MB	800000-FFFFFFH*	1MW/4WI	4-7	800H

* Top 128KB or 256KB accessible only via EMS.

Table 5-7. Memory Configuration Address Ranges and Interleaving Sequences Nonencoded RAS Only

Physical Configuration	Address Ranges	Map Mode	Banks	Page Interleave Size
17 = 4x256KB, 2x4MB	000000-0FFFFFH	256KW/2WI	0, 1	400H
17 = 4x256KB, 2x4MB	100000-8FFFFFH	4MW/P	2	—
18 = 2x1MB, 2x4MB	000000-7FFFFFH	4MW/P	1	—
18 = 2x1MB, 2x4MB	800000-9FFFFFH	1MW/P	0	—
19 = 4x4MB	000000-FFFFFFH*	4MW/2WI	0, 1	1000H

* Top 128KB or 256KB accessible only via EMS.

Automatic Memory Configuration Detection

Typical system BIOS implementations include automatic tests to determine memory mode and configuration without requiring user intervention. The following operational characteristics may be useful for these BIOS algorithims:

- External hardware can be encoded RAS (SRA), nonencoded RAS (SRA), or MRA. Encoded RAS hardware can be detected by testing Bank 0 in both encoded RAS and nonencoded RAS (SRA) modes. Bank 0 is inaccessible in SRA mode if the external hardware is encoded RAS, but is accessible in either SRA or MRA mode with nonencoded RAS hardware.
- If encoded RAS hardware is detected, configuration 16H (8 banks of 1MW) can be used to detect 256K or 1MB DRAM size. If address bit 10 is ignored, i.e., the same physical data is accessible regardless of A10, then the DRAM is 256K. Address bit 10 corresponds to an address increment of 400H.
- With nonencoded RAS hardware, Bank 1 (if not empty) can be used to detect MRA or SRA hardware. If the external hardware is MRA, but SRA mode is used, or vice versa, Bank 1 is inaccessible. However, if Bank 1 is empty, there is probably no simple way to differentiate between SRA and MRA. Schemes involving DMA requests are possible but probably are not general enough to work in all possible system configurations (MRA or SRA mode must still be setup correctly for the DMA channels to operate properly).
- 256K DRAM size can be detected in Banks 0-3 using configuration code 0EH (4 banks of 1MW). If address bit 10 is ignored, i.e., same memory data accessed regardless of A10, then the DRAM is 256K. Address bit 10 corresponds to an address increment of 400H.
- If Bank 3 is empty and 256KW has been found in Banks 0 and 1, then Bank 2 can be tested for 4MW using configuration 17H. Otherwise, if Banks 2 and 3 are both empty, then Banks 0 and 1 can be tested for 4MW using configuration 19H. In either case, address bit 11 being ignored indicates 1MB DRAM size. Address bit 11 corresponds to an address increment of 800H.

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 a 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 the 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 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 rewritten. 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 ICR41H 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 nonpipelined, 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 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 AT bus memory resources residing in the first 1MB 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 address 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; local DRAM writes can be claimed also, if ICR 63H bit 0 is set. One use for this feature is directly reading and writing cache data RAM in a diagnostic test.
- AT bus memory reads or writes, I/O reads or writes, and interrupt acknowledge.

When using Early READY or LBA modes, the CPU -NA input must be tied high, causing the CPU to operate in nonpipelined 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 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 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, 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

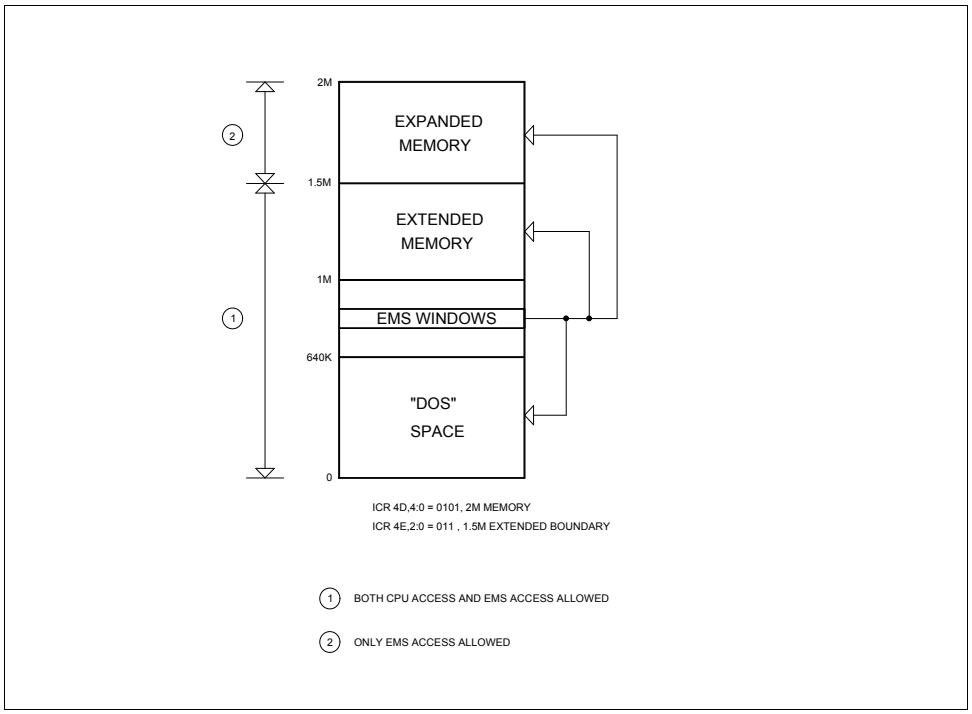
The 82C836 fully supports the LIM EMS 4.0 and 3.2 specifications. EMS allows operating systems and software applications to access memory above the 1MB DOS limit through a page mapping scheme managed by an EMS driver. The SCATsx architecture allows up to 64KB of address space in the first 1MB to be remapped, 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 because these have fixed physical addresses. The target pages are referred to as logical pages because these have logical addresses only —until mapped into a physical page within the page frame.
- 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 5-3.

Figure 5-3. Example of EMS/Extended Memory



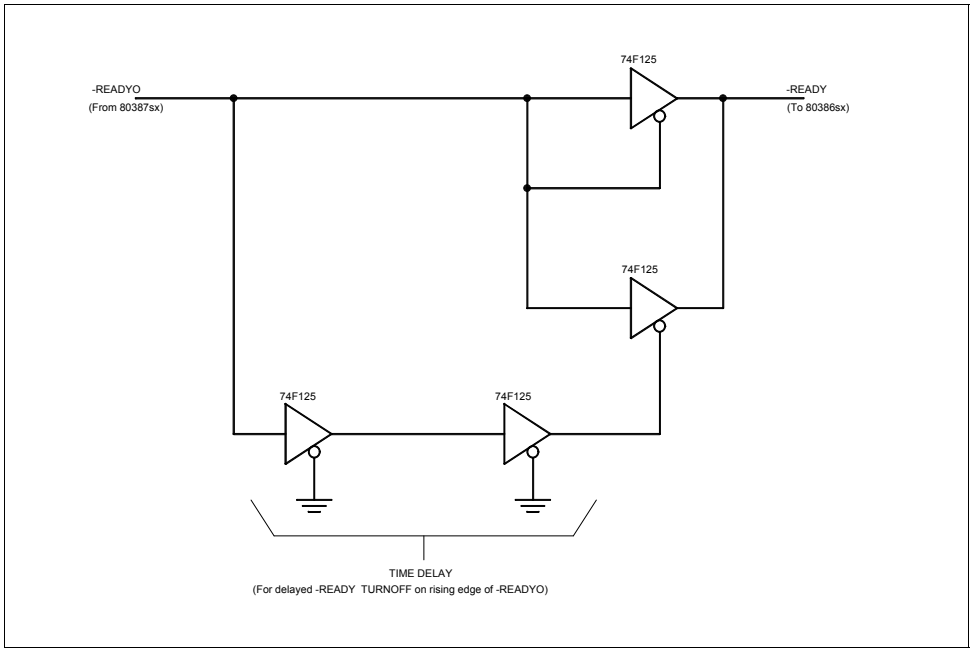
EMS paging and the EMS I/O ports are normally managed by application software or an EMS driver that allows applications to access more RAM than is normally allowed by DOS. More information on software implementation can be obtained from various Intel® and Microsoft® publications.

Numeric Coprocessor 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 invaraiably 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 does not differentiate between protected and real modes.)
- Automatically detects the presence or absence of the 80387sx 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 inteface, such as that shown in Figure 5-4, 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.
- 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.

Figure 5-4. -READYO Interface



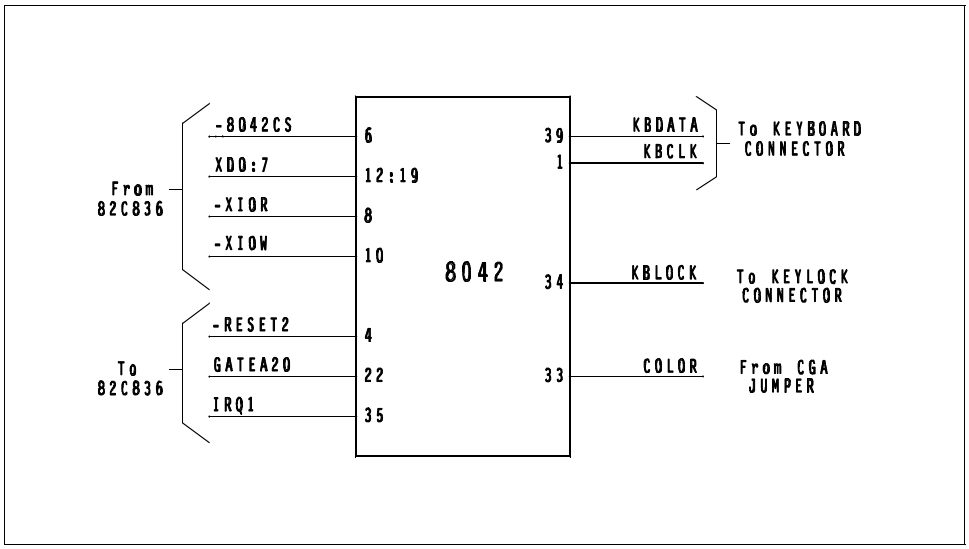
Additional considerations are as follows:

- 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 remaining I/O cycles 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 preset, a low-order data bit, such as D0, should be pulled down instead of pulled up (see Appendix A, System Schematics).

Keyboard Controller Interface—External 8042

The 82C836 relies on an external 8042 to handle keyboard operations (see Figure 5-5). The clock for the 8042 may be derived from BUSCLK or OSC and should have a frequency between 6MHz and 10MHz 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 GATE A20 and Fast CPU Reset functions available via port 92H. The keyboard controller also supports the keylock and CGA strap functions.

Figure 5-5. Keyboard Controller



Section 6

Real Time Clock and Interval Timer Registers

Real Time Clock 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 -EXTRTC 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 Vcc pins should be connected to battery-backed power
- MFP4 is used as the 32KHz clock input
- MFP5 is used as the power sense input.

When an external RTC is used:

- 82C836 Vcc 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
- -RTCCS should be connected to PWRGOOD through a diode (see Figure 3-2 shown earlier).

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 6-1 shows the address map for the real time clock.

Table 6-1. Address Map for the Real Time Clock

Column	Function	Column	Function
00	Seconds	09	Year
01	Seconds alarm	0A	Register A
02	Minutes	0B	Register B
03	Minutes Alarm	0C	Register C
04	Hours	0D	Register C
05	Hours alarm	0E	Register D
06	Day of the week	0F	User RAM
07	Day of the month	10-7F	User RAM
08	Month		

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 could potentially intervene between the output to port 70H and the subsequent I/O to port 71H, overwriting the port 70H value.

Real Time Clock Address Map

Table 6-1 (shown above) identifies 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 pupose 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 infomration is accomplished by writing to these locations. Information is stored in these locations in binary-coded decimal (BCD) format.

Before initialization of the internal register can be performed, the SET bit in register B should be set to a one to prevent Real Time Clock updates from occuring. 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 6-2. 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)	01-12 (AM), 81-92 (PM)
	Hours (24 hour mode)	00-23
5	Hours Alarm (12 hour mode)	01-12 (AM), 81-92 (PM)
	Hours Alarm (24 hour mode)	00-23
6	Day of Week	01-07
7	Day of Month	01-31
8	Month	01-12
9	Year	00-99

Table 6-2 above 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 set to one.

During uptdates, 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 peroidic interrupt.

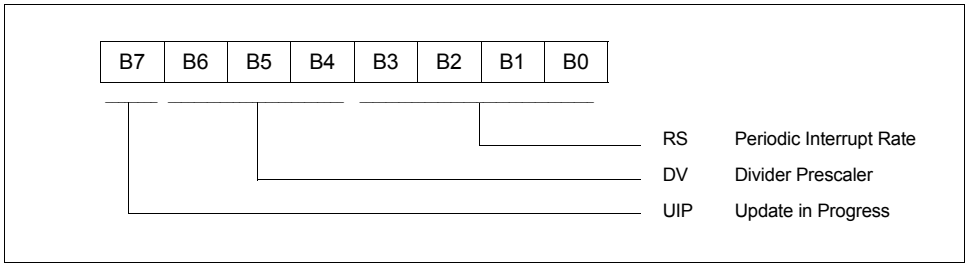
Static RAM

The 114 bytes of RAM for 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 nonvolatile 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 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

Figure 6-1. Register A—Address 0AH (All Bits Except UIP are Read/Write)

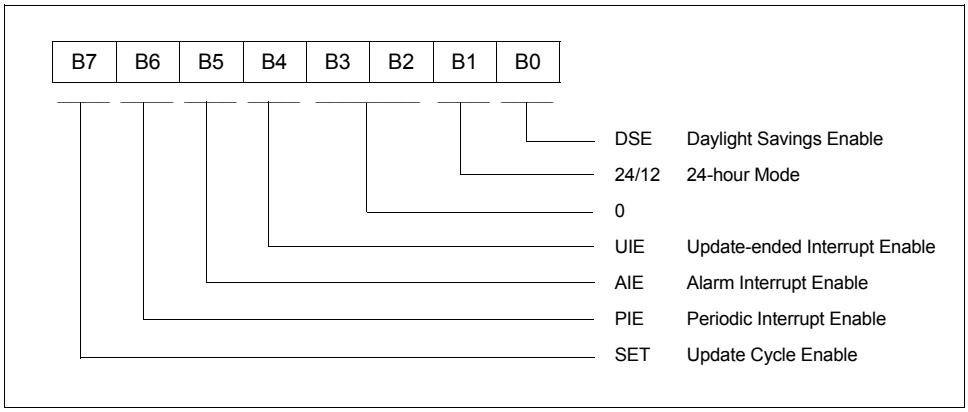


bits: B0-B3 RS These bits control the periodic interrupt rate. The periodic interrupt is derived from the divider/prescaler in the Real Time Clock and is separate from the alarm interrupt. Both the alarm and periodic interrupts use the same interrupt channel in the interrupt controller. Use of the periodic interrupt allows the generation of interrupt rates higher than one per second. Below are the interrupt rates for which the Real Time Clock may be programmed.

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

B7	UIP	Update In Progress flag is a status bit used to indicate when an update cycle is about to take place. A one indicates 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.
B4-B6	DV	These bits control the Divider Prescaler on the Real Time Clock. While the 82C836 RTC can operate at frequencies higher than 32.768KHz, this is not recommended for battery-powered operation due to the increased power consumption at these frequencies. OSCI frequencies and modes are: • 4.194304MHz in operate mode when DV2 = 0, DV1 = 0 and DV0 = 0 • 1.048576MHz in operate mode when DV2 = 0, DV1 = 0 and DV0 = 1 • 32.768KHz in operate mode when DV2 = 0, DV1 = 1 and DV0 = 0 The divider is reset when DV2 = 1 and DV1 = 1, regardless of DV0.

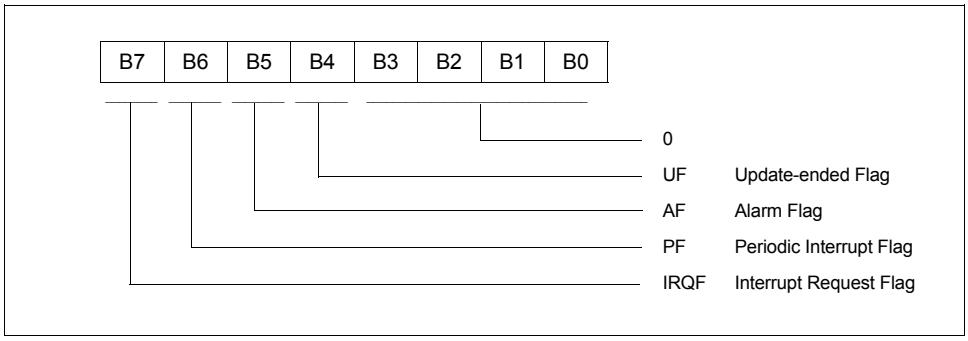
Figure 6-2. Register B—Address 0BH (Read Only)



bits:	B0	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.
	B1	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.
	B2-B3	0	Read as zeros.

B4	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.
B5	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.
B6	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.
B7	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.

Figure 6-3. Register C—Address 0CH (Read Only)



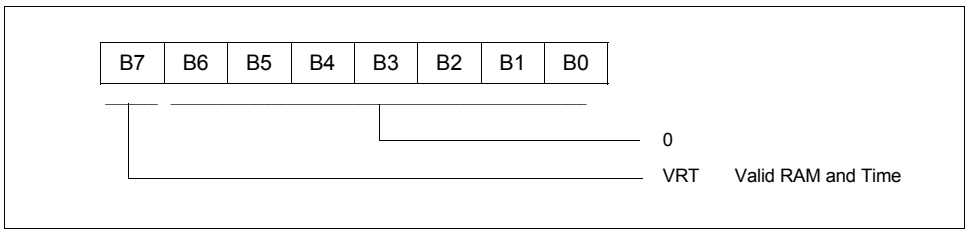
bits: B0-B3		Read as zeros.
B4	UF	The Update-ended Flag bit is set after each cycle. When the UIE bit is a one, the one in UF causes the IRQF bit to be a one asserting IRQ. UF is cleared by a register read or by a RESET.
B5	AF	A one appears in the AlarmFlag bit whenever a match has occurred between the 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.
B6	PF	The Periodic 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.

B7 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 activation the -PS input pin. Writing to this register has no effect on the contents.

Figure 6-4. Register D—Address 0DH (Read Only)



bits: B0-B6 0 Read as zeros.

B7 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 Vcc to the device and allows the user to determine whether the registers have been intialized since power was applied to the device. PWRGOOD 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.

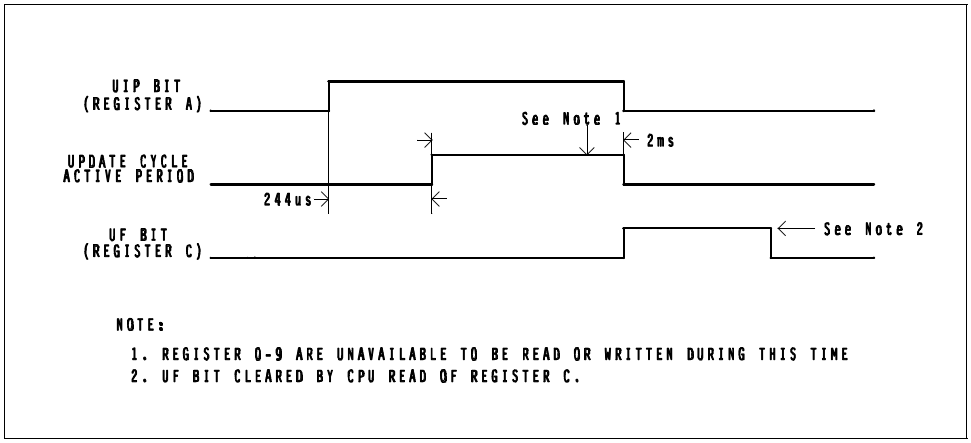
Update Cycle

During normal operation, the Real Time Clock performs an update cycle once every second. The performance of an update cycle is contigent upon the divider bits DV<0:2> not 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 register, an alarm is issued. An interrupt is issued if the alarm and interrupt control bits are enabled.

While 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 register 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 (UIP) bit 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 6-5 illustrates the update cycle.

Figure 6-5. 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.

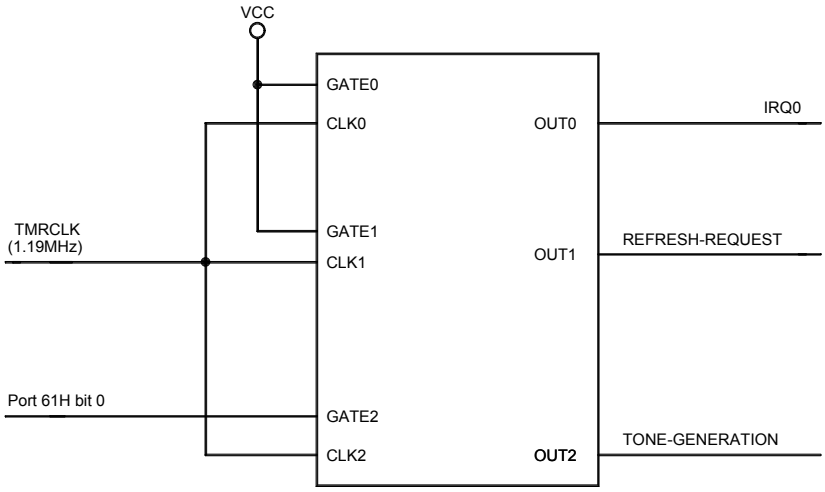
The first method is to read register A, determine the state of the UIP bit. If the UIP is zero, perform the read or write opertation. For this method to work successfully, the entire read or write operation (including any interrupt service routines which might occur) must not require more than 244 µs 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 comlete 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 6-6).

Figure 6-6. Programmable Timer Control



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 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 0-2) 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 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 the the lack of an external hardware trigger signal. Counter 2 can be operated in any of six modes listed as follows.

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

All three counters in the CTC are driven from a common clock TMRCLK (1.19MHz) derived by dividing OSC1 (14.31818MHz) by 12. Counter zero output (OUT0) is connected to IRQ0 of INTC1 (see Section 7, Interrupt Controller) 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 (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 61H, bit 0), and an OUT signal. The OUT signal's state and function are controlled by the Counter Mode and condition of the CE.

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, 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 65536 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. 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.

Programming the Counter Timer Controller

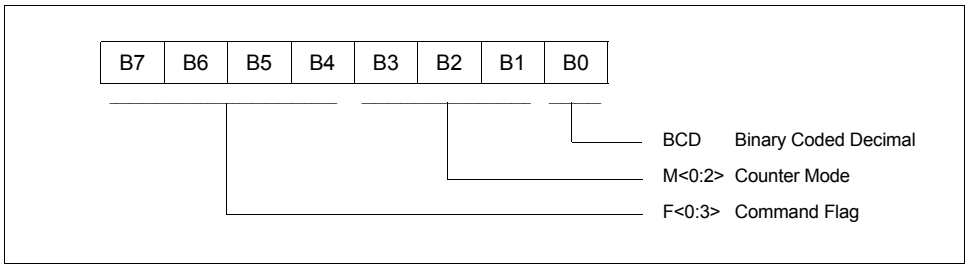
After power-up, the conditions of the CTC control register, counter register, CE, and the output of all counters is undefined. Each counter must be programmed before it can be used.

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 6-3. Figure 6-7 shows the Control Word structure. The control word is a write-only location.

Table 6-3. Counter Timer Control I/O Addresses

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

Figure 6-7. Control Word—Address 043H



bits: B0 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.

B1-B3 M<0:3> These three bits determine the counter's mode during Read/Write counter commands, or select the counter during a Read-Back command. Bits 1-3 become "do not care" during Latch Counter commands.

B4-B7 F<0:3> These four bits determine the command to be performed as shown below.

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 2 (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 2 (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

Read/Write Counter Command

When writing to a counter, two convensions 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 = 0 and M0 = 0
- Select Mode 5 when M2 = 1 and M1 = 0 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.

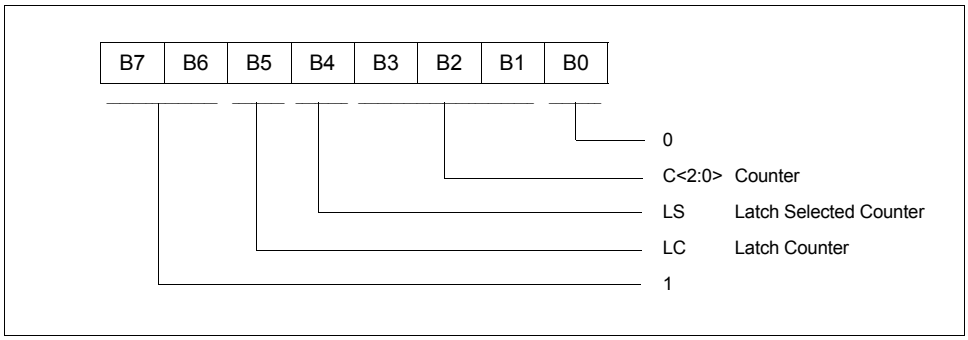
Latch Counter commands may be issued to more than one counter before reading the first counter to which the command was issued. Also, multiple Latch Counter commands issued to the same counter without reading the counter cause all but the first command to be ignored.

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 (NC) flag of the selected counter(s).

The format of the Read-Back command is shown in Figure 6-8.

Figure 6-8. Read-Back Command Format



bits: B0 0 Read as zero.

B1-B3 C<2:0> Writing a one in bit 3 causes Counter 3 to latch one or both of the registers speicified 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.

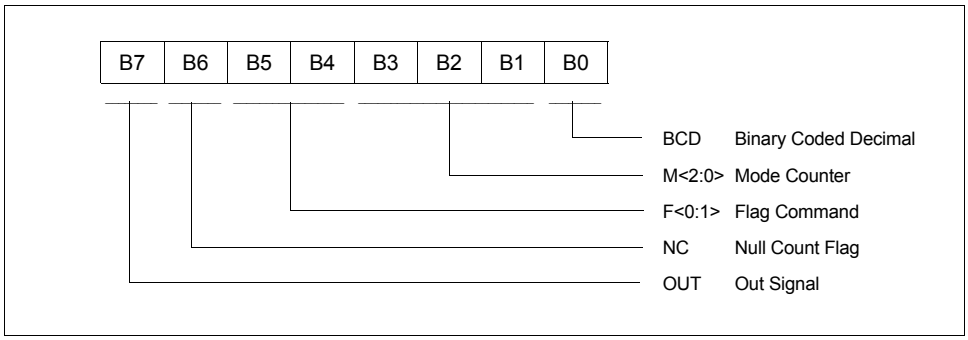
If LS = LC = 0, status is returned on the next read from the counter. The next one or two reads (depending on whether the counter is programmed to transfer one or two bytes) from the counter result in the count being returned.

B4	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.
B5	LC	Writing a zero in bit 5 causes the selected counter(s) to latch to the state of the CE in COL and COH.
B6-B7	1	Read as ones.

Status Byte

The format of the Status Byte command is shown in Figure 6-9.

Figure 6-9. Status Byte Command



bits:	B0	BCD	Bit 0 indicates the CE is in operation in BCD format.
	B2-B3	M<0:2>	These three bits reflect the mode of the counter and are interpreted in the same manner as in Write command operations.
	B4-B5	F<0:1>	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, low byte, or both must be transferred during counter read/write operations.
	B6	NC	Bit 6 contains the condition of the Null Count flag. This flag is used to indicate 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 whevever the counter is loaded from the counter input registers.
	B7	OUT	Bit 7 contains the state of the OUT signal of the counter.

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 modes, 0, 2, 3, and 4 are relevant to all counters.

The following terms are defined for describing CTC operation.

- TMRCLK pulse—a clock equivalent to OSC1 (14.31818 MHz) 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—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. 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).

  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 Retriggerable 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 retriggered.

- 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 immediately forces OUT2 high. A trigger releads 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 retriggered by software.

- Mode 5—Hardware Triggered Strobe

  Writing the Control Word causes OUT2 to go high initially. Counting is started by a trigger. The expiration of the initial count causes OUT2 to go low for one TMRCLK cycle. GATE2 = 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 6-4.

Table 6-4. Gate Pin Function

Mode	Conditions		
	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	—

Section 7

Interrupt Controller

The 82C836 incorporates two programmable interrupt controllers 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 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 7-1 shows the interrupt levels used by the system board and I/O channel. The interrupts are shown by priority, starting with the highest level.

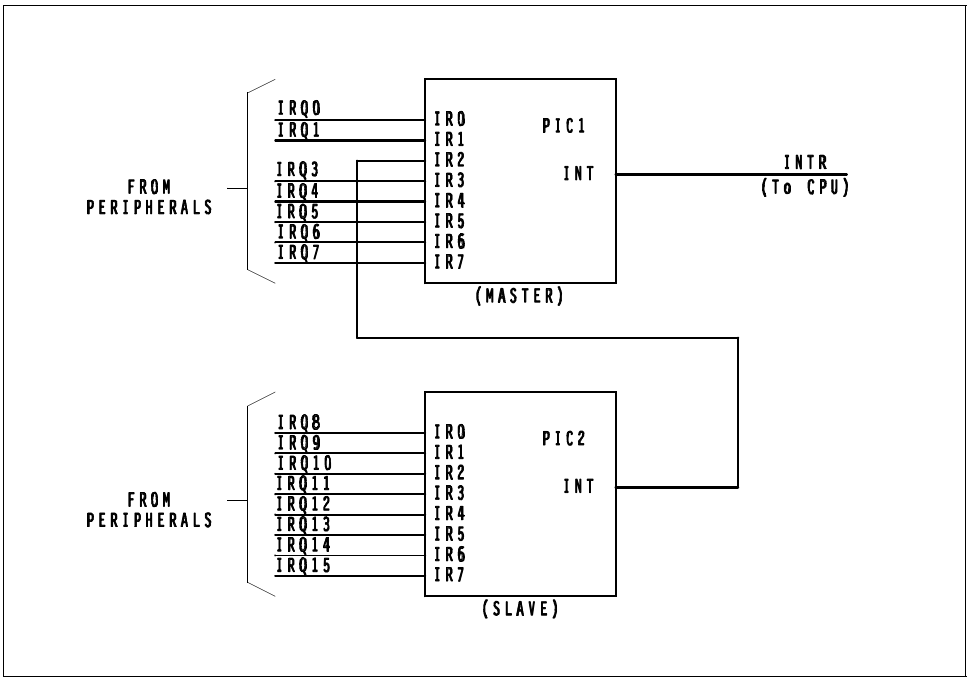
Table 7-1. Interrupt Levels for System Board

Interrupt	System Board	I/O Channel
NMI	Parity check	IOCHCK
IRQ0	Timer	Not available
IRQ1	Keyboard	Not available
IRQ8*	Real Time Clock	Not avaiable
IRQ9	Not used	Available
IRQ10	Not used	Available
IRQ11	Not used	Available
IRQ12	Not used	Available
IRQ13	Coprocessor	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 and Floppy Disk	Available
IRQ7	Parellel 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 7-1) for proper operation of all 16 interrupt channels. INTC1 is located at addresses 020H-021H and is configured for Master operation in Cascade Mode. INTC2 is a slave device 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 7-1. Cascaded Interrupt Controllers



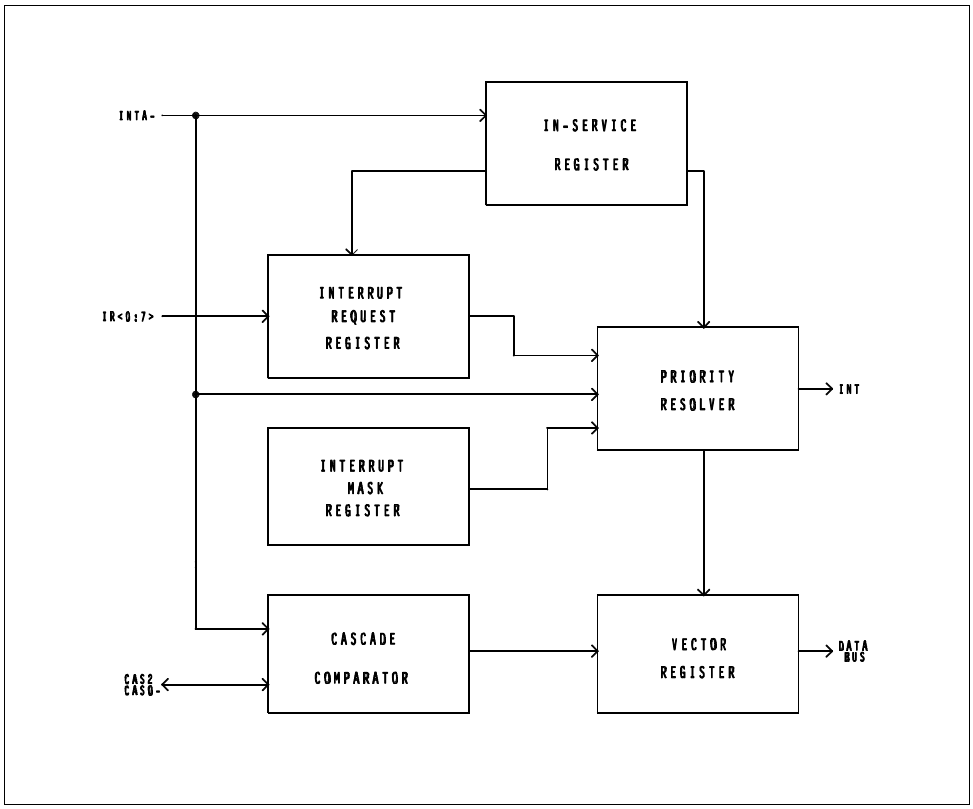
Two additional interconnections are made to the interrupt request inputs of the interrupt controllers. The output of Timer 0 in the Counter/Timer subsystem is connected to Channel 0 (IR0) of INTC1. An Interrupt request from the Real Time Clock is connected to Channel 0 (IR0) of INTC2.

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

Controller Operation

Figure 7-2 shows 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 requesting service. Interrupt Request Register bits are labeled using the channel name IR<0:7>. The In-Service Register (ISR) contains all the channels currently being serviced (more than one channel can be in service at the same 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 7-2. 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 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 occurring 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 7-2. Note that V<3:7> is programmable by writing to Initialization Control Word 2.
6. At the end of the second INTA cycle, the ISR bit is cleared if the Automatic End-Of-Interrupt (AEOI) mode is selected. Otherwise, the IRS bit must be cleared by an End-Of-Interrupt (EOI) command from the CPU at the end of the interrupt service routine.

Table 7-2. Interrupt Vector Format

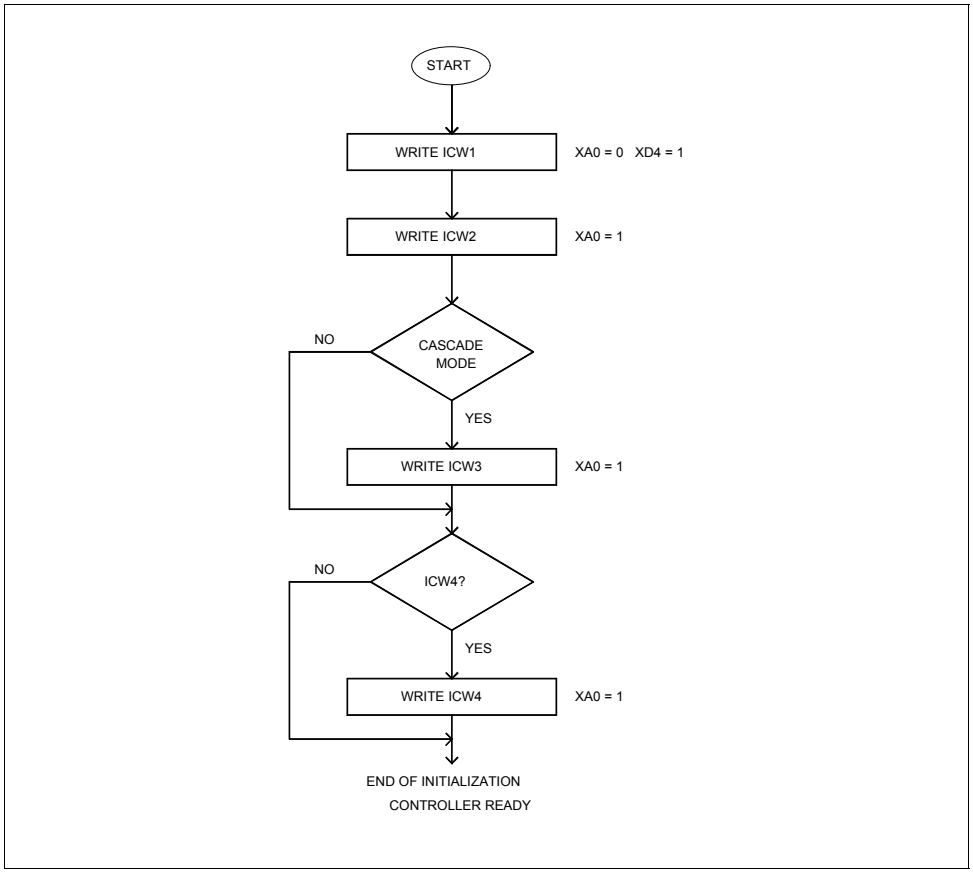
Bits	Vectors							
IR7	V7	V6	V5	V4	V3	1	1	1
IR6	V7	V6	V5	V4	V3	1	1	0
IR5	V7	V6	V5	V4	V3	1	0	1
IR4	V7	V6	V5	V4	V3	1	0	0
IR3	V7	V6	V5	V4	V3	0	1	1
IR2	V7	V6	V5	V4	V3	0	1	0
IR1	V7	V6	V5	V4	V3	0	0	1
IR0	V7	V6	V5	V4	V3	0	0	0

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.

End-Of-Interrupt

End-Of-Interrupt (EOI) is defined as the condition causing 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 (nonspecific EOI).

Figure 7-3. Initialization Sequence



The 82C836 can determine the correct ISR bit to reset when operated in modes that don't 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 an IMR bit, is not cleared by a nonspecific EOI command.

Optionally, the interrupt controller can generate an Automatic End-Of-Interrupt (AEOI) on the trailing edge of the second INTA cycle.

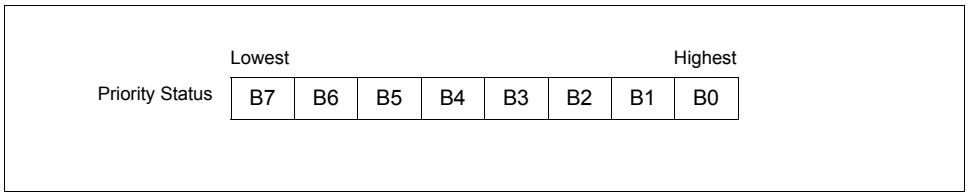
Priority Assignment

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 the following Figure.

Figure 7-4. Fixed Priority Mode

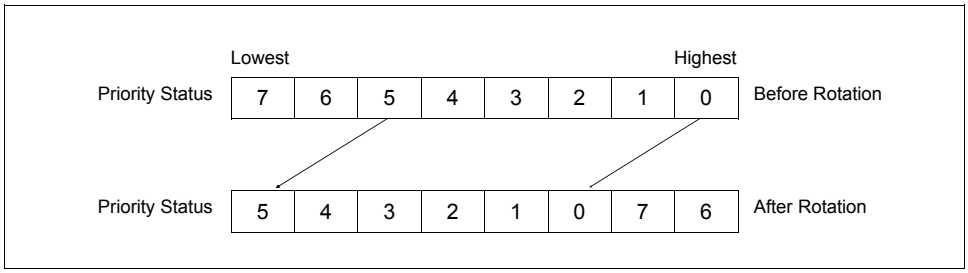


Nesting allows interrupts with higher priorities to generate interrupt requests prior to the completion of the interrupt in service. When an interrupt is acknowledged, priority is resolved, the highest priority request's vector is placed on the bus, and the ISR bit for that channel is set. This bit remains 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 reassign priority levels by issuing a command that redefines the highest priority channel. Figure 7-5 illustrates the result of a specific rotation command, assigning highest priority to Channel 5.

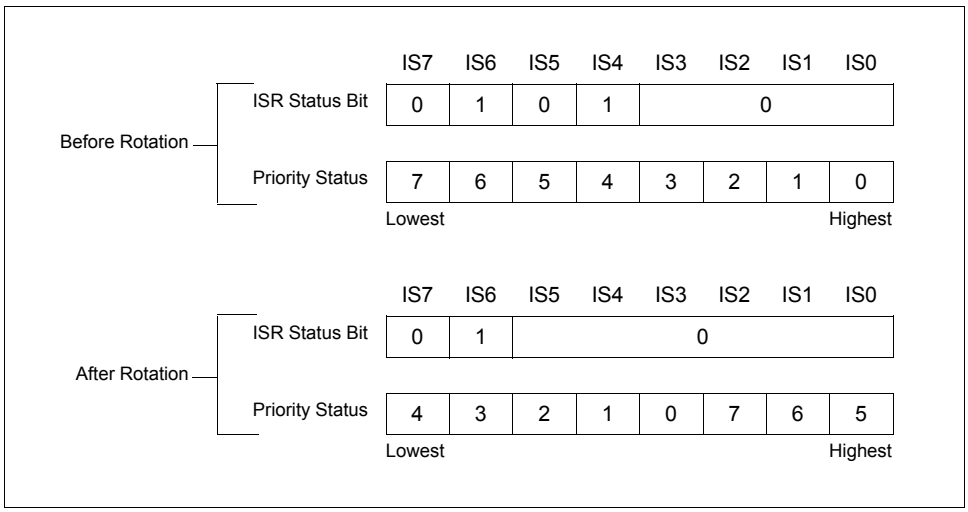
Figure 7-5. Specific Rotation Mode



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).

Figure 7-6. 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).

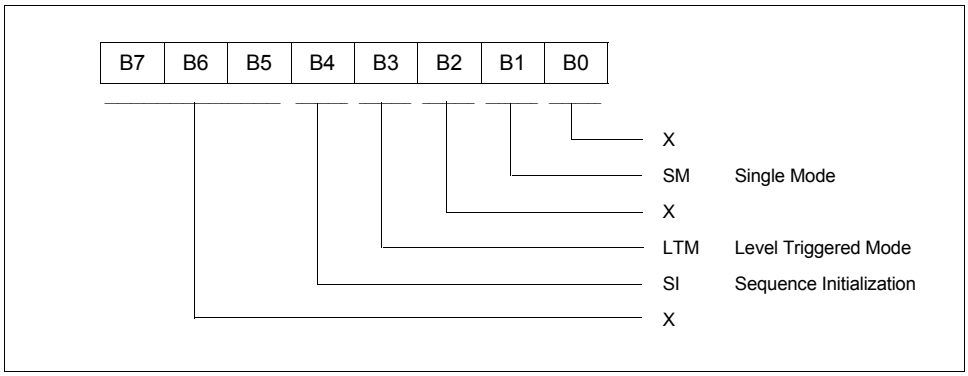
Initialization Command Words

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.

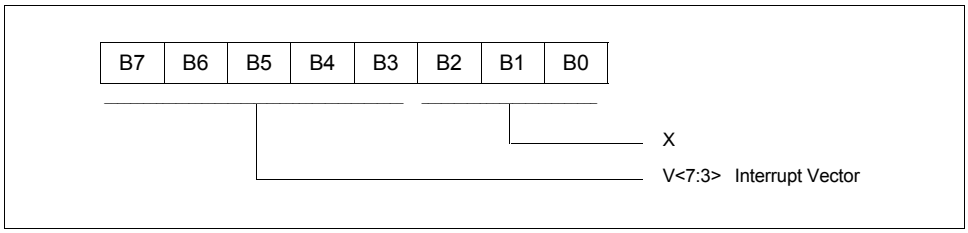
The next three I/Os write to address 021H (0A1H) will load ICW2-ICW4. See Figure 7-3 shown earlier for a flow chart of the initialization sequence. The initialization sequence can be terminated at any point (all four bytes must be written for the controller to be properly initialized) by writing to address 020H (0A0H) with a zero in data bit 4. Note that this causes OCW2 or OCW3 to be written.

Figure 7-7. ICW1—Address 020H (0A0H) Write Only



bits:	B0	X	This bit is ignored.
	B1	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 higest priority IR pending is from an INTC2 input. INTC1 and INTC2 must be programmed for Cascade mode for both devices to operate.
	B2	X	This bit is ignored.
	B3	LTM	Bit 3 selects the 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 occuring.
	B4	SI	Bit 4 indicates to the interrupt controller that an initialization sequence is starting and must be a one to write ICW1.
	B5-B7	X	These bits are ignored.

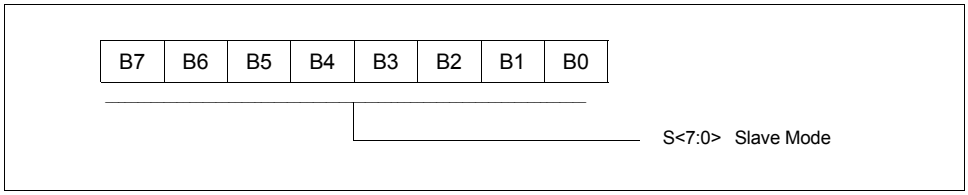
Figure 7-8. ICW2—Address 021H (0A1H) Write Only



bits: B0-B2 X These bits are ignored.

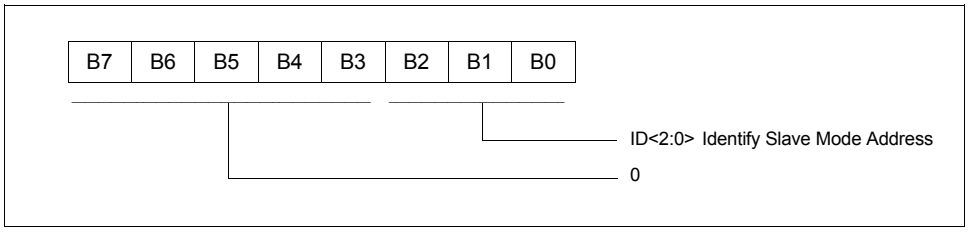
B3-B7 V<7:3> These bits are the upper five bits of the interrupt vector and are programmable by the CPU. The lower three bits of the vector are generated by the Priorty Resolver during INTA. INTC1 and INTC2 need not be programmed with the same value in ICW2.

Figure 7-9. ICW3 Format for INTC1—Address 021H Write Only



bits: B0-B7 S<7:0> Select which IR inputs have Slave mode controllers connected. ICW3 in INTC1 must be written with 04H for INTC2 to function.

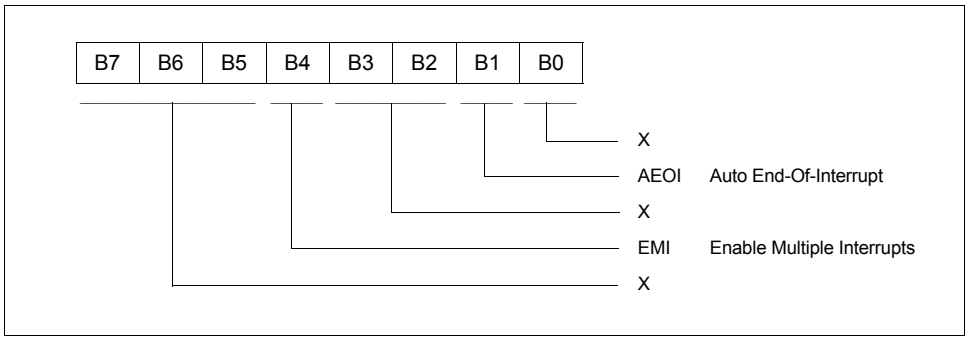
Figure 7-10. ICW3 Format for INTC2—Address 0A1H Write Only



bits: B0-B2 ID<2:0> 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 B3-B7 should be zero.

B3-B7 0 These bits are zeros.

Figure 7-11. ICW4—Address 021H (0A1H) Write Only

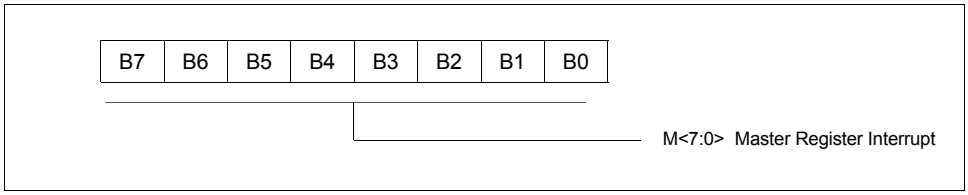


bits:	B0	X	This bit does not matter.
	B1	AEOI	Auto End-Of-Interrupt is enabled when ICW4 is written with a zero in both interrupts. The interrupt controller performs a non-specific EOI on the trailing edge of the second INTA cycle. Note that this function should not be used in a device with fully nested interrupts, unless the device is a cascade master.
	B2-B3	X	These bits are ignored.
	B4	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 nonspecific EOI command to INTC2 and check its In-Service Register for zero when exiting an interrupt service routine. If zero, a nonspecific EOI command should be sent to INTC1. If nonzero, no command is issued.
	B5-B7	X	These bits are ignored.

Operational Command Words

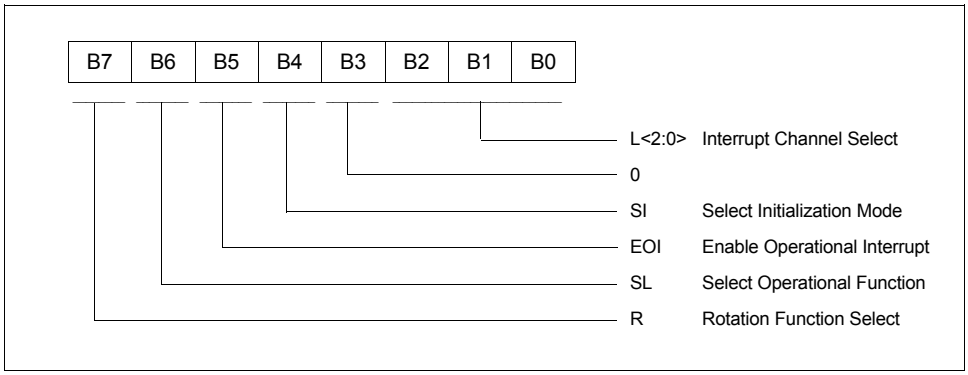
Operational Command Word One (OCW1) is located at address 021H (0A1H) and may be written any time the controller is in Initialization mode. Operational Command Words Two and Three (OCW2 and OCW3) are located at address 020H (0A0H). Writing to address 020H (0A0H) with a zero in bit 4 places the controller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

Figure 7-12. OCW—Address 021H (0A1H) Read/Write



bits: B0-B7 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 7-13. OCW2—Address 020H (0A0H) Write Only



bits: B0-B3 L<2:0> These three bits are internally decoded to select which interrupt channel is to be affected by the specific command. L<2:0> must be valid during three of the four specific cycles (see SL).

B3 0 This bit is zero.

B4 SI Writing a zero in this bit position takes the interrupt controller out of initialize mode and writes OCW2 or OCW3.

B5 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.

- Nonspecific 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 nonspecific EOI when R = 1 and SL = 0 and EOI = 1
- Rotate on specific EOI when R = 1 and SL = 1 and EOI = 1

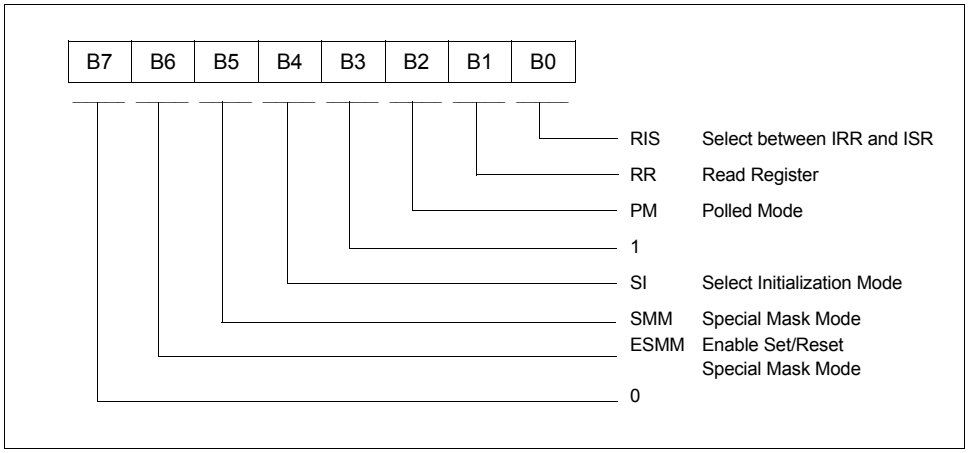
B6 SL This bit, in conjunction with R and EOI, selects operational function. Writing a one in this bit position causes a specific or immediate function to occur. All specific commands require L<2:0> to be valid, except no operation.

- 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.

B7 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. (Note: This function is disabled by writing a zero to all three bit positions.) When R = 1 and SL = 0 and EOI = 0.
- Rotate on nonspecific 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.

Figure 7-14. OCW3—Address 020H (0A0H) Write Only



bits:

B0	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.
B1	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 XD<7:0>. Asserting PM forces RR reset.
B2	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 bit 2-0. The IRR remains frozen until the read cycle is completed, at which time the PM bit is reset.
B3	1	This bit is set to one.
B4	SI	See SI shown in OCW2.
B5	SMM	If ESMM and SMM are both 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; a zero enables interrupts on the associated channel by causing the Priority Resolver to ignore the condition for the ISR.
B6	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.
B7	0	This bit is set to zero.

Section 8

DMA Controller

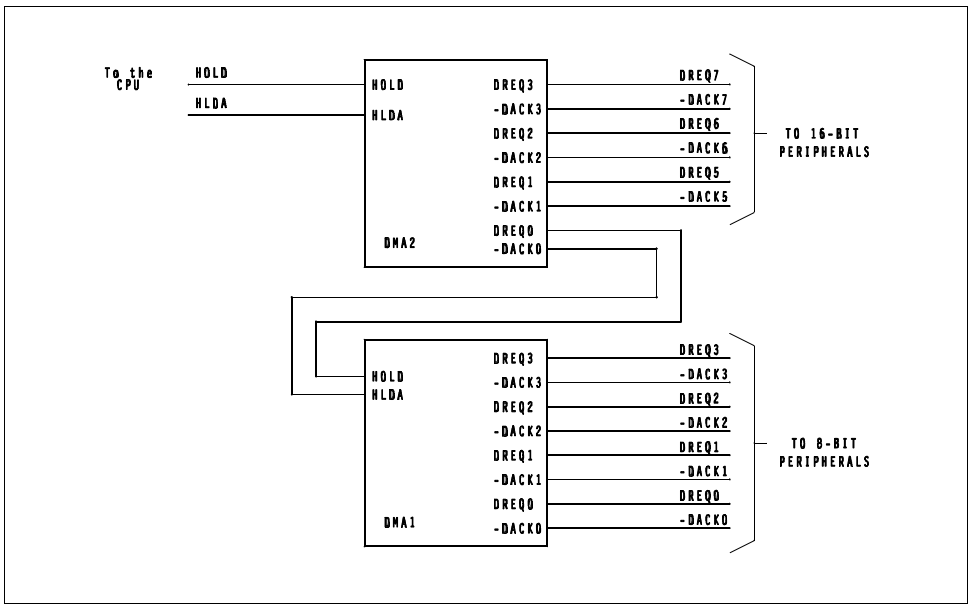
Memory Refresh and DMA functions are implemented within the 82C836 as independent bus masters, i.e., the refresh controller is separate from the 8237-compatible DMA controllers. The 82C836 arbitrates refresh and DMA functions with internal logic and gains control of the local bus via the HOLD/HLDA protocol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

The 82C836 contains two DMA controllers 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 interconnetion for the two DMA devices, thereby maintaining IBM PC/AT compatibility.

Figure 8-1 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 8-1. Cascaded DMA Controllers



The DMA clock, internal to the 82C836, controls the DMA transfer rate and timing. It is derived from the BUSCLK and is selectable as either BUSCLK/2 (default, AT-compatible) or BUSCLK. 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 1.19MHz (OSC1/12) clock. The refresh period is set by programming timer channel 1 of the 8254 timer/counter. Table 8-1 shows the usage of DMA levels on the I/O channel. The DMA requests are shown by priority, starting with the highest level.

Table 8-1. 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

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 reinitialize without programming. The following description of the DMA subsystem pertains to both DMA1 and DMA2 unless otherwise noted.

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 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 datasheet only because the 82C836 implements the same 8237-compatible functionality found in the PC/AT.

DMA Operations

During normal operation of the 82C836, the DMA subsystem is in either an Idle condition, a Program condition, or an Active condition. In 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, the 82C836 neither latches data from, or 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.

Idle Condition

When no device is requesting service, the DMA is in an Idle condition keeping 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, 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 to be read or written. 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 flip-flop toggles each time a read or write occurs to any of the word count or address registers in the DMA. The internal flip-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 the Idle condition 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 the compressed timing is selected. Once a ready condition is detected, the DMA enters S4, where both commands are deasserted. In

Burst mode and Demand mode, subsequent cycles begin in S2, unless the intermediate address bits require updating. In these subsequent cycles, the lower address bits are changed in S2. The DMA can be programmed on a channel-by-channel basis to operate in one of four modes. The four mode are described as follows:

- 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 deasserts HRQ and releases the bus after the transfer is complete. After HLDA is inactive, the 82C836 again asserts HRQ and executes another cycle in the same cycle; 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 auto-initialization option is enabled, the channel reinitialized 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 Tranfer 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 with limited buffering availability. The perpheral 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 register. Once DREQ is deasserted, 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 8-1 (shown earlier), portrays the cascade interconnection between the two levels of DMA devices. Note, 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 controller, begin with the device which is actually generating HRQ to the system (first level device) and then proceed to the second level device. 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. This 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 aserting -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 attempt to output the contents of the Temporary register on the XD0-7 data lines during the write portion of the cycle, so 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 the CPU can initialize 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 useful for diagnostics. In this type of transfer, the DMA operates as if it is performing a read or write transfer by generating HRQ, address, 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 contain a bit that causes the channel to reinitialize 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 uninitialized.

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 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 re-evaluated 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 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 8-2.

Table 8-2. DMA Page Register Function I/O Ports

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

I/O port 80H is normally used externally for diagnotic LEDs, and is updated by the AT-compatible BIOS during the 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 following 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 generated T/C, suspends operation on that channel, sets the appropriate Request Mask bit or Auto-Initialize, and continues.

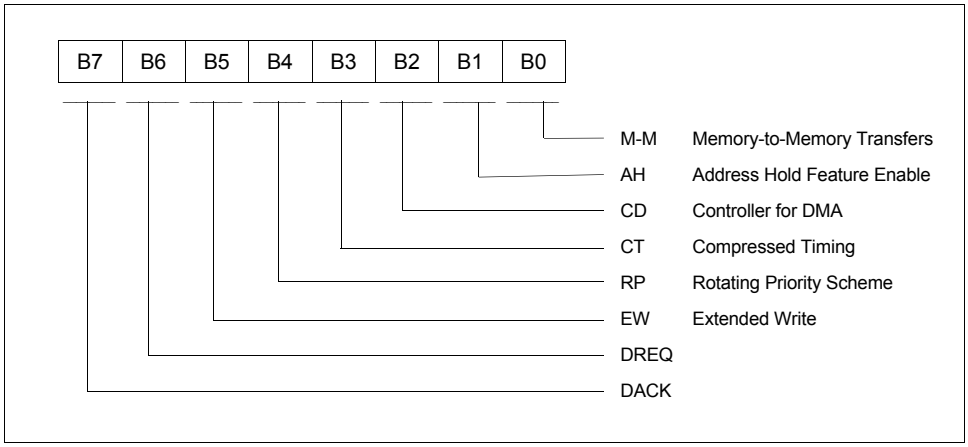
Base Word Count Register

This register preserves the initial value of the Current Word Count register. It is also a write-only register 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. See Figure 8-2 for the Command register format.

Figure 8-2. Command Register



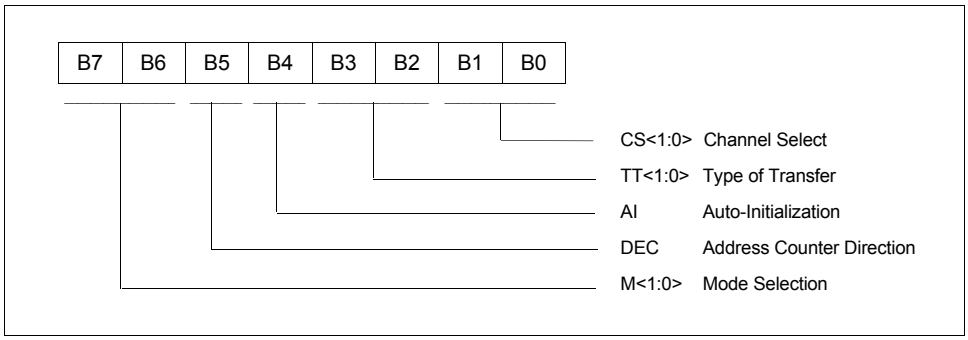
bits:	B0	M-M	A one in the bit 0 position enables Channel 0 and Channel 1 to be used for memory-to-memory transfers.
	B1	AH	Writing a one to bit 1 enables the address hold feature in Channel 0 when performing memory-to-memory transfers.
	B2	CD	Bit 2 is the master disable for the DMA controller. Writing 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 occuring.
	B3	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.
	B4	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.
	B5	EW	Extended Write is enabled by writing a one to bit 5, causing the write command to be asserted one DMA cycle earlier during a transfer. The read and write commands both begin in state S2 when enabled.
	B6	DREQ	DREQ active level is determined by bit 6. Writing a one in this bit position causes DREQ to become active low.
	B7	DACK	DACK active level is determined by bit 7. Programming a one in this bit position makes DACK an active high signal.

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 8-3.

Figure 8-3. Mode Register



bits: B0-B1 CS<1:0> 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

B2-B3 TT<1:0> 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

B4 AI The Auto-Initialization function is enabled by writing a one in bit 4 of the Mode register.

B5 DEC Determines direction of the address counter. A one in bit 5 decrements the address after each transfer.

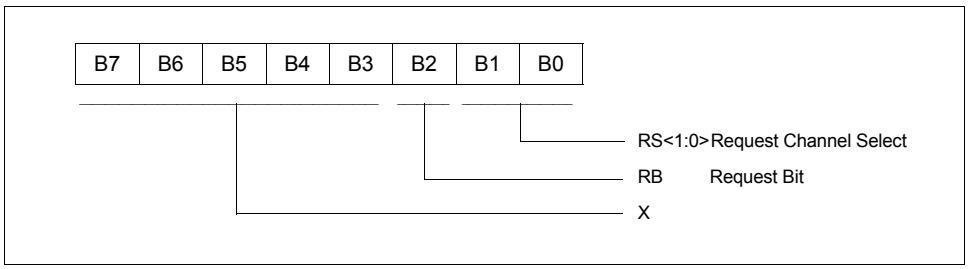
B6-B7 M<1:0> Mode selection for each channel is accomplished by bit 6 and 7. The type of modes are as follows:

- 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 = 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.

Figure 8-4. Request Register (Write Operaton)



bits: B0-B1 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.

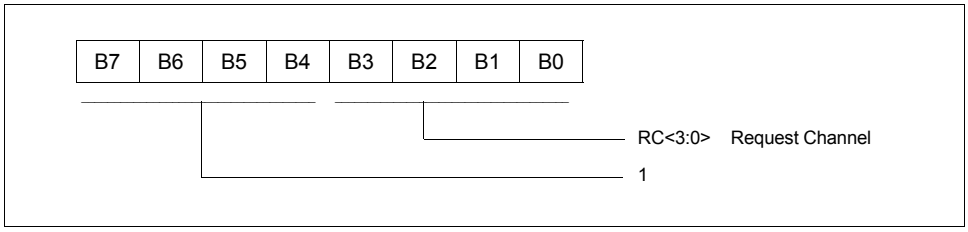
- Channel 0 Select when RS1 = 0 and RS0 = 0
- Channel 1 Select when RS1 = 0 and RS0 = 1
- Channel 2 Select when RS1 = 1 and RS0 = 0
- Channel 3 Select when RS1 = 1 and RS0 = 1

B2 RB The request bit is set by writing a one to bit 2. RS1-RS0 select which bit (channel) is to be manipulated.

B3-B7 X These bits are ignored.

The format for the Request register read operation is shown in Figure 8-5.

Figure 8-5. Request Register Read Format



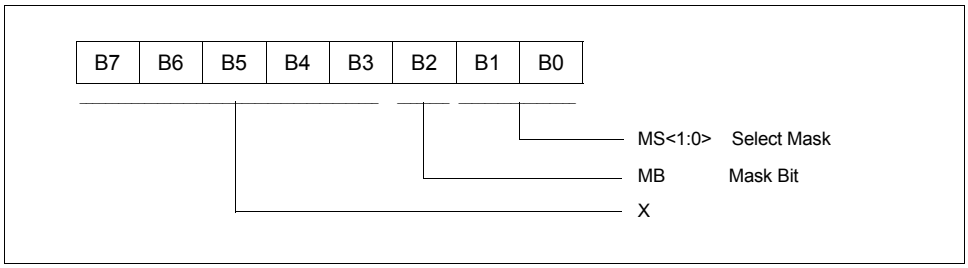
bits: B0-B3 RC<3:0> During a Request register read, the state of the request bit associated with each channel is returned in bits 0 through 3 of the byte. The bit position corresponds to the channel number.

B4-B7 1 These bits are read as ones.

Request Mask Register

The Request Mask register is a set of four bits that are used to inhibit external DMA requests from generating transfers 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 as follows:

Figure 8-6. Request Mask Register—Write Single Mask Bit



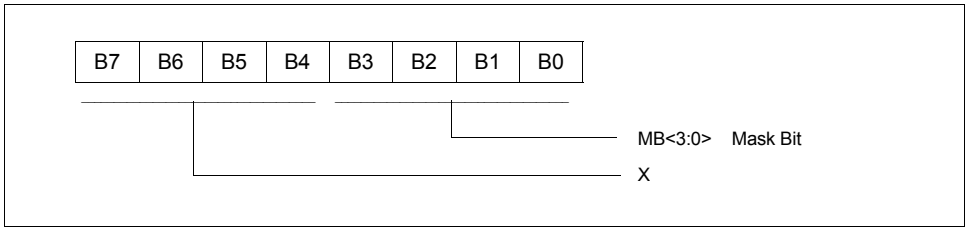
bits: B0-B1 MS<0:1> These two bits select the specific mask bit that is to be set or reset.

- Channel 0 Select when MS1 = 0 and MS0 = 0
- Channel 1 Select when MS1 = 0 and MS0 = 1
- Channel 2 Select when MS1 = 1 and MS0 = 0
- Channel 3 Select when MS1 = 1 and MS0 = 1

B2	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.
B3-B7	X	These bits are ignored.

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 8-7.

Figure 8-7. Request Mask Register—Write All Mask Bits

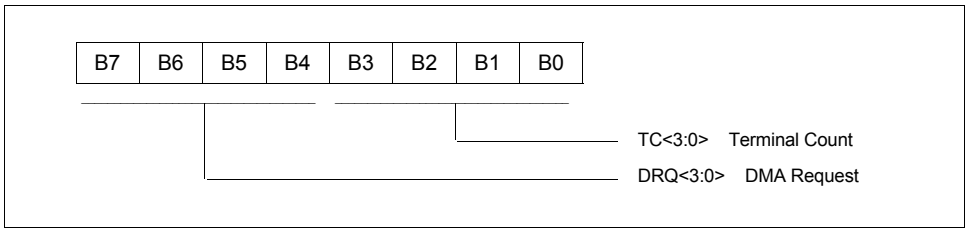


bits:	MB<3:0>	Each bit position in the field represents the mask bit of a channel. The mask bit number corresponds to the channel number associated with the mask bit. All four mask bits are set following a RESET or a Master Clear command. Individual channel mask bits are set as a result of terminal count being reached, if Auto-Initialize is disabled. The entire register can be cleared, enabling all four channels, be performing a Clear Mask register operation.
	B4-B7	These bits are ignored.

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 bits. The channel number corresponds to the bit position. Refer to Figure 8-8.

Figure 8-8. Status Register (Read Only)



bits: B0-B3 TC<3:0> Terminal count reached.

B4-B7 DRQ<3:0> DMA request pending

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 XD0-7. 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. Decriptions of the five special commands follows:

- 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.

Section 9

Address Maps

I/O Address Maps

The following Tables identify the I/O Ports used in the 82C836. For information regarding reserved locations, consult the IBM PC/AT Technical Reference Manual.

Table 9-1. DMA Controller I

I/O Address	Read/Write	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 9-2. Interrupt Controller I

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

Table 9-3. Internal Configuration Registers

I/O Address	Read/Write	Description
0022H	W/O	Internal Configuration Register Index (See Table 9-4.)
0023H	R/W	Internal Configuration Register Data

Table 9-4. Index Registers

ICR Index	Read/Write	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
60H	R/W	Laptop Features
61H	R/W	Fast Video Control
62H	R/W	Fast Video RAM Enable
63H	R/W	High Performance Refresh
64H	R/W	CAS Timing for DMA/Master

Table 9-5. Timer Counter Registers

I/O Address	Read/Write	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 9-6. Miscellaneous I/O Registers

I/O Address	Read/Write	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 9-7. DMA Page Registers

I/O Address	Read/Write	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 9-8. Interrupt Controller II

I/O Address	Read/Write	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 9-9. DMA Controller II

I/O Address	Read/Write	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 Requrest 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 Bits

Table 9-10. Coprocessor Registers

I/O Address	Read/Write	Description
00F0H	W	Clear Coprocessor Busy
00F1H	W	Not Used in the 82C836

Table 9-11. EMS Page Registers

I/O Address	Read/Write	Description
02x8H†	R/W	EMS Page Register, A21-A14
02x9H†	R/W	EMS Page Register, Enable and A23-A22
02xAH†	R/W	EMS Page Register Select

† x = 0 or 1 depending on ICR 4F Bit 0.

Memory Address Map

The following Table identifies the memory addresses used in the 82C836. For more detailed information about memory addressing, refer to the IBM PC/AT Technical Reference Manual.

Table 9-12. Memory Address Map

I/O Address	Read/Write	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

Section 10

Configuration Registers

Internal configuration registers (ICRs) are accessed by addressing an index register at port 22H and immediately writing (or reading) data to (or from) port 23H. The 82C836 contains ICRs at index values 01H, 40H - 4FH, and 60H - 64H. All ICRs are read/write unless 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.

Table 10-1. Index 01H—DMA Wait State Control

Bit	Name	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

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.

Table 10-2. Index 40H—82C836 Version, Read Only

Bit	Name	Description
7-4	Family Type	Identifies specific part in pin-compatible family. SCATsx denoted by 0001.
3-0	Device Revision Code	Identifies revision level of the 82C836 silicon, starting with 0000. The 82C836B has revision code 0100; the 82C836A has code 0011.

Table 10-3. Index 41H—Channel Environment

Bit	Name	Description
7	Fast Refresh	This bit controls an improvement in HOLD timing to reduce total HLDA time during refresh. Up to 1% performance improvement possible. 0 = Enable fast refresh (default) 1 = Disable; follow revision 1 refresh timing This bit is intended to remain zero except for test purposes.
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 read not "claimed" by early READY. 1 = Enable* 0 = Disable (default)
5	LBA Enable	Allows external devices to assert LBA (0WS pin) during the first T2 after T1 to prevent the 82C836 from starting a memory cycle. Enables Early Wait State for local memory read 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	Enable 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 1MB. 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)
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 = 104ns 01 = 210ns 10 = 280ns (default) 11 = 350ns

* Bits 5 and 6 must remain zero if CPU pipeline mode is enabled.

Index 42H and Index 43H are reserved.

Index 44H controls whether peripherals are accessed on the XD-bus or SD-bus. The SDIRH and SDIRL SD-bus controls are affected by bits in this register.

Table 10-4. Index 44H—Peripheral Control

Bit	Name	Description
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.
5	Game Port on the XD-bus	This bit specifies whether or not the game port is resident of 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 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 is 03F8H-03FFH. 0 = Serial port 1 on the SDS-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 X D 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. 10K pull-up resistors should be used on XD0-15 in systems that use XD-bus peripherals so read cycles from unused XD-bus addresses will receive 0FFH da ta.

Table 10-5. Index 45H—Miscellaneous Status, Read Only

Bit	Name	Description
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 is default).
6	GATEA20 From 8042	This bit indicates the current state of the GATEA20 input from the 8042 keyboard controller.
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.
2-0	Sense Lines 2-0	These lines (2-0) sense the power-up state of DACK lines (2-0), respectively (noninverted). These lines can be used for power-up 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 controls sleep mode and PROCCLK frequency for power conservation.

Table 10-6. Index 46H—Power Management

Bit	Name	Description
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.

Table 10-6. Index 46H—Power Management (continued)

Bit	Name	Description
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	High Shadow RAM Enable	This bit causes memory accesses in the "High ROM" area (see ICR 4EH, bit 4) to be remapped into shadow RAM at the address obtained by forcing A23-20 to 0H. This feature is useful in laptops in which shadow RAM is used for BIOS, and it is desired to have CPU instruction fetches following CPU reset go to 0FFFF0H in the shadowed BIOS area instead of going to ROM at FFFFF0H (CPU reset without XRST is generally used to return to real mode from protected mode). Do not set this bit unless shadow RAM has been enabled in the translated address range. SCATsx will not perform a ROM or AT bus access in response to a translated high ROM read or write, regardless of other ICR bit settings. Shadow RAM write protection, if enabled, applies to translated high ROM writes as well as other types of writes. EMS has no effect on the translation, even if the translated address happens to fall within the internal EMS page frame; the high ROM translation bypasses the EMS mechanism. An Early Wait State always occurs on translated high ROM accesses. Master accesses to high ROM are affected by this bit in the same way as CPU accesses. 0 = No high ROM translation (default) 1 = Enable high ROM translation to shadow RAM This bit is always cleared by a logic low level on PWRGOOD, even if stand-by mode is enabled (see ICR 60H). Consequently, this bit cannot affect high ROM accesses immediately following system reset via PWRGOOD.
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
1-0	Sleep Freq	These bits select the frequency of the PROCCLK signal to the CPU when in sleep mode. 00 = Stopped (default). See ICR 64H bits 4-3. 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 Register 47H is reserved.

Index Register 48H determines the address ranges included in the -ROMCS output. The power-on default address range is 0F0000H-0FFFFFH. The range from FC0000H to FFFFFFH may also be enabled, depending on ICR 4EH bit 4 and ICR 46H bit 5. 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, accesses to ROM will always be 16-bits wide (two ROMs are 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 as follows:

1 = Enable ROM
0 = Disable ROM

Do not enable ROM and shadow RAM in the same range at the same time.

Table 10-7. Index 48H—ROM Enable

Bit	Description
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 is used to write protect RAM in the address ranges specified as follows:

1 = Read only
0 = Read/Write

Default for all bits is 0.

Table 10-8. Index 49H—RAM Write Protect

Bit	Description
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 controls whether shadow RAM is enabled or disabled in the 16KB address ranges specified as follows:

1 = Enable RAM
0 = Disable RAM

Default for all bits is 0.

Table 10-9. Index 4AH—Shadow RAM Enable 1

Bit	Description
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 Register 4BH controls whether shadow RAM is enabled or disabled in the 16KB address ranges specified as follows:

1 = Enable RAM
0 = Disable RAM

Default for all bits is 0.

Table 10-10. Index 4BH—Shadow RAM Enable 2

Bit	Description
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 Register 4CH controls whether shadow RAM is enabled or disabled in the 16KB address ranges specified as follows:

1 = Enable RAM
0 = Disable RAM

The default for all bits is 0.

Table 10-11. Index 4CH—Shadow RAM Enable 3

Bit	Description
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 sh adow RAM when using memory configuration #03H (640K + 384K).

Table 10-12. 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 enable (default) 1 = RAS timeout disabled
5	CAS Wait State for Read	Allows the CAS Wait State (bit 7 above) to be inserted on local memory reads, but not on writes. 0 = Wait State controlled by bit 7 above (default) 1 = CAS Extend Wait State Enabled on memory read, regardless of bit 7; wait state for write still controlled by bit 7.
4-0	CFG4-0	These bits specify the DRAM configuration as described in Section 5, System Interface, subsection titled DRAM Interface. The default value after reset is 00001 (512KB total DRAM).

Table 10-13. Index 4EH—Extended Memory Boundary

Bit	Name	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	MRA Enable	Enables the "Multiple RAS Active " mode for higher system performance. 0 = Single RAS active mode (default) 1 = Multi RAS active mode. Also enables multiplexed DREQs and encoded DACKs
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 = Disabled 040000H-09FFFFH DRAM range.
4	High ROM Enable	This bit should be set to one for PC/AT compatibility (see also ICR 46H bit 5, high shadow RAM enable). 0 = Enable FC0000H and above (256K) as ROM (default) 1 = Enable FE0000H and above (128K) as ROM.

Table 10-13. Index 4EH—Extended Memory Boundary (continued)

Bit	Name	Description
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 3	2	1	0	Extended Memory Boundary
0	0	0	0	No limit (default); entire DRAM accessible
0	0	0	1	1MB (i.e., no extended memory)
0	0	1	0	1.25MB
0	0	1	1	1.5MB
0	1	0	0	2MB
0	1	0	1	3MB
0	1	1	0	4MB
0	1	1	1	5MB
1	0	0	0	7MB
1	0	0	1	8MB
1	0	1	0	9MB
1	0	1	1	10MB
1	1	0	0	11MB
1	1	0	1	12MB
1	1	1	0	13MB
1	1	1	1	15MB

Table 10-14. Index 4FH—EMS Control Register

Bit	Name	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.
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.

The purpose of Index 60H is to allow DRAM refresh to continue during power-down. Consequently, bits 7-3 are preserved during power-down instead of cleared by PWRGOOD. To initialize this ICR to its default value, a logic low level on the Power Sense (MFP5) input is used while PWRGOOD is low.

Table 10-15. Index 60H—Laptop Features

Bit	Name	Description
7	CAS-Before-RAS Refresh Enable	0 = RAS only refresh (default) 1 = CAS before RAS refresh
6	Stand-by Refresh Enable (also known as Stand-by mode)	This mode allows local DRAM refreshing to continue during power-down. No refresh address is generated during power-down, so CAS-before-RAS refresh should be enabled and DRAMs that support CAS-before-RAS refresh should be used. Stand-by refresh relies on the 32KHz input clock (MFP4 pin), which isn't available with external RTC (MFP4 becomes IRQ8). Consequently, stand-by refresh cannot be used with external RTC. 0 = No refresh during power-down (default) 1 = Refresh continues during power-down This bit, when set to one, has the following additional effects: ICR 4DH bits 4-0 (memory configuration) are preserved during power-down. ICR 4EH bit 6 (MRA enable) is preserved during power-down. DACK signals are no longer sampled following PWRGOOD rise. PWRGOOD fall (powering down) takes effect only after stand-by refresh is operational to maintain refresh continuity. For assured recovery from potential lockup conditions following initial power turn-on or software malfunction, the 82C836B contains a built-in 32KHz based timeout and external RTC interlock to clear this bit automatically under abnormal conditions. PWRGOOD can then take effect in the normal manner without standby mode
5-4	Stand-by Refresh Interval	Allows reduced power consumption for stand-by refresh during power-down via reduced DRAM refresh rate if compatible DRAMs are used. The reduced refresh rate takes effect only during power-down and only if stand-by refresh is enabled. 00 = 15µs (default; half cycle of 32.768KHz input clock) 01 = 61µs (2 cycles of 32.768KHz input clock) 10 = 122µs (4 cycles of 32.768KHz input clock) 11 = 244µs (8 cycles of 32.768KHz input clock)
3	Refresh Time Base (Read Only)	Indicates whether or not the system has switched from stand-by refresh to normal. If stand-by mode is enabled, the system switches from 14.3MHz refresh timing to 32KHz timing during power-down. When power comes back up, the BIOS must reprogram Timer Channel 1 (see Section 6, Programmable Interval Timer) to get refresh request running again. The 82C836 then switches back to 14.3MHz based refresh. BIOS should not try to access DRAM until the switch back to 14.3MHz timing has occurred. The switching of the refresh time base is done in a double-synchronized manner to prevent any lost or aborted refresh cycles. 0 = 14.3MHz based refresh 1 = 32KHz based refresh
2-0	—	Reserved. Write as 0.

Index Register 61H controls the following special modes for video access:

- Fast 8-bit I/O access —cycle timing equivalent to 16-bit cycles, but using only the low byte of the data bus for data transfer.
- Fast 8-bit memory access —cycle timing equivalent to 16-bit cycles, but using only the low byte of the data bus for data transfer.
- Forced -MEMCS16 for 16-bit memory access. 82C836B asserts -MEMCS16 and performs a 16-bit cycle, eliminating the need for equivalent logic in the video subsystem.

In all cases, IOCHRDY can be used to insert bus wait states.

Table 10-16. Index 61H—Fast Video Control

Bit	Name	Description
7-6	—	Reserved. Write as 0.
5	I/O 3D0-3DFH Enable	0 = Normal operation (default) 1 = Fast 8-bit I/O
4	I/O 3C0-3CFH Enable	0 = Normal operation (default) 1 = Fast 8-bit I/O
3	I/O 3B0-3BBH Enable	The parallel port I/O address range, 3BC-3BFH, is not affected by this bit. 0 = Normal operation (default) 1 = Fast 8-bit I/O
2	Fast 8-bit I/O Enable	Enables the fast 8-bit I/O mode for I/O accesses in the ranges selected via bits 3, 4, and 5 above. Bits 3, 4, and 5 have no effect if bit 2 is zero. 0 = Normal operation (default) 1 = Fast 8-bit I/O
1	Force MEMCS16.	0 = Normal operation (default) 1 = 82C836B asserts -MEMCS16 and performs 16-bit cycle timing and protocol for all memory accesses in the ranges selected via ICR 62H. Do not set both bit 1 and bit 0 to one at the same time.
0	Fast 8-bit Memory Enable	Enables the fast 8-bit mode for memory accesses in the ranges selected via ICR 62H. 0 = Normal operation (default) 1 = Fast 8-bit memory access.

Index 62H enables the memory address ranges selected below for either fast 8-bit memory timing or forced MEMCS16 mode, depending on the bit settings in ICR 61H.

0 = Range not selected (default)
1 = Range selected

Table 10-17. Index 62H—Fast Video RAM Enable

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

Table 10-18. Index 63H—High Performance Refresh

Bit	Name	Description
7	Hidden Refresh Enable	Causes system-initiated refresh cycles to be performed by inserting CPU wait states as needed instead of using HOLD/HLDA protocol. In a cache-based system, this allows the CPU to continue performing cache read hit cycles simultaneously with a refresh cycle in local memory and on the AT bus. If the CPU tries to access local memory or the AT bus while a hidden refresh cycle is in progress, the CPU is delayed via wait states until the refresh is completed. 0 = Normal HOLD/HLDA refresh (default) 1 = Hidden refresh
6	Refresh on idle	This bit causes hidden refresh cycles (if enabled) to be initiated only following a CPU Idle state (no -ADS since last -READY, and HLDA inactive). Depending on system workloads, waiting for CPU Idle states may slightly increase the probability of successfully hiding refresh cycles during cache read hits. 0 = Hidden refresh doesn't wait for CPU Idle (default) 1 = Hidden refresh waits for CPU Idle (or timeout) If no CPU Idle states occur within the normal time limit between refresh cycles, then refresh is initiated after the current CPU cycle terminates (-READY), without waiting for a CPU Idle state. This insures DRAM refresh requirements will be met even if the CPU is unusually busy.
5	AT Refresh Disable	Allows AT bus refresh to be disabled during hidden refresh. The total time needed to perform a hidden refresh is greatly reduced in this mode, since local memory requires far less time for a refresh than the AT bus. The result is a significant performance benefit in cache-based systems if AT bus refresh is not needed (non-cache systems will not benefit as much beause of the performance effect of running exclusively in nonpipeline mode). 0 = AT refresh enabled during hidden refresh (default) 1 = AT refresh disabled during hidden (non-HLDA) refresh -REF and -MEMR are not generated during hidden refresh if AT refresh is disabled. However, any refreshes performed while HLDA is active for DMA or Master cycles are still performed in the normal manner, including refresh on the AT bus.

Table 10-18. Index 63H—High Performance Refresh (continued)

Bit	Name	Description
4-3	Hidden Refresh Timing	Adjusts hidden refresh timing according to CPU speed. Refresh timing during HLDA is not affected. 00 = Reserved 01 = 16MHz CPU speed (default) 10 = 20MHz or 25MHz CPU speed 11 = Reserved
2	Refresh Delay	Enables extra delay from one refresh to the next when an add-on card bus master holds -REF active long enough to cause multiple back-to-back refresh cycles. As in a PC/AT, refresh address remains the same for each subsequent refresh cycle occurring while -REF remains continuously asserted. 0 = One microsecond delay from -XMEMR active to next -XMEMR active for multiple refresh in Master mode (default) 1 = No delay added; same timing as in revision 1. Normally, this bit should remain zero to prevent subsequent refreshes from starting prematurely when a Master keeps -REF asserted longer than necessary for one refresh cycle (or when -REF has unusually long rise time due to insufficient pull-up resistance).
1	—	Reserved. Write as 0.
0	Write Through Disable	In cache-based SCATsx systems, memory writes normally go directly to DRAM without any delay as compared to non-cache systems; local memory writes cannot be "claimed" by the cache controller. This bit changes memory write protocol so that memory writes can be claimed by external logic i.e., no write cycle performed in memory. In a cache-based system, this feature generally will be useful only for diagnostic purposes. This features may also be useful in certain high-performance, non-cache systems that rely on dynamic cycle claiming on writes as well as reads. 0 = Normal write-through protocol (default); no early wait state 1 = Write-through disable; writes can be externally claimed; early wait state inserted. Only local memory writes are affected by this bit. Local memory reads and all AT bus memory or I/O accesses are already subject to cycle claiming (ICR 41H bits 5 and 6). If this bit is set to one, ICR 41H bit 5 or 6 (or both) should also be set to one.

For timing reasons, hidden refresh mode of Index 63H requires that the CPU operate exclusively in nonpipelined mode; the -NA signal to the CPU must be held high continuously. The -NA output from SCATsx is not affected (if SCATsx is programmed for -STCYC mode on the NA/STCYC pin, the -STCYC pulse is delayed until the hidden refresh is complete).

The CPU address bus remains available to the CPU during hidden refresh, so AT bus refresh must either be turned off (see bit 5 in ICR 63H) or the AT bus refresh address must be provided by other external logic. The 82C835A cache controller is designed to provide the AT bus refresh address during hidden refresh. SCATsx continues to generate -REF and -XMEMR during hidden refresh if AT bus refresh is enabled, and the 82C835 can be programmed to use these signals to generate the AT bus refresh address.

Timing considerations also require all externally claimed cycles to have zero wait states. For cycles claimed via early READY, as with the 82C835 cache controller, this will always be the case automatically. Cycle claiming via -LBA is effectively unusable when hidden refresh is enabled unless -READY is asserted on the same T-state as -LBA.

If HLDA is already active for a DMA or Master cycle, normal refresh cycles occur as needed without waiting for HLDA to end. The refresh requests internally sent to the hidden refresh logic are reduced accordingly. Also, during sleep mode, normal HOLD/HLDA refresh is performed instead of hidden refresh.

Table 10-19. Index 64H—CAS Timing for DMA/Master

Bit	Name	Description
7	Respond During Reset	Determines whether or not the 82C836B will perform a normal cycle in response to CPU activity occurring while CPU reset is active. The CPU can initiate one or more bus cycles after CPURST has been asserted. The 82C836B can either ignore these cycles or respond normally as if no CPU reset is in progress. 0 = Ignore CPU cycles starting after CPURST goes active (default) 1 = Respond as usual to CPU cycles starting after CPURST goes active.
6-5	—	Reserved. Write as 0.
4-3	Stopped Clock Sleep Phase	These bits select the exact phase within a T-state in which PROCCLK stops during sleep mode when a sleep frequency of zero has been selected. This will allow maximum flexibility with static CPUs. Zero-frequency sleep mode should be used only with a static CPU. 00 = PROCCLK low, 1st half of T-state (default) 01 = PROCCLK low, 2nd half of T-state 10 = PROCCLK high, 1st half of T-state 11 = PROCCLK high, 2nd half of T-state
2-0	CAS delay for DMA/Master Write	These bits determine when CAS is asserted during DMA or Master write cycles to local memory. CAS assertion during reads is not affected. The minimum delay from valid column address to assertion of CAS is one PROCCLK. Up to four additional PROCCLKs can be added, for a maximum of five PROCCLKs total. 000 = No added delay 001 = 1 PROCCLK delay added 010 = 2 PROCCLKs delay added 011 = 3 PROCCLKs delay added (default) 100 = 4 PROCCLKs delay added 101 = Reserved 110 = Reserved 111 = Reserved For 25MHz operation, 011 should be optimum. For 20MHz operation, 010 should be optimum. For 16MHz operation, 001 should be optimum. The timing of CAS is most critical relative to valid write data and parity.

The only known case where Index Register 64H bit 7 makes any difference in system operation is when running certain diagnostic programs that use CPURST on a system having an 82C835 cache controller, but with cache operation disabled (CPU reset without XRST is often used to return the CPU to real mode from protected mode). The 82C835 needs to see READY in response to every ADS, including ADS pulses occuring during CPU reset. Setting this bit to one allows the 82C836B to finish every CPU cycle started during CPU reset, including assertion of READY. This bit might need to be zero if software attempts to perform a cache read miss (cache enabled) or memory write operation during CPU reset. Normally, however, software will halt the CPU or execute a "jump to self" loop (cache read hits) while waiting for CPU reset to take effect. Thus, it should be safe always to set bit 7 to one.

EMS Page Registers

This section lists the bit assignments for the three I/O ports used to access the four EMS page registers.

I/O port 208H (or 218H) accesses bits 21 through 14 of the address of the 16KB target page to be mapped into the corresponding page window in the 64KB remappable address range. The page register accessed by this port is selected by port 20AH (or 21AH).

Table 10-20. Port 208H—Target Page Address

Bit	Description	Bit	Description
7	A21 of target page address	3	A17 of target page address
6	A20 of target page address	2	A16 of target page address
5	A19 of target page address	1	A15 of target page address
4	A18 of target page address	0	A14 of target page address

Port 209H (0r 219H) accesses bits 23 and 22 of the address of the 16KB target page mapped into the selected page window of the 64KB remappable address range. The page register accessed by this port is selected by port 20AH (or 21AH).

Table 10-21. Port 209H—Page Enable

Bit	Name	Description
7	Page Enable	"Page disable" means that the page window in the 64KB remappable address range is not remapped but is treated as ordinary non-EMS memory. 1 = page Enable 0 = page Disable.
6-2	—	Reserved.
1	—	A23 of target page address
0	—	A22 of target page address

Table 10-22. Port 20AH—EMS Page Select

Bit	Name	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 64KB remappable address range follow in ascending order.

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

Table 10-23. Port 61H—Control Port/Status

Bit	Name	Description
7	Parity Error (read only)	This bit (if 1) indicates that a parity error has been detected during a local memory read. Causes NMI if Port 70H bit 7 is 0.
6	IOCHCK (read only)	This bit (if 1) indicates an I/O channel check has occured (usually a parity error) on the system I/O channel. Causes NMI if port 70H bit 7 is 0.
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 enable (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). 1 = Output is enabled, i.e., speaker tone on. 0 = Output is forced low (default)
0	TMR 2 Gate (read/write)	This bit gates the clock input for timer channel 2 (speaker tone). 1 = Channel 2 timer clock enabled. 0 = Channel 2 timer clock disabled (default).

Port 70H 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.

Table 10-24. Port 70H—RTC/CMOS Index and NMI Mask

Bit	Name	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 is 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.

Port 71H 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 earlier.

Table 10-25. Port 71H—RTC/CMOS Data

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

Port 92H 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.

Table 10-26. Port 92H—System Control

Bit	Name	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.
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 occured. The main reason for resetting the CPU alone is to return from protected mode in an 80286-compatible manner.

Section 11

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

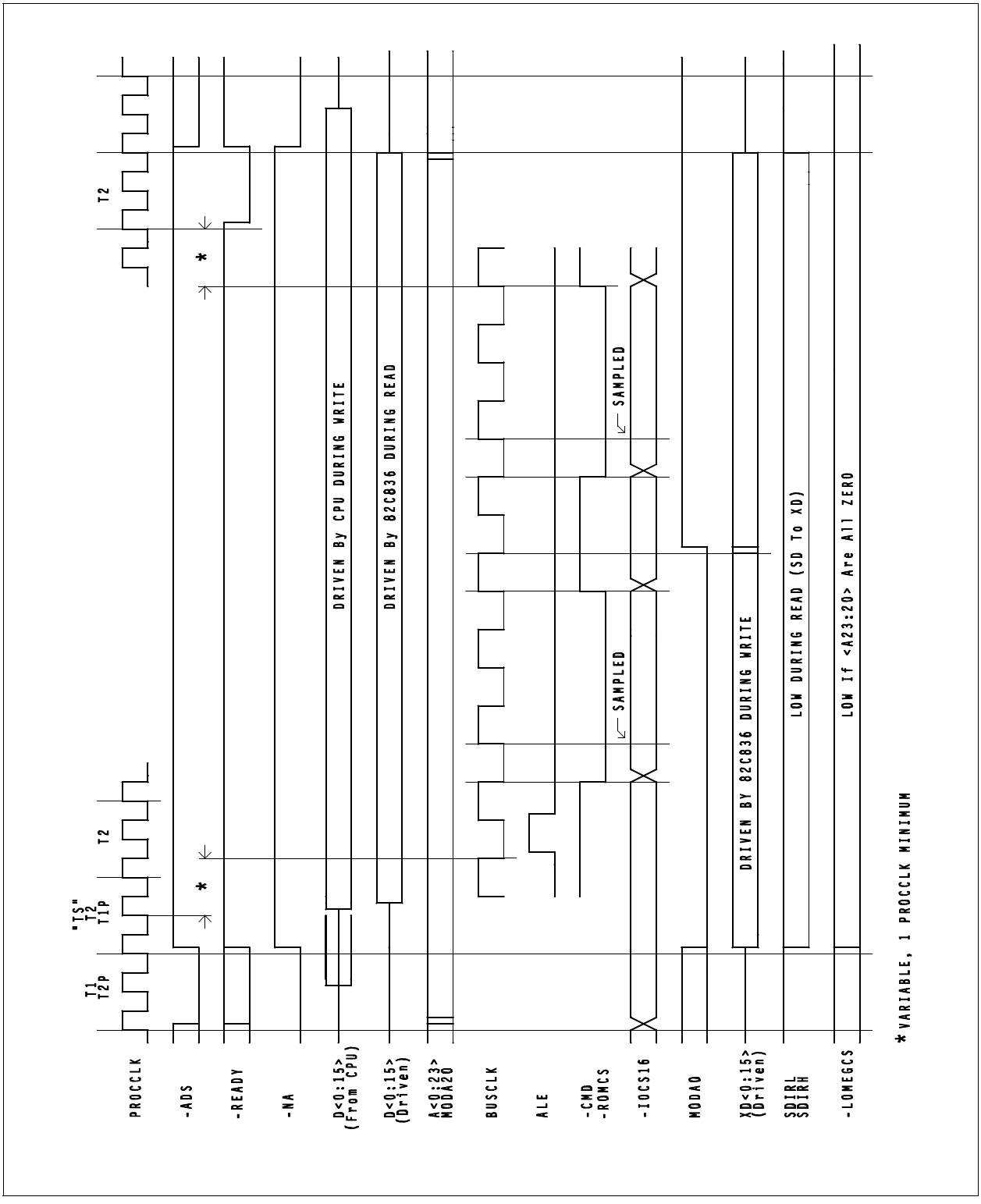
Figure 11-1 shows a bus convert read or write. -CMD refers to the following possibilities:

- -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 1MB 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 assererted.

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).

Figure 11-1. CPU Access to AT-Bus



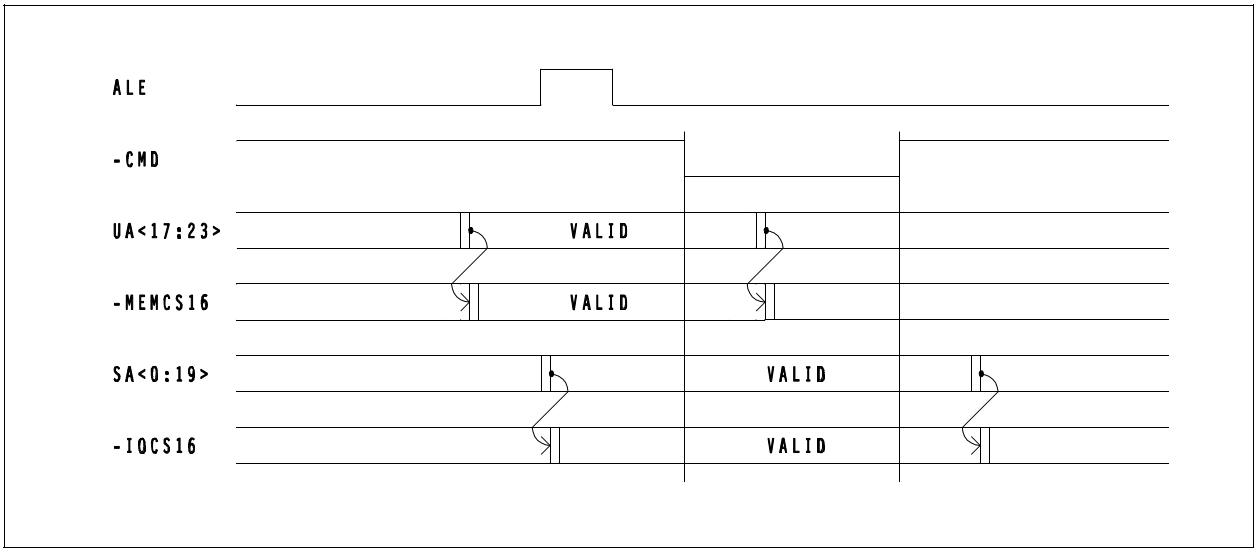
The delay from ALE to the start of command, and the width of the command, depend on the type of cycle, as shown in the following Table.

Table 11-1. Delays From ALE to Start Command in BUSCLK Cycles

Cycle Type	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 11-1 depicts 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. 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 11-2.

Figure 11-2. -MEMCS16 and -IOCS16 Timing



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 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 as follows:

- 1.0 cycle of PROCCLK if BUSCLK = CXIN/4
- 1.5 cycles of PROCCLK if BUSCLK = CXIN/5
- 2.0 cycles of PROCCLK if BUSCLK = CXIN/6

The minimum delay from command rise to -READY fall is one cycle of PROCCLK in all cases. -READY, in turn, is always synchronized to the last T2 or T2P of the bus cycle.

Bus conversion 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.

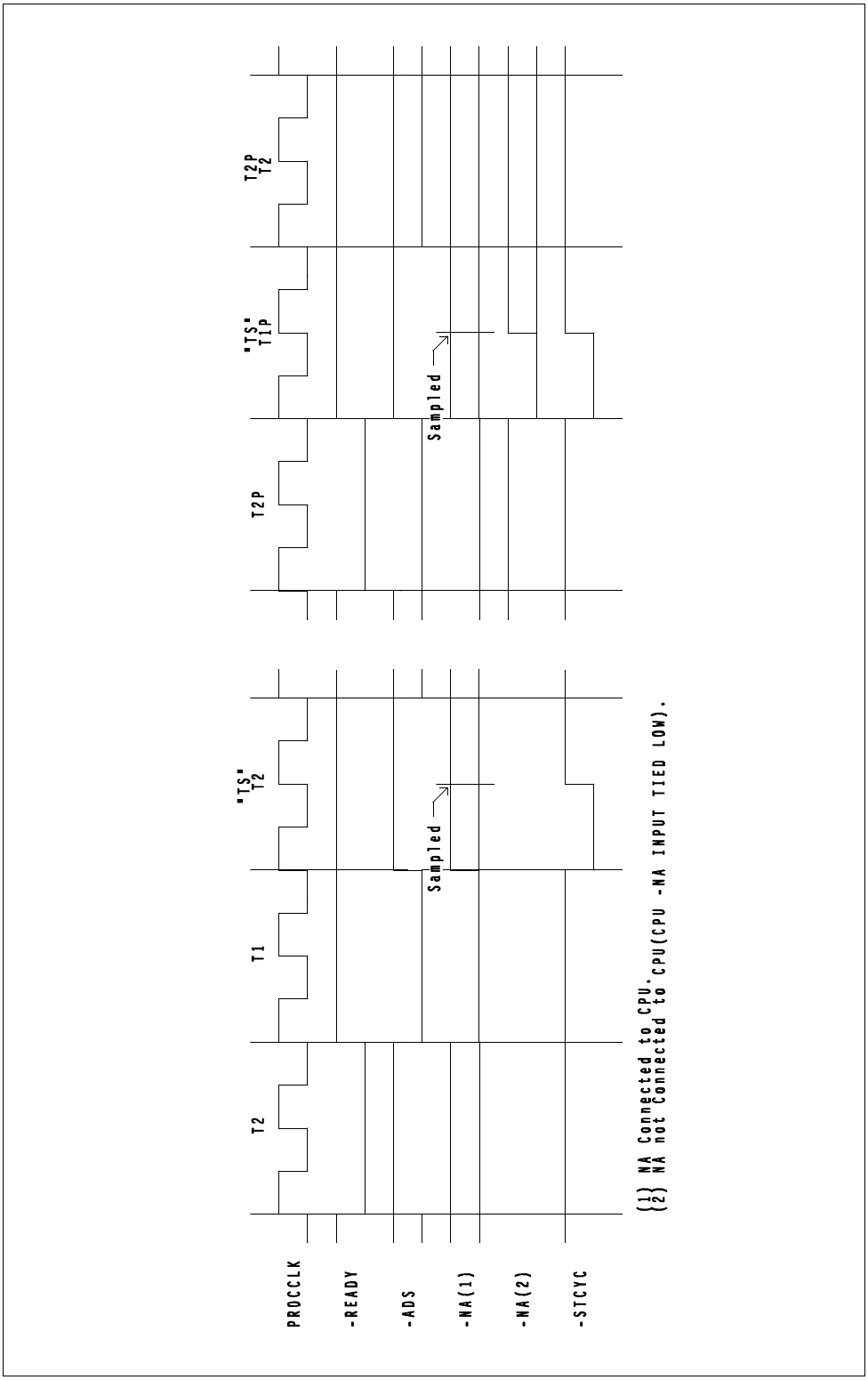
Normal bus conversion occurs on any CPU-generated memory or I/O read or write in which the CPU is requesting 16 bits of data to or from an 8-bit resource at an even memory or I/O address. (If the address is odd, the CPU itself automatically performs two separate byte transfers for a word operand.)

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 cycle is always nonpipelined, as shown in Figure 11-3, 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 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 11-3.
- 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).

Figure 11-3. -NA/-STCYC Timing



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. 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 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 unit the end of TS insead 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.

-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 resouce, 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 causing the 82C836 to reset the CPU include:

- 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).

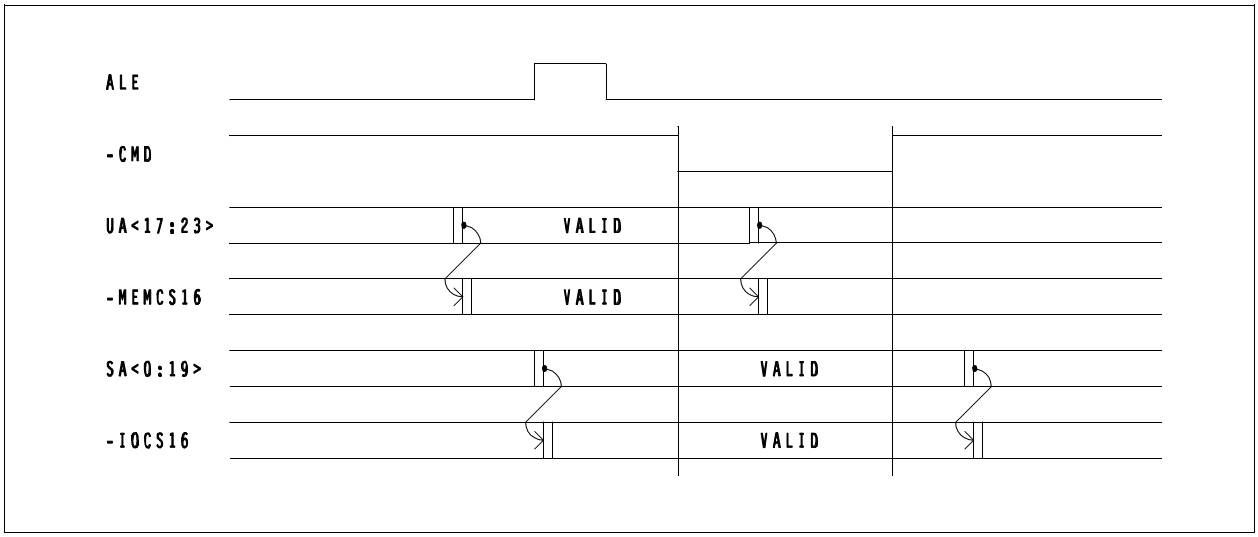
Memory write operations to an area programmed as EPROM (-ROMCS) result in write cycles on the AT bus. -XMEMW is generated. -ROMCS remains inactive (high) during write operations.

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

-MEMCS16 and -IOCS16 Timing

Figure 11-4 describes 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. Similary, -IOCS16 will be an unlatched decode of the latched address bits, SA0-19 (possibly gated with the I/O command signal).

Figure 11-4. -MEMCS16 and -IOCS16 Timing



When an attached memory resouce 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 by different AT bus add-on cards.

As previously shown in Figure 11-1, 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 transtion 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 samples -IOCS16 on the first rising edge of BUSCLK after I/O command goes active (low), -IOCS16 must remain valid until the end of the I/O command.

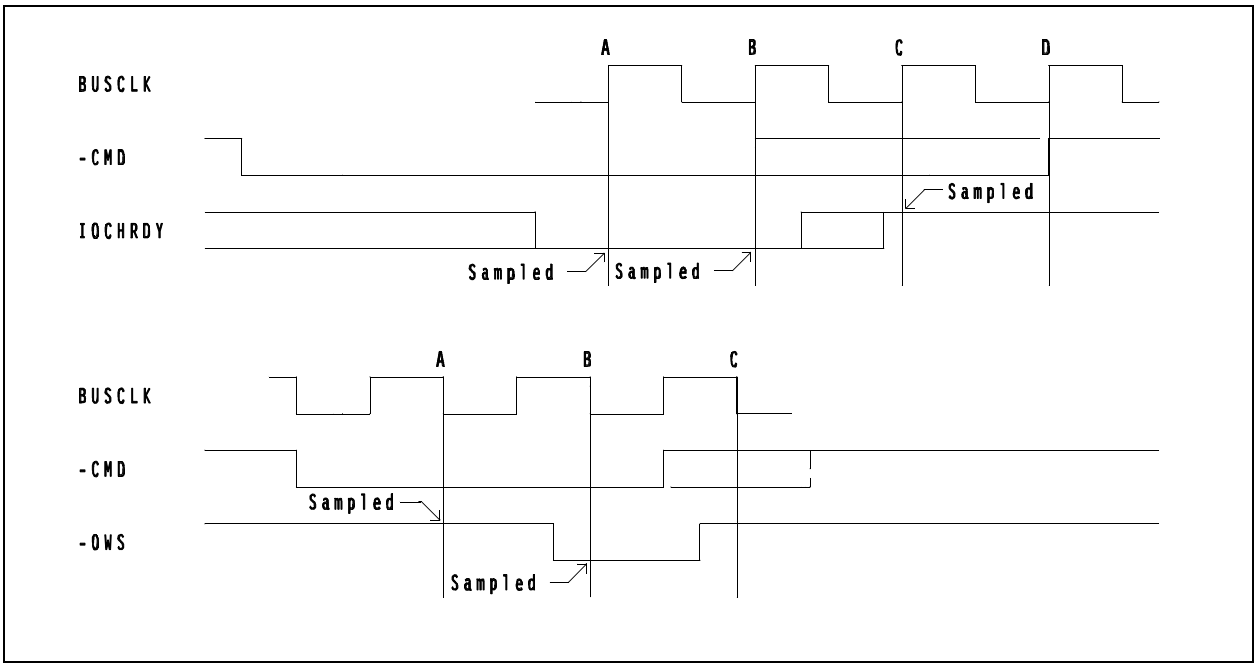
Although -MEMCS16 and -IOCS16 normally are asserted by 16-bit resources on the AT bus, the 82C836 can assert either or both of these signals in the following cases:

- 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 either 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 OWS

Figure 11-5 shows the effect 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 11-5. IOCHRDY and -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 of 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 11-3 shown earlier identifies 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 nonpipelined 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 as 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 preceeding 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 T1 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 nonpipelined 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 11-6 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. Nonpipelined cycles follow the same timing except for an added T1 state at the beginning.

The -CAS active time is three PROCCLK cycles for read, and 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 the read data will remain valid long enough for the cache data RAM to capture it (cache read miss cycle). See the section titled Early READY and LBA Modes for other cache related timing.

-MWE is normally low so 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 nonpipelined, the falling edge of -MWE is delayed until the middle of the next T-state.

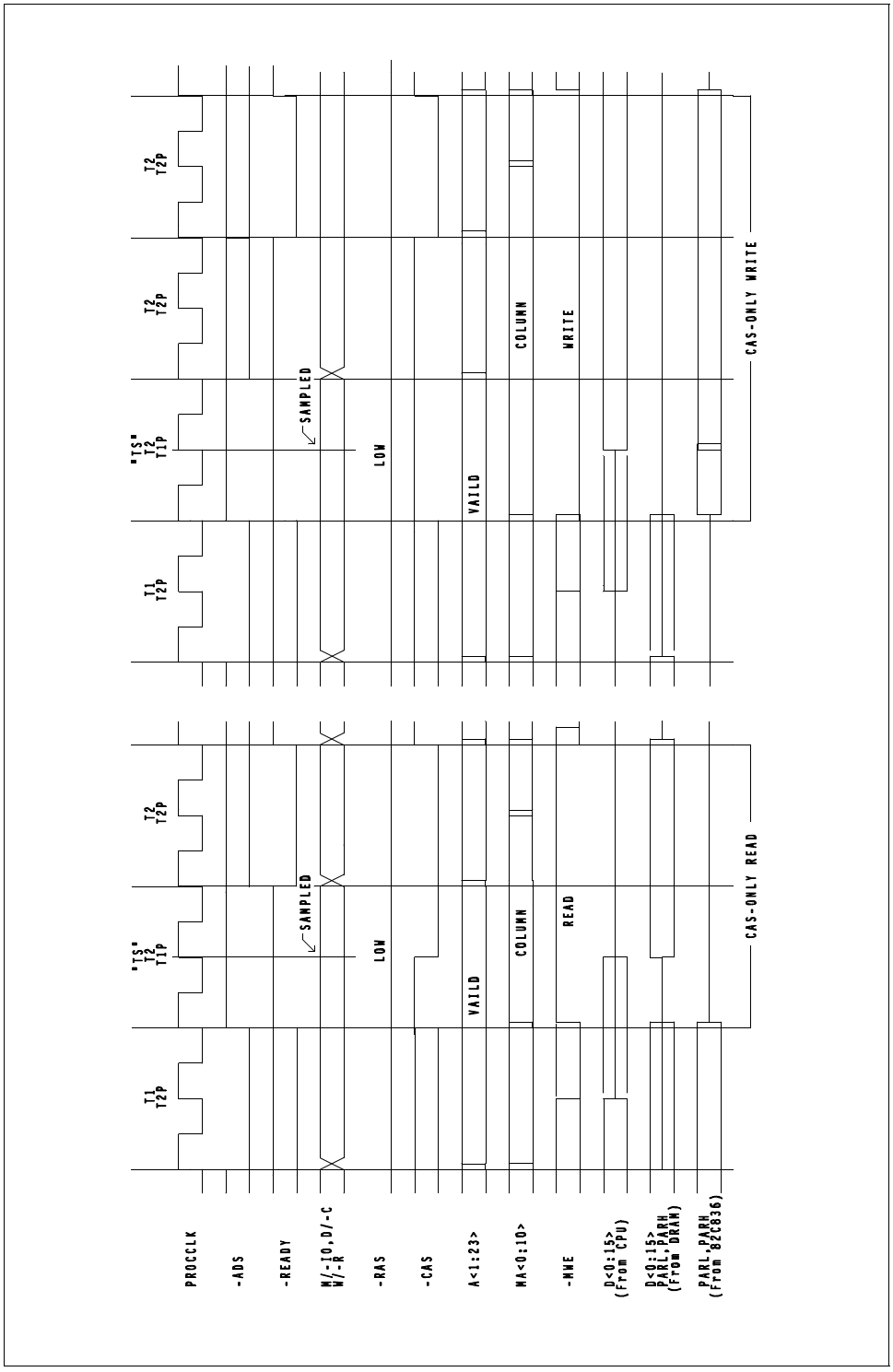
If the "-CAS Extend" wait state is enabled, one additional T-state is inserted in the -CAS active time.

If internal EMS, Early READY mode or LBA 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.

The CPU status signals (M/-IO, D/-C, W/-R) are latched internally at mid-TS.

Figure 11-6. -CAS-Only DRAM Access by CPU



Local DRAM Bank Switch (SRA Mode)

Figure 11-7 shows CPU accesses to local DRAM in zero wait state mode with -RAS initially high. These timing relationships apply when the DRAM bank 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 nonencoded RAS consists of T1P, T2P, T2P (three T-states). For encoded RAS, an extra T2P is needed. For nonpipelined 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.

Local DRAM RAS High Cycle (MRA Mode)

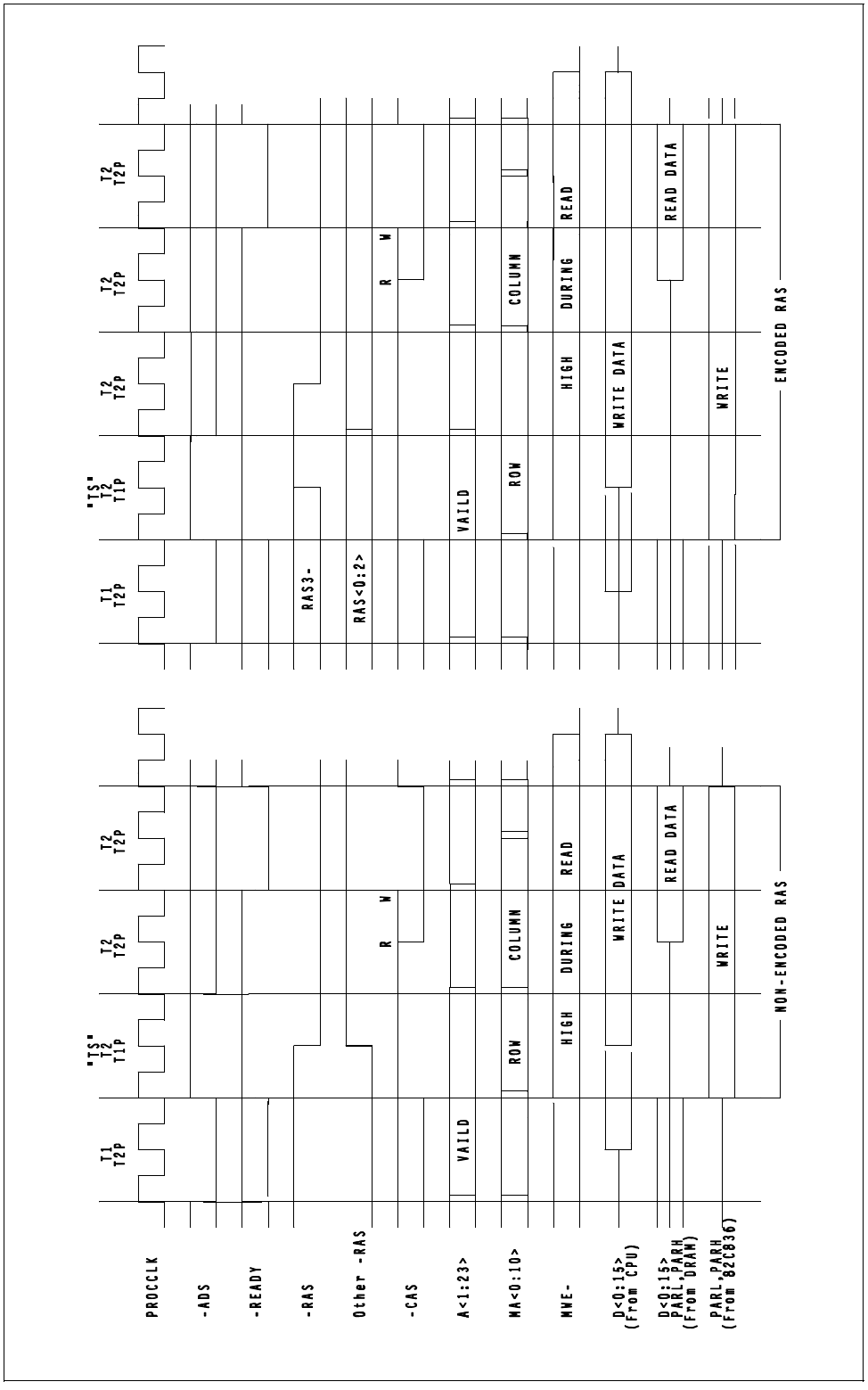
In MRA mode, bank-switch timing differs from SRA mode as follows:

RAS for the new bank frequently is already low, and the row address previously latched by the DRAMs is already valid. In that case, bank switch timing is exactly the same as page hit timing shown in Figure 11-6. The fastest possible bank-switch read in MRA mode is two T-states, while the fastest possible bank-switch write in MRA mode is three T-states (CPU running in pipeline mode).

If RAS for the new bank is not already low, then the cycle follows the nonencoded RAS timing shown in Figure 11-7, except that RAS for the previously accessed bank remains low instead of going high at mid-TS.

MRA mode, therefore, saves one T-state over SRA mode for bank-switch (page-hit) DRAM reads. Since the majority of DRAM accesses will be reads (mostly code fetches), and will be DRAM page-hits, the total reduction in T-states can be substantial. This the key performance advantage of MRA mode over SRA.

Figure 11-7. Local DRAM Bank Switch



Maximum Wait State Page Miss

Figure 11-8 shows the fully-burdened local DRAM cycle i.e., the longest possible cycle (eight T-states). It is a page miss cycle, meaning 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.

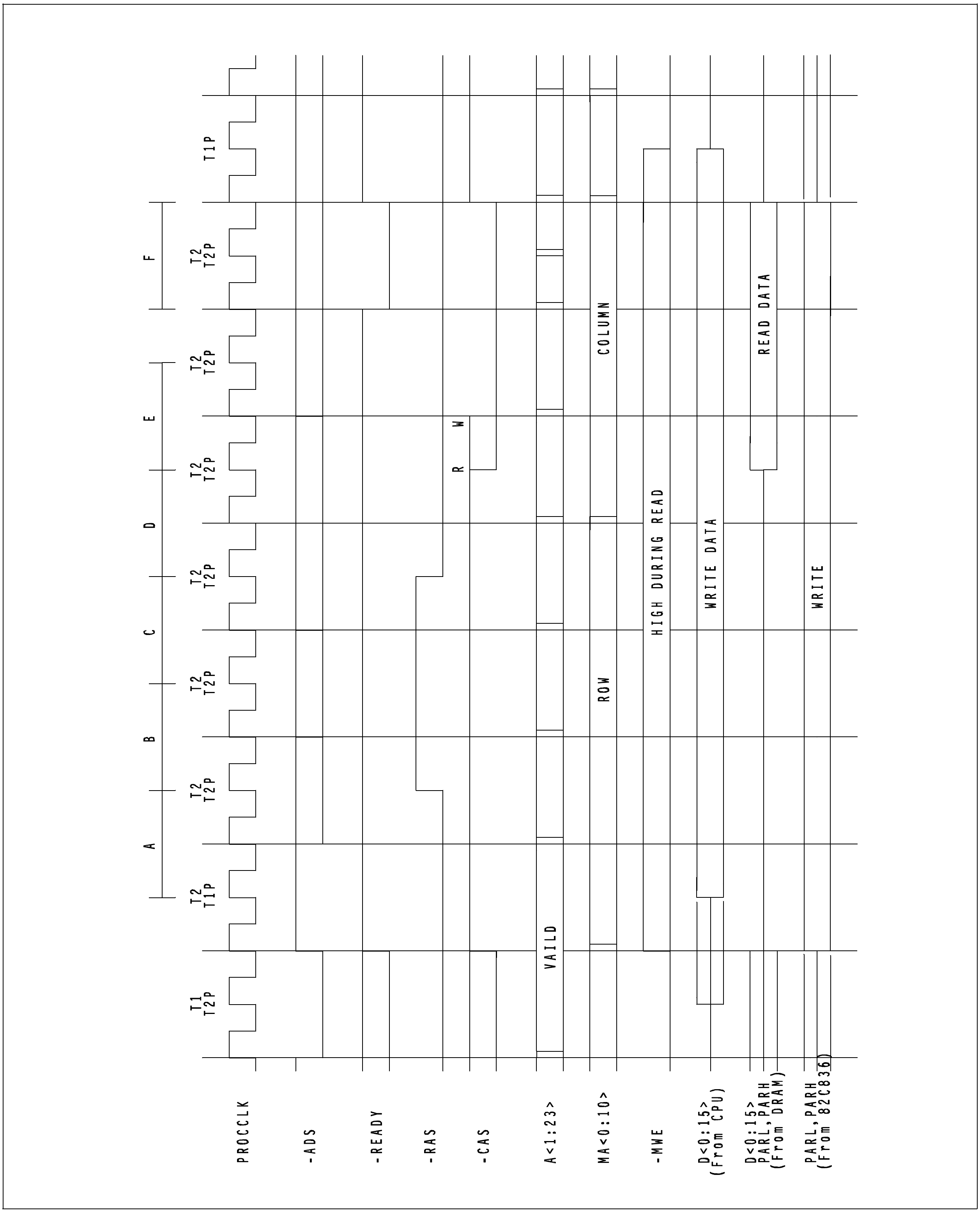
- State T1 occurs only if the CPU ended the previous cycle in nonpipeline mode.
- Interval A is the Early Wait State, applicable only when internal EMS, Early READY or LBA modes are enabled.
- Intervals B and C are needed for -RAS precharge time. For an encoded -RAS bank switch, only B is needed. For -CAS only cycles, neither B or 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 states affect 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.

Figure 11-8. Maximum Wait State Page Miss



Cache Mode Write Cycles

Figure 11-9 shows local DRAM write timing when early READY and/or LBA mode is enabled. The diagram shows a page hit write (zero wait states) followed by a page miss write (three wait states).

A RAS high write (not shown) has the same timing as a page miss write except the "RAS Precharge" interval, two T-states in duration, is deleted, resulting in a one wait state cycle. In a RAS high write, the falling edge of RAS occurs at the middle of the first T2 instead of the middle of the third T2.

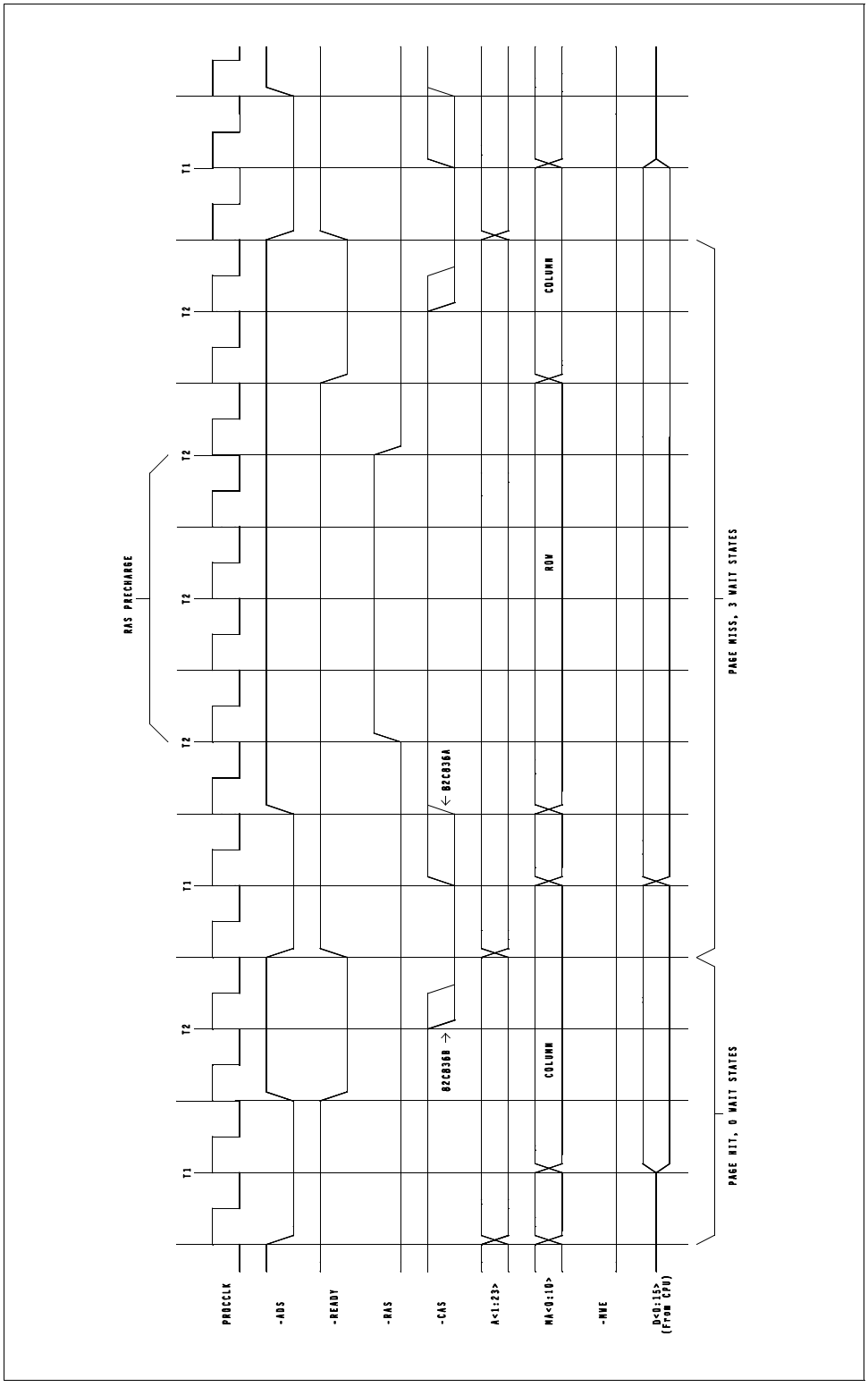
There is a slight difference in CAS timing between the 82C836B and the 82C836A. CAS timing for the 82C836A is shown in dashed lines. 82C836B timing improves the write data hold delay after CAS goes active.

The ability to perform page hit writes in zero wait states provides a significant performance benefit in cache-based systems. Since all writes go to DRAM (except in a diagnostic mode), streamlining write cycles to the same zero wait state performance as cache hit reads should result in a level of performance approaching buffered write-through or write-back cache architectures. Actual performance depends on the number of DRAM banks, DRAM page size, and ratio of DRAM page hits to page misses on RAS high cycles.

Zero wait state write timing is made possible by allowing CAS to extend into the T1 state of the next cycle. This also relies on the fact that nonpipeline CPU mode must be used when early READY and LBA modes are enabled in SCATsx.

If ICR 63H bit zero is set to one, an early wait state is inserted after T1 to allow time for external logic to claim the cycle via Early READY or LBA.

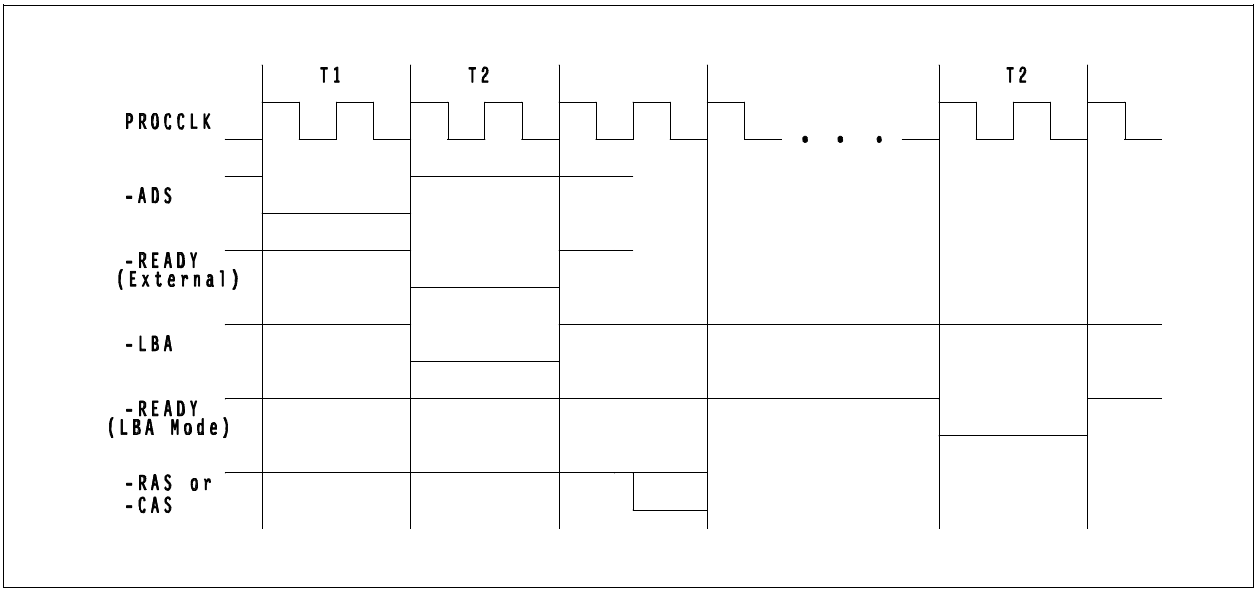
Figure 11-9. Cache Mode Write Cycles



Early READY and LBA Modes

Figure 11-10 shows how external logic can claim a CPU generated cycle by asserting -LBA (0WS pin) 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.

Figure 11-10. Early READY and -LBA



Any CPU generated cycle, except a local memory write, can be claimed in this fashion, including interrupt acknowledge cycles. Local memory writes normally cannot be claimed by external logic and do not cause an early wait state. However, even local memory writes can be claimed (and there will be an early wait state) if ICR 63H bit 0 is set to one.

When these cycle claiming capabilities are used, the CPU must run in nonpipelined 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.

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. -LBA is ignored and is "don't care" at all times other than at the end of the first T2 after T1.

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.

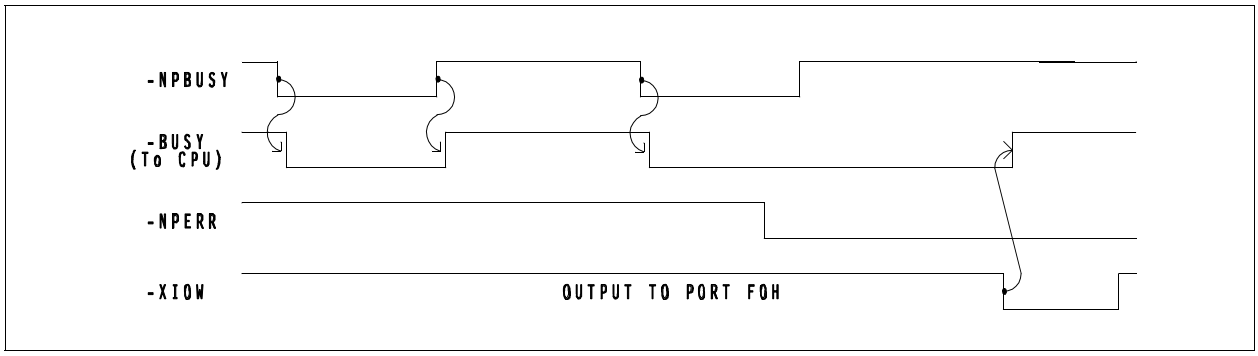
The net performance effect of local cache is as follows:

- 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-state 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.
- Page hit writes can occur in zero wait states, as compared to one wait state minimum in noncache modes.

Coprocessor Timing

Figure 11-11 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 prot 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. The -NPERR signal from the coprocessor is eventually cleared by I/O writes to the coprocessor sometime after the output to port F0H.

Figure 11-11. 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 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 tristated. This allows plenty of time for the floating CPU data bus to stablize 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. 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. Zero wait state cycles are possible when the coprocessor generates -READY.

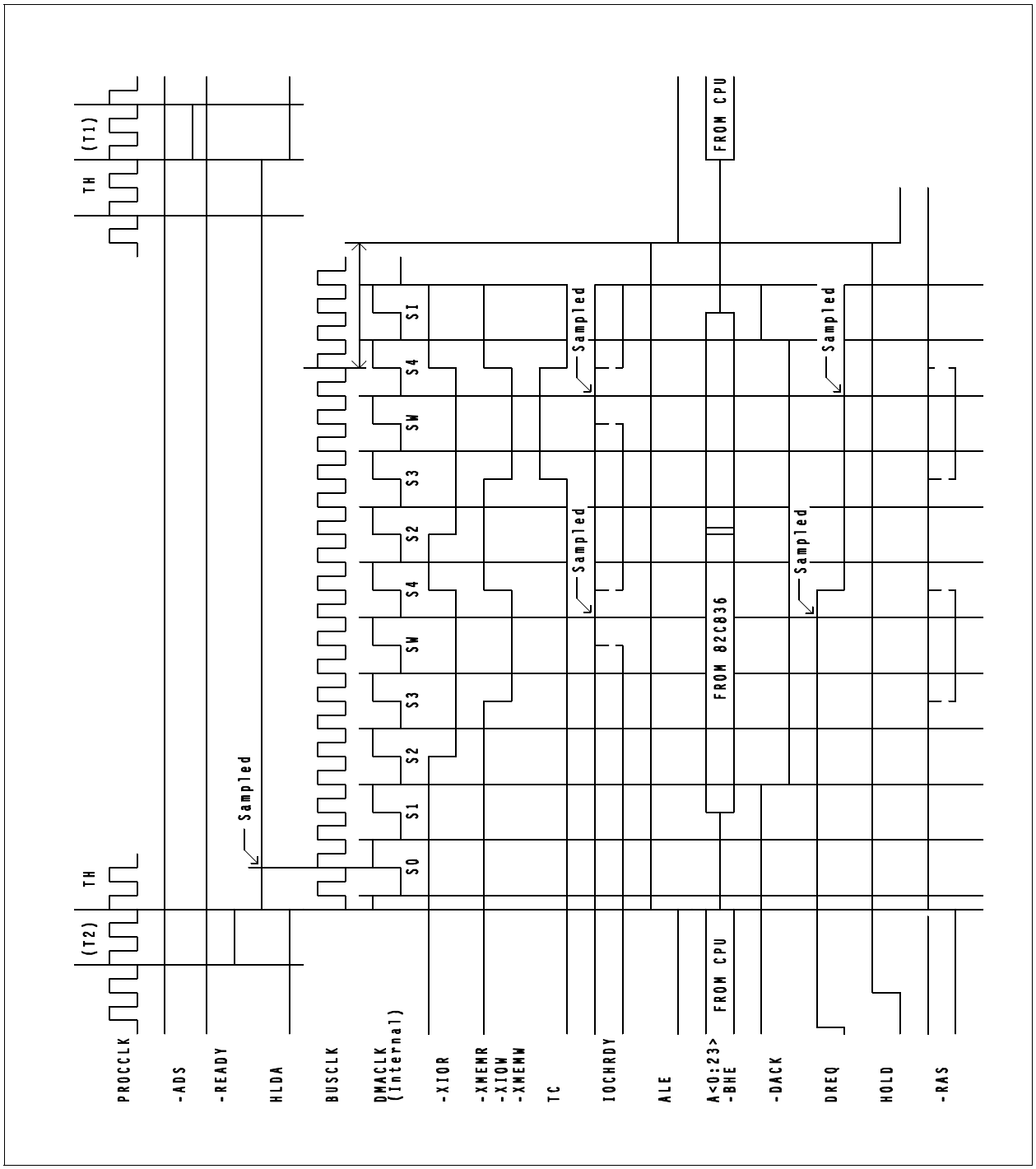
Since the coprocessor interface is tighly coupled to the CPU (rather than using an -XIOR/-XIOW interface), the coprocessor is not accessible by add-on card bus masters.

DMA Timing

Figure 11-12 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 deasserted.
- Some time after detecting HLDA, the 82C836 generates the DMA memory address and issues the appripriate -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 the address on the bus is not an I/O address.

Figure 11-12. DMA Timing



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 11-12, shown earlier, identifies hardware defaulting 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.

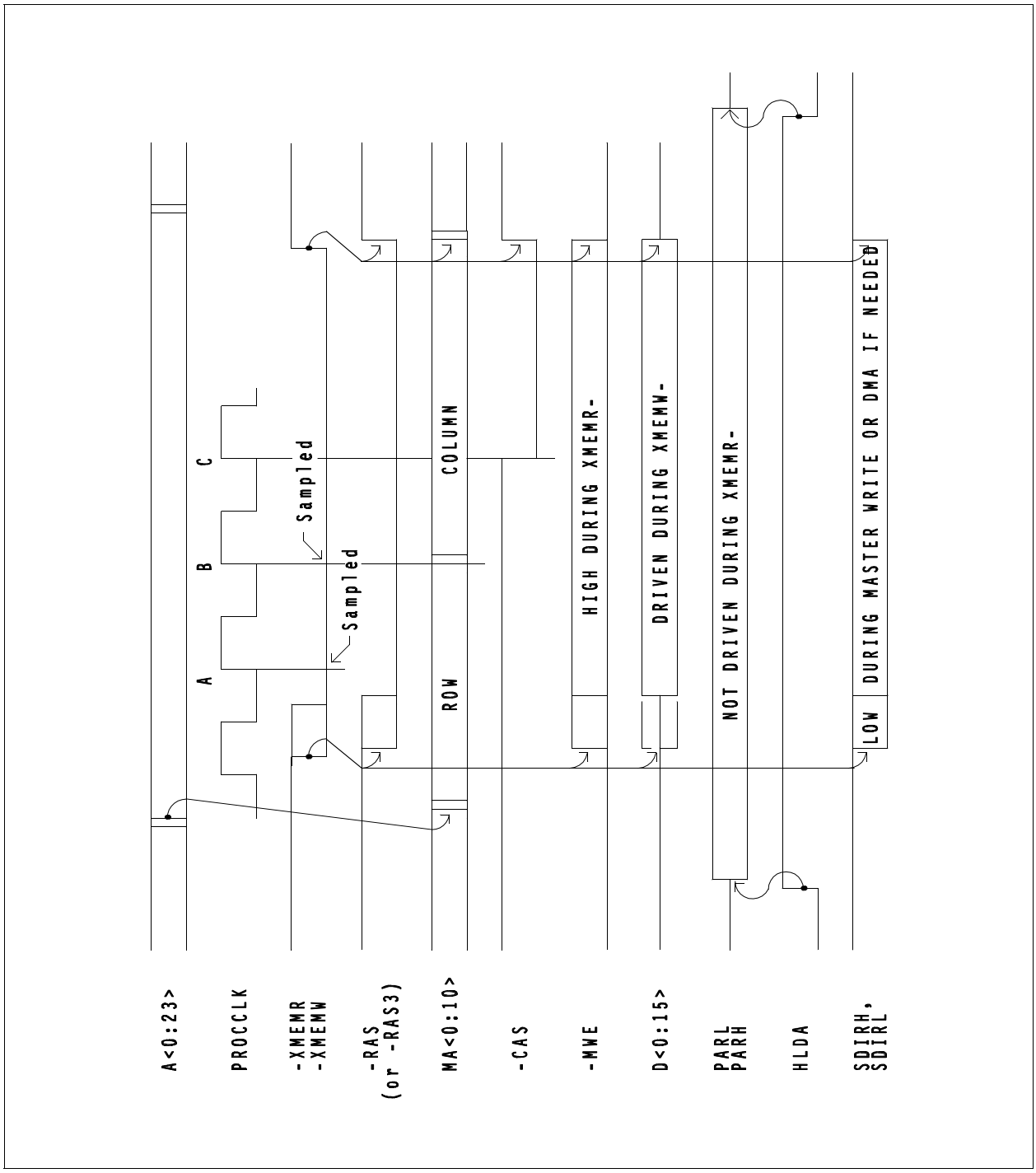
Using compressed mode, the S3 states can be eliminated. S3 events, in that case, occur in S2 instead of S3. SW states are not affected.

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

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 11-13.

Figure 11-13. DMA and 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 11-15.

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 14.3MHz OSC clock and an additional minimum delay of one complete cycle of PROCCLK. The involvement of the 14.3MHz 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.

DRQ/DACK Scanning in MRA Mode

Figure 11-14 shows how DRQ scanning and DACK encoding are performed in Multiple RAS Active (MRA) mode. The seven DRQ signals are continuously scanned, two at a time, and latched inside the 82C836. One pair of DRQ signals is scanned every 70ns (based on the 14.3MHz clock source) using an external 74F153 multiplexer. If the 82C836 internal 8237A-compatible logic decides to recognize one of the internally latched DRQ signals, scanning stops and locks onto the corresponding DRQ signal. At about the same time, the DACK signal issued by the internal 8237A-compatible logic is encoded and generated on the DACKA-C lines. Finally, -DACKEN goes active (low), enabling the external 74ALS138 decoder to reconstruct the appropriate DACK output.

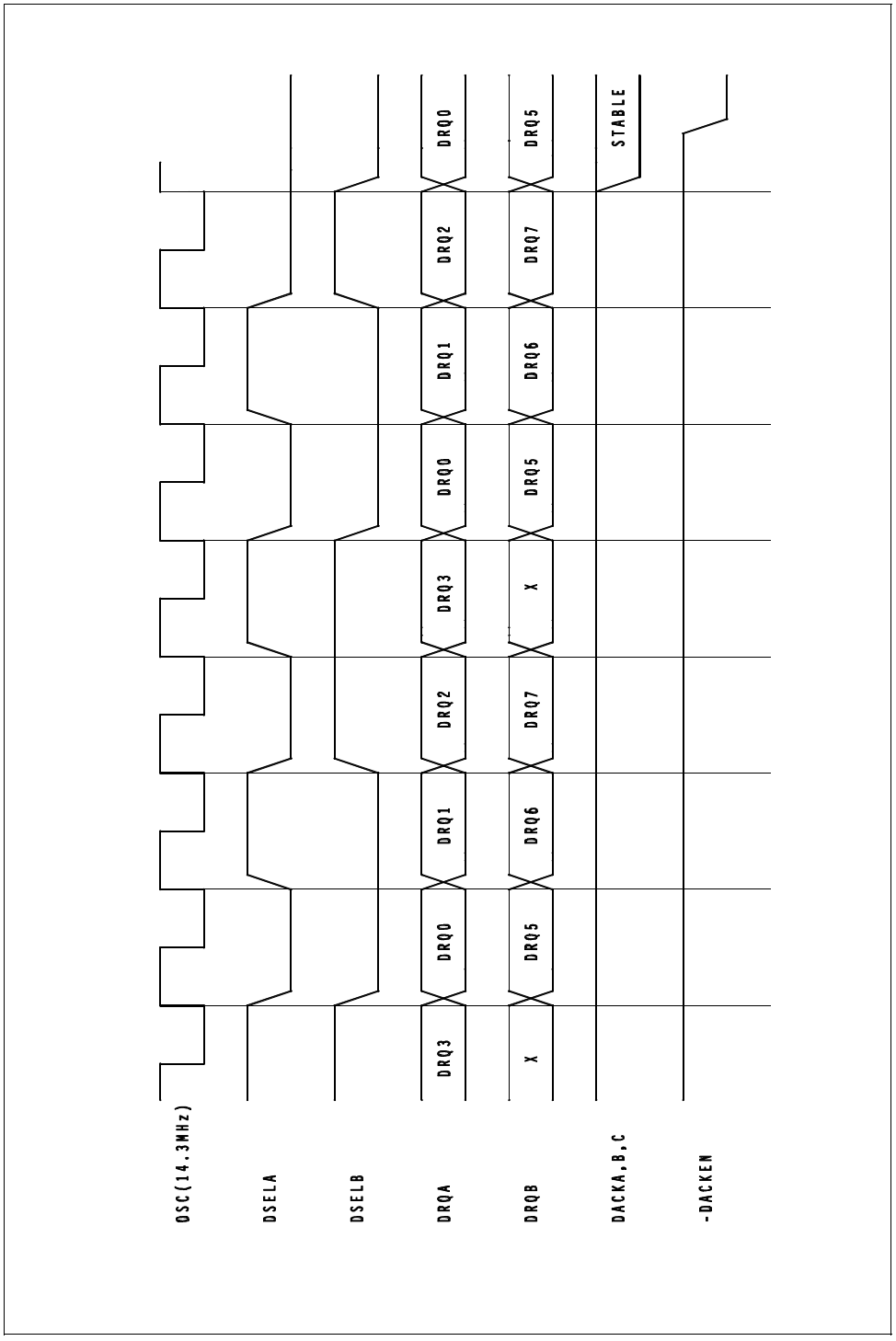
The scanning mechanism remains frozen on the selected DRQ signal until the DMA transfer is completed and the corresponding DACK signal is deasserted. Usually this will occur in response to external logic deasserting the DRQ signal. Scanning then resumes from the lock-in point and continues until the next DRQ signal is recognized by the internal 8237A-compatible logic.

It is possible for more than one DRQ signal to have been scanned and latched inside the 82C836 before the internal 8237A-compatible logic decides to recognize one of the internally latched DRQs. In fact, if a DMA channel has not been enabled by software, a latched DRQ for the disabled DMA channel will never be recognized by the 8237A-compatible until software enables the channel logic. Meanwhile, other DMA channels are free to function normally as programmed by the software.

Figure 11-14 illustrates the DRQ scanning process followed by recognition of DRQ0 or DRQ5.

Only four cycles of the 14.3 MHz clock are needed to completely scan and internally latch all seven DRQ signals.

Figure 11-14. DRQ/DACK Scanning in MRA Mode

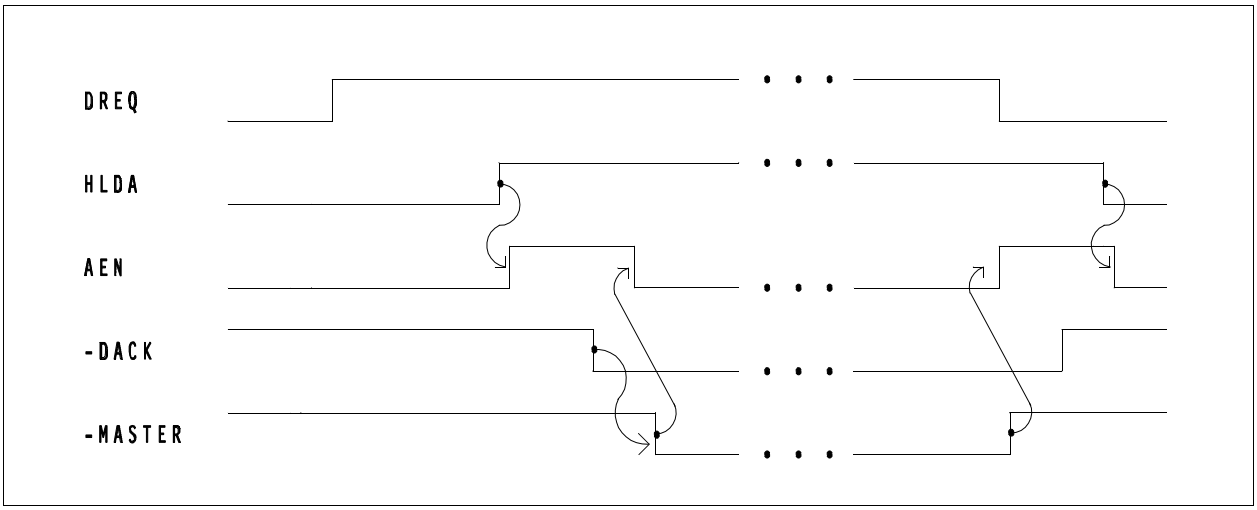


Master Arbitration

Figure 11-15 shows how an add-on card bus master obtains, and later releases, control of the buses. The general AT-compatible protocol is as follows:

- 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 deasserting -MASTER and DREQ.

Figure 11-15. Master Arbitration



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 11-13) 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). On Master writes, additional clock cycles can be programmed between B and C (see ICR 64H).
- -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 (HLDA and 14MHz-Based)

Figure 11-16 shows CAS-before-RAS using HOLD/HLDA protocol and the 14.3MHz time base. It applies to either system initiated refresh or master initiated refresh while power is good (PWRGOOD high). Refreshing ceases during power down unless standby refresh has been enabled (see ICR 60H). Laptop applications using the 82C836 can use standby refresh to preserve DRAM contents during power down. The basic 82C836 refresh protocol is as follows:

- If a system initiated refresh immediately follows a DRAM write in early READY or LBA mode (as depicted in Figure 11-16), CAS goes high at the middle of the first TH, and RAS goes high at the end of the first TH. In all other cases of system initiated refresh, HLDA causes all RAS lines to go high.

- Two PROCCLK cycles later, -REFRESH is asserted. For Master intitated 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 14.3MHz clock.
- The width of -XMEMR is programmable (see ICR 41H). For AT-compatibility, the width should be set to four clocks (280ns).
- The width of -XMEMR can also be extended, one 14.3MHz clock cycle at a time, by driving IOCHRDY low.
- -MWE is the inverse of -CAS; it goes high while -CAS is low.

The delay from HLDA rise to -REF fall during CAS-before-RAS refresh allows for CAS precharge time. The worst-case for CAS precharge is when a zero wait state cache-mode write is immediately followed by a refresh cycle. CAS can extend past the end of -READY by up to one full PROCCLK cycle in that case, as shown in Figure 11-16 (see also Figure 11-9).

When using nonencoded 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 assertion.

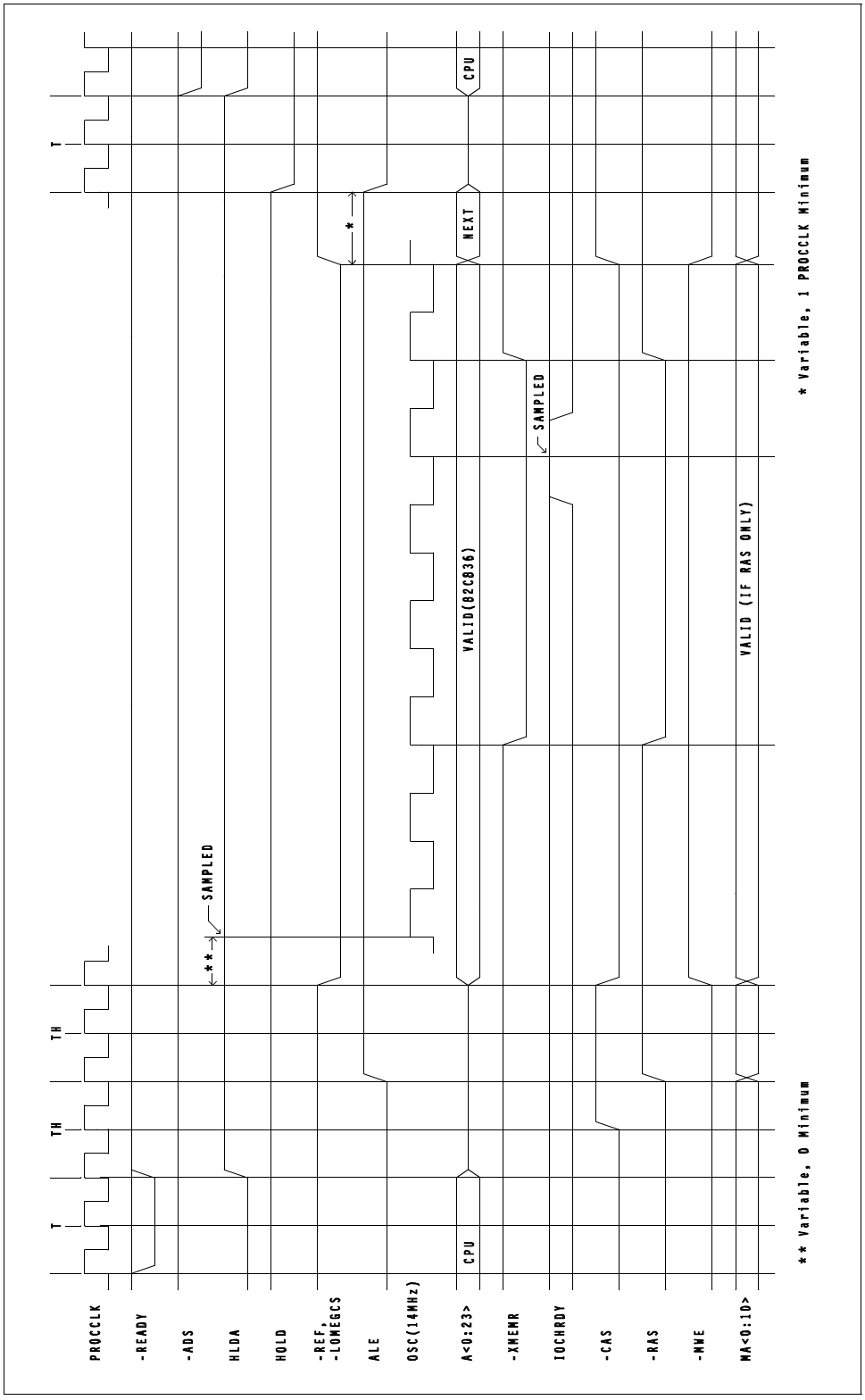
A0-10 contain the refresh address, incremented at the end of -REF (rising edge). A16-23 contain the Refresh Page register value, normally zero (programmable). A11-15 are undefined. The DRAM uses its own internal refresh address counter during CAS-before-RAS refresh, but the refresh address generated by the 82C836 is still needed on the AT bus.

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

When using RAS-only refresh instead of CAS-before-RAS refresh (see ICR 60H), refresh timing differs from the foregoing as follows:

- -CAS remains high during RAS-only refresh.
- -MWE remains low during RAS-only refresh.
- The two PROCCLK delay between -RAS rise and -REF fall is deleted; -REF goes low on the same PROCCLK edge on which -RAS goes high.
- The DRAMs rely on the 82C836 to provide the refresh address during RAS-only refresh.

Figure 11-16. Refresh Timing (HLDA/14MHz-Based)



Hidden Refresh (PROCCLK-Based, No HLDA)

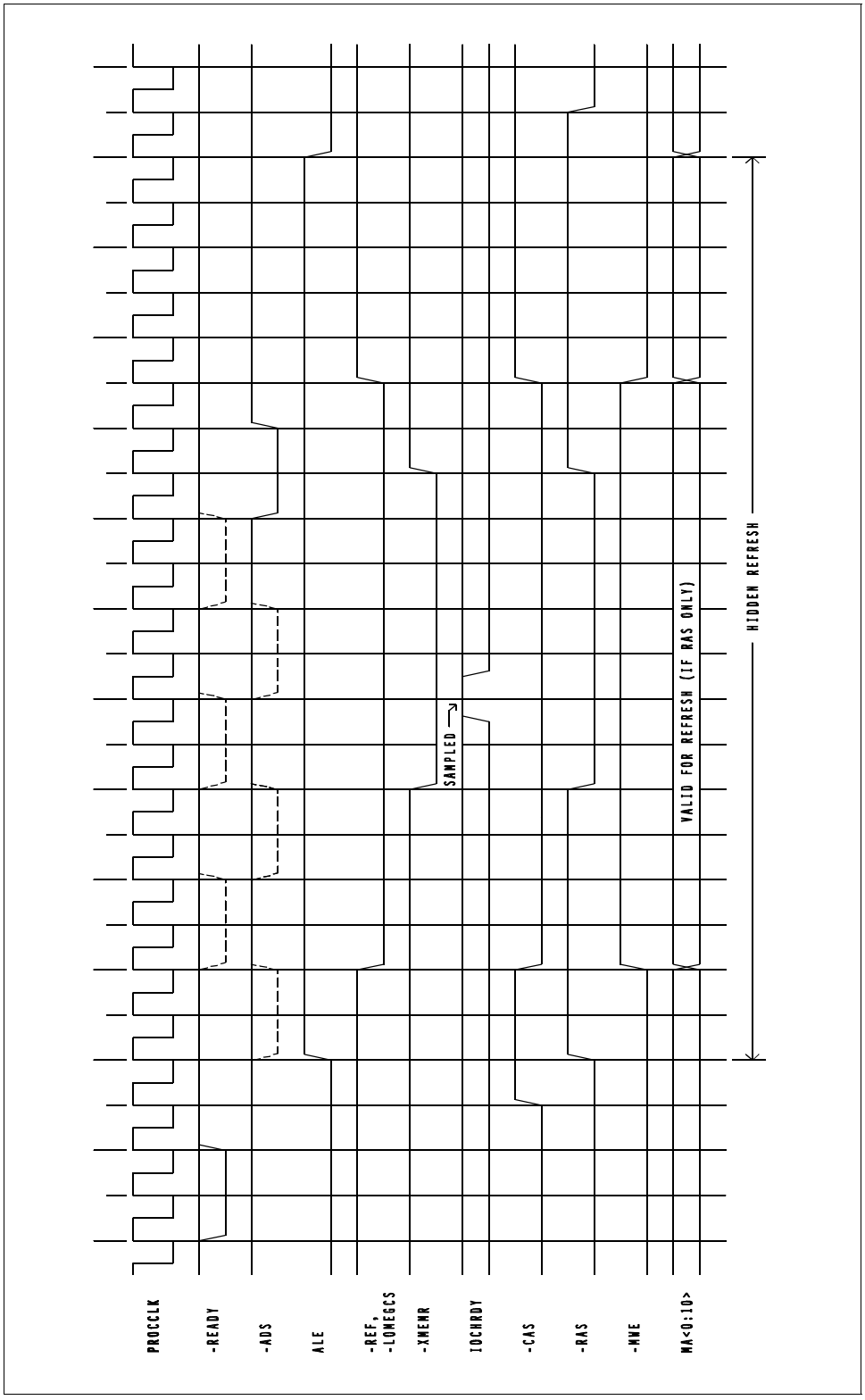
Figure 11-17 shows hidden refresh timing. In hidden refresh, the CPU is allowed to continue operating during refresh cycles. HOLD and HLDA are not used. Obviously, the CPU cannot be allowed to access local memory or the AT bus during refresh. So, the main usefulness of hidden refresh is in cache based systems to allow cache read hits to continue during refresh. The basic protocol for hidden refresh is as follows:

- If a CPU cycle is already in progress when the 82C836 needs to perform a refresh cycle, the 82C836 waits for the CPU cycle to end (indicated by -READY). If no CPU cycle is in progress (-READY has occurred but no new -ADS yet), the hidden refresh cycle can begin immediately.
- The 82C836B asserts ALE throughout the hidden refresh cycle.
- Two PROCCLK cycles after ALE goes high, -REF and -LOMEGCS go low. -CAS goes low on the same PROCCLK edge as -REF. The delay from ALE to -REF allows time for CAS precharge when using CAS-before-RAS refresh.
- After additional delay, -XMEMR and -RAS go low for a few PROCCLKs, then back high. The 82C836 provides the refresh address for the DRAMs using MA0-10, but other external logic must provide the refresh address for the AT bus.
- As in HLDA based CAS-before-RAS refresh cycles, -MWE goes high while -CAS is low.
- A few PROCCLKs after the end of -XMEMR, -REF goes high.
- Finally, after further delay, ALE goes low. If the CPU attempted to start a local memory or AT bus access during the hidden refresh cycle, CPU wait states are inserted as needed to allow the refresh to complete. When ALE goes low, the waiting CPU cycle is allowed to proceed. The T1 state for the waiting CPU can overlap the hidden refresh cycle, but the normal T2 states begin after ALE goes low. For example, Figure 11-17 shows a local memory write being delayed until the end of the hidden refresh. -RAS goes low for the memory write in the middle of the first T-state after the end of the hidden refresh.

Since hidden refresh is useful only in cache-based systems, which must use nonpipeline mode, hidden refresh is designed to work in nonpipeline mode only.

-MWE needs to go high during CAS-before-RAS refresh to prevent the DRAMs from going into Test mode. This is incompatible with the use of MD buffers if -MWE is used as the direction control for the buffers; a high logic level on -MWE would cause the MD buffers to drive the CPU local data bus, interfering with cache read hit activity. Thus, in a cache-based system using hidden refresh, MD buffers should not be used. Worst case system timing and loading analysis shows that MD buffers aren't needed in any case, in any cache or non-cache configuration; worst-case system timing margins are better without MD buffers.

Figure 11-17. Hidden Refresh (PROCCLK-Based, No HLDA)



The following programmable options are available with hidden refresh (ICR 63H):

- RAS-only Refresh —In RAS-only refresh, -CAS remains high and -MWE remains low.
- AT Refresh OFF —In this mode, ALE, -REF and -XMEMR remain inactive during hidden refresh. IOCHRDY has no effect. Only the local DRAM is refreshed, not the AT bus, and the total refresh cycle time is correspondingly shorter (see Table 11-1). For compatibility with many AT bus add-in cards, AT refresh should not be turned off.
- Refresh on Idle —Enabling this option prevents hidden refresh from starting until after a CPU idle state. Normally, a hidden refresh can begin as early as the next T-state following -READY, as shown in Figure 11-17. Refresh on idle prevents a hidden refresh from starting if the next T-state after -READY is a T1-state (-ADS asserted). In most systems, refresh on idle will probably have no measurable performance effect and should remain disabled.

The exact number of PROCCLK cycles between various events in hidden refresh depends on the CPU speed as programmed in ICR 63H. The PROCCLK counts also depend on whether AT refresh is enabled or disabled, as in the following Table.

Table 11-2. PROCCLK Cycles and Hidden Refresh

Events	16MHz		20MHz		25MHz	
ALE rise to -REF/-CAS fall	2	2†	2	2†	2	2†
-REF/-CAS fall to -XMEMR/-RAS fall	4	1†	5	2†	7	2†
-XMEMR/-RAS fall to IOCHRDY sampling	2	—	3	—	4	—
IOCHRDY sampled high to -XMEMR/-RAS rise	5	—	6	—	7	—
Total -XMEMR/-RAS low time without IOCHRDY	7	4†	9	4†	11	4†
-XMEMR/-RAS rise to -REF/-CAS rise	2	1†	3	1†	3	1†
-REF/-CAS rise to ALE fall	4‡	4†	4‡	5†	4‡	5†
Total hidden refresh time without IOCHRDY	20	12†	24	14†	28	14†

† Denotes AT refresh OFF.
‡ Plus one PROCCLK if needed to make ALE fall at end of T-state.

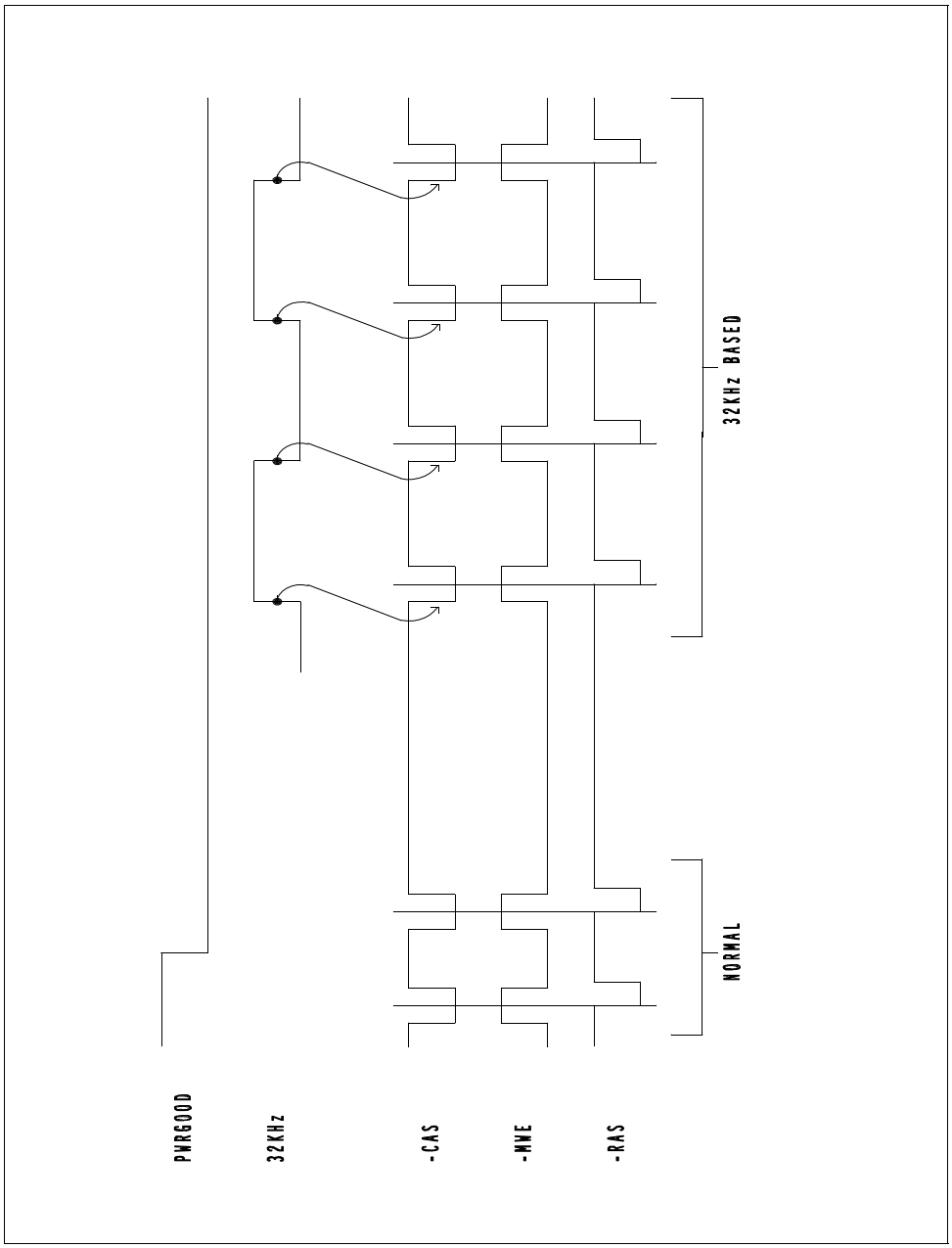
Standby Refresh (32KHz-Based)

In addition to HLDA refresh (14.3MHz based, CAS-before-RAS or RAS-only) and hidden refresh, the 82C836B also supports standby refresh (useful only in CAS-before-RAS mode). Standby refresh operates only if it has been enabled via ICR 60H. It allows the contents of DRAM to be preserved during power down. Figure 11-18 shows standby refresh timing:

- PWRGOOD going low indicates the imminent loss of main power. The 82C836 may complete one more normal refresh (HLDA or hidden), but must then switch to standby refresh as quickly as possible.

- Each standby refresh cycle starts with -CAS going low and -MWE going high. After a short delay, -RAS goes low.
- After a further delay, -RAS goes back high. After more delay, -CAS goes high and -MWE goes low.

Figure 11-18. Standby Refresh (32KHz-Based)



The various time delays during standby refresh are generated via asynchronous gate delays.

Standby refresh is based on the 32KHz time base. The 32KHz clock should be designed to operate on battery power (along with the 82C836) when main power has been turned off. Each edge of the 32KHz clock triggers a CAS-before-RAS refresh cycle. The DRAMs, of course, must also operate on battery power in order to preserve their contents during main power interruptions. There should be either no buffering of -RAS, -CAS, and -MWE to the DRAMs, or battery backed buffers should be used.

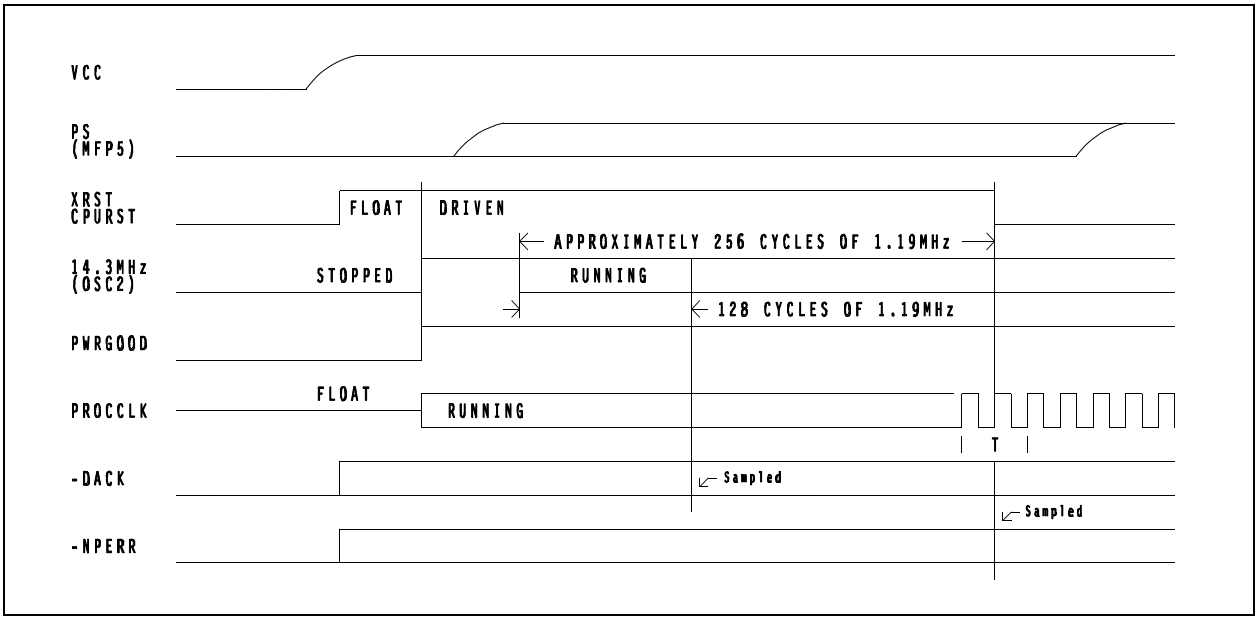
As shown in Figure 11-18, the transition from normal refresh to standby refresh occurs automatically in response to PWRGOOD going low. The transition back to normal refresh after PWRGOOD goes back high is somewhat more complicated. First, BIOS must restart normal refresh by appropriately programming the 14.3MHz based refresh timer. After the refresh timer starts running again, 32KHz based refresh is stopped and 14.3MHz based refresh starts up.

For laptop applications, both normal refresh and standby refresh should use CAS-before-RAS protocol in which the DRAMs generate their own refresh addresses internally.

Power Turn-On and System Reset

Figure 11-19 shows what happens immediately following power turn-on. Except for VCC and PS, this sequence applies to the cycling of PWRGOOD by means of a hardware reset switch as well as power turn-on.

Figure 11-19. Power-On



Vcc typically ramps up very slowly (perhaps 100ms or more) relative to the high speed of the 82C836 and CPU. At around 2 or 3VDC, 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 Vcc reaches its valid operational value (4.75VDC minimum).

While PWRGOOD is still low: All outputs except the 14.3MHz 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 14.3MHz driver is held at a continuous low 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 14.3MHz oscillator is allowed to start running. There will typcially be a significant start-up time for the 14.3MHz oscillator, potentially 1000 microseconds or longer.

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

There is, therefore, an interval of approximately 200 microseconds during which all clocks are running and both resets are still high. Allowing BUSCLK and 14.3MHz 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 the section titled 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.

The foregoing operation is changed as follows when standby mode is enabled:

- If Power Sense (MFP5) is low at any time while PWRGOOD is low, standby mode is cleared and the system reacts to PWRGOOD exactly as described above. When using standby mode, the 82C836 and PS (MFP5) signal must remain powered up even when PWRGOOD is low. In normal system operation, a battery will do this. In a test environment with no battery connected, standby mode can be tested by driving PWRGOOD low without actually turning power off.
- When PWRGOOD goes low while standby mode is enabled, the 82C836 attempts to switch to standby refresh (32KHz based). PWRGOOD is internally blocked from having any further effect until the switch to 32 KHz based standby refresh is successful. To prevent disruption of memory refresh, at least one additional normal refresh must occur before the 82C836 switches to standby refresh. There is a 32 KHz based timeout to prevent a possible system lockup if normal refresh has stopped running for any reason (such as erroneous programming by the software). If the timeout elapses before the switch to standby refresh occurs, indicating that there is

a major malfunction in memory refresh, then the standby mode is cleared completely and the system recovers as if standby mode had not been enabled. The timeout interval is approximately 15 refresh periods at the currently programmed standby refresh rate (ICR 60H, bits 4 and 5).

- Since refresh requires the use of -RAS, -CAS, MA, and -MWE signals, these signals remain active during standby refresh instead of being tristated. (The MA lines are driven to a continuous logic high level. No refresh address is generated; rather, CAS-before-RAS refresh must be used along with standby refresh.) The ICR bits for memory configuration and MRA enable are also preserved during power down if standby mode is enabled. See the section titled Standby Power Management and Laptop Support for more precise definition of which outputs remain actively driven during power-down standby mode.
- Since DACK signals become CAS lines during MRA mode, DACK sampling during XRST (see Figure 11-19) is suspended when powering back up from standby mode. The previously sampled state of the DACK lines at the time of initial power application (before standby mode was enabled) is preserved. (The 82C836B has an internal hardware interlock to prevent standby mode from becoming or remaining enabled when external RTC is enabled. This interlock is necessary because the 32KHz input is cut off when external RTC is enabled. Without 32KHz, standby mode cannot function and the system could lock up if standby mode is enabled.)

In a manufactuirng or test environment, it often happens that the system is powered up initially with no battery connected. To keep the external PS (MFP5) circuit as simple as possible, the 82C836B automatically ignores the PS input when PWRGOOD is high. As soon as PWRGOOD goes high, the 82C836B operates as if PS were high even if PS is actually still low. (The 82C836A lacks this feature and requires PS to be high before XRST goes low. In addition, the 82C836A will malfunction if PS goes low at any time while PWRGOOD is high and XRST is low.)

In a normal operating environment, a battery will have been connected before turning on the main power supply. In that case, PS (MFP5) will already be high before PWRGOOD goes high, and PS will remain high during subsequent power-down periods. For the 82C836B, the external PS (MFP5) circuit only needs to insure a brief interval (100ns should be adequate) of low time on PS (MFP5) after Vcc to the 82C836 has reached 3.0V (see Figure 3-3).

Section 12

System Characteristics

Physical Characteristics

Tables 12-1 through 12-3 lists the physical characteristics of the 82C836 chipset.

Table 12-1. Absolute Maximum Ratings

Symbol	Parameter	Min.	Max.	Units
V_{CC}	Supply Voltage	—	7.0	V
V_I	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

Permanent device damage may occur if absolute maximum ratings are exceeded. Functional operation sh ould be restricted to the conditions described under Operating Conditions.

Table 12-2. Operating Conditions

Symbol	Parameter	Min.	Max.	Units	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	

Table 12-3. Capacitive Characteristics

Symbol	Parameter	Min.	Max.	Units	Conditions
C_{IN}	Input Capacitance	—	10	pF	fC = 1MHz
$C_{I/O}$	I/O Capacitance	—	20	pF	Unmeasured pins grounded
C_{OUT}	Output Capacitance	—	20	pF	Unmeasured pins grounded

DC Characteristics

Table 12-4 summarizes the DC characteristics of the 82C836 chipset.

Table 12-4. DC Characteristics

Symbol	Parameter	Min.	Max.	Units	Conditions
V_{IL}	Input Low Voltage	—	0.8	V	—
V_{IH}	Input High Voltage	2.0		V	—
V_{OL}	Output Low Voltage Drive for pins: -MWE, -XIOR, -XIOW, -XMEMR, -XMEMW,ALE, -REFRESH, PROCCLK	—	0.4	V	I_{OL} = 8mA
V_{OL}	Output Low Voltage Drive for pins: A0-23, MODA0, MODA20, MA0-9, RAS0-3, -CASH, -CASL, PARH, PARL, BUSCLK, D0-15, XD0-15, -DACKn, -IOCS16, -MEMCS16, -NA, -BHE, -READY, XRST	—	0.4	V	I_{OL} = 4mA
V_{OL}	Drive for all other pins	—	0.4	V	I_{OL} = 2mA
V_{OH}	Output High Voltage Drive for pins: -MWE, -XIOR, -XIOW, -XMEMR, -XMEMW, ALE, -REFRESH, PROCCLK	2.4	—	V	I_{OH} = -8mA
V_{OH}	Output High Voltage Drive for pins: A0-23, MODA0, MODA20, MA0-9, RAS0-3, -CASH, -CASL, PARH, PARL, BUSCLK, D0-15, XD0-15, -DACKn, -IOCS16, -MEMCS16, -NA, -BHE, -READY, XRST	2.4	—	V	IOH = 4mA
V_{OH}	Drive for all other pins	2.4	—	V	IOH = -2mA
V_{CL}	PROCCLK Output Low	—	0.4	V	I_{OL} = 4.0 mA
V_{CH}	PROCCLK Output High	—	V_{CC} -0.8	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	—	140	mA	C_{XIN} = 50 MHz
I_{CCSB}	V_{CC} Standby Supply Current	—	50	µA	PWRGOOD low, standby refresh is disabled

AC Characteristics 16- and 20MHz

All timing parameters are specified under capactive load of 50pf, Vcc 4.75V to 5.25V, and temperature of 0 to 70 ° C, and allow operation at 16MHz with 100ns DRAM or 20MHz with 80ns DRAM without a CAS-extended wait state. All the units discussed are in nanoseconds and are subject to change.

Unless otherwise noted, the following configuration options are also supported:

- Memory configurations up to 8 banks using Encoded RAS (SRA mode)
- Memory configurations up to 4 banks using either SRA or MRA mode, nonencoded RAS
- Cache or non-cache
- Nonlatching buffer between SA-bus and CPU local address bus

Formula Specifications refer to the minimum or maximum result of a specified calculation involving other chip parameters. For any particular chip, formula specifications express a tracking relationship over temperature and voltage for the parameters involved in the formula.

Note that 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.

The following terminology is used in the parameter descriptions:

Command	=	-XIOR, -XIOW, -XMEMR or -XMEMW
Rise	=	low-to-high transition
Fall	=	high-to-low transition
Float	=	transition to high-impedance state
Turn-on	=	a signal coming out of “float” and driven high or low
Active	=	high for high-true signals, or low for low-true
Inactive	=	high for low-true signals, or low for high-true

To assure reliable local memory operation under worst-case conditions, the DRAMs should meet the specifications shown in the following table. Most DRAMs utilizing CMOS technology can readily meet these specifications.

Table 12-5. DRAM Parameter Specifications

Parameter	16MHz		20MHz	
	Min.	Max.	Min.	Max.
Read data valid from RAS active (tRAC)	—	100	—	80
Read data valid from CAS active (tCAC)	—	35	—	25
Read data valid from column address (tAA)	—	55	—	45
CAS active time (tCAS)	35	—	25	—
CAS precharge time in page mode (tCP)	15	—	15	—
CAS cycle time in page mode (tPC)	85	—	70	—
CAS fall to RAS rise (tRSH)	40	—	25	—

CPU to Local Memory

Tables 12-6 through 12-8 summarizes CPU to memory specification requirements.

Table 12-6. CPU to Local Memory—Output Responses

Symbol	Parameters	16MHz		20MHz	
		Min.	Max.	Min.	Max.
t100	RAS active from PROCCLK rise	—	33	—	23
t101	RAS inactive from PROCCLK rise	—	33	—	32
t102	CAS active from PROCCLK rise	—	23	—	23
t103	CAS inactive from PROCCLK rise	—	28	—	27
t104	Row address valid from PROCCLK rise	—	40	—	32
t105	Row address hold from PROCCLK rise	0	—	0	—
t106	Row address valid from A0-23 in	—	50	—	50
t107	Column address valid from PROCCLK rise	—	19	—	19
t108	Column address hold from PROCCLK rise	0	—	0	—
t109	Column address valid from A0-23 in	—	50	—	47
t110	-MWE fall from -CAS rise	0	60	0	48*
t111	-MWE rise from PROCCLK rise	—	33	—	32
t112	PARL, PARH valid from D0-15 write data in	—	55	—	55
t113	-READY delay from PROCCLK rise	4	40	4	38

* 60ns maximum for 82C836A-20.

Table 12-7. CPU to Local Memory—Formula Specifications

			16MHz		20MHz	
Symbol	Critical Path	Formula	Min.	Max.	Min.	Max.
te101	RAS precharge	t101-t100	—	39	—	19
te102	RAS-to-CAS delay	t100-t102	—	24	—	15
te103	CAS precharge	t103-t102	—	16	—	10
te103a	CAS rise to RAS fall	t103-t100	—	7	—	6
te104	Row address setup before RAS	t104-t100	—	20	—	14
te105	Row address hold after RAS	t100-t105	—	7	—	5
te106	Row address setup before RAS	t106-t100	—	46	—	34
te107	Column address setup before CAS	t107-t102	—	17	—	11
te108	Column address hold after CAS	t102-t108	—	8	—	2
te109	Column address setup before CAS	t109-t102	—	43	—	31
te111	MWE rise before CAS fall	t111-t102	—	16	—	10*
te112	Write parity setup before CAS	t112-t102	—	10	—	10

* 14ns maximum for 82C836A-20.

Table 12-8. CPU to Local Memory—Input Requirements

		16MHz		20MHz	
Symbol	Parameters	Min.	Max.	Min.	Max.
t120	PARL, PARH setup before PROCCLK rise during read	9	—	9	—
t121	PARL, PARH hold after PROCCLK rise during read	6	—	6	—
t122	D0-15 setup before PROCCLK rise during read	9	—	9	—
t123	D0-15 hold after PROCCLK rise during read	6	—	6	—

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

Tables 12-9 through 12-11 are the specification requirements for the CPU to access the AT-bus, the on-board I/O, and ROM.

Table 12-9. CPU to AT-Bus, On-board I/O, and ROM—Output Responses

		16MHz		20MHz	
Symbol	Parameters	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	4	40	4	31
t133	ALE rise from BUSCLK low	—	25	—	25
t134	ALE fall from BUSCLK high	—	20	—	15
t135	Command and -ROMCS active from BUSCLK	—	30	—	30
t136	Command and -ROMCS inactive from BUSCLK rise	—	25	—	25
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	—	35	—	35
t140	MODA20 delay from HOLD fall	—	59	—	50
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 CPU write or DMA/Master read	—	30	—	30
t146	XD0-7 delay from D8-15 during CPU write or DMA	—	70	—	70
t147	D0-15 delay from XD0-15 during CPU read or DMA/Master write	—	25	—	25
t148	D8-15 delay from XD0-7 during CPU read or DMA	—	45	—	40

Table 12-10. CPU to AT-Bus, On-board I/O, and ROM—Formula Specifications

			16MHz		20MHz	
Symbol	Critical Path	Formula	Min.	Max.	Min.	Max.
te133	ALE width	t133-t134	—	10	—	10
te137	MODA0 hold after command rise	t136-t137	—	16	—	16
te139	MEMCS16 setup before ALE fall	t139-t134	—	24**	—	15**
te145	Master read, access from CAS	t145+t102	—	38*	—	38*
te145a	Master read, access from Column Address	t145+t107	—	34	—	34

* 36ns maximum for 82C836A.
** 20ns maximum for 82C836A-20. 29ns maximum for 836A-16.

Table 12-11. CPU to AT-Bus, On-board I/O, and ROM—Input Requirements

		16MHz		20MHz	
Symbol	Parameters	Min.	Max.	Min.	Max.
t160	-ADS setup before PROCCLK rise*	24	—	24	—
t161	-ADS hold after PROCCLK rise*	4	—	4	—
t162	A0-23 setup before PROCCLK rise †	57	—	45	—
t163	A0-23 hold after PROCCLK rise ‡	4	—	4	—
t164	-BHE setup before PROCCLK rise †	39	—	39	—
t165	-BHE hold after PROCCLK rise ‡	4	—	4	—
t166	-DC, W/-R and M/-IO setup before PROCCLK rise †	39	—	39	—
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	—
t173	-IOCS16 hold after Command inactive	0	—	0	—
t174	-MEMCS16 setup before ALE fall	33**	—	30**	—
t175	-MEMCS16 hold after ALE fall	10	—	10	—
t176	XD0-15 valid before read command rise	25	—	20	—
t177	-IOCS16 setup before BUSCLK rise (first BUSCLK rise after I/O command fall)	0	—	0	—

\+ Certain input parameters, as noted, are nonrestrictive. If these parameters are violated, the signal may not be recognized until the subsequent clocking period. The parameter specifies only the condition needed to guarantee recognition of the signal on the particular clock edge.
‡ At end of TS (TS is the T-state in which -ADS changes from low to high).
† At mid-TS.
* At start of TS.
** 28ns maximum for 82C836A at 16MHz. 25ns maximum for 82C836A at 20MHz.

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

Tables 12-12 through 12-14 are the specification requirements for DMA to access the AT-bus, on-board I/O, and ROM.

Table 12-12. DMA to AT-Bus, On-board I/O, and ROM—Output Responses

		16MHz		20MHz	
Symbol	Parameters	Min.	Max.	Min.	Max.
t190	-DACKn or -DACKEN delay from BUSCLK rise	—	50	—	50
t191	ALE rise from HLDA rise	—	75	—	70
t192	DMA Address delay from BUSCLK rise	—	40	—	35
t194	Command fall from BUSCLK rise	—	45	—	40
t195	Command rise from BUSCLK rise	—	45	—	40
t197	TC rise from BUSCLK rise	—	40	—	40
t198	TC fall from BUSCLK rise	—	40	—	35
t199	LOMEGCS delay from A16-23	—	35	—	35
t200	Hold delay from PROCCLK rise at start of T-state	5	36	5	33
t201	DSELA, B delay from OSC2 rise in MRA mode	—	35	—	35
t202	DACKA, B, C valid before DACKEN fall in MRA mode	0	—	0	—
t203	DACKA, B, C hold after DACKEN rise in MRA mode	0	—	0	—

Note: DMA address refers to A0-23, MODA0, MODA20, and -BHE.

Table 12-13. DMA to AT-Bus, On-board I/O, and ROM—Formula Specifications

			16MHz		20MHz	
Symbol	Critical Path	Formula	Min.	Max.	Min.	Max.
te194	Command low time	t194-t195	—	10	—	10
te195	Command high time	t195-t194	—	10	—	10

Table 12-14. DMA to AT-Bus, On-board I/O, and ROM—Input Requirements

		16MHz		20MHz	
Symbol	Parameters	Min.	Max.	Min.	Max.
t210	HLDA setup before BUSCLK rise †	20	—	15	—
t212	DREQ setup before BUSCLK rise †	20	—	15	—
t213	DREQ hold after BUSCLK rise †	20	—	15	—
t214	IOCHRDY setup before BUSCLK rise †	15	—	10	—
t215	IOCHRDY hold after BUSCLK rise †	7	—	5	—

† Certain input parameters, as noted, are nonrestrictive. The signal may not be recognized until the subsequent clocking period if these parameters are violated. The parameter specifies only the condition needed to guara ntee recognition of the signal on a particular clock edge.

DMA and AT-Bus Master Access to Local Memory

Tables 12-15 through 12-17 are the specification requirements for the DMA and the AT bus master to access local memory. Note: For XD, D, and PAR timing, use t145-t148 and t112. For row/column change over and CAS active, use t102, t105, and t107.

Table 12-15. DMA and AT-Bus Master Access to Local Memory—Output Responses

		16MHz		20MHz	
Symbol	Parameters	Min.	Max.	Min.	Max.
t230	-RAS high from HLDA rise	—	60	—	60
t231	A0-23 and -BHE float from -MASTER active	—	60	—	60
t232	Command float from -MASTER active	—	35	—	30
t233	-MWE rise from -XMEMR fall	—	45	—	40
t234	-MWE fall from -CAS rise	2	60	2	60
t235	-RAS inactive from -XMEMR or -XMEMW high	—	45	—	42
t236	-CAS inactive from -XMEMR or -XMEMW high	—	55	—	50
t239	-RAS active from XMEMR or XMEMW	—	36	—	36
t240	A20 valid from MODA20 during MASTER access	—	25	—	25
t241	A0 valid from MODA0 during MASTER access	—	24	—	24
t242	SDIRL, H fall from -XMEMW fall during Master write	—	40	—	40

Table 12-16. DMA and AT-Bus Master Access to Local Memory—Formula Specifications

			16MHz		20MHz	
Symbol	Critical Path	Formula	Min.	Max.	Min.	Max.
te230	RAS precharge before refresh	t230-t263-t265	—	27	—	27
te233	MWE rise before CAS fall (read)	t233-t102	—	35	—	24
te235	RAS precharge, Master	t235-t239	—	16	—	16
te239	Master read, access from RAS	t239+t145	—	51	—	51
te240	Row address to RAS	t240-t239+t106	—	25	—	25
te242	Master write, SDIR fall	t242+t147	—	65	—	65

Table 12-17. DMA and AT-Bus Master Access to Local Memory—Input Requirements

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

Refresh

The refresh timing specifications and requirements are shown in Tables 12-18 through 12-20.

Table 12-18. Refresh—Output Responses

		16MHz		20MHz	
Symbol	Parameters	Min.	Max.	Min.	Max.
t260	-REFRESH active from HLDA	—	71	—	71
t261	-REFRESH float from OSC2 rise	—	55	—	55
t262	Refresh address valid from -REFRESH active †	—	45**	—	45**
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

† Refresh address refers to MODA0, A0-9, and MA0-9.
** 40ns maximum for 836A.

Table 12-18. Refresh—Output Responses (continued)

		16MHz		20MHz	
Symbol	Parameters	Min.	Max.	Min.	Max.
t269	LOMEGCS delay from -REFRESH	—	44	—	44
t270	-CAS fall from HLDA rise in CAS-RAS HLDA refresh	*	—	*	—
t271	-REFRESH active from PROCCLK in Hidden Refresh, or in HLDA refresh in Early READY mode following memory write	—	50	—	50
t272	-XMEMR fall from PROCCLK in Hidden Refresh	—	50	—	50
t273	-XMEMR rise from PROCCLK in Hidden Refresh	—	50	—	50

* Not applicable to 82C836B. 10ns minimum for 82C836A.

Table 12-19. Refresh—Formula Specifications

			16MHz		20MHz	
Symbol	Critical Path	Formula	Min.	Max.	Min.	Max.
te260	REF to XMEMR delay in HLDA refresh	t260-t263	—	16	—	16
te270	CAS precharge in CAS-RAS refresh	t270-t103	*	—	*	—
te271	REF to XMEMR in Early READY mode	t271-t263	—	16	—	16
te271a	REF to XMEMR delay in Hidden Refresh	t271-t272	—	5**	—	5**
te272	XMEMR low time in Hidden Refresh	t272-t273	—	15***	—	15***
te273	Address hold after XMEMR rise	t273-t271	—	50	—	38

* Not applicable to 82C836B. 9ns minimum for 82C836A.
** 1ns maximum for 82C836A.
*** 4ns maximum for 836A.

Table 12-20. Refresh—Input Requirements

		16MHz		20MHz	
Symbol	Parameters	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 nonrestrictive. If these parameters are violated, the signa l may not be recognized until the subsequent clocking period. The parameter specifies only the conditions needed to guarant ee recognition of the signal on a particular clock edge.

Miscellaneous Parameters

Miscellaneous timing specifications are shown in Tables 12-21 through 12-23.

Table 12-21. Miscellaneous Parameters—Output Responses

		16MHz		20MHz	
Symbol	Parameters	Min.	Max.	Min.	Max.
t290	XRST inactive from PROCCLK rise	4	18	4	13
t291	CPURST inactive from PROCCLK rise	4	18	4	13
t292	BUSCLK delay from PROCCLK (all ratios)	—	25	—	20
t293	-BUSY delay from -NPBUSY	—	25	—	25
t294	-BUSY inactive from -XIOW fall (output to port F0H)	—	35	—	30
t295	-8042CS, -RTTCS delay from PROCCLK rise	—	75	—	70
t296	PROCCLK rise from CXIN fall	—	26	—	26
t297	PROCCLK fall from CXIN rise	—	27*	—	27*

* 26ns maximum for 82C836A.

Table 12-22. Miscellaneous Parameters—Formula Specifications

			16MHz		20MHz	
Symbol	Critical Path	Formula	Min.	Max.	Min.	Max.
te296	PROCCLK high symmetry	t296-t297	—	3	—	2
te297	PROCCLK low symmetry	t297-t296	—	3	—	2

Table 12-23. Miscellaneous Parameters—Input Requirements

		16MHz		20MHz	
Symbol	Parameters	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 during coprocessor I/O cycles	19	—	12	—
t318	-READY hold after PROCCLK rise during coprocessor I/O cycles	4	—	4	—

Local Bus Access and Cache

The local bus access and cache timing specifications are shown in Tables 12-24 through 12-27.

Table 12-24. Local Bus Access and Cache—Output Responses

		16MHz		20MHz	
Symbol	Parameters	Min.	Max.	Min.	Max.
t342	-CAS inactive from -READY rise	1	—	1	—
t343	-CAS fall from PROCCLK FALL during local memory write in Early READY mode	—	*	—	*

* Not applicable to 82C836B. 22ns maximum for 82C836A-16. 20ns maximum for 82C836A-20.

Table 12-25. Local Bus Access and Cache—Formula Specifications

			16MHz		20MHz	
Symbol	Critical Path	Formula	Min.	Max.	Min.	Max.
te343	Parity setup in Early READY write	t112-t343	—	*	—	*

* Not applicable to 82C836B. 26ns maximum for 82C836A-16. 14ns maximum for 82C836A-20.

Table 12-26. Local Bus Access and Cache—Input Requirements

		16MHz		20MHz	
Symbol	Parameters	Min.	Max.	Min.	Max.
t350	-LBA setup before PROCCLK rise	32	—	20	—
t351	-LBA hold after PROCCLK rise	3	—	2	—
t352	-READY setup before PROCCLK rise during cache read hit or LBA cycle	19	—	15	—
t353	-READY hold after PROCCLK rise during cache read hit or LBA cycle	4	—	4	—

Standy Refresh

Standby refresh timing specifications are shown in Table 12-27.

Table 12-27. Standby RefreshOutput Responses

		16MHz		20MHz	
Symbol	Parameters	Min.	Max.	Min.	Max.
t370	-CAS fall from 32KHz rise or fall during standby refresh	—	200	—	200
t371	RAS0, 3 fall after -CAS fall during standby refresh	12	—	12	—
t372	-CAS rise after RAS1, 2 fall during standby refresh	47	2000	47	2000
t373	-RAS low time during standby refresh	110	2000	110	2000
t374	-MWE rise before -RAS fall in 32KHz mode	15	—	15	—

AC Characteristics 25MHz

The following parameters are for 25MHz operation in these system configurations:

- Memory configurations up to 4 banks in MRA mode only
- Nonencoded RAS; no buffers on CAS or MD lines
- 80ns DRAM, one wait state; or 60ns DRAM, zero wait state
- Nonlatching buffer between SA-bus and CPU local address bus
- Cache (PEAKsx) or non-cache.

Unless otherwise specified, the following timing parameters are defined under capacitive load of 50pF, Vcc 4.75V to 5.25V, and temperature of 0 to 70 ° C. All the units discussed are in nanoseconds. Also, these AC specifications are subject to change.

Formula Specifications refer to the minimum or maximum result of a specified calculation involving other chip parameters. For any particular chip, formula specifications express a "tracking" relationship over temperature and voltage for the parameters involved in the formula.

CPU to Local Memory

Tables 12-28 through12-30 provides the timing requirements for the CPU to access local memory.

Table 12-28. CPU to Local Memory—Output Responses

Symbol	Parameters	Min.	Max.
t100	RAS active from PROCCLK rise	—	23
t101	RAS inactive from PROCCLK rise	—	26
t102	CAS active from PROCCLK rise	—	23
t103	CAS inactive from PROCCLK rise	—	27
t104	Row address valid from PROCCLK rise	—	32*
t105	Row address hold from PROCCLK rise	0	—
t106	Row address valid from A0-23 in	—	50
t107	Column address valid from PROCCLK rise	—	19
t108	Column address hold from PROCCLK rise	0	—
t109	Column address valid from A0-23 in	—	47
t110	-MWE fall from -CAS rise	0	36**
t111	-MWE rise from PROCCLK rise (C_L = 30pF)	—	26
t112	PARL, PARH valid from D0-15 write data in	—	55
t113	-READY delay from PROCCLK rise	4	31

* 30ns maximum for 82C836A.
** 58ns maximum for 82C836A.

Table 12-29. CPU to Local Memory—Formula Specifications

Symbol	Critical Path	Formula	Max.
te101	RAS precharge	t101-t100	19**
te102	RAS to CAS delay	t100-t102	9
te103	CAS precharge	t103-t102	7
te103a	CAS rise to RAS fall	t103-t100	6
te104	Row address setup before RAS	t104-t100	11
te105	Row address hold after RAS	t100-t105	5
te106	Row address setup before RAS	t106-t100	28
te107	Column address setup before CAS	t107-t102	6
te108	Column address hold after CAS	t102-t108	2
te109	Column address setup before CAS	t109-t102	25
te111	MWE rise before CAS fall	t111-t102	6*
te112	Write parity setup before CAS	t112-t102	10

* 13ns maximum for 82C836A.
** 5ns maximum for 82C836A.

Table 12-30. CPU to Local Memory—Input Requirements

Symbol	Parameters	Min.	Max.
t120	PARL, PARH setup before PROCCLK rise during read	7	—
t121	PARL, PARH hold after PROCCLK rise during read	5	—
t122	D0-15 setup before PROCCLK rise during read	7	—
t123	D0-15 hold after PROCCLK rise during read	5	—

CPU to AT-Bus, On-Board I/O, and ROM

Tables 12-31 through 12-33 shows the CPU AT bus, on-board I/O, and ROM accesses.

Table 12-31. CPU to AT-Bus, On-Board I/O, and ROM—Output Responses

Symbol	Parameters	Min.	Max.
t130	LOMEGCS delay from PROCCLK rise	—	35
t131	-NA delay from PROCCLK rise (C_L = 25pF)	—	15
t132	-READY delay from PROCCLK rise	4	31
t133	ALE rise from BUSCLK low	—	25
t134	ALE fall from BUSCLK high	—	15
t135	Command and -ROMCS active from BUSCLK	—	30
t136	Command and -ROMCS inactive from BUSCLK rise	—	25
t137	MODA0 delay from PROCCLK rise	—	30
t138	MODA0 rise from BUSCLK fall during bus convert	—	20
t139	MODA20 delay from A20 (if MODA20 enabled)	—	35
t140	MODA20 delay from HOLD fall	—	40
t141	SDIRL, H delay from PROCCLK rise	—	50
t142	XD0-15 delay from PROCCLK rise during write	—	50
t143	XD0-15 turn-on delay from PROCCLK rise	5	—
t144	XD0-15 turn-off delay from PROCCLK rise	—	50
t145	XD0-15 delay from D0-15 during CPU write or Master read	—	30
t146	XD0-7 delay from D8-15 during CPU write (byte swap)	—	70
t147	D0-15 delay from XD0-15 during CPU read or DMA/Master write	—	25
t148	D8-15 delay from XD0-7 during CPU read (byte swap) or DMA	—	40

Table 12-32. CPU to AT Bus and On-Board I/O and ROM—Formula Specifications

Symbol	Critical Path	Formula	Max.
te133	ALE width	t133-t134	10
te137	MODA0 hold after command rise	t136-t137	16
te139	MEMCS16 setup before ALE fall	t139-t134	15**
te145	Master read, access from CAS	t145+t102	38*
te145a	Master read, access from column address	t145+t107	34

* 36ns maximum for 82C836A.
** 20ns maximum for 82C836A.

Table 12-33. CPU to AT-Bus, On-Board I/O, and ROM—Input Requirements

Symbol	Parameters	Min.	Max.
t160	-ADS setup before PROCCLK rise**	23	—
t161	-ADS hold after PROCCLK rise**	4	—
t162	A0-23 setup before PROCCLK rise †	39	—
t163	A0-23 hold after PROCCLK rise ‡	4	—
t164	-BHE setup before PROCCLK rise †	39	—
t165	-BHE hold after PROCCLK rise ‡	4	—
t166	-DC, W/-R and M/-IO setup before PROCCLK rise †	39	—
t167	-DC, W/-R and M/-IO hold after PROCCLK rise ‡	4	—
t168	IOCHRDY setup before BUSCLK rise *	15	—
t169	IOCHRDY hold after BUSCLK rise *	15	—
t170	-0WS setup before BUSCLK fall *	10	—
t171	-0WS hold after BUSCLK fall *	20	—
t173	-IOCS16 hold after Command inactive	0	—
t174	-MEMCS16 setup before ALE fall	30***	—
t175	-MEMCS16 hold after ALE fall	10	—
t176	XD0-15 valid before read command rise	20	—
t177	-IOCS16 setup before BUSCLK rise (first BUSCLK rise after I/O command fall)	0	—

† These parameters are referenced to middle of T-state in which -ADS changes from low to high (TS).
‡ Referenced to end of TS.
* Nonrestrictive
** At start of TS.
*** 25ns minimum for 82C836A.

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

Tables 12-34 through 12-36 shows the DMA to AT bus, on-board I/O, and ROM accesses.

Table 12-34. 197DMA to AT-Bus, On-Board I/O, and ROM—Output Responses

Symbol	Parameters	Min.	Max.
t190	-DACKn or -DACKEN delay from BUSCLK rise	—	46
t191	ALE rise from HLDA rise	—	70
t192	DMA Address delay from BUSCLK rise. Note: DMA address refers to A0-23, MODA0, MODA20, and -BHE	—	35
t194	Command fall from BUSCLK rise	—	40
t195	Command rise from BUSCLK rise	—	40
t197	TC rise from BUSCLK rise	—	40
t198	TC fall from BUSCLK rise	—	35
t199	LOMEGCS delay from A16-23	—	35
t200	HOLD delay from PROCCLK rise at start of T-state	5	25
t201	DSELA, B delay from OSC2 rise (MRA mode)	—	35
t202	DACKA, B, C valid before DACKEN fall (MRA mode)	0	—
t203	DACKA, B, C hold after DACKEN rise (MRA mode)	0	—

Table 12-35. DMA to AT-Bus, On-Board I/O, and ROM—Formula Specifications

Symbol	Critical Path	Formula	Max.
te194	Command low time	t194-t195	10
te195	Command high time	t195-t194	10

Table 12-36. DMA to AT-Bus, On-Board I/O, and ROM—Input Requirements

Symbol	Parameters	Min.	Max.
t210	HLDA setup before BUSCLK rise †	15	—
t212	DREQ setup before BUSCLK rise †	15	—
t213	DREQ hold after BUSCLK rise †	15	—
t214	IOCHRDY setup before BUSCLK rise †	10	—
t215	IOCHRDY hold after BUSCLK rise †	5	—

† These parameters are nonrestrictive. The signal may not be recognized until the subsequent clocking period if these parameters are violated. The parameter specifies only the condition needed to guarantee recognition of the signal on a particular clock edge.

DMA and AT-Bus Master Access to Local Memory

The DMA and AT bus master access to local memory are shown in Tables 12-37 through 12-39.

Table 12-37. DMA and AT-Bus Master Access to Local Memory—Output Responses

Symbol	Parameters	Min.	Max.
t230	-RAS high from HLDA rise	—	50
t231	A0-23 and -BHE float from -MASTER active	—	60
t232	Command float from -MASTER active	—	30
t233	-MWE rise from -XMEMR fall	—	40*
t234	-MWE fall from -CAS rise	2	60
t235	-RAS inactive from -XMEMR or -XMEMW high	—	42
t236	-CAS inactive from -XMEMR or -XMEMW high	—	50
t239	-RAS active from XMEMR or XMEMW	—	36
t240	A20 valid from MODA20 during MASTER access	—	25
t241	A0 valid from MODA0 during MASTER	—	19
t242	SDIRL, H fall from -XMEMW fall (Master Write)	—	40

* 35ns maximum for 82C836A.

Table 12-38. DMA and AT-Bus Master Access to Local Memory—Formula Specifications

Symbol	Critical Path	Formula	Max.
te230	RAS precharge before refresh	t230-t263-t265	27
te233	MWE rise before CAS fall (read)	t233-t102	18
te235	RAS precharge, Master	t235-t239	16
te239	Master read, access from RAS	t239+t145	51
te240	Row address setup before RAS	t240-t239+t106	25
te242	Master write, SDIR fall	t242+t147	65

Table 12-39. DMA and AT-Bus Master Access to Local Memory—Input Requirements

Symbol	Parameters	Min.	Max.
t250	PARL, PARH setup before -XMEMR rise during memory read	20	—
t251	PARL, PARH hold after -XMEMR rise during memory read	0	—

Refresh

The refresh timing specifications are shown in Tables 12-40 through 12-42

Table 12-40. Refresh—Output Responses

Symbol	Parameters	Min.	Max.
t260	-REFRESH active from HLDA	—	71
t261	-REFRESH float from OSC2 rise	—	55
t262	Refresh address valid from -REFRESH active. Note: Refresh address refers to MODA0, A0-9 and MA0-9	—	45*
t263	-XMEMR active from OSC2 rise	—	55
t264	-XMEMR inactive from OSC2 rise	—	50
t265	-RAS0, -RAS3 active from -XMEMR fall	—	30
t266	-RAS0, -RAS3 inactive from -XMEMR rise	—	25
t267	-RAS1, -RAS2 active from -XMEMR fall	—	70
t268	-RAS1, -RAS2 inactive from -XMEMR rise	—	70
t269	LOMEGCS delay from -REFRESH	—	40
t270	-CAS fall from HLDA rise in CAS-RAS HLDA refesh	**	—
t271	-REFRESH active from PROCCLK in Hidden Refresh, or in HLDA refresh in Early READY mode following memory write	—	50
t272	-XMEMR fall from PROCCLK in Hidden Refresh	—	50
t273	-XMEMR rise from PROCCLK in Hidden Refresh	—	50

* 40ns maximum for 82C836A.
** Not applicable to 82C836B. 10ns minimum for 82C836A.

Table 12-41. Refresh—Formula Specifications

Symbol	Critical Path	Formula	Min.	Max.
te260	REF to XMEMR delay in HLDA refresh	t260-t263	—	16
te261	REF float after XMEMR rise	t261-t264	—	49
te270	CAS precharge in CAS-RAS refresh	t270-t103	*	—
te271	REF to XMEMR in early READY mode	t271-t263	—	16
te271a	REF to XMEMR delay in hidden refresh	t271-t272	—	5**
te272	XMEMR low time in hidden refresh	t272-t273	—	15***
te273	Address hold after XMEMR rise	t273-t271	—	23

* Not applicable for 82C836B. 9ns minimum for 82C836A.
** 1ns maximum for 82C836A.
*** 4ns maximum for 836A.

Table 12-42. Refresh—Input Requirements

Symbol	Parameters	Min.	Max.
t280	IOCHRDY setup before OSC2 rise †	15	—
t281	IOCHRDY hold after OSC2 rise †	5	—
t282	-REFRESH setup before OSC2 rise †	20	—

† These parameters are nonrestrictive. The signal may not be recognized until the subsequent clock ing period if these parameters are violated. The parameter specifies only the conditions needed to guarantee re cognition of the signal on a particular clock edge.

Miscellaneous Parameters

Miscellaneous timing specifications are shown in Tables 12-43 through 12-45.

Table 12-43. Miscellaneous Parameters—Output Responses

Symbol	Parameters	Min.	Max.
t290	XRST inactive from PROCCLK rise (C_L = 30pF)	4	12
t291	CPURST inactive from PROCCLK rise (C_L = 30pF)	4	12
t292	BUSCLK delay from PROCCLK (all ratios)	—	20
t293	-BUSY delay from -NPBUSY	—	25
t294	-BUSY inactive from -XIOW fall (output to port F0H)	—	30
t295	-8042CS, -RTTCS delay from PROCCLK rise	—	70
t296	PROCCLK rise from CXIN fall	—	26
t297	PROCCLK fall from CXIN rise	—	27*

* 26ns maximum for 82C836A.

Table 12-44. Miscellaneous Parameters—Formula Specifications

Symbol	Critical Path	Formula	Min.	Max.
te296	PROCCLK high symmetry	t296-t297	—	2
te297	PROCCLK low symmetry	t297-t296	—	2

Table 12-45. Miscellaneous Parameters—Input Requirements

Symbol	Parameters	Min.	Max.
t310	CXIN cycle time	20	—
t311	CXIN low time	9	—
t312	CXIN high time	9	—
t313	-NPBUSY active pulse width	30	—
t314	-NPERR fall after -BUSY fall	0	—
t315	-NPERR fall before -NPBUSY rise	25	—
t316	-NPERR active pulse width	30	—
t317	-READY setup before PROCCLK rise	9	—
t318	-READY hold after PROCCLK rise	4	—

Note: t317 and t318 apply to coprocessor accesses only.

Local Bus Access and Cache

The local bus access and cache timing specifications are shown in Tables 12-46 through 12-48.

Table 12-46. Local Bus Access and Cache—Output Responses

Symbol	Parameters	Min.	Max.
t342	-CAS inactive from -READY rise	1	—
t343	-CAS fall from PROCCLK fall during local memory write in Early READY mode	—	*

* Not applicable for 82C836B. 20ns maximum for 82C836A.

Table 12-47. Local Bus Access and Cache—Formula Specifications

Symbol	Critical Path	Formula	Max.
te343	Parity setup in early READY write	t112-t343	*

* Not applicable for 82C836B. 14ns maximum for 82C836A.

Table 12-48. Local Bus Access and Cache—Input Requirements

Symbol	Parameters	Min.	Max.
t350	-LBA setup before PROCCLK rise	15	—
t351	-LBA hold after PROCCLK rise	2	—
t352	-READY setup before PROCCLK rise	14	—
t353	-READY hold after PROCCLK rise	4	—

Standby Refresh

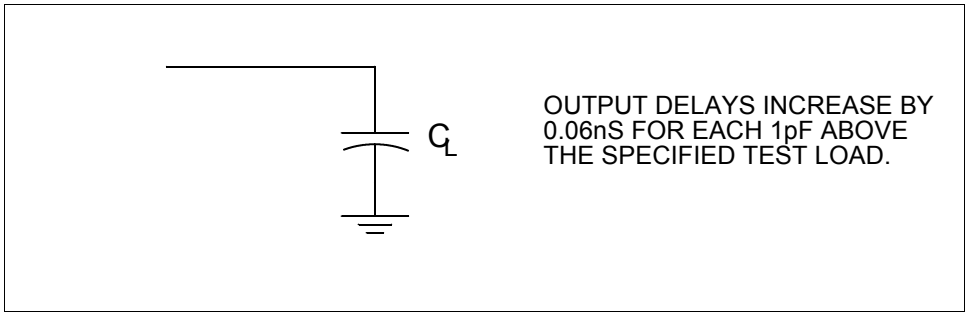
Standby refresh timing specifications are shown in Table 12-49.

Table 12-49. Output Responses—Standby Refresh

Symbol	Parameters	Min.	Max.
t370	-CAS fall from 32KHz rise or fall	—	200
t371	-RAS0, 3 fall after -CAS fall	12	—
t372	-CAS rise after -RAS1, 2 fall	47	2000
t373	-RAS low time	110	2000
t374	-MWE rise before -RAS fall	15	—

Figure 12-1 shows the circuit load for the AC specifications of the 82C836.

Figure 12-1. Load Circuit



Section 13

Timing Diagrams

This section provides diagrammatic representations of each AC timing parameter. Refer to *Section 11, System Timing Relationships* for further details of event sequences and clock cycle counts.

Figure 13-1. *Timing Waveforms*

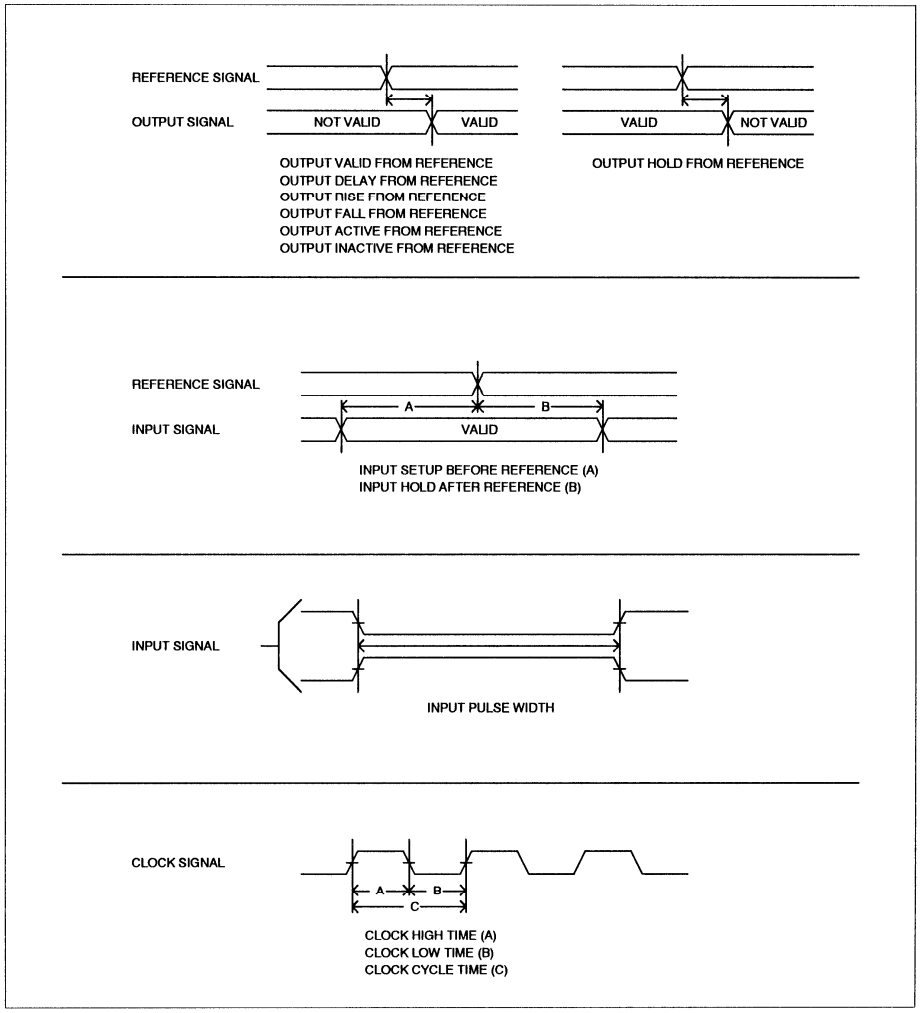


Figure 13-2. CPU to Local Memory—Output Responses and Input Requirements

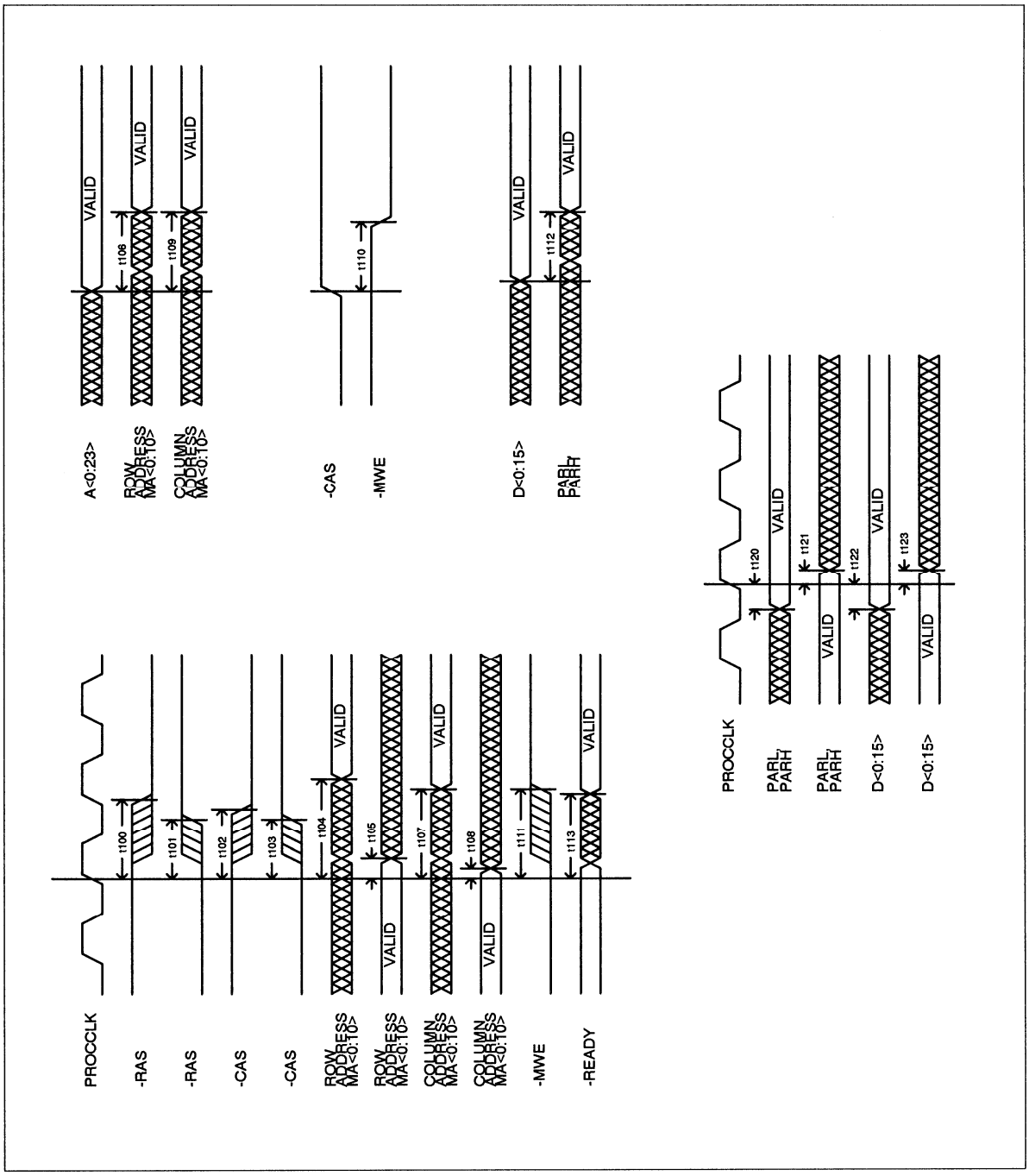


Figure 13-3. CPU AT-Bus, On-Board I/O and ROM—Output Responses

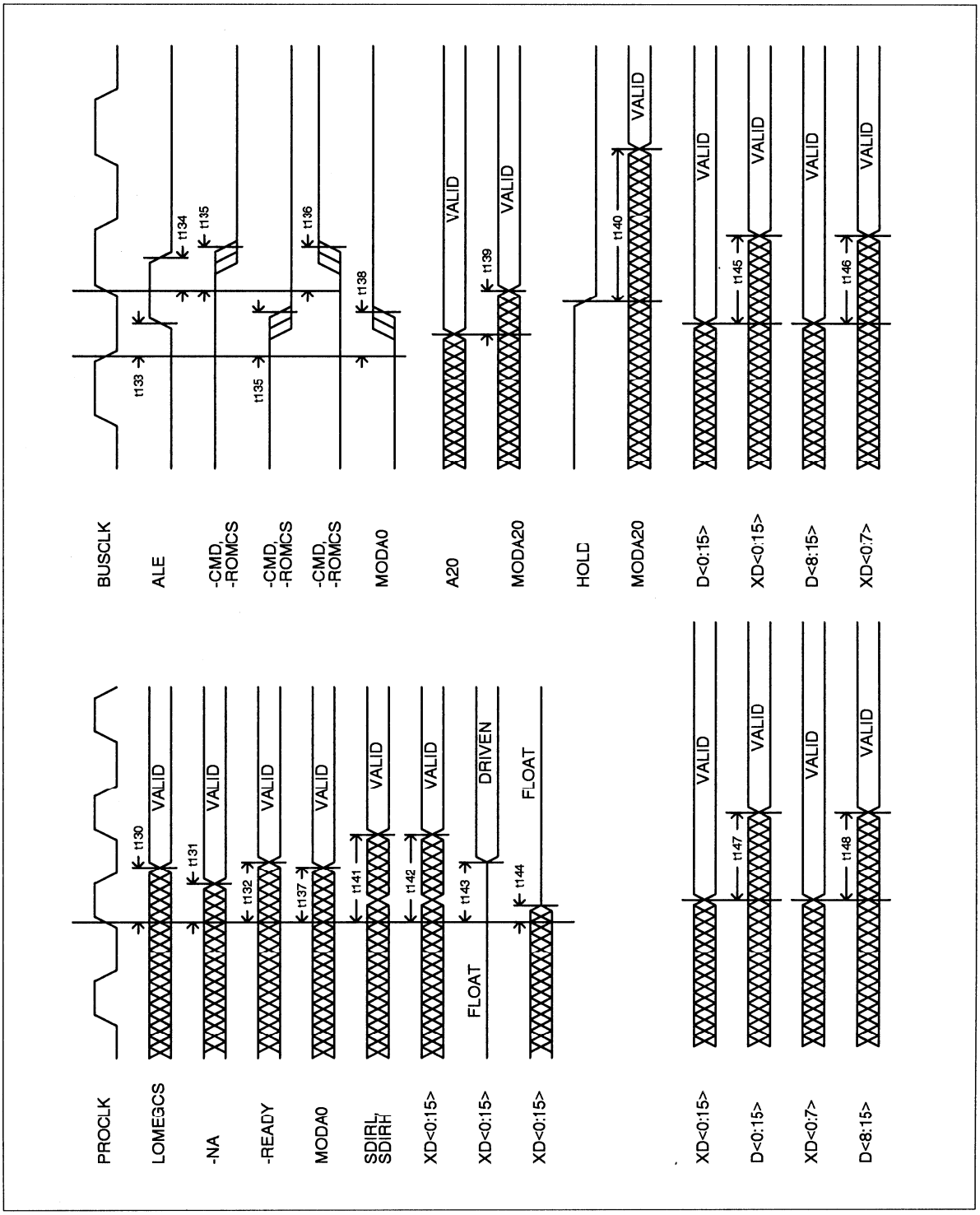


Figure 13-4. *CPU to AT-Bus, On-Board I/O and ROM—Input Requirements*

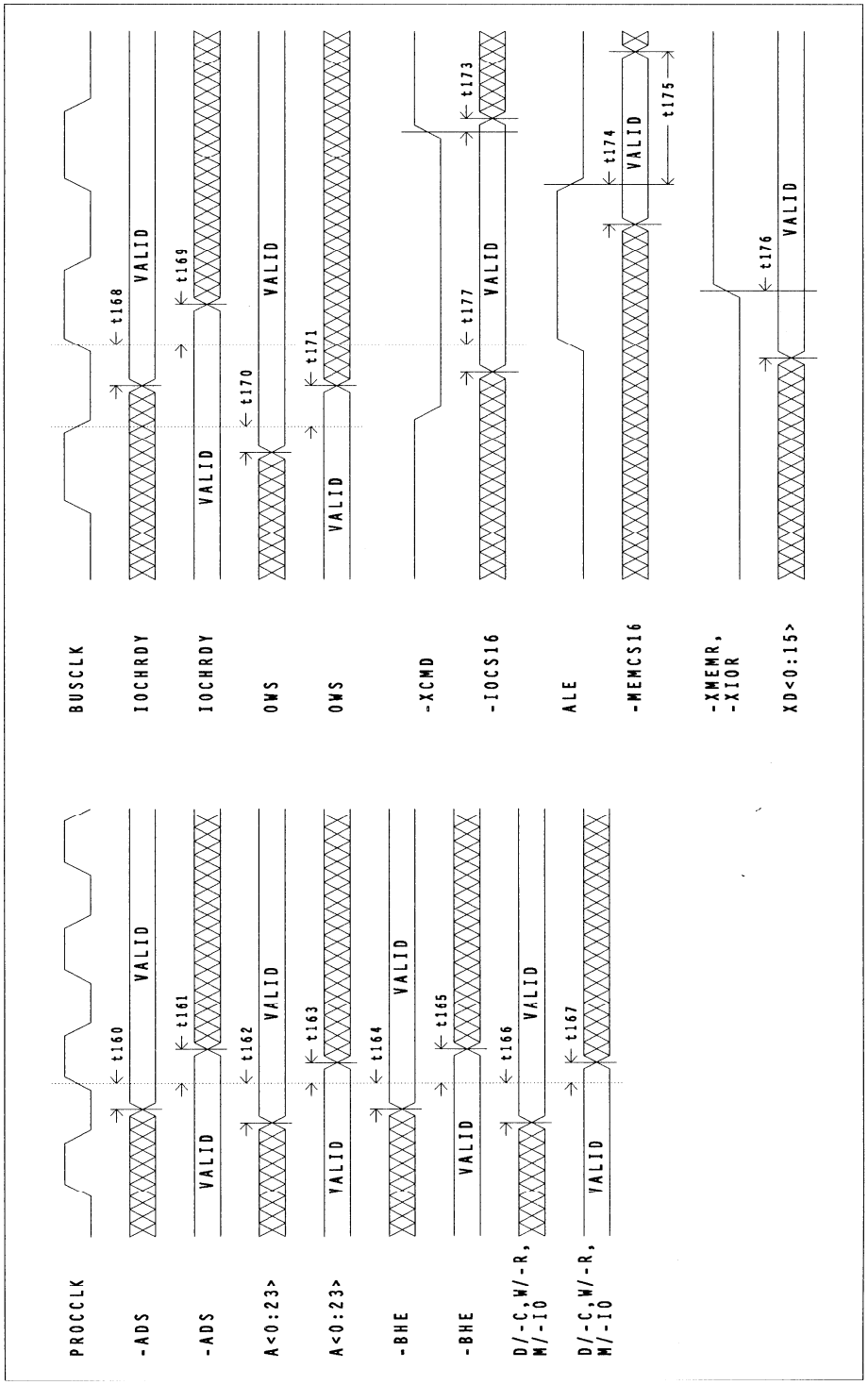


Figure 13-5. *DMA to AT-Bus, On-Board I/O, and ROM—Output Responses*

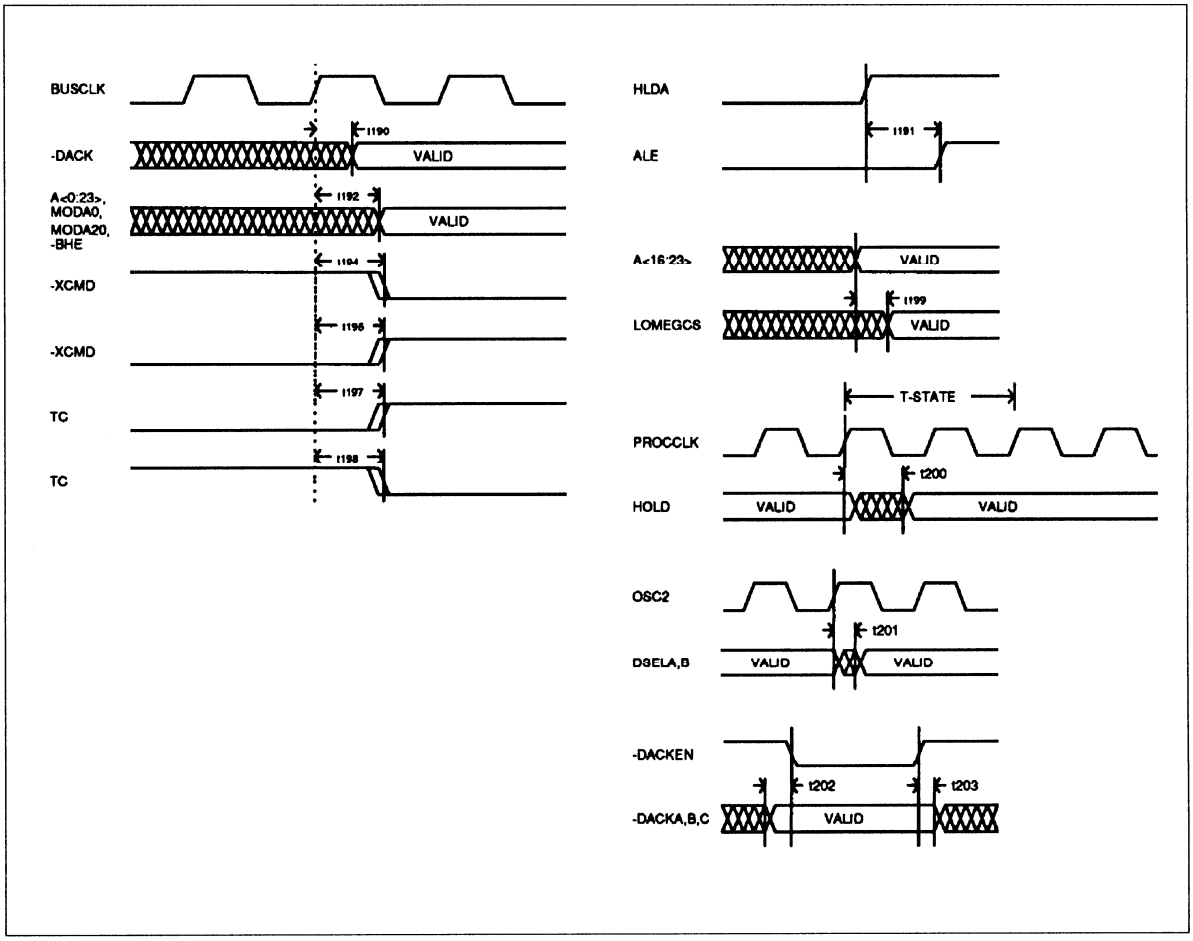


Figure 13-6. DMA to AT-Bus, On-Board I/O and ROM—Input Requirements

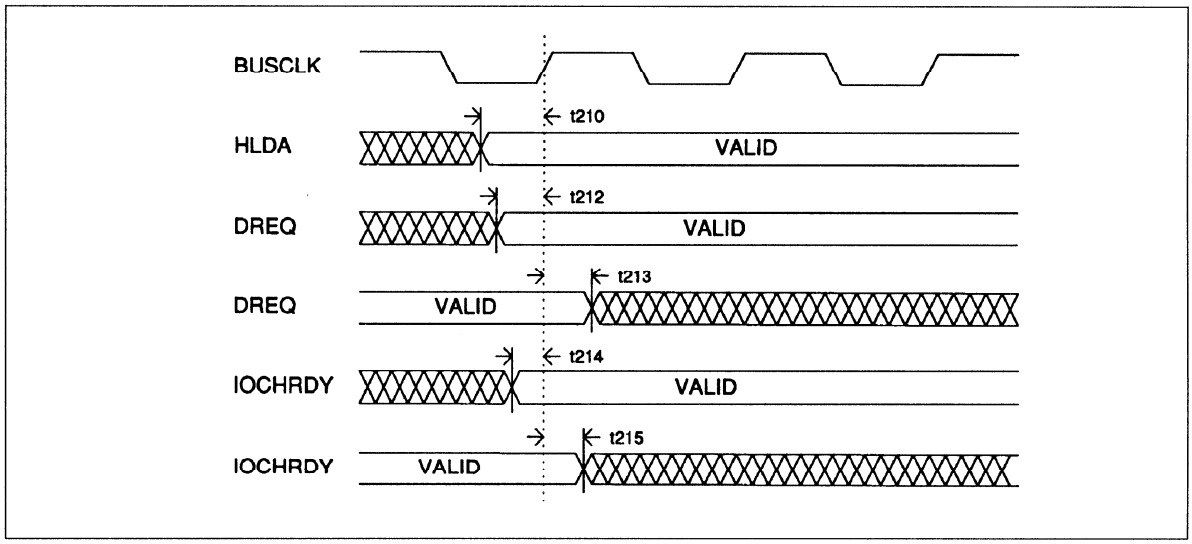


Figure 13-7. DMA and AT-Bus Master Access to Local Memory—Output Responses

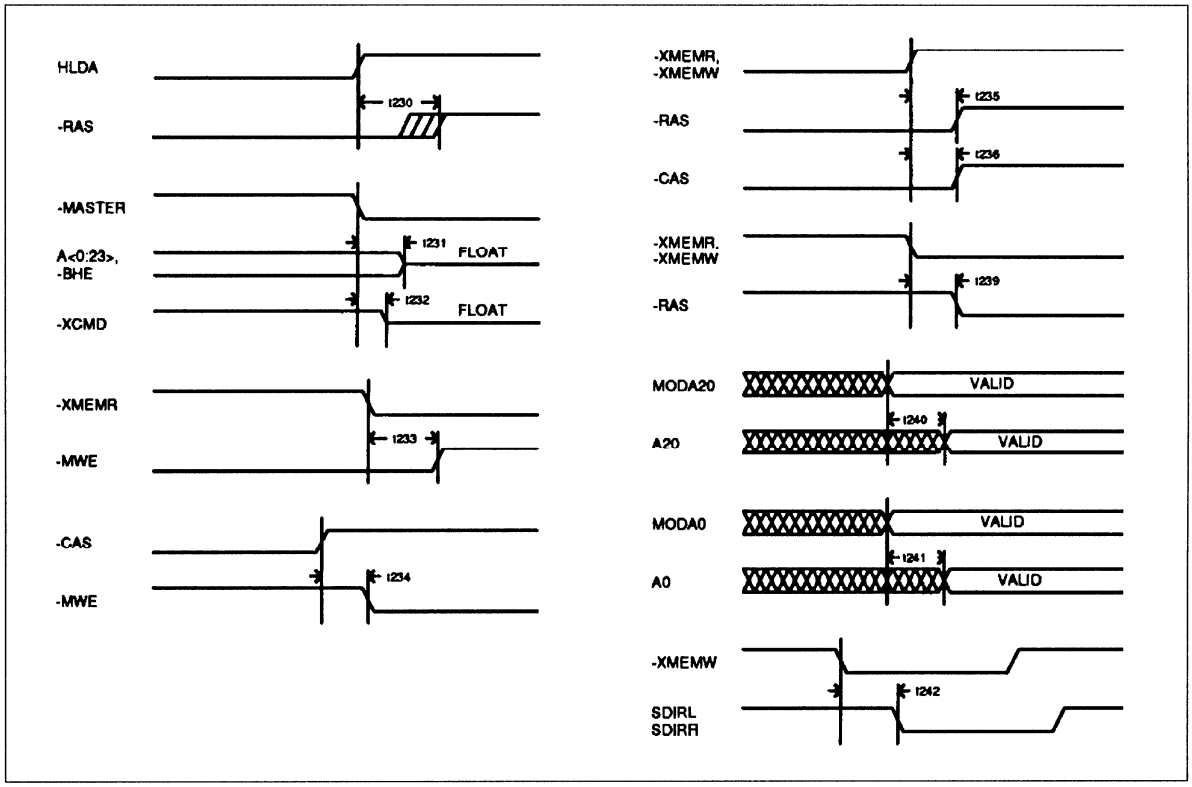


Figure 13-8. DMA and AT-Bus Master Access to Local Memory—Input Requirements

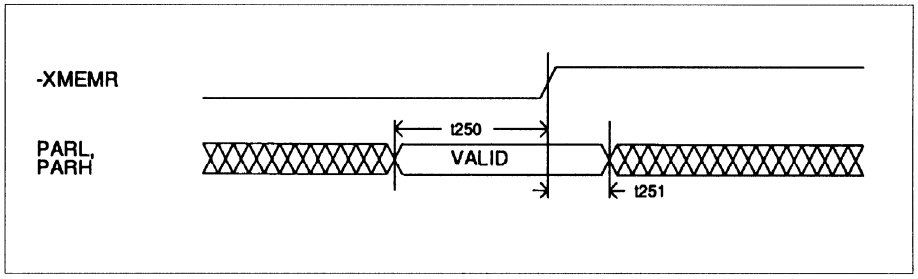


Figure 13-9. Refresh—Input Requirements

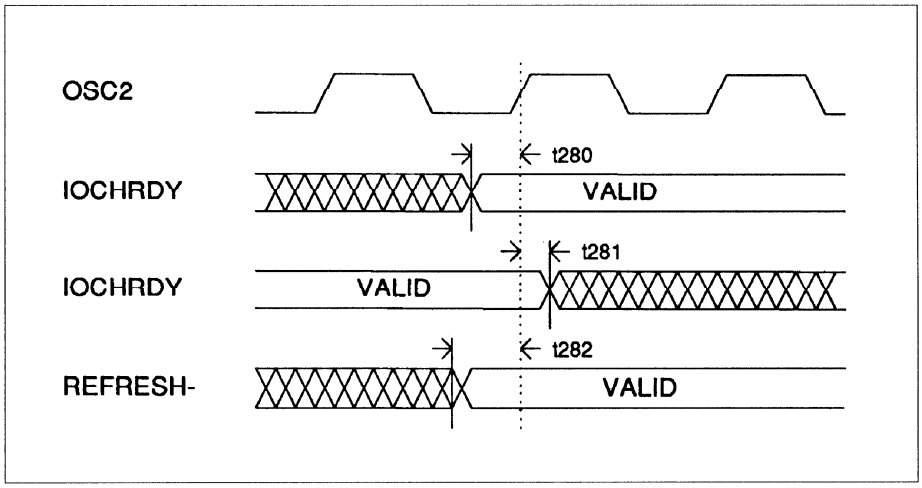


Figure 13-10. *Refresh—Output Responses*

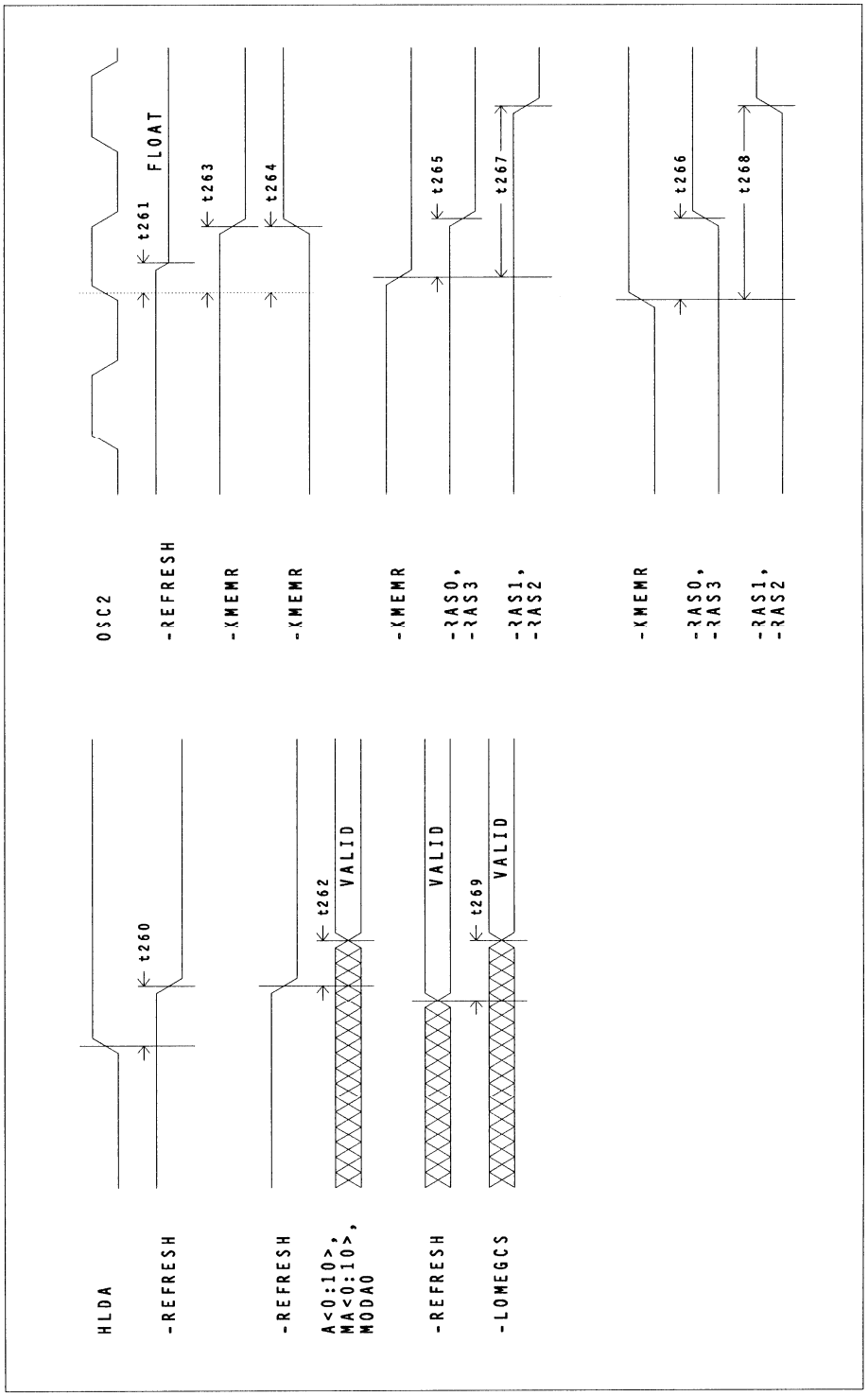


Figure 13-11. *Miscellaneous Parameters—Output Responses*

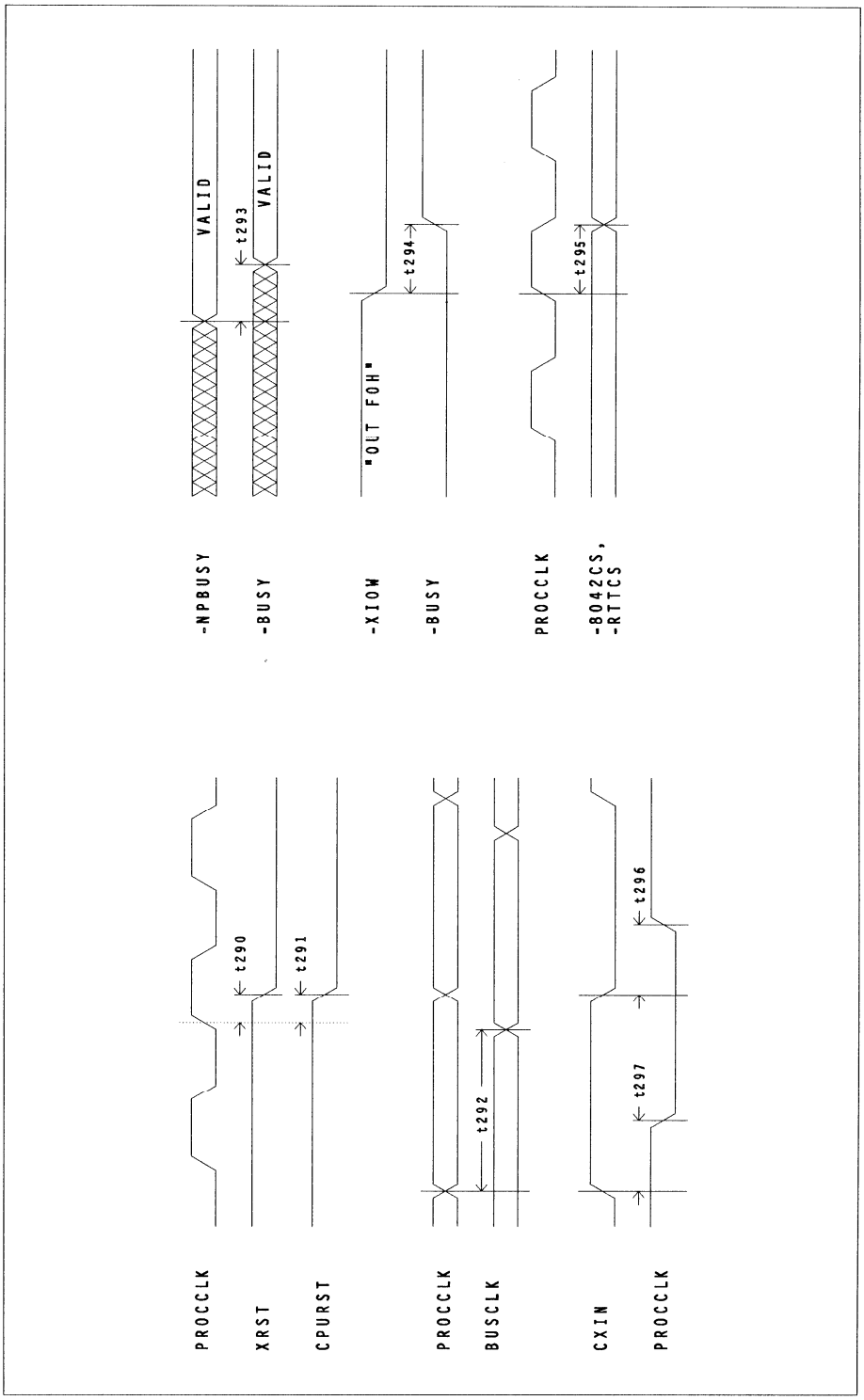


Figure 13-12. Miscellaneous Parameters—Input Requirements

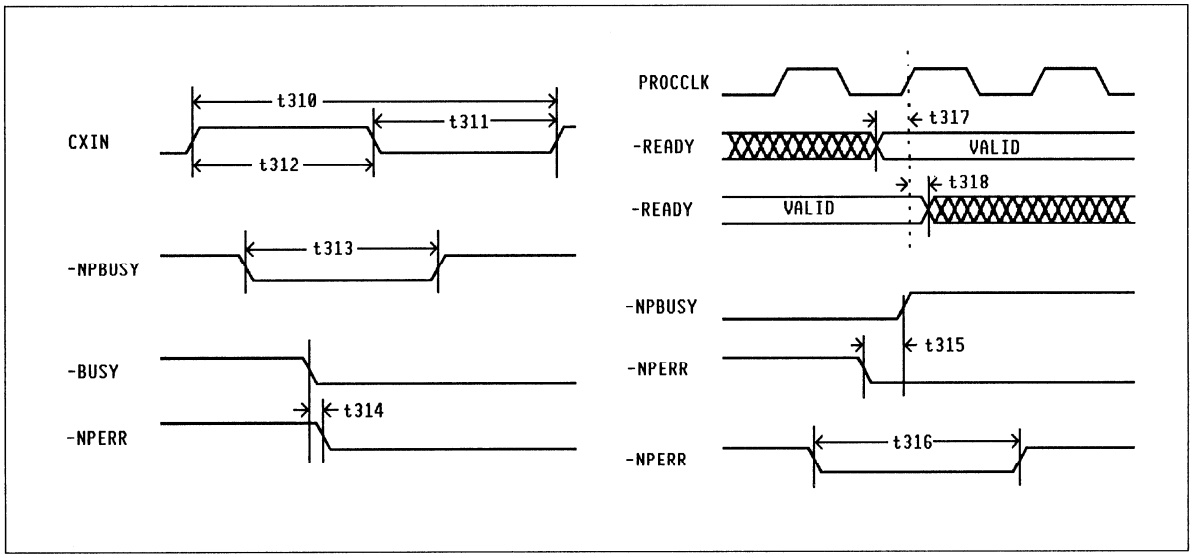


Figure 13-13. Local Bus Access and Cache—Output Responses

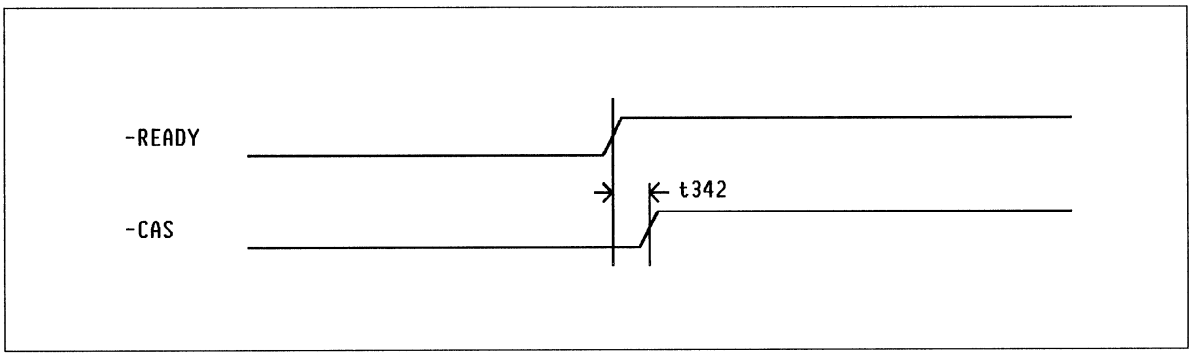


Figure 13-14. *Local Bus Access and Cache—Input Requirements*

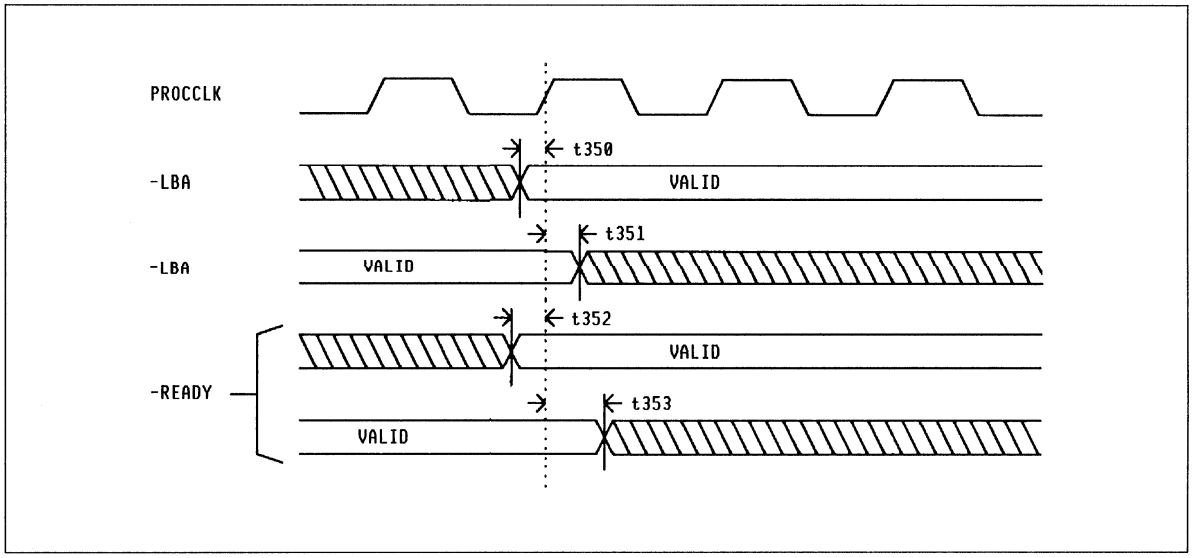
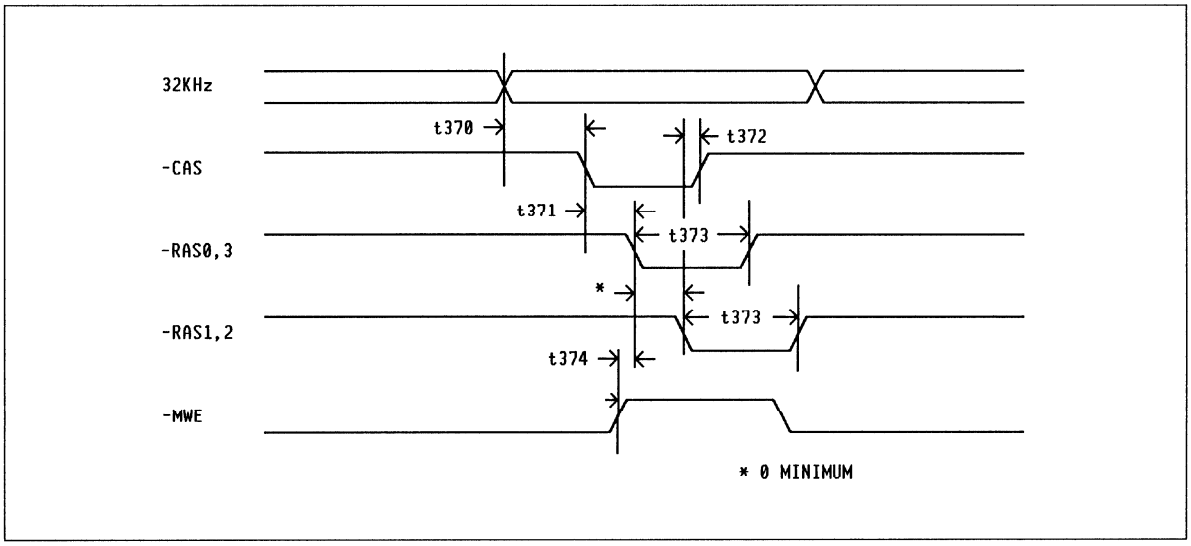


Figure 13-15. *Standby Refresh—Output Responses*

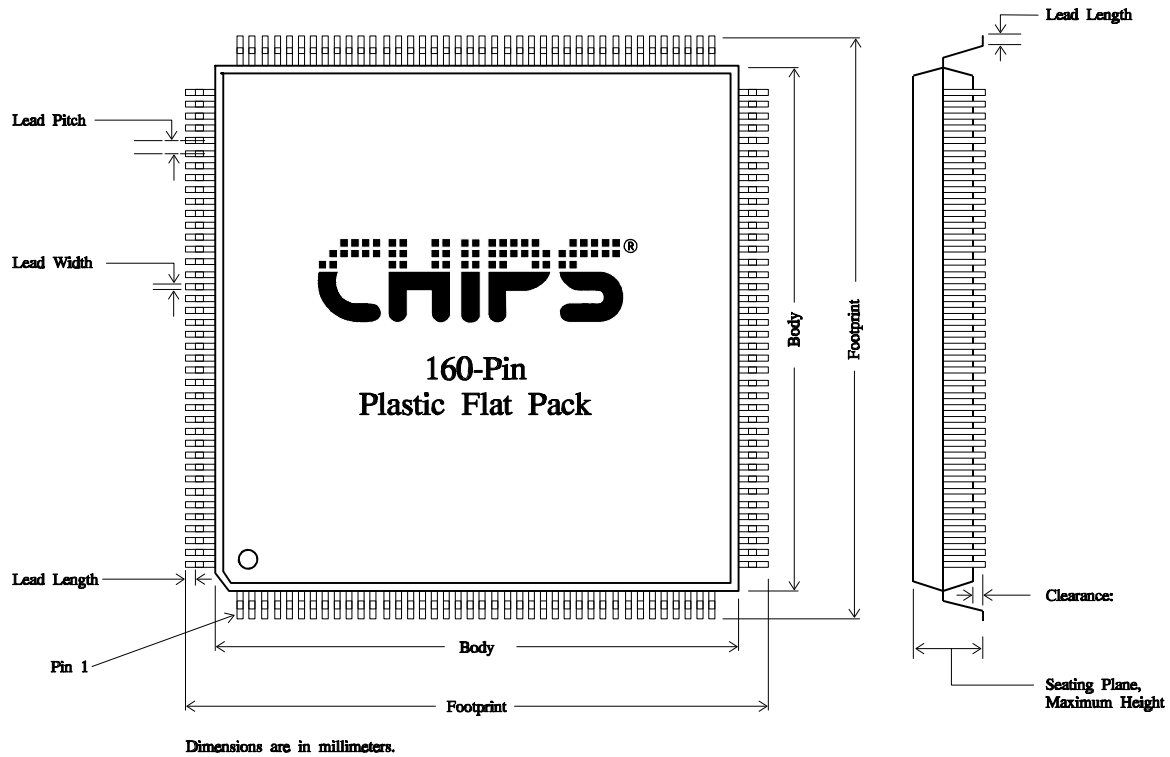


Section 14

Mechanical Specifications

The 82C836 CHIPSet is packaged in a 160-pin plastic flat pack. The dimensions are shown in Figure 14-1.

Figure 14-1. Package Dimensions



Part	Lead Pitch	Lead Width	Lead Length	Body	Footprint	Maximum Height	Clearance	Package
82C836A	0.65 TYP.	0.30 ±0.10	0.80 ±0.20	28.00 ±0.20	32.00 ±0.40	4.45	0.35/0.65	T
82C836B	0.65 TYP.	0.34 ±0.09	0.80 ±0.20	28.00 ±0.20	32.00 ±0.30	4.00	0/0.25	S

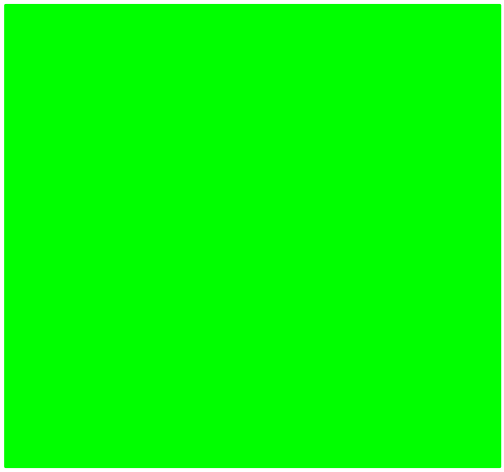
Appendix A

Differences Between 82C836A and 82C836B

The 82C836B corrects various anomalies found in the 82C836A. Refer to Product Alert 204.2/2-92 for details regarding these anomalies. The additional differences between 82C836A and 82C836B are as follows:

For the 82C836A:

- ICRs 61H and 62H are reserved. Fast video mode is not supported.
- ICR 46H bit 5 is reserved. There is no option to remap high ROM addresses above 15MB to shadow RAM below 1MB.
- ICR 64H bit 7 is reserved. The 82C836A always ignores any CPU cycles that begin during CPU reset.
- The 82C836A does not support hidden refresh.
- During DRAM write operations in early READY mode, the 82C836A delays CAS assertion by one half-cycle of PROCCLK compared to the 82C836B. Although this allows extra margin for write data and parity setup before CAS falls, it also reduces the worst-case timing margin for write data hold after CAS falls. The 82C836B greatly reduces the parity generation delay as well as improving CAS timing to allow an increased worst-case margin for DRAM write data hold time.
- Several differences in AC timing parameters are noted in Section 12, System Characteristics.



Chips and Technologies, Inc.
2950 Zanker Road
San Jose, California 95134
Phone: 408-434-0600
FAX: 408-894-2080

Publication No.: DS96.3
Stock No.: 010836-003
Revision No.: 3.0