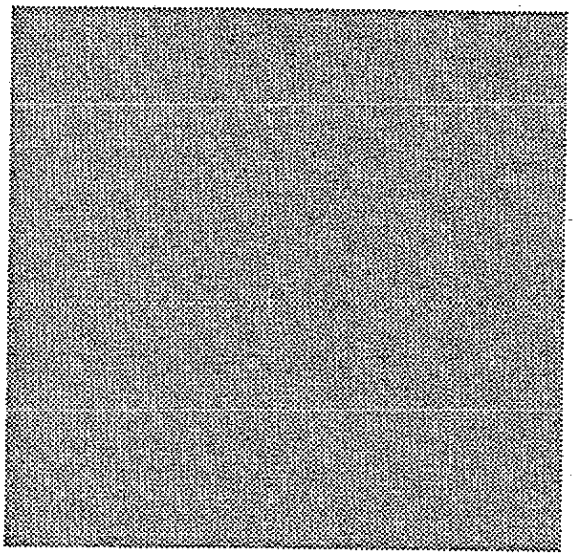
CS82310
CHIPSet



PEAK/DM

Data Book

February 1991

R E L I M I N A R Y

CHIPS®

CHIPS and Technologies, Inc.
3050 Zanker Road
San Jose, California 95134
Phone: 408-434-0600

Contents

Volume I: PEAK/DM Overview

Volume II: 82C351 CPU/Cache/DRAM Controller

List of Figures

Volume I: PEAK/DM Overview

Volume II: 82C351 CPU/Cache/DRAM Controller

List of Tables

Volume II: 82C351 CPU/Cache/DRAM Controller

Volume III: 82C355 Data Buffer

Volume IV: 82C356 Peripheral Controller

List of Registers

Volume II: 82C351 CPU/Cache/DRAM Controller

Section 2: 351 Configuration Registers

Index Registers

I/O Port Registers

Volume IV: 82C356 Peripheral Controller

Section 2: 356 Configuration Registers

Index Registers

I/O Port Registers

VOLUME

PEAK/DM Overview

Section 1

Introduction

Section 1

Introduction

1.1 Features

- Supports 25 and 33 MHz 386DX cache based systems
- 100% IBM PC/AT compatible
- Supports asynchronous AT bus timing through independent clock or division of CPU clock
- Requires only 16 IC's plus memory for a 386/AT cache based PC/AT
- Includes an integrated CPU/Cache/DRAM/bus controller
- Enhanced performance of the 386DX and memory system due to simultaneous activation of cache and DRAM accesses by the integrated controller
  - Zero wait state non-pipelined read hit access
  - Buffered write through DRAM update scheme to minimize write cycle penalty
  - Supports 32KB, 64KB, 128KB and 256KB direct mapped cache
  - Supports 4 blocks (of variable size from 4KB to 4MB) of main memory as programmable non-cacheable address space
- Utilizes tightly coupled 386DX interface
- 80387 and Weitek 3167 coprocessor interface logic
- Supports up to 128MB of local memory through flexible memory architecture
  - Programmable wait states and RAS precharge time for each block (2 pair of banks)
  - Supports 256Kb, 1Mb, and 4Mb DRAMS in configurations of up to 4 blocks (8 banks)
  - Supports staggered RAS during refresh
  - Supports hidden refresh
- Supports shadowing of BIOS EPROMs
- Supports dual 8-bit or single 16-bit EPROMs
- Features integrated high speed cache tag comparator
- Provides built in hardware for direct tag RAM testing
- Provides posted or non-posted writes

1.2 82C351 CPU/Cache/DRAM/Controller

By integrating both the cache and DRAM control functions in one chip, the 82C351 supports simultaneous activation of cache and DRAM accesses; thereby minimizing the cache miss penalty. It has hardware support to allow the user to designate up to four blocks (of variable size from 4KB to 4MB) of main memory as non-cacheable address space.

The 82C351 cache controller supports a direct mapped cache architecture and cache sizes of 32KB, 64KB, 128KB, or 256KB. Memory write updates are implemented using a buffered write-through scheme. The 82C351 is available in a 160 pin PFP package.

1.3 82C355 Data Buffer

The 82C355 bus controller contains the data buffers used to interface between the local and system memory buses and a path for the AT data bus. In addition to having high current bus drive, it also performs conversions between the different sized data paths and provides parity generation and checking. The 82C355 is available in a 120 pin PFP package.

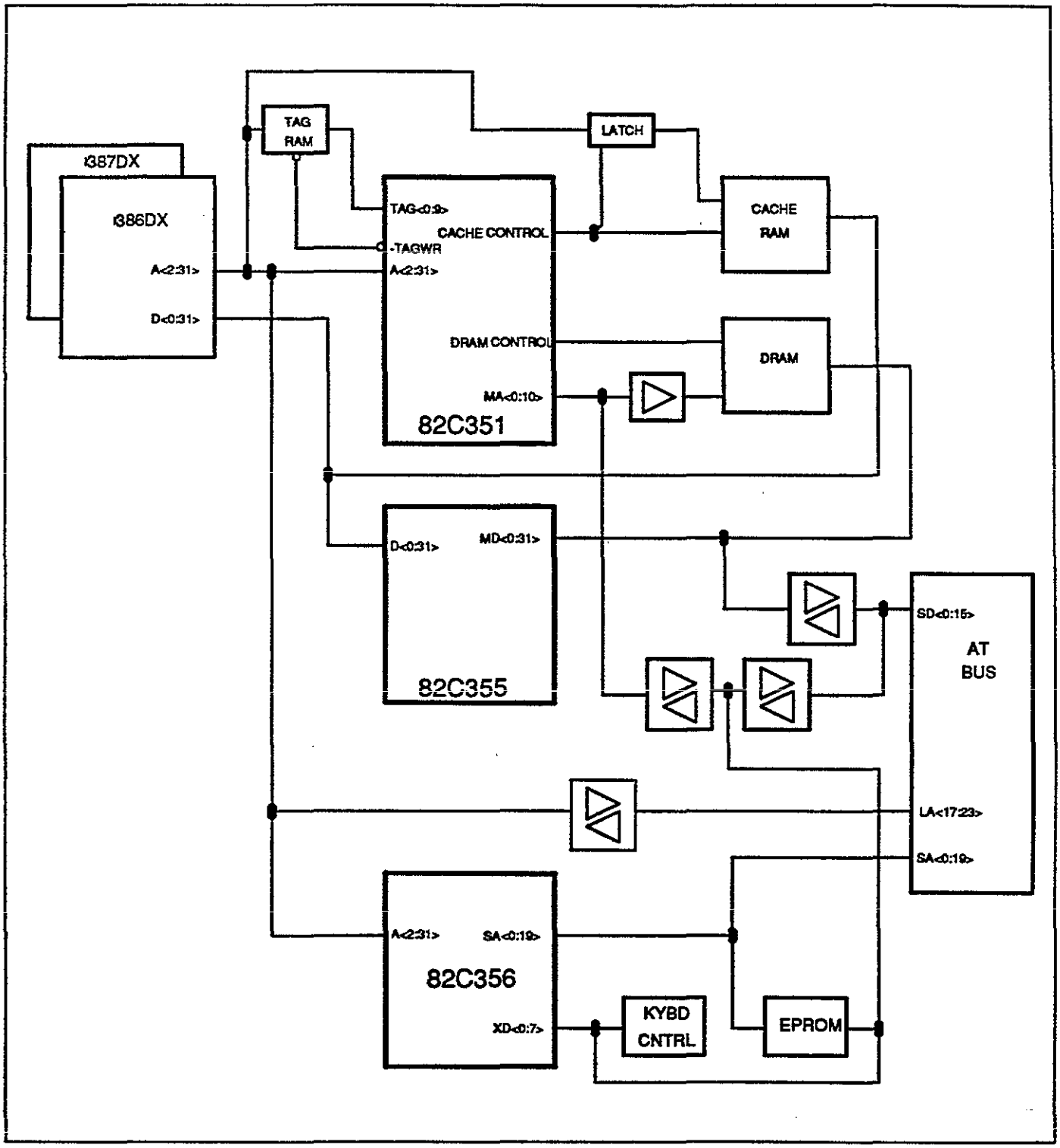
1.4 82C356 Peripheral Controller

The 82C356 peripheral controller contains the address buffers used to interface between the processor address bus (A<23:2>) and the system address bus (SA<19:0>). It also contains an equivalent 82C206 integrated peripheral controller that incorporates:

- Two 8237 DMA controllers
- Two 8259 interrupt controllers
- One 8254 timer/counter
- One MC146818 real time clock
- Several TTL/SSI interface logic chips.

The 82C356 is available in a 144 pin PFP package.

Figure 1-1. PEAK/DM System Block Diagram



This page is intentionally left blank.

Section 2

Manual Conventions

- 2.1 What You Need to Know
- 2.2 About This Manual
- 2.3 Typographical Conventions

Section 2

Manual Conventions

2.1 What You Need to Know

We assume you understand basic computer concepts, operations, components, and terminology. We also assume you understand the architecture, structure, and operation of an IBM PC/AT; as well as how the various components relate to one another to effect overall operation. We further assume you have an understanding of the various memory types and operations.

Based on these assumptions, the explanations presented in this manual therefore assume you have a working knowledge of component parts and their operation outside our CHIPSet. With the explanation of our CHIPSet, you can, with your understanding of the other components, utilize our CHIPSet to enhance the overall performance and excellence of the product.

If you are unfamiliar with these concepts, please consult the appropriate documentation to enhance your understanding of the explanations presented in this manual.

2.2 About This Manual

Here's what you'll find in this manual:

Volume I: PEAK/DM Overview

- *Section 1: Introduction* provides basic information about the PEAK/DM.
- *Section 2: Manual Conventions* gives an overview of this manual.

Volume II: 82C351 CPU/Cache/DRAM Controller

- *Section 1: 351 Functional Description* identifies the functioning of the 82C351.
- *Section 2: 351 Configuration Registers* details the internal registers within the Peak/DM chipset.
- *Section 3: 351 Pin Descriptions* provides tables describing each pin and its function.
- *Section 4: 351 Physical Characteristics* reports the range of absolute maximum ratings, operating conditions, and capacitive characteristics.
- *Section 5: 351 DC/AC Characteristics* identifies the DC and AC Characteristics associated with the 82C351.
- *Section 6: 351 Timing Diagrams* details the timings of the 82C351.

- *Section 7: 351 Physical Dimensions* identifies the dimensions of the 82C351 PFP package.

Volume III: 82C355 Data Buffer

- *Section 1: 355 Functional Description* describes the 82C355 chip including an in-depth functional description.
- *Section 2: 355 Pin Descriptions* provides tables describing each pin and its function.
- *Section 3: 355 Physical Characteristics* reports the range of absolute maximum ratings, operating conditions, and capacitive characteristics.
- *Section 4: 355 DC/AC Characteristics* identifies the DC and AC Characteristics associated with the 82C355.
- *Section 5: 355 Timing Diagrams* details the timings of the 82C355.
- *Section 6: 355 Physical Dimensions* identifies the dimensions of the 82C355 PFP package.

Volume IV: 82C356 Peripheral Controller

- *Section 1: 356 Functional Description* provides an in-depth functional description of the 82C356 chip.
- *Section 2: 356 Configuration Registers* details the internal registers within the Peak/DM chipset.
- *Section 3: 356 Pin Descriptions* provides tables describing each pin and its function.
- *Section 4: 356 Physical Characteristics* reports the range of absolute maximum ratings, operating conditions, and capacitive characteristics.
- *Section 5: 356 DC/AC Characteristics* identifies the DC and AC Characteristics associated with the 82C356.
- *Section 6: 356 Timing Diagrams* details the timings of the 82C356.
- *Section 7: 356 Physical Dimensions* identifies the dimensions of the 82C356 PFP package.

2.3 Typographical Conventions

The following conventions are used throughout this manual:

- REGnH identifies the internal register of index n in hexadecimal notation.
- REGnH <x:y> indicates the bit field from bit x to bit y with index n in hexadecimal notation.
- (xxx) signifies the default value after power on Reset; where xxx are register bits.
- Signal names are identified in full caps; for example, READY.
- A dash before the signal name indicates an active low signal; for example, -READY.
- Section and Volume titles are shown in *italic* type for example; *Section 7.2 Clock Input Timing* is found in *Volume IV: 82C356 Peripheral Controller*.

The following terms are used throughout this document:

- MB = megabyte
- Mb = megabit
- KB = kilobyte
- Kb = kilobit
- MHz = megahertz
- ns = nanoseconds

This page is intentionally left blank.

VOLUME

82C351 CPU/Cache/ DRAM Controller

Section 1

351 Functional Description

Section 1

351 Functional Description

1.1 Features

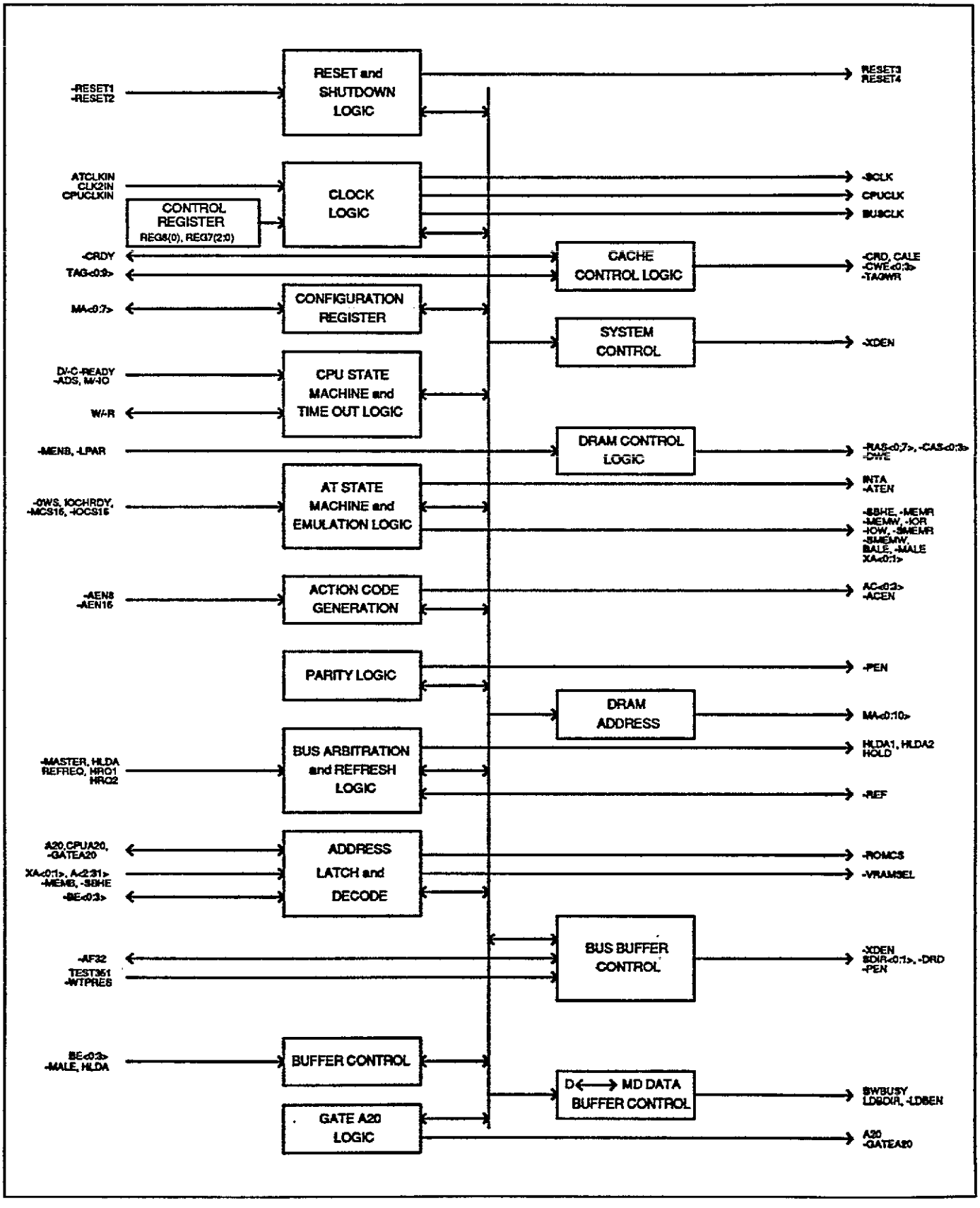
The 82C351 interfaces directly with the 386 DX and controls all bus accesses including CPU, DRAM, SRAM (cache) and AT bus cycles. The 82C351 includes an integrated Cache and DRAM controller designed to minimize cache miss penalties. The 82C351 provides registers for controlling various system functions such as DRAM and SRAM timing, system configuration, non-cacheable memory areas, and refresh. It also includes action code generation for the bus conversion logic contained in the 82C355.

The 82C351 is part of a cost-effective solution for maximizing the performance of a 386 DX based system. This is accomplished by providing support for high-speed cache memory that is used to hold the most recently accessed code and data. With the use of a cache memory subsystem, the 386 DX is capable of performing memory bus cycles without additional wait states. Most memory requests can be supplied by this fast cache memory. Cache updating is performed using a buffered write-through system. Using this system, write accesses to the slower main memory are buffered, so the CPU can start the next cycle before the write cycle is completed. The cache memory has a direct-mapped organization.

The 82C351 provides clock circuitry to supply both the high-speed CPU clock, and the relatively low speed AT bus clock. Since the CPU can operate at clock rates of over four times the AT bus clock, the 82C351 is capable of producing an AT bus clock that is a division of the CPU clock, or an AT bus clock derived from an independent clock source.

The 82C351 is also designed with independent programmable controls for the AT bus command timing and wait state generation for memory and I/O accesses.

Figure 1-1. 82C351 Block Diagram



1.2 Functional Subsystem

The 82C351 CPU/Cache/DRAM Controller consists of the following functional subsystems as illustrated in Figure 1-1.

- Reset and shutdown logic
- AT and CPU clock selection logic
- Action code generation, buffer control logic, and data conversion logic
- Control Logic for:
  - CPU
  - DRAM (local) memory and SRAM (cache) memory
  - AT bus accesses
  - Bus arbitration
  - 0WS or 1WS buffered write cycles
- Memory control logic for:
  - DRAM accesses
  - Cache memory accesses
  - Refresh cycles
  - EPROM Accesses
  - Shadow RAM support
- Index registers for system control
- Fast RESET
- Fast GATEA20

Reset and Shutdown Logic

The 82C351 has two reset inputs, -RESET1 and -RESET2; and two reset outputs, RESET3 and RESET4. -RESET1 is used to indicate a request for power-On reset, and is usually connected to the POWERGOOD signal from the power supply. -RESET1 is also used to reset the keyboard controller; and when active, the 82C351 issues RESET3 and RESET4 for a system reset. Both RESET3 and RESET4 are synchronized with CPUCLKIN and meet the setup and hold time requirements for the 386DX.

-RESET2 is generated by the keyboard controller in response to an instruction from the system and is used to switch the CPU from protected mode to real mode. -RESET2 activates only RESET3 which is used to reset the CPU. RESET3 is also activated by the 82C351 when a shutdown condition is detected. After a shutdown condition is detected, RESET3 is asserted and is held high for at least 64 CPUCLKIN cycles and then deasserted. An active RESET3, resulting from a shutdown, is synchronized with respect to CPUCLKIN to ensure proper CPU operation.

The methods described above for resetting the processor in order to return to real mode are relatively slow. The 386DX can switch from protected to real modes without using the slower keyboard -RESET2; however, most available software is designed for 80286 or 8086 machines and requires the keyboard

controller to generate a CPU reset. For this reason, the 82C351 includes logic to initiate a fast reset, to be used in place of the methods previously described. The fast reset can be invoked by a 0 to 1 transition of bit 0 in port 92H, or bit 5 in index register 60H. If port 92H is used, bit 6 in index 2BH must be set to enable the port. RESET3 is asserted for at least 64 CPUCLKIN cycles during either a power-On reset, a fast reset, or a keyboard controller reset.

RESET4 is used to reset the 82C356, the AT bus, and the math coprocessor. It is synchronized with respect to CPUCLKIN and is asserted for at least 64 CPUCLKIN cycles.

1.3 Clock Logic

The 82C351 has a flexible clock selection system as shown in Figure 1-2. This system generates clocks for both the processor and the AT bus. The inputs to this clock selection logic are CLK2IN and ATCLKIN. CLK2IN is driven from a packaged oscillator circuit running at twice the 386DX rated frequency. ATCLKIN, if present, is usually 16MHz, to provide an 8MHz AT bus clock. The output of the AT bus clock selector is BUSCLK. BUSCLK is equal to one half the internal BCLK. BCLK can be a divisor (2, 3, 4, or 5) of CLK2IN or equal to ATCLKIN. As shown in Table 1-1, index Reg 07H can be programmed to select the desired clock for BCLK.

Table 1-1. CPU and AT Clock Selection

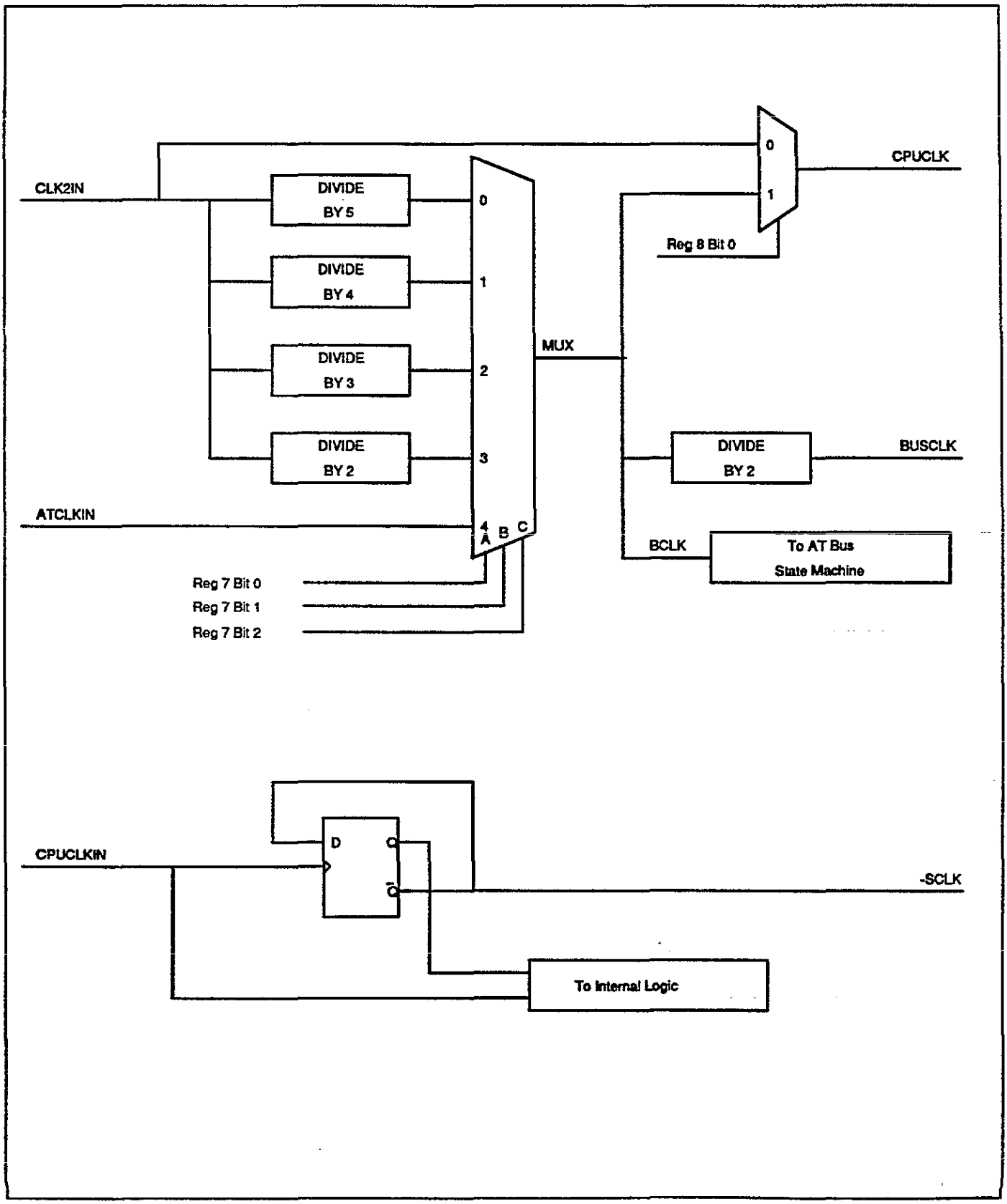
Reg 8H <0>	Reg 7H <2:0>	BCLK	CPUCLK
0	000	CLK2IN/5	CLK2IN
0	001	CLK2IN/4	CLK2IN
0	010	CLK2IN/3	CLK2IN
0	011	CLK2IN/2	CLK2IN
0	100	ATCLKIN	CLK2IN
1	100	ATCLKIN	ATCLKIN

Table 1-1 lists all possible clocks modes. Note that ATCLKIN should be a lower frequency than CLK2IN. Table 1-2 is a listing of the different combinations available with PEAK/DM that yield a compatible AT clock frequency. Some add-in cards may not function properly if the AT Bus clock is greater than 8MHz.

Table 1-2. BUSCLK Frequencies

CLK2IN	Divisor	BCLK	BUSCLK
50 MHz	3	16.66 MHz	8.33 MHz
50 MHz	4	12.50 MHz	6.25 MHz
66 MHz	4	16.50 MHz	8.25 MHz
66 MHz	5	13.20 MHz	6.60 MHz

Figure 1-2. 82C351 Clock Selection



The output of the processor clock selector is CPUCLK. This signal is routed to an output pin and is not used inside the 82C351 except for the clock select circuit. CPUCLK is software selectable between CLK2IN and BCLK. Index Register 08H, bit 0, selects the CPUCLK source. Normally, CLK2IN should be selected as the processor clock (CPUCLKIN) to allow the processor to operate at its maximum rated speed. BCLK may be used to generate the CPU clock when there is a need to slow down the processor for timing dependent code execution. There is a separate CPUCLKIN pin on the 82C351 that is used internally. This pin should always be connected to the same clock as the CPU.

Once clock selection options are set, clock switching occurs with a clean transition. During clock switching, no phase of CPUCLK is less than the minimum or greater than the maximum value specified for the 386DX. The default settings for CPUCLK and BCLK are CLK2IN and CLK2IN/5, respectively.

1.4 Action Code Generation and Data Conversion Logic

The AT bus control logic performs data conversions for CPU accesses to devices that are not on either the local CPU or memory bus. The data bus conversions are performed for 16 and 8-bit read or write operations. If required, 32-bit transfers to or from the CPU are broken into multiple 16 or 8-bit cycles. Action codes are generated to control the buffers within the 82C355. Byte addresses XA<0:1> are generated for CPU AT bus cycles to drive the lower two bits of the AT address bus.

The 82C351 uses -MCS16, -IOCS16, and -BE<0:3> to determine the size of the data being transferred between the CPU and the AT bus. -MCS16 indicates 16-bit memory accesses are required while -IOCS16 indicates 16-bit I/O accesses are required. If -MCS16 or -IOCS16 are not asserted, 8-bit transfers are assumed. The request is converted into 1, 2, 3, or 4 AT bus cycles based on the CPU byte enable signals. For example, when -MCS16 is asserted, the AT bus control logic converts a 32-bit memory access into two 16-bit AT bus accesses.

When performing the data conversions, a 4-bit action code AC<0:3>, is generated to control the buffers in the 82C355 for the alignment of data paths and to control the direction between the D and MD data buses. The definitions for the action codes are given in the functional description of the 82C355. For data accesses to and from the SD bus, SDIR0 controls the buffer direction for SD<0:7> and SDIR1 controls the buffer direction for SD<8:15>.

1.5 CPU, Local Memory, AT Bus Control and Arbitration Logic

The 82C351 contains logic to control all accesses initiated by the CPU, DMA/Master, or refresh request logic to the DRAMs (local memory), SRAMs (cache memory), or the AT bus.

The CPU and Local Memory (DRAM) logic controls all the accesses to the local bus (DRAM and cache memory), while the AT bus logic controls non-local bus accesses. The CPU and DRAM logic supports only 32-bit transfers between the CPU and the system memory where no bus conversions are required. Thus, the -BS16 input of the 386DX is not used in PEAK/DM based systems and should be pulled high.

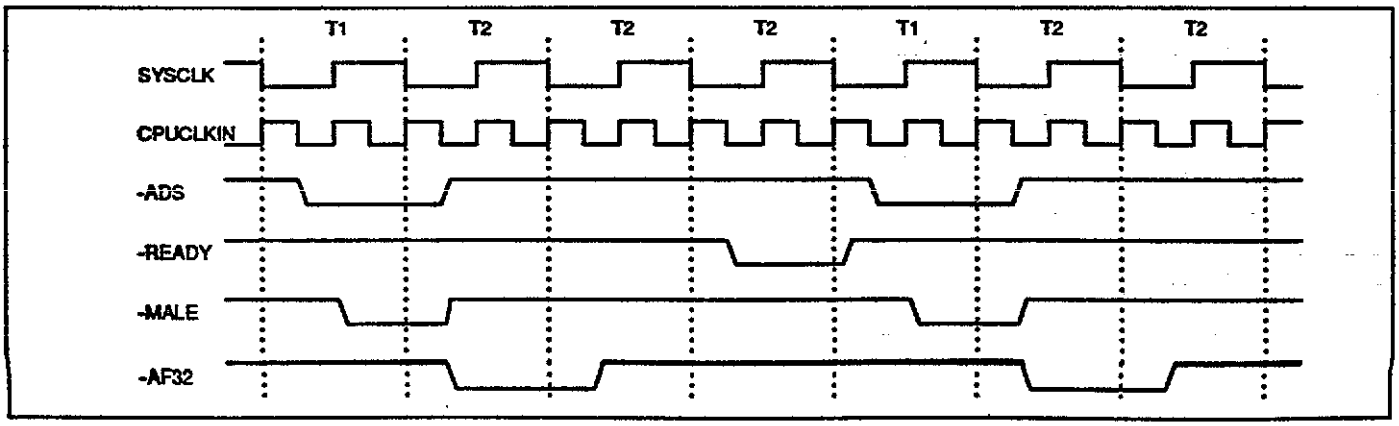
The AT bus logic is responsible for all non-local bus accesses and controls the AT bus for proper bus conversions. In a traditional AT compatible design, the AT bus clock (BCLK/2 or BUSCLK) and the CPU clock (CPUCLK) are the same. In PEAK/DM, it is not necessary for the CPUCLK and BUSCLK to share the same clock. The CPU and local memory logic can be driven by CPUCLKIN, while the AT bus can run off BCLK. This allows the CPU to operate at much higher frequencies, while the AT bus operates at an AT-compatible 8 MHz speed. By allowing the different clock source for the CPU and the AT bus, high performance systems can be designed compatible with standard AT bus peripherals. PEAK/DM also has the capability of deriving the AT bus clock from CLK2IN. CLK2IN can be divided by 2, 3, 4, or 5 to yield BCLK (see section *1.3 Clock Logic* for more information.)

1.6 CPU Control Logic

The interface to the 386 DX processor requires proper interpretation of the status lines upon assertion of -ADS, and the generation of -READY upon completion of the requested operation. The CPU control logic monitors -ADS, M/-IO, W/-R, and D/-C to determine the CPU's cycle type and its starting point. This logic generates -MALE to indicate the start of a new CPU cycle.

A local memory access is distinguished from a non-local access by the assertion of -AF32. -AF32 is generated by either the 82C351 for local cycles, or the Weitek coprocessor for Weitek coprocessor cycles. -AF32 is also an input to the 82C351 to indicate that a local device is in control of the bus such as the Weitek coprocessor. At the end of T2, after the generation of -MALE, the CPU control logic samples -AF32. If -AF32 is asserted, the current cycle is considered a local cycle and the AT bus logic is not activated. If -AF32 is inactive, then the control is passed to the AT bus logic. The CPU control logic and the processor then wait for -READY to be generated by the AT bus control logic. Figure 1-3 shows the timing relationship for -MALE and -AF32 with respect to CPUCLKIN, -ADS, and -READY.

Figure 1-3. -MALE and -AF32 Generation for Local Bus Cycles (2 WS Cycle)



Local Memory Control

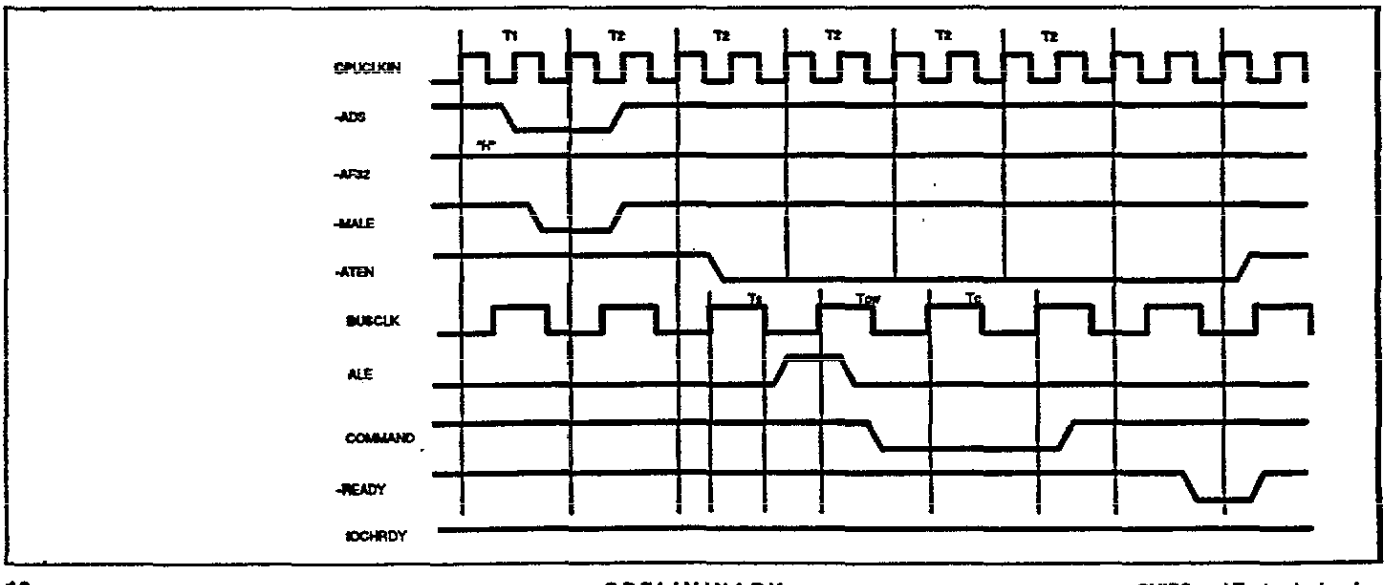
If -AF32 is asserted by the 82C351 at the start of a memory cycle, then the 82C351 starts a local memory cycle and does not allow the AT bus control logic to begin. The local memory control logic generates the corresponding DRAM control signals. It can be programmed to insert wait states in units of two CPUCLKIN cycles to extend the memory cycle so that slow DRAMs may be used to implement the memory subsystem. At the end of the cycle, the local memory control logic generates -READY to terminate the local cycle. If -AF32 is asserted externally, then neither a local or AT cycle will occur. The device that generated -AF32 must generate -READY to end the cycle.

AT Bus Control

The AT bus control logic is invoked whenever -AF32 is not active at the start of the cycle. This logic is driven by BCLK, which is two times the frequency of the AT system clock, BUSCLK. The 82C351 performs the necessary synchronization of the control and status signals between the AT bus and the processor. The 82C351 supports 8, 16, or 32-bit transfers between the processor and the 8 or 16-bit memory or I/O devices located on the AT bus.

An AT bus cycle is initiated by asserting -MALE that is decoded from the CPU status signals and is terminated by asserting -READY. During memory cycles, on the falling edge of BALE, -MCS16 is sampled to determine the bus size required. During I/O cycles, -IOCS16 is sampled on the falling edge of BALE. After a programmed number of delays, the AT logic then enters the command phase. The command signals for the memory or I/O cycles remain active until the programmed number of wait states are executed. At this point, IOCHRDY is sampled. If IOCHRDY is active, the command signals become inactive during the next BUSCLK cycle. If IOCHRDY is not active, the command signals are extended for an additional cycle and IOCHRDY is sampled again. This process continues indefinitely until IOCHRDY becomes active. Figure 1-4 shows the AT bus timing.

Figure 1-4. AT Bus Timing - One Wait State

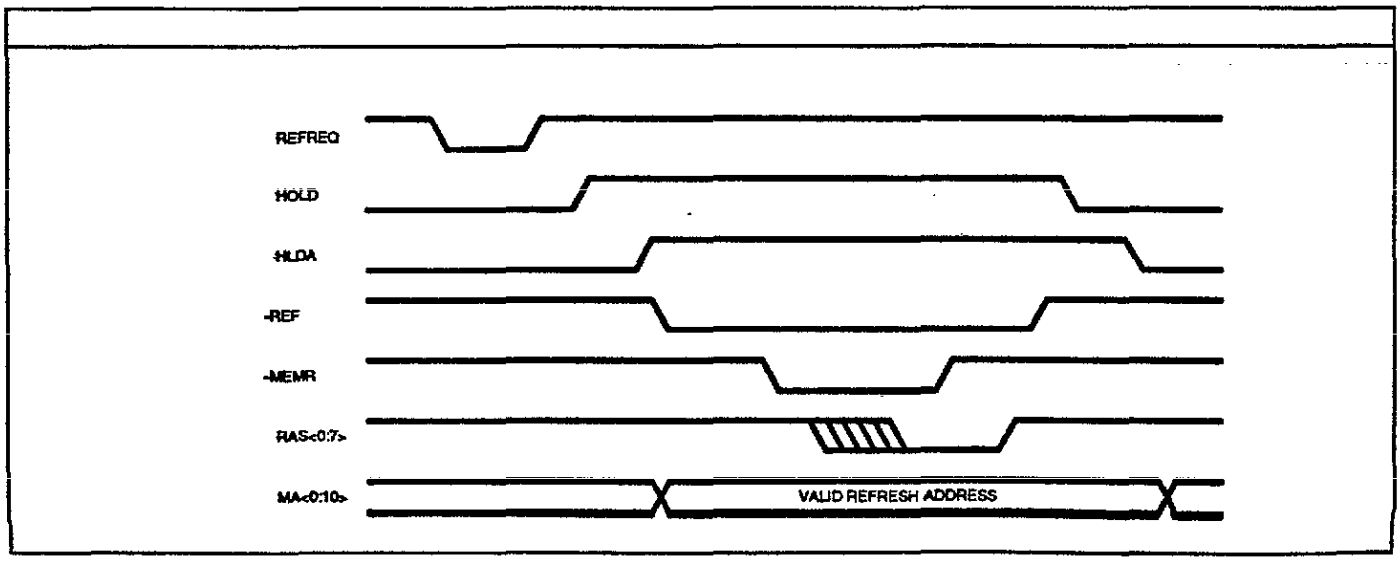


The AT bus control logic provides programmable registers to control the status and command phases of different AT bus cycles. These registers allow low speed AT bus peripherals to operate properly. The 82C351 can be programmed to insert wait states in units of BUSCLK and to delay the generation of -IOR, -IOW, -MEMR, -MEMW commands in one half units of BUSCLK (BCLK) within the selected wait states. AT bus wait states can be controlled independently for 8 and 16-bit accesses and the command delays for 8-bit memory, 16-bit memory or I/O cycles. The command cycle is terminated by detecting -0WS or IOCHRDY active. A command delay is one BCLK delay between the end of BALE and the start of a command. Command delays reduce the total time a command is active. Wait states are additional BUSCLK cycles added to the time a command is active. A command active for 0 wait states with 0 command delays would be 1 BUSCLK long. In addition, extra address hold time can be added for AT adapter cards that may need it to operate reliably. This feature, when enabled, delays the next cycle from starting by one T state. This leaves the address for the current cycle valid for an extended period of time.

Bus Arbitration

The 82C351 controls bus activity and provides arbitration between the CPU, DMA, bus master devices, and DRAM refresh logic. The 82C351 arbitrates between HRQ1, HRQ2, and REFREQ in a non-preemptive manner by generating a HOLD request to the CPU. The CPU relinquishes the bus by issuing HLDA. The 82C351 responds by issuing HLDA1, HLDA2, or -REF depending on which device prevailed during arbitration. During an AT style refresh cycle, the refresh logic has control of the bus until -REF goes inactive. -MEMR is asserted low during the refresh cycle and the refresh address is placed on the MA<0:10> bus by the refresh address counter. In an AT-compatible design during a DMA cycle, the DMA controller has control of the bus until HRQ1 becomes inactive. Figure 1-5 shows the timing for bus arbitration.

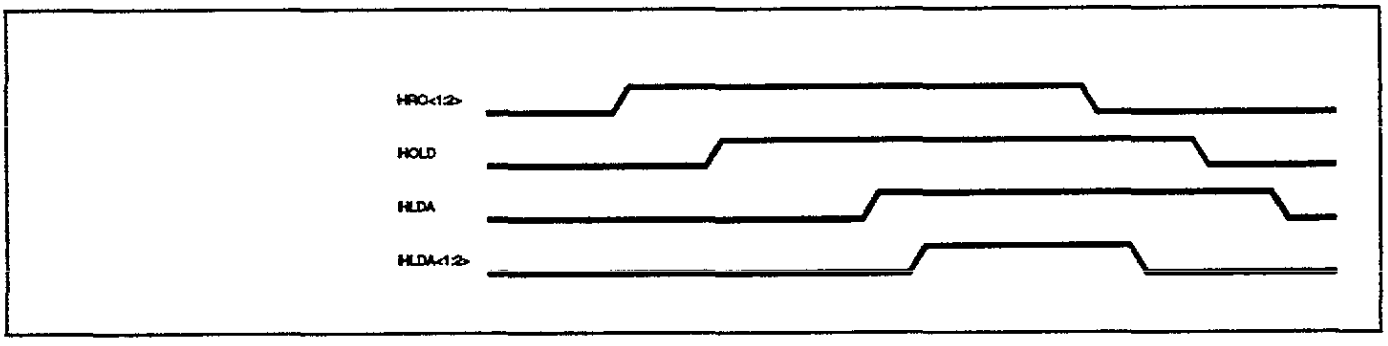
Figure 1-5. Bus Arbitration (REFRESH)



The rising edge of REFREQ sets an internal flag in the 82C351. A HOLD request is then generated, synchronous to the CPU Clock. When the CPU finishes with its current activity, it releases the bus and activates hold acknowledge (HLDA.) The 82C351 then begins the AT style refresh cycle by asserting -REF. After -REF is activated, the 82C351 places the refresh address on the MA<0:9> bus and activates -MEMR. After the refresh cycle is completed, -REF and -MEMR are driven to inactive states. HOLD to the CPU returns to inactive state and the CPU regains control of the bus.

Figure 1-6 shows a DMA cycle arbitration. If a DMA device requests control of the bus, it activates HRQ1 or HRQ2. The 82C351 receives HRQ1 or HRQ2 and issues HLDA1 or HLDA2 and relinquishes the bus. Once the DMA device is no longer asserting HRQ1 or HRQ2, the 82C351 deasserts HLDA1 or HLDA2 and returns control to the CPU.

Figure 1-6. DMA Cycle



1.7 Bus Timeout Logic

An optional Ready timeout is available in the 82C351. If index Reg 26H <2> is a 1, the feature is enabled. If any device being accessed by the CPU does not respond with -READY within 128 clock cycles, the 82C351 AT bus control logic will generate -READY itself if the option is enabled.

1.8 Memory Control Logic

The 82C351 provides all the logic necessary to interface to both local and cache memories. It utilizes a direct-mapped cache organization of up to 256KB and conventional DRAM accesses up to 128MB. It also contains the necessary logic for refreshes, DMA cycles, accessing the EPROMs, and Shadow RAM.

The 82C351 provides the control logic for generating -RAS<0:7> , -CAS<0:3>, MA<0:10> and -DWE to control DRAM accesses. It also provides BWBUSY, LDBDIR, and -LDBEN signals to control the direction of the data flow between the D and MD buses through the 82C355. Figure 1-7 (shown in the section titled *Refresh Cycle*) identifies the basic timing for a DRAM write access.

The 82C351 supports 256KB, 1MB, and 4MB DRAMs in configurations of up to 4 blocks of 2 banks each. The data width of a bank is four bytes plus 4 parity bits (one parity bit for each byte) or 36-bits. Each block has its own set of configuration registers. These configuration registers are used to define the type of memory, the number of wait states (for a DRAM cycle), the minimum RAS pre-charge time, the duration of a refresh RAS pulse, and the starting address within a block. Each block can have a different combination of these options. For example, block 0 can have 256K DRAMs running with 5 wait states while block 1 has 1MB DRAMS running with 4 wait states. The minimum amount of local memory is 1MB if 256Kb memory devices are used (If 1Mb DRAMs are installed, the minimum amount of local memory is 4MB and if 4Mb DRAMs are employed, 16MB). Since the starting address is for an entire block, all DRAM used in a block should be of the same size so the memory will be contiguous. All DRAM access and refresh control signals are presented by the 82C351 with programmable configurations for 256Kx1, 256Kx4, 1Mx1, 1Mx4, 4Mx1, and 4Mx4 devices. Parity generation and checking is implemented in conjunction with the 82C355 data buffer. Index 28 bit 7 in the 82C351 determines if parity is enabled (-PEN is low) or disabled (-PEN is high). If -PEN to the 82C355 is low and a parity error is detected, -LPAR is driven active to the 82C351. The address of the memory location where the parity error occured is latched into index Reg 28H<0:2> and Reg 29H<0:7>.

The 82C351 controls the DRAM memory accesses from four sources: CPU, DMA request, refresh request or a master. These accesses are arbitrated based on the inputs HLDA1, HLDA2 and -REF and are handled by their own control logic. The refresh logic is in control whenever -REF is active. When HLDA1 or HLDA2 is active, the DMA logic is in control. In all other cases, the CPU control logic is in control for valid DRAM memory accesses as defined by the memory map in the configuration registers. The arbitration is non-preemptive; the current active control logic always runs to completion prior to relinquishing control.

The CPU initiated accesses are decoded according to the memory map defined in the configuration registers. The accesses to the local memory are conventional DRAM accesses. The DRAM access begins by strobing in a valid row address with -RAS, while -CAS remains high. The addresses on the MA<0:10> bus are then changed from row addresses to column addresses and strobed in by -CAS, with -RAS remaining low. This is the beginning of a DRAM cycle. The specific type of cycle is determined by the state of the DRAM write enable pin, -DWE. When high, -DWE instructs the DRAMs to read the data on the memory data bus. When -DWE is low, the DRAMs place their data on the memory data bus. The DRAM cycle is terminated when both -RAS and -CAS are returned to a high state. The 82C351 then generates -READY to terminate the current CPU cycle. -DWE should be buffered and connected to each DRAM write enable input.

Bit 3, in index register 2F, controls the location in a local DRAM cycle that -RAS and -CAS are asserted. When this bit is 0, Late RAS mode is selected; and when the bit is 1, Early RAS mode is selected. With Late RAS mode, -RAS is asserted at the end of the first T2 state. -CAS is asserted in the middle of the third T2 state. With Early RAS mode, -RAS is asserted in the middle of the first T2 state. -CAS is asserted at the end of the second T2 state. Early RAS mode

provides more data access time for the DRAM, but requires the Tag SRAMs to deliver the cache address sooner. Therefore, slower DRAM could be used, but faster Tag SRAMs would be needed. Also, since -RAS is asserted earlier, additional time between cycles may be added by the internal logic to meet the minimum -RAS precharge time of the DRAM.

DMA accesses are initiated by asserting HLDA1. The signal combinations -MEMR and -IOW or -MEMW and -IOR, determine if the DMA cycle is a memory read or write access to or from an I/O device. Byte accesses to the local memory are controlled by the four -CAS lines, CAS<0:3>. -CAS0 selects the least significant byte, while -CAS3 selects the most significant byte within the 32-bit wide local memory. To accomplished this, the 82C351 generates the byte enable coded -CAS lines based on the XA0, XA1, and SBHE signals. The DMA control logic makes one memory access per DMA bus cycle and does not attempt to pack or unpack data transfers to make full 32-bit transfers.

First Megabyte of Memory

The initial memory default is for 0K of local memory. Index register 2AH bit 0, when set to a 1, enables the first 512K of local memory. If memory is to be located from this point to the end of the first megabyte, then bit 1 of index register 08H should be set to a 1 which allows index registers 0CH through 0FH to control this address space. Registers 0CH through 0FH determine whether the memory located between 080000H and 0FFFFFH is located on and controlled by the system board, or on the I/O channel. Each bit of the registers controls a 16KB address space. In addition, there is a register that controls whether the accesses to four 64KB areas will be to ROM or to RAM between 0C0000H and 0FFFFFH. The RAM, if selected, can be write protected if desired. This index register, 09H, is for controlling the Low Boot Space. Bits <7:4> are used to select if the RAM is readable and writeable or read only. Bits <3:0> determine if ROM is enabled in any of these four locations.

These seven registers control how the first megabyte of address space is utilized. For instance, assume a system has this configuration:

- 1MB DRAM on the system board
- System ROM BIOS
- EGA/VGA video board installed in I/O channel with BIOS and RAM

The registers would be set as follows:

Register	Description	Bits <7:0>	Function
08H	Initial Memory	X X X X X X 1 X	Enables control by register OCH to 0FH
09H	Low Boot Space	1 0 0 1 0 0 0 0	All ROM disabled, read only RAM at C0000H - CFFFFH and F0000H - FFFFFH
0CH	Mem Enable Map 80000H - 9FFFFH	0 0 0 0 0 0 0 0	RAM at 80000H - 9FFFFH on board
0DH	Mem Enable Map A0000H - BFFFFH	1 1 1 1 1 1 1 1	RAM at A0000H - BFFFFH on channel
0EH	Mem Enable Map C0000H - DFFFFH	0 0 0 0 0 0 0 0	RAM at C0000H - DFFFFH on board
0FH	Mem Enable Map E0000H - FFFFFH	0 0 0 0 0 0 0 0	RAM at E0000H - FFFFFH on board
2AH	Mem Enable Map 00000H - 7FFFFH	X X X X X X X 1	Enable low 512KB local memory

This example is intended to show how the registers controlling the memory enable map and Low boot space are used. Other registers are used to select the DRAM timing and configuration and are explained in *Section 2: 351 Configuration Registers*. Also, a specific sequence must be used in order to move a BIOS to RAM; this is explained in the section titled *Shadow RAM*.

Cache Access

The 82C351 supports 32KB, 64KB, 128KB or 256KB of direct-mapped cache memory. The 82C351 provides all the logic for generating -CRD, -CWE<0:3> and CALE to control the cache memory. For a detailed description of the cache organization, structure and usage refer to *Section 1.10 Cache Concepts*.

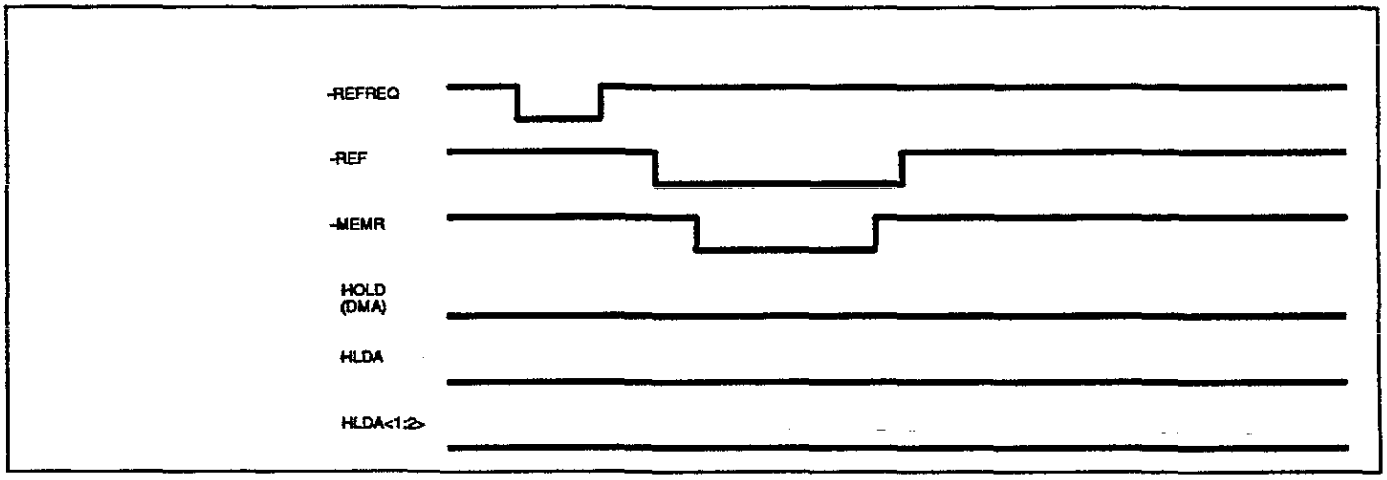
Refresh Cycle

The 82C351 provides two different methods of performing DRAM refreshes, AT style refresh and Hidden refresh.

When using AT style refresh, the 82C351 arbitrates for the bus after receiving REFREQ, and generates HOLD request to the CPU. The CPU relinquishes the bus by issuing HLDA. The 82C351 responds by issuing -REF and starts the refresh cycle. -RAS only refresh is performed by strobing a row address while -CAS remains high. The independent RAS pulses (-RAS<0:7>), are staggered in order to reduce noise caused by driving them all active simultaneously.

With Hidden refresh, DRAMs are refreshed without the 82C351 implementing the HOLD and HLDA sequence. With Hidden refresh, the CPU is allowed to access the cache memory for a read hit during a refresh cycle. If a local memory, AT or DMA cycle is initiated while a hidden refresh cycle is occurring, wait states are inserted until the refresh cycle is complete. The delayed cycle is then allowed to complete. On the other hand, if a local memory, AT or DMA cycle is active when a hidden refresh cycle would normally start, that cycle is allowed to finish and the refresh happens afterwards. Figure 1-7 shows the hidden refresh timing.

Figure 1-7. Hidden Refresh

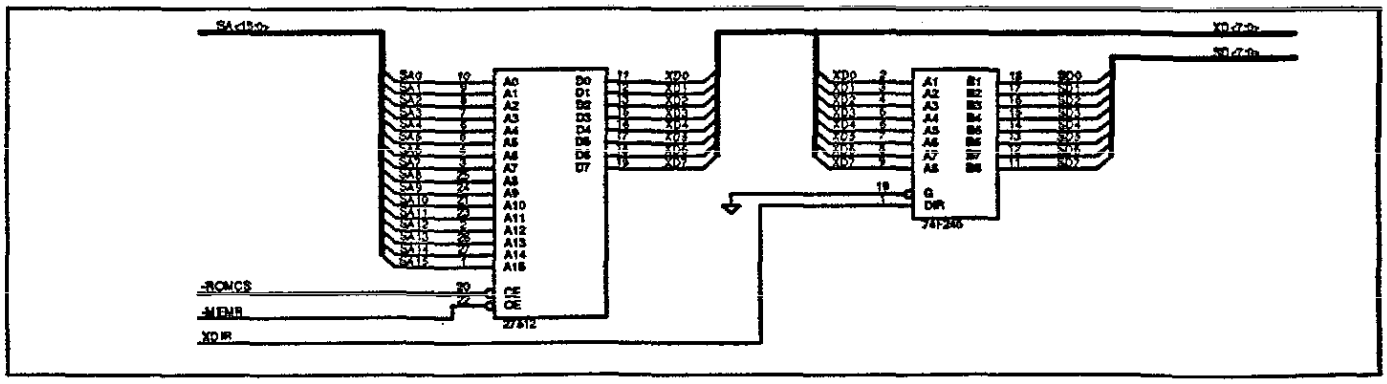


1.9 EPROM Control Logic

The 82C351 provides control logic to generate -ROMCS for ROM accesses. The AT bus control logic generates -READY for this cycle. -ROMCS can be enabled for any of the 64KB sections starting at 0C0000H, 0D0000H, 0E0000H, and 0F0000H with index register 09H bits 3 to 0, respectively. Index register 2CH bits 3 to 0 control -ROMCS for the 64KB sections starting at 0FC0000H, 0FD0000H, 0FE0000H, and 0FF0000H. -ROMCS is connected to any ROMs located in these address ranges that are located on the system board. -ROMCS is also connected to the 82C356 which controls the direction of data flow between the *SD* and *XD* busses. When a ROM access occurs in one of the above address spaces, and -ROMCS is enabled for that area, the 82C351 asserts -ROMCS. The 82C356 uses XDIR to direct ROM data from the XD bus to the SD bus. If, on the other hand, one of the above areas is to be used for RAM, as with the shadowing of BIOS ROMS, the registers 09H or 2CH would be configured so -ROMCS is not asserted for accesses to that section. Accesses to this RAM would be handled by the local memory control logic instead of the AT bus control logic.

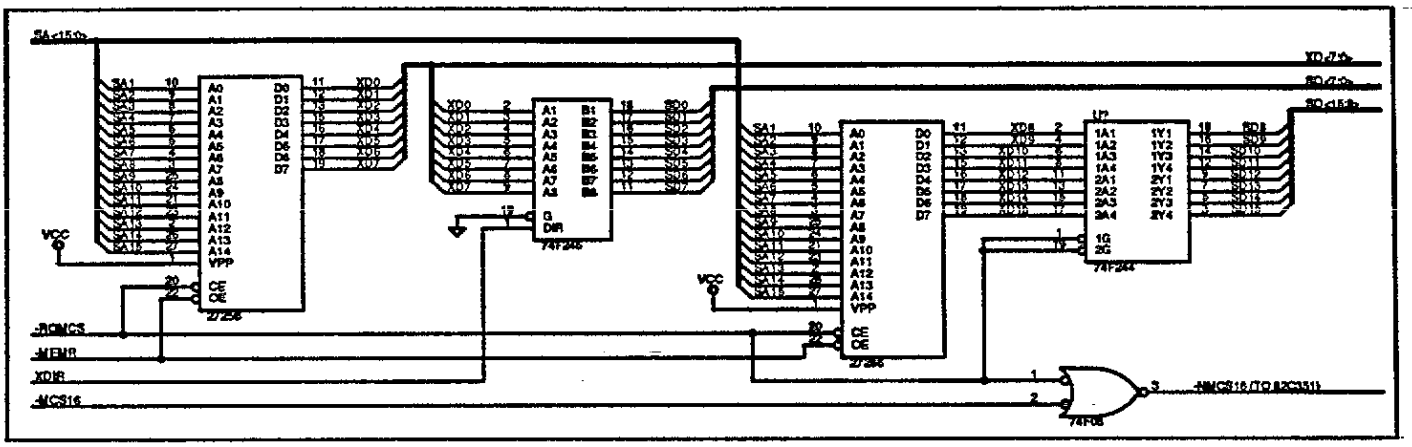
When a single ROM is used, it is attached to the XD<0:7> bus via a bi-directional buffer whose direction is controlled by XDIR from the 82C356. When the processor accesses the ROM, four successive byte transfers are initiated to complete the 32-bit read.

Figure 1-8. Single EPROM Connection Example



Dual ROMs are often used where the code is split, with the even (low) addresses in the first ROM, and the odd (high) addresses in the second ROM. In this case, logic should be added to drive -MCS16 low to the 82C351 whenever -ROMCS is asserted, in order for the 82C351 to perform two 16-bit accesses to complete the 32-bit read. The low byte ROM should be attached to the XD<0:7> bus. The high byte ROM should be connected to a newly created XD<8:15> bus that connects to the SD<8:15> bus via a buffer that is enabled by -ROMCS.

Figure 1-9. Dual EPROM Connection Example



Shadow RAM

In order to enhance system performance, it is preferable to execute BIOS code from RAM rather than from slower EPROMs. The 82C351 provides a feature called Shadow RAM, that when enabled, allows BIOS code to be executed from system RAM resident at the same physical address as the BIOS EPROM. Software should transfer code stored in the BIOS EPROMs to the system RAM before enabling the Shadow RAM feature. This feature significantly improves the performance in applications that use BIOS calls extensively. The Shadow RAM feature is invoked by enabling certain bits in the ROM enable register and RAM mapping registers (Reg 09H, 0EH, and 0FH). Once the BIOS is shadowed, that portion of RAM is write protected.

The sequence for turning on the shadow RAM for Video BIOS ROM is as follows:

1. Enable the ROM space in register 09H and make the RAM read/writable.
2. Disable the local memory DRAM at 0C0000H to 0CFFFFH using register 0EH<4-7>.
3. Read a byte of the Video ROM into a temporary register.
4. Enable the corresponding local memory at 0C0000H to 0CFFFFH by using register 0EH<4-7>.
5. Write the data from the temporary register into the memory. Since the memory located at 0C0000H to 0CFFFFH is now enabled the memory write cycle will be directed to the local system memory instead of the video memory.

6. Go back to step 2 until the entire BIOS is transferred from ROM into RAM.
7. Upon completion of the data transfer from ROM to RAM, program register 09H to disable the ROM and make the RAM read only.

The sequence for turning on the shadow RAM for system BIOS ROM is as follows:

1. Enable the ROM space in register 09H and make the RAM read/writable.
2. Enable the corresponding local memory by using register 0FH.
3. Read the content of the ROM and write to local memory. Since the memory write cycle does not generate -ROMCS, data is written to RAM.
4. Upon completion of the data transfer from ROM to RAM, program the register 09H to disable the ROM and make the RAM read only.

OS/2 Optimization

The PEAK/DM architecture features OS/2 optimization using fast RESET and fast GATEA20. OS/2 makes frequent DOS calls while operating in 80386 protected mode. In order to service these DOS calls efficiently, the CPU needs to switch from protected to real mode. For a PC/AT architecture, the keyboard controller is used to reset the CPU (switch to real mode) and to activate GATEA20. Since this is a slow process, the 82C351 contains a register at I/O port 92H used to generate a fast CPU reset, and a fast GATEA20. If GATEA20 is low (0 in port 92H), the address line A20 is forced to 0; otherwise address line A20 is passed through. Port 92 is accessed directly instead of using the indexed register method. The port can be enabled or disabled by writing a 1 or 0, respectively, to index Reg 2B bit 6.

Coprocessor Handling

The 82C351 identifies a 387DX cycle by decoding A31 and M/-IO. A one wait state -READY is generated for all 387DX cycles. The 82C351 can be programmed to either send coprocessor cycles to the AT bus, or not to generate -READY at all. Sending the 387DX cycles to the AT bus is done when the 387DX is not present and a value should be forced onto the data bus. Not generating -READY allows external logic to assert -READY. The option for generating -READY for a 387DX cycle is present in index Reg 2BH <5:4>. Coprocessor error handling is not performed within the 82C351. External logic is required to perform this function.

Following is an example of the logic necessary to support the 387DX. This can be contained in a single programable logic device and contains all of the logic necessary to interface the 387DX and Weitek 3167 with the exception of the following, which are performed on other chips:

- Port F0 decode — Performed in the 82C356
- 387DX READY logic — Performed in the 82C351
- WT3167 READY logic — Performed by the WT3167 directly
- WT3167 bus cycle claiming — -AF32 signal from the WT3167 to the 82C351

The PLD has the following features:

- Uses a standard speed (35nS) 16L8 PLD
- Provides the 387DX -BUSY signal, which consists of the 387DX -BUSY, "latched Busy", and "Toggle busy"
- Provides IRQ13 from both the 387DX circuit and the WT3167
- Provides the 386DX -ERROR signal to indicate whether the 387DX is present at each 386DX reset
- Provides the PEREQ signal to the 386DX
- Auto-detection of the 387DX and WT3167; no jumpers required
- Allows the 387DX and WT3167 to be installed at the same time
- Reduces the number of pull-up and pull-down resistors required by gating signals off inside the PLD if the respective coprocessor is not present.

-387ERROR and -WTPRES are used by the PLD to determine which coprocessor is present, and should have 10K pull-up resistors attached. PEREQ, -387BUSY, and WTINTR do not need resistors, since these inputs are ignored when the respective coprocessor is not present.

Outputs

386PEREQ — If the 387DX is not present, the PEREQ signal is always low. This prevents the need for a pull down resistor. If the 387 DX is present, 386PEREQ is the OR of the 387DX PEREQ and a signal which goes active following a 387DX error, after the 387DX becomes non-busy. This is required to clear out an internal flag in the 386DX, and is a result of the non-standard co-processor hookup for AT compatibles. The "error" portion of this signal is the AND of 387IRQ13 and -387BUSY. When the 387DX error occurs, 387IRQ13 goes active until cleared by the interrupt service routine. -BUSY will go high several clock cycles after the interrupt goes high.

387PRESENT — This signal is used at various places within the PLD. A 387DX drives -ERROR low when it is RESET, and keeps it low until it is initialized. If the 387DX is not there, a pull-up keeps 387ERROR high. The PLD term is a transparent latch with -ERROR as the data input, and RESET4 as the gate. When RESET4 (which occurs on a power up reset or RESET button push) is active (high), the latch is transparent. The output pin is inverted from the input, so a 1 indicates the 387DX is present.

-386ERROR — The 386DX samples the -ERROR pin sometime between coming out of RESET and the first -ADS to determine whether a 387DX is present. If it is low, the 387DX is assumed present. If it is high, either a 80287 or no coprocessor is present. The 386DX uses this information to determine whether to use a 16 or 32-bit data bus when communicating with the coprocessor. In a "normal," 386DX/387DX hook-up, the 387DX -ERROR pin is attached directly to the 386DX. This is not possible for an AT compatible system for two reasons:

The 386DX -ERROR signal must be high during coprocessor operation since the AT-compatible scheme uses and reports errors differently.

The 386DX must be informed of the 387DX presence EVERY time it is reset, including times when this is done to take it out of protected mode. Note that the 387DX is NOT reset during these resets, and will not drive -ERROR low.

The circuit drives -386ERROR high at all times except the time from RESET3 (386DX reset) going active until the next -ADS. During this time the inverted state of 387PRESENT is driven (low if the 387DX is present, high otherwise). An RS type latch is used inside the PLD for this to occur.

-BLOCKBUSY — This signal is used internally. It is low from the beginning of RESET3 until -ADS goes low for the first time. It is used to block busy during that time, to prevent the 386DX from performing a self test when coming out of protected mode via RESET3.

-386BUSY — This signal is always high if a 387DX is not present. It is also forced high by the -BLOCKBUSY signal to prevent unwanted self tests (which would reduce performance when the 386DX is reset to exit protected mode. If the 387DX is present, the signal is the OR of:

- The 387DX BUSY- signal
- The "Toggle BUSY" function
- The "Latched BUSY" signal (actually 387IRQ13)
- "TOGGLE BUSY" toggles the busy signal periodically when the 387DX is not present. This prevents system hangs with certain software packages that search for the presence of the coprocessor in "non-Intel suggested" ways.
- "Latched BUSY" is part of what is required by the AT compatible coprocessor hook-up. It holds -BUSY active to the 386DX during an error condition, preventing the 386DX from sending a new instruction to the coprocessor before the error status can be read.

387IRQ13 — This signal is the output of a flip-flop whose input is 387BUSY (inverted -387BUSY) and whose clock is -387ERROR. The flip-flop is cleared when RESET4 is active or INTCLR is high (which occurs during I/O writes to port 0F0). The output is held low if 387PRESENT is inactive.

IRQ13 — This pin is the OR of the 387DX interrupt and the WT3167 interrupt. The WT3167 busy is blocked off if the WT3167 is not present. The 387DX interrupt will not occur if the 387DX is not present (it is gated off in the 387IRQ13 term of this PLD).

Following is a pinout of the PLD and the logic equations necessary to provide the above features for this example:

Table 1-3. PLD Pinouts and Logic Equations

Inputs	Outputs
Pin 1 = 387PEREQ	Pin 19 = 386PEREQ
Pin 2 = 387 -BUSY	Pin 18 = IRQ13
Pin 3 = -387ERROR	Pin 17 = 387PRESENT
Pin 4 = -ADS	Pin 16 = -386ERROR
Pin 5 = INTCLR	Pin 15 = FLOP
Pin 6 = RESET4	Pin 14 = 387IRQ13
Pin 7 = RESET3	Pin 13 = -BLOCKBUSY
Pin 8 = -REFRESH	Pin 12 = -386BUSY
Pin 9 = -WTPRES	
Pin 11 = WTINTR	

Logic Equations

```
386PEREQ        !387BUSY * 387IRQ13 * 387PRESENT
              + 387PEREQ * 387PRESENT;
387PRESENT      !(!387ERROR * RESET4
              + !RESET4 * !387PRESENT
              + !387ERROR * !387PRESENT);
386ERROR        RESET3 * 387PRESENT
              + 387PRESENT * !ADS * 386ERROR;
BLOCKBUSY       RESET3 + !ADS * BLOCKBSY;
386BUSY         387BUSY * !BLOCKBUSY *
                387PRESENT
              + REFRESH * !BLOCKBUSY *
                !387PRESENT
              + 387IRQ13 * !BLOCKBUSY;
FLOP            !(!387BUSY * !387ERROR
              + 387ERROR * !FLOP
              + !387BUSY * !FLOP
              + INTCLR
              + RESET4);
387IRQ13        !(!FLOP * 387ERROR
              + !387ERROR * !387IRQ13
              + !387IRQ13 * !FLOP
              + INTCLR
              + RESET4
              + !387PRESENT);
IRQ13           387IRQ13 + WTINTR * WTPRES;
```

Note The "!" character denotes the signal in its inactive state (i.e. !ADS would indicate -ADS high since -ADS was active low in the input list. The "*" character indicates logical AND, while the "+" character indicates logical OR.

During a Weitek coprocessor cycle, the Weitek coprocessor generates -AF32. If the Weitek WT3167 coprocessor is addressed and is not present (-AF32 is not active), then the 82C351 generates -READY to terminate it. When the Weitek WT3167 is present and is addressed, it generates -AF32 and -READY. The WT3167 memory address range is C0000000H to C1FFFFFFH.

1.10 Cache Concepts

Cache memory optimizes processor performance and enhances bus bandwidth. Cache is fast memory where recently used code and data is stored. A cache system reduces the average memory access time because it holds the most often requested code and data. The effectiveness of the cache is determined by its size, physical organization, cache replacement algorithm, and the behavior of the program. When a cache satisfies the processor access requirements, the overhead resulting from accessing the slower main memory is eliminated. The cache allows the system to operate at the speed of static memories while maintaining the economic advantages of slower main memory storage.

Cache Size

The cache size is one of the most important parameters in terms of both cost and performance trade-offs. Cache miss rates reduce asymptotically with the size of the cache. For example, if the cache size is 32KB, then the performance of programs that are less than 32KB which fit completely in the cache are not be significantly effected when the cache size is 64KB. If the program is larger, then increasing the cache size likely reduces cache misses.

Cache Organization

The basic characteristic of a cache is the fast access time. Therefore, very little or no time must be wasted when searching for words in the cache. For maximum efficiency, the cache is sub-divided into many smaller blocks of storage called lines.

Each line has an address associated with it, that must be stored and compared against the address of the memory request. Addresses are kept as entries (one per line) in a directory that establishes the correspondence between the data in the cache and the particular fragment of main memory that is represented.

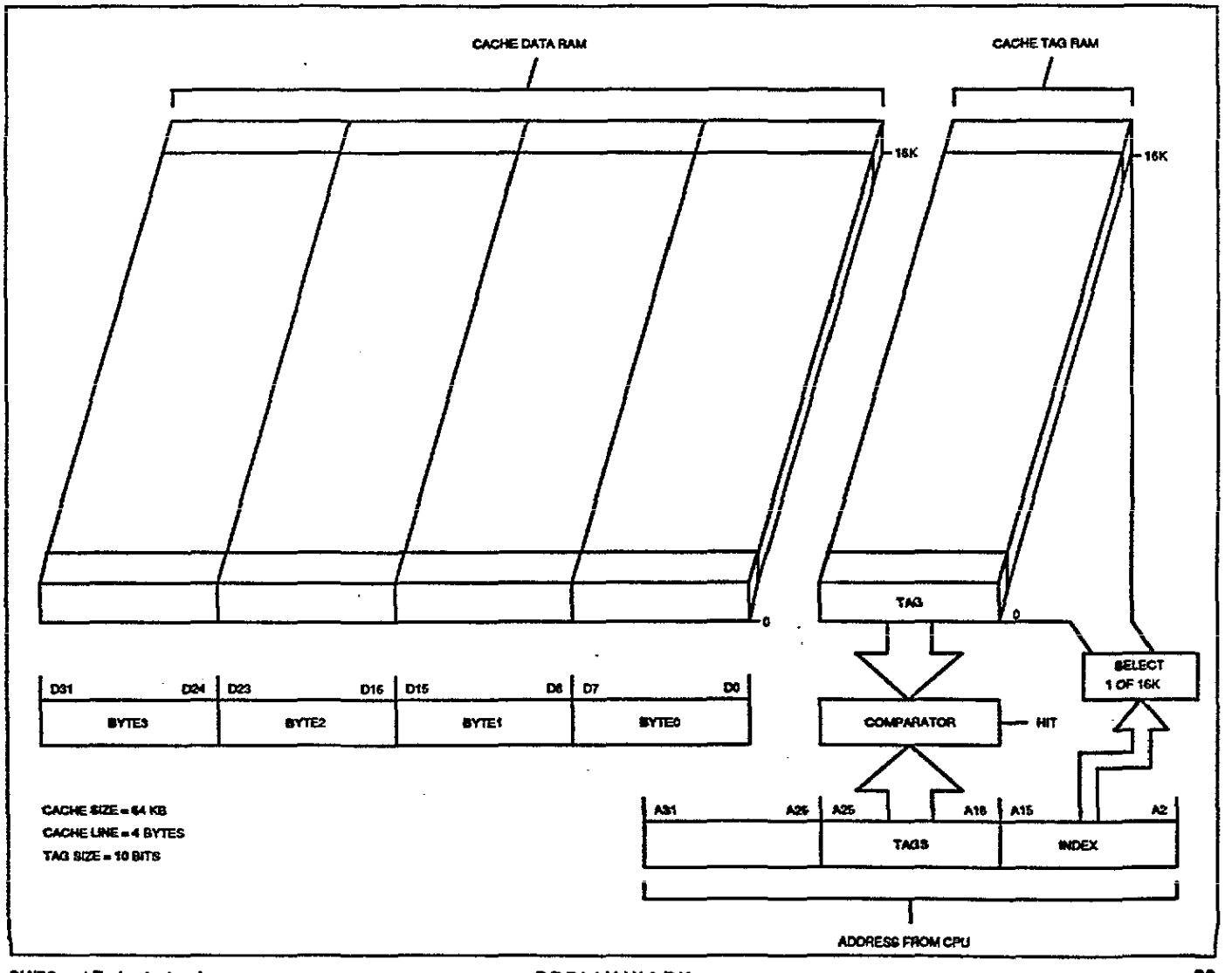
If the size of the lines are as small as possible, then the cache directory becomes large since there is a cache directory entry for each line in the cache. Doubling the size of a cache line, while holding the cache size fixed, reduces the size of the directory by a factor of two because two items (sub-lines) in the same line share the same directory entry. The transformation of data from main memory to cache is referred to as the mapping process. Having a large number of directory entries provides a smaller granularity to the address space. This results in higher performance than larger line sizes. Since external tag directories can use 8K by 8 and 16K by 4 SRAMs, it is relatively inexpensive to implement a large directory. Single chip cache controllers must sacrafice granularity because of the limits of the RAM size on the chip.

Direct-mapped Cache

In a direct-mapped cache every memory location can be stored in one location in cache. The lower order address presented by the processor (determined by the depth of the tag directory) is used as an index to select one of the entries in the tag directory. The higher processor address bits (starting after the lower order addresses and extending upward to a point determined by the width of the tag) are compared with the contents of the tag directory entry to determine if it is a hit or a miss.

Figure 1-10 shows the implementation of a direct-mapped cache. In this example, the 64K direct-map implementation has tag entries 10-bits wide. To construct this cache, 64KB of cache memory, 16K X 10-bits wide of cache directory, and one 10-bit comparator are required. When the processor presents the address for the next instruction, address lines A2 thru A15 are used as an index to select one entry from the cache directory. The value contained in the tag directory is compared with A16 thru A25. If there is a match, then the data is fetched from the cache. If a miss is detected, then the data is obtained from main memory.

Figure 1-10. Direct Mapped Cache Architecture (64KB)



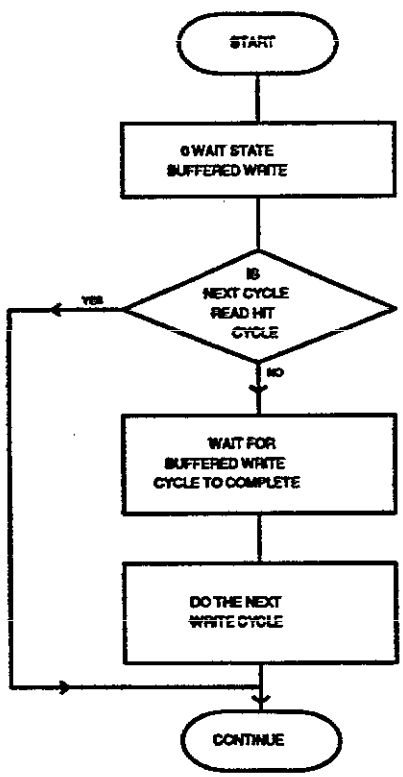
Cache Updating

The cache maintains a copy of the most recently used code or data from main memory. The data in the cache should be identical to the main memory. When the cache memory is modified, the data in the main memory has to be updated as well.

In a write-through system, the controller copies the data into memory as well as the cache. This ensures that the main memory and the cache contain the same data. The draw back with this type of implementation is that each write cycle is treated as a write miss cycle, and the CPU has to wait for the slow memory to be updated.

By using the posted (or buffered) write-through scheme, the write operation penalty can be reduced. During the write operation, data is written into a temporary buffer. The CPU is released to begin a new cycle before the write cycle to the main memory is completed. If a write access is followed by a read hit cycle, the cache access is performed while the main memory is being updated. However, if a write cycle is followed by another write cycle or a read miss cycle, the processor has to wait for the completion of the previous cycle. The posted write through is by far the most popular implementation. The flow chart for posted write is shown in figure 1-11. The 82C351 supports a one stage posted write through scheme to update the cache. To enable Posted Writes, index register 2F bit 1 should be set to 0. Bit 2 of this register determines if there will or will not be additional wait states inserted between writes.

Figure 1-11. Buffered Write Algorithm



Cache Coherency

The buffered write through ensures, under normal operations, that the data present in the cache mirrors the data in the main memory. But in a system environment, other bus masters and slave DMA devices access main memory and modify the contents. Cache controllers that have a built in mechanism to update the corresponding cache contents are said to maintain coherency. Many schemes are used to maintain cache coherency, the easiest being to invalidate (as in the Intel 82385) or flush the cache during DMA operation. Though convenient, this degrades the performance, as all subsequent memory accesses will be misses until the cache is filled with new data. Another method of maintaining cache coherency is by ensuring all accesses to the main memory go through the cache controller. This method is used in the 82C351.

1.11 Cache Functional Overview

The functional overview outlines the cache architecture of the 82C351. The 82C351 performs as an integrated CPU/Cache/DRAM memory controller in a 386DX based system. The 82C351 Cache Controller portion is designed to be a cost-effective solution for achieving the full performance of a 386DX based system. This is accomplished by maintaining the most recently accessed code and data in high-speed memory (cache memory); such that most memory requests can be satisfied from this memory. If the data resides in the cache memory (hit), the data is returned to the CPU, without wait states. If the data is not present in the cache memory (miss), then it is retrieved from the slower main memory with wait states.

During a write operation, the data is held in a temporary buffer, and the CPU is released to start a new cycle before the write cycle to main memory is completed. However, if another write cycle or a read miss cycle is performed, additional wait states are inserted until the previous cycle is completed.

The effectiveness of the cache is determined predominantly by the size and organization of the cache, the hit and miss access times, and the dynamic behavior of the program. An efficient cache organization results in a high hit rate. The majority of the accesses are to cache and are completed without wait states. Very few accesses are to the main memory. Consequently, the average access time approaches that of the fast cache memory.

The 82C351 integrates the control logic required to support an external 32KB, 64KB, 128KB or 256KB cache. The direct-mapped cache directory can be 8, 9, or 10-bits wide and maps up to 128MB of memory space (although 256KB cache with 10 bit wide cache directory should be able to map 256MB of address space, the 82C351 has a 128MB limit). The amount of memory that is cacheable is determined by the size of the cache and the width of the cache directory. As either of these items increase in size, so does the cacheable memory limit. If for instance the cache directory is 16KB deep, the cache can be 16K double words or 64KB in size. This cache would use the address bits A2 to A15. If the cache directory is 10 bits wide, then the largest cacheable memory will be specified by address bits A2 to A15 plus A16 to A25 (the 10 cache directory bits). This would allow 64MB of main memory to be cacheable.

Any memory not cacheable can still be used but all accesses will be misses and handled by the DRAM control logic.

Tag RAM Testing

Two internal registers are used to read and write the tag data. The lower 8-bits of the tag are accessed through index 23H, while the upper two bits are accessed through index register 24H. A memory write to the test window causes the data in the register to be written to the external tag SRAM. A memory read from the test window causes the tag data from the SRAM to be latched in the register, which may later be read by the CPU. Consecutive tag locations are read by accessing every 4th location in the test window, regardless of whether words or bytes are read.

Initializing the Tag

To initialize the tag, the cache must be turned on, but READ HITs disabled. This forces all accesses to be misses. The tags and SRAMs are still written to for cacheable areas, just as if the cache was fully operating. The processor must perform a read from all tag locations to fill the cache with good data. It may do this by reading any block of 256KB of (128KB, 64KB, or 32KB depending on the cache size) consecutive cacheable memory locations. Once this is done, the cache READ HITs may be enabled and the cache becomes operable. The block used to initialize the tag must not be made non-cacheable afterward.

Flushing the cache involves performing the initialization sequence over again. Note, the cache should be off (or READ HITs disabled) when changing the memory map (DRAM configuration registers). The cache must be flushed after changing the registers.

Cache Data RAM Testing

The data RAM is tested using a special testing window. This window is an area in the memory map below 1 MB, which is the same size as the cache data RAM. The address of the window is specified by Reg 22H bits <7:3>, which specify address bits A19-15, respectively. For a 64KB cache, A15 is not included in the decode; for 128KB cache, A16 is not used for decode; and for 256KB, A17 is not used. Bit 0 of Reg 22H is set to 1 so -CAS signals for DRAM are turned off and the read/write are from data RAM. While testing the data RAM, only the test window should be made cacheable by setting bit 2 of Reg 22H = 1. This feature allows the code to remain in the DRAM and thus the program would not crash during testing.

Physical Address Field Assignment

The 386DX supports 32 address bits (30 address bits and four byte enables). While the 82C351 monitors all 30 bits, it supports a maximum of 128MB of physical memory (A0 thru A26).

Table 1-4 identifies the Tag assignments. The CPU address assignment of Tag bits 2-8 is fixed. However, Tag bits 0, 1, and 9 change as different chache sizes are selected.

Table 1-4. Tag Assignments

Cache Size	CPU Address Bits												
	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
32K				Tag 8	Tag 7	Tag 6	Tag 5	Tag 4	Tag 3	Tag 2	Tag 1	Tag 0	Tag 9
64K			Tag 9	Tag 8	Tag 7	Tag 6	Tag 5	Tag 4	Tag 3	Tag 2	Tag 1	Tag 0	
128K		Tag 0	Tag 9	Tag 8	Tag 7	Tag 6	Tag 5	Tag 4	Tag 3	Tag 2	Tag 1		
256K	Tag 1	Tag 0	Tag 9	Tag 8	Tag 7	Tag 6	Tag 5	Tag 4	Tag 3	Tag 2			

Table 1-5 shows which Tag bits from the 82C351 should be connected to the Tag SRAM for each combination of cache size and Tag width. Any unused 82C351 Tag pins should be pulled up with a 10K resistor to Vcc.

Table 1-5. Physical Tag RAM Connections

	82C351 Tag Bits to Connect
32KB Cache	
8-bit	0-6, 9
9-bit	0-7, 9
10-bit	all
64KB Cache	
8-bit	0-7
9-bit	0-8
10-bit	all
128KB Cache	
8-bit	1-8
9-bit	1-9
10-bit	all
256KB Cache	
8-bit	2-9
9-bit	0, 2-9
10-bit	all

1.12 Cache Operation

This section illustrates the different cache cycles:

Read hit cycle	Data available in the cache
Read miss cycle	Data is not available in the cache
Write hit/miss cycle	Both main and cache memory are updated
DMA cycle	Cache updated for write hit cycles only

Read Hit Operation

When a physical address is presented to the 82C351, it uses the line index field (A2 thru A15 for a 64KB cache) to select a line from the cache directory which is stored in the TAG RAM. The TAG stored in this location is compared against the processor's upper address lines. If there is a valid comparison indicating a cache hit, the processor is returned -READY. Once the processor recognizes ready assertion it completes the read cycle of the address location in cache memory.

Read Miss Operation

When a cache read miss occurs (the requested data is not in the cache memory) the main memory (DRAM) provides the data to the CPU, and the cache RAM simultaneously. The data provided to the cache memory at that address location is updated to the new value of the data, and at the same time, the address provided by the CPU is written into the TAG RAM to indicate that this address location is now valid for cache accesses.

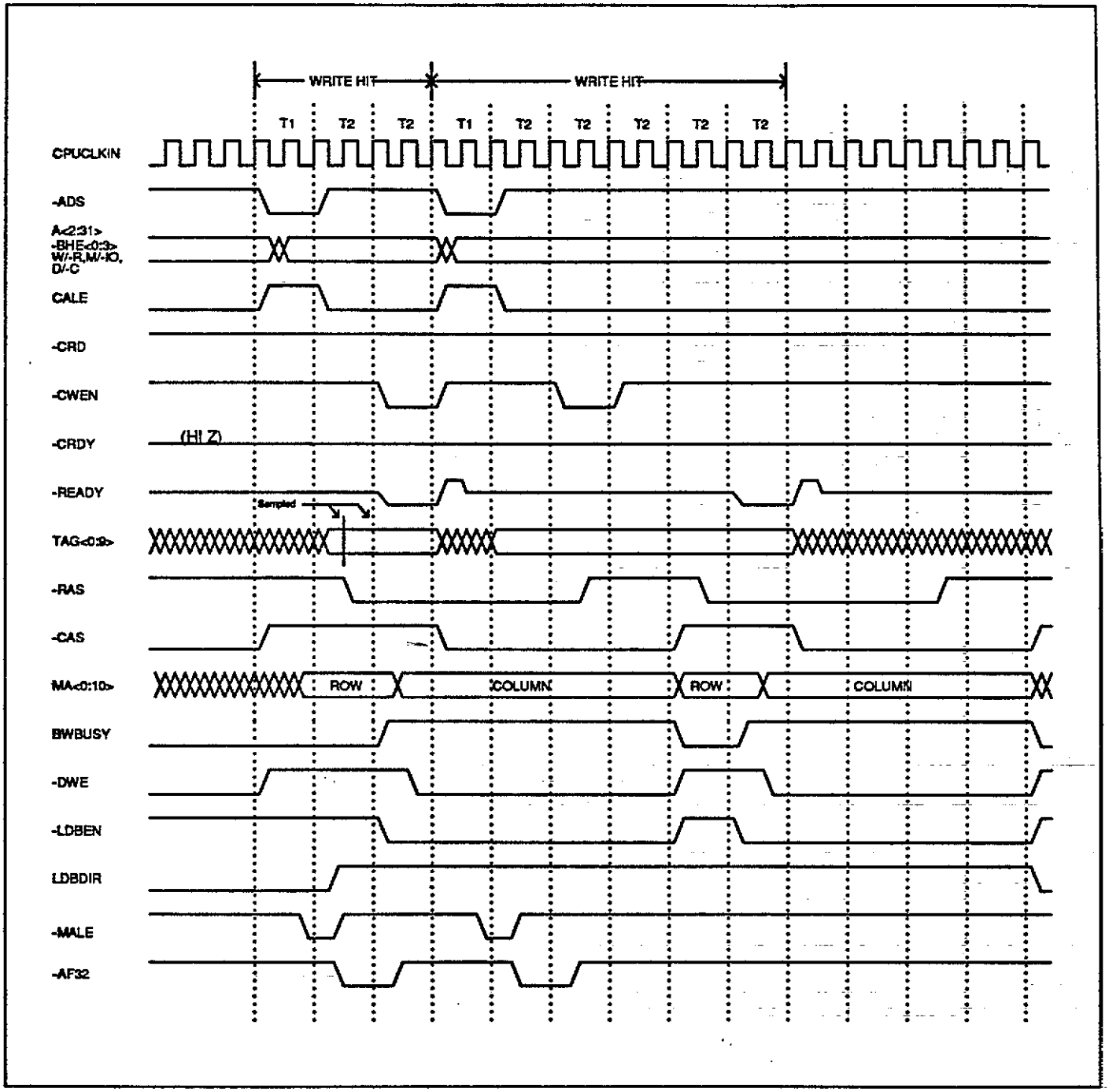
Write Hit Operation

When the CPU initiates a write cycle, a DRAM cycle also started immediately; independent of cache hit or miss result. If a cache hit happens in a write cycle, that is the corresponding data location in DRAM is also in cache memory, both DRAM and cache memory are updated simultaneously. Therefore, SRAM and DRAM write cycle are initiated at the same time. SRAM cycles are much faster than DRAM cycles and additional wait states are required to complete the DRAM cycles. Instead of having the CPU wait for the completion of the DRAM cycle, a register between CPU and the main memory temporarily holds the data for DRAM so the processor can continue.

If a cache miss occurs in a write cycle, no cache write operation occurs; only data in the main memory is updated.

If any of the cycles, immediately following the first write cycle, requires another DRAM access while the main memory is still busy with the first write cycle, then additional wait states are asserted until the DRAM has completed the previous operation. Figure 1-12 shows the timing diagram for write cycle followed by another write cycle.

Figure 1-12. Back to Back Writes Early RAS, Four Wait States



1.13 DMA Operation

For local memory read cycles initiated by DMA or other masters on the bus, regardless of cache hit or miss, cache is not accessed. All memory read cycles are directed to main memory and the data buffer (82C355) receives data from the D bus, drives the MD, and routes the data to the SD bus via data buffers.

For a local memory write cycle, the 82C355 buffer's direction is from SD bus toward MD and D busses. A cache miss causes the data to be written into the DRAM only and a cache hit writes the data into the DRAM and SRAM to maintain coherency.

Non-Cacheable Regions

The 82C351 provides non-cacheable registers REG30 thru REG39 to allow the system designer to set aside up to 4 blocks of variable size of main memory as non-cacheable. This non-cacheable area is required by the system as some memory areas should not be cached. Some cache controllers achieve this by externally decoding the memory address which is to be non-cached. This requires an external fast PLD; and also, once the PLD is programmed the non-cacheable area is fixed. In 82C351 index registers are provided to make any memory non-cacheable without using an external device. The index registers 30H thru 39H are used for programming the starting address of non-cacheable areas and their size. The non-cacheable regions may range from 4KB to 4MB in size.

1.14 Index Registers

Since there are a great number of configuration and diagnostics registers in the PEAK/DM CHIPSet, an indexing scheme is used to reduce the number of I/O addresses required to configure the system. The system control logic generates -XDEN to access the configuration and diagnostics registers. -XDEN is issued for I/O accesses at 22H, 23H, and 92H. For the 82C351, it directs the XD<0:7> lines to the MA<0:7> address lines with an external buffer. Located at 22H and 23H are the index and data registers, respectively. The index register selects an internal register, while the data register is used to read or write information to the selected internal register. The index and data registers for the PEAK/DM are accessed as follows:

1. Write the address of the index register to be accessed to port 22H.
2. To read the value of the index register read I/O port 23.
3. To write data to the index register write to I/O port 23.

Every time an index register is accessed, port 22H should be written. For example, if index register 20H is read (by writing to port 22H and reading port 23H), then to write to index register 20H, port 22H should be rewritten with 20H before writing to port 23H. Port 92H can be accessed directly without using indexing.

Section 2

351 Configuration Registers

Section 2

351 Configuration Registers

2.1 Index Registers

Index Registers accesses are through I/O ports 22 and 23. This port is active only if index 2B bit 6 is a 1.

Note Please note that values identified in parentheses ( ) signify the defaiut value. Setting values to other than the specified designation may cause system malfunction.

Index 04H *Revision Registers (READ ONLY)*

Index	Bits	Values and Functions
04	7-4	(0000): Undefined
	3-0	Revision Level

Index 05H *AT Bus Command Delays (READ/WRITE)*

Index	Bits	Values and Functions
05H	7:6	(00): Reserved
	5:4	AT bus 16 bit memory command delay
		00 : 0 BCLK delay
		01 : 1 BCLK delay
		10 : 2 BCLK delay
		11 : 3 BCLK delay
	3:2	AT bus 8 bit memory command delay
		00 : 0 BCLK delay
		01 : 1 BCLK delay
		10 : 2 BCLK delay
		11 : 3 BCLK delay
	1:0	AT bus I/O command delay
		00 : 0 BCLK delay
		01 : 1 BCLK delay
		10 : 2 BCLK delay
		11 : 3 BCLK delay

Index 06H *AT Bus Wait States (READ/WRITE)*

Index	Bits	Values and Functions
06H	7:6	(00): Reserved
	5:4	16-bit AT bus wait states
		00 : 3 BUSCLK wait states
		01 : 2 BUSCLK wait states
		10 : 1 BUSCLK wait states
		11 : 0 BUSCLK wait states
	3:2	8-bit AT bus wait states
		00 : 5 BUSCLK wait states
		01 : 4 BUSCLK wait states
		10 : 3 BUSCLK wait states
		11 : 2 BUSCLK wait states
	1	AT bus address hold time. (This feature is used if an AT card requires extra address time for reliable operation. When enabled, the next cycle is delayed by one T state which leaves the address valid for this extra time.)
		0 : Disable extra address hold time on AT bus
		1 : Enable extra address hold time on AT bus
	0	Reserved

Index 07H *AT Clock Source Select (READ/WRITE)*

Index	Bits	Values and Functions
07H	7:3	(00000): Reserved
	2:0	Bus clock Source Select (BCLK). The actual BUSCLK output is half of the BCLK source select
		000 : CLK2IN/5
		001 : CLK2IN/4
		010 : CLK2IN/3
		011 : CLK2IN/2
		100 : ATCLK

Index 08H *Initial Memory CPU Speed (READ/WRITE)*

Index	Bits	Values and Functions
08H	7:3	(00000): Reserved
	2	-AF32 assertion control
		(0) : Does not generate -AF32 for addresses above 16MB. Should only be used if external logic can recognize addresses above 16MB.
		1 : Generates -AF32 for addresses above 16MB. Unless real physical memory is installed for that address range, no memory cycle starts and no -READY is given to the CPU.
	1	Initial memory
		(0) : Only 512K memory enabled. Registers 0C-0F are ignored. Register 2A can also effect the bottom 512K.
		1 : Registers 0C-0F can enable the memory above 512K.
	0	CPUCLK (CPU Clock) select
		(0) : Selects CLK2IN
		1 : Selects AT control logic clock (BCLK)

Index 09H *Low Boot Space RAM/ROM (READ/WRITE)*

Bits 7 to 4 write protect RAM located in the BIOS area in 64KB blocks. Bits 3 to 0 enable the substitution of the BIOS ROM located below 1MB with RAM at the same location in 64KB blocks. This should be done after BIOS code is copied from the ROM and the RAM locations have been protected using bits 7 to 4. Make sure registers 0C-0F are enabled if RAM accesses are required.

Index	Bits	Values and Functions
09H	7	64KB RAM at 768K C0000-CFFFFH (VIDEO)
		(0) : 64KB of RAM at 768K is read/write
		1 : 64KB of RAM at 768K is read only
	6	64KB RAM at 832K D0000-DFFFFH
		(0) : 64KB of RAM at 832K is read/write
		1 : 64KB of RAM at 832K is read only
	5	64KB RAM at 896K E0000-EFFFFH
		(0) : 64KB of RAM at 896K is read/write
		1 : 64KB of RAM at 896K is read only
	4	64KB RAM at 960K F0000-FFFFFH (SYSTEM)
		(0) : 64KB of RAM at 960K is read/write
		1 : 64KB of RAM at 960K is read only
	3	64KB ROM at 768K C0000-CFFFFH (VIDEO)
		(0) : 64KB of ROM at 768K is disabled
		1 : 64KB of ROM at 768K is enabled
	2	64KB ROM at 832K D0000-DFFFFH
		(0) : 64KB of ROM at 832K is disabled
		1 : 64KB of ROM at 832K is enabled
	1	64KB ROM at 896K E0000-EFFFFH
		(0) : 64KB of ROM at 896K is disabled
		1 : 64KB of ROM at 896K is enabled
	0	64KB ROM at 960K F0000-FFFFFH (SYSTEM)
		(0) : 64KB of ROM at 960K is disabled
		1 : 64KB of ROM at 960K is enabled

Index 0CH *Memory Enable Map 080000-09FFFFH (READ/WRITE)*

Bit 0 enables the lowest and bit 7 the highest 16K block in the memory area 080000 to 09FFFFH. For Example, bit 0 controls the 16K block from 512K to 528K. This permits 16K blocks of memory to be disabled allowing ROMs, memory expansion schemes (EMS, EEMS or XMA) or memory mapped I/O devices to reside within the lower 1MB address space.

Index	Bits	Values and Functions
0CH	7:0	Enable bits for eight 16K blocks of memory.
		0 : Address is on
		(1) : Address is on the I/O channel; 16K block disabled.

Index 0DH *Memory Enable Map 0A0000-0BFFFFH (READ/WRITE)*

Index	Bits	Values and Functions
0DH	7:0	Enable bits for eight 16K blocks of memory.
		0 : Address is on or controlled by the system board: 16K block enabled.
		(1) : Address is on the I/O channel; 16K block disabled.

Index 0EH *Memory enable Map 0C0000-0DFFFFH (READ/WRITE)*

Index	Bits	Values and Functions
0EH	7:0	Enable bits for eight 16K blocks of memory.
		0 : Address is on or controlled by the system board: 16K block enabled.
		(1) : Address is on the I/O channel; 16K block disabled.

Index 0FH *Memory enable Map 0E0000-0FFFFFH (READ/WRITE)*

Index	Bits	Values and Functions
0FH	7:0	Enable bits for eight 16K blocks of memory.
		0 : Address is on or controlled by the system board: 16K block enabled.
		(1) : Address is on the I/O channel; 16K block disabled.

Index 10H *Block 0 Type and Starting Address (READ/WRITE)*

Index	Bits	Values and Functions
10H	7:6	DRAM type in Banks 0/1
		00 : Bank0/1 disabled
		(01) : 256Kx1 or 256Kx4 DRAMs
		10 : 1Mx1 or 1Mx4 DRAMs
		11 : 4Mx1 or 4Mx4 DRAMs
	5:0	(000000): Starting address for Block 0 <A26:A21>.
	256K usage	A<26:21> (2MB per pair of banks)
	1M usage	A<26:23>(8MB per pair of banks)
	4M usage	A<26:25>(32MB per pair of banks)

Index 11H *Block 0 RAM timing (READ/WRITE)*

Index	Bits	Values and Functions
11H	7:6	DRAM wait states
		00 : 3 wait states
		01 : 4 wait states
		(10) : 5 wait states
		11 : Reserved
	5	(0) : Reserved - write 0
	4:3	-RAS<0:1> precharge time
		00 : 4 CPUCLKIN cycles
		01 : 6 CPUCLKIN cycles
		10 : 8 CPUCLKIN cycles
		(11) : 8 CPUCLKIN cycles
	2:1	-RAS<0:1> Refresh pulse width
		00 : 4 CPUCLKIN cycles
		01 : 5 CPUCLKIN cycles
		10 : 6 CPUCLKIN cycles
		(11) : 7 CPUCLKIN cycles
	0	(0) : Reserved - write 0

Index 12H — *Block 1 Type and Starting Address (READ/WRITE)*

Index	Bits	Values and Functions
12H	7:6	DRAM type in Banks 2/3
		00 : Bank 2/3 disabled
		(01) : 256Kx1 or 256Kx4 DRAMs
		10 : 1Mx1 or 1Mx4 DRAMs
		11 : 4Mx1 or 4Mx4 DRAMs
	5:0	(000000): Starting address for Block 1
	256K usage	A<26:21>(2MB per pair of banks)
	1M usage	A<26:23> (8MB per pair of banks)
	4M usage	A<26:25> (32MB per pair of bank)

Index 13H — *Block 1 RAM Timing (READ/WRITE)*

Index	Bits	Values and Functions
13H	7:6	DRAM wait states
		00 : 3 wait states
		01 : 4 wait states
		(10) : 5 wait states
		11 : Reserved
	5	(0) : Reserved - write 0
	4:3	-RAS<2:3> precharge time
		00 : 4 CPUCLKIN cycles
		01 : 6 CPUCLKIN cycles
		10 : 8 CPUCLKIN cycles
		(11) : 8 CPUCLKIN cycles
	2:1	-RAS<2:3> Refresh pulse width
		00 : 4 CPUCLKIN cycles
		01 : 5 CPUCLKIN cycles
		10 : 6 CPUCLKIN cycles
		(11) : 7 CPUCLKIN cycles
	0	(0) : Reserved - write 0

Index 14H — *Block 2 Type and Starting Address (READ/WRITE)*

Index	Bits	Values and Functions
14H	7:6	DRAM type in Banks 4/5
		00 : Bank 4/5 disabled
		(01) : 256Kx1 or 256Kx4 DRAMs
		10 : 1Mx1 or 1Mx4 DRAMs
		11 : 4Mx1 or 4Mx4 DRAMs
	5:0	(0000): Starting address for Block 2
	256K usage	A<26:21> (2MB per pair of banks)
	1M usage	A<26:23> (8MB per pair of banks)
	4M usage	A<26:25> (32MB per pair of bank)

Index 15H — *Block 2 RAM Timing (READ/WRITE)*

Index	Bits	Values and Functions
15H	7:6	DRAM wait states
		00 : 3 wait states
		01 : 4 wait states
		(10) : 5 wait states
		11 : Reserved
	5	(0) : Reserved - write 0
	4:3	-RAS<4:5> precharge time
		00 : 4 CPUCLKIN cycles
		01 : 6 CPUCLKIN cycles
		10 : 8 CPUCLKIN cycles
		(11) : 8 CPUCLKIN cycles
	2:1	-RAS<4:5> Refresh pulse width
		00 : 4 CPUCLKIN cycles
		01 : 5 CPUCLKIN cycles
		10 : 6 CPUCLKIN cycles
		(11) : 7 CPUCLKIN cycles
	0	(0) : Reserved - write 0

Index 16H — *Block 3 Type and Starting Address (READ/WRITE)*

Index	Bits	Values and Functions
16H	7:6	DRAM type in Banks 6/7
		00 : Bank 6/7 disabled
		(01) : 256Kx1 or 256Kx4 DRAMs
		10 : 1Mx1 or 1Mx4 DRAMs
		11 : 4Mx1 or 4Mx4 DRAMs
	5:0	(000000): Starting address for Block 3
	256K usage	A<26:21> (2MB per pair of banks)
	1M usage	A<26:23> (8MB per pair of banks)
	4M usage	A<26:25> (32MB per pair of bank)

Index 17H — *Block 3 RAM Timing (READ/WRITE)*

Index	Bits	Values and Functions
17H	7:6	DRAM wait states
		00 : 3 wait states
		01 : 4 wait states
		(10) : 5 wait states
		11 : Reserved
	5	(0) : Reserved - write 0
	4:3	-RAS<6:7> precharge time
		00 : 4 CPUCLKIN cycles
		01 : 6 CPUCLKIN cycles
		10 : 8 CPUCLKIN cycles
		(11) : 8 CPUCLKIN cycles
	2:1	-RAS<6:7> Refersh pulse width
		00 : 4 CPUCLKIN cycles
		01 : 5 CPUCLKIN cycles
		10 : 6 CPUCLKIN cycles
		(11) : 7 CPUCLKIN cycles
	0	(0) : Reserved - write 0

Index 18H — *Number of Populated Banks within a Block (READ/WRITE)*

Index	Bits	Values and Functions
18H	7:4	(0000): Reserved
	3	Block 3 (Bank 6/7)
		(0) : Bank 6 populated only
		1 : Banks 6 and Bank 7 populated
	2	Block 2 (Bank 4/5)
		(0) : Bank 4 populated only
		1 : Banks 4 and Bank 5 populated
	1	Block 1 (Bank 2/3)
		(0) : Bank 2 populated only
		1 : Banks 2 and Bank 3 populated
	0	Block 0 (Bank 0/1)
		(0) : Bank 0 populated only
		1 : Banks 0 and Bank 1 populated

Index 20H — *Cache Control (READ/WRITE)*

Index	Bits	Values and Functions
20H	7	Cache enable bit
		(0) : cache is disabled
		1 : cache is enabled
	6	Force Read miss cycle. This bit is used for initializing the data and tag SRAMs. When set to 1 all accesses are to DRAM, but the tag and data SRAMs are still updated for cacheable areas as if the cache is fully operating. Any block of 256KB (or size of the cache) consecutive locations are then read, followed by setting this bit to 0. The 256KB block used to initialize the tag must not be made non-cacheable afterward.
		(0) : No forced read miss cycle (normal operation).
		1 : Forced read miss cycle enabled.
	5	Freeze cache directory
		(0) : Normal operation
		1 : Freeze cache directory. A cache read miss will not cause a tag RAM update and change of data in the cache data RAM. Instead a normal DRAM read operation will be performed. A cache write hit will update the cache data RAM.
	4	(0): Reserved
	3	Refresh type selection
		(0) : Hidden refresh
		1 : AT Style refresh
	2:0	(000): Reserved

Index 21H — *Cache Size (READ/WRITE)*

Index	Bits	Values and Functions
21H	7:6	(00): Reserved
		(0): Reserved - write 0
	5:4	Cache size
		(00) : 32K bytes
		01 : 64K bytes
		10 : 128K bytes
		11 : 256K bytes
	3	(0): Reserved
	2:1	Address Tag Width. These bits define the number of bits in an address tag. The tag RAM can be 8 to 10 bits wide.
		(00) : 8 bit tag
		01 : 9 bit tag
		10 : 10 bit tag
		11 : Reserved
	0	(0): Reserved - write 0

Index 22H — *Cache and Tag Test (READ/WRITE)*

The Cache data and tag RAM can be tested using a test window. The address of the test window is specified by bits <7:3> which corresponds to address lines A19-A15 (below 1MB). When bit 0 is set to 1 the access to DRAM is cut off for the test window. Reads and writes can then be performed on the data RAM alone.

Index	Bits	Values and Functions
22H	7:3	Address A19-A15 for test window (for 64K cache the address A15 is not used.
	2	Test window cacheable. The test window can be made cacheable and the rest of the memory non-cacheable by setting this bit to 1. Setting this bit to 0 makes all the memory cacheable.
		(0) : Disable test window
		1 : Enable test window
	1	Tag RAM test mode
		(0) : Disabled
		1 : Enabled
	0	Cache data RAM test mode.
		(0) : Disabled
		1 : Enabled

Index 23H — *Tag <0:7>Test Data Port Register (READ/WRITE)*

Index	Bits	Values and Functions
23H	7:0	When tag test mode is enabled (bit 1 of reg 22 = 1) the data can be written to tag RAM through this register. Writes to the test window causes the value in this register to be it written into the tag bits <7:0>. Reads from the test window causes the tag bits <7:0> to be written into this register.

Index 24H *Tag <8:9> Test Data Port Register (READ/WRITE)*

Index	Bits	Values and Functions
24H		This register is similar to index register 23H, and is used as a data port for testing the tag window. Bits <9:8> are read or written through this register.
	7:2	(000000): Reserved
	1:0	Correspond to tag bits 9:8

Index 26H *Ready Timeout Control (READ/WRITE)*

Index	Bits	Values and Functions
26H		This register controls the Ready Timeout option. If any device accessed by the CPU does not respond with -READY within 128 clock cycles, the 82C351 AT control logic will generate -READY itself.
	7:3	Reserved (See *Volume IV: 82C356 Peripheral Controller, Section 2: 356 Configuration Registers*)
	2	-READY Timeout
		(0) : -READY Timeout disabled
		1 : -READY Timeout enabled
	1:0	(00): Reserved (See *Volume IV: 82C356 Peripheral Controller, Section 2: 356 Configuration Registers*)

Index 28H *Parity Error Status Register (READ/WRITE)*

Index	Bits	Values and Functions
28H	7	Parity check enable
		(0) : Parity check enabled (-PEN is low)
		1 : Parity is disabled (-PEN is high)
	6:3	(0000): Reserved
	2:0	Error address bits A<26:24>. When a local parity error occurs the address A<26:24> are latched into bits <2:0>. These bits are read only.

Index 29H *Parity error Address Register (READ ONLY)*

When a local memory parity error occurs, A<23:16> are latched into this register.

Index	Bits	Values and Functions
29	7:0	Parity error address A<23:16>

Index 2AH *Memory Enable Map 0-07FFFFH (READ/WRITE)*

Index	Bits	Values and Functions
2AH	7:1	(0000000): Reserved
	0	512K local memory enable
		(0) : Disable 512K on local memory
		1 : Enable 512K on local memory

Index 2BH — *Miscellaneous Control (READ/WRITE)*

Index	Bits	Values and Functions
2BH	7	(0): Reserved
	6	Enable port 92H
		(0) : Disable port 92H
		1 : Enable port 92H
	5:4	Ready generation for 387 DX cycles
		(00) : 82C351 generates ready after 1 wait state
		01 : 82C351 runs AT cycle for all NPX cycles
		10 : 82C351 does not generate ready at all
		11 : Reserved
	3	Weitek Present bit (READ ONLY)
		1 : Weitek coprocessor present
		(0) : Weitek coprocessor not present
	2:0	(000) : Reserved

Index 2CH — *Middle Boot Space RAM/ROM Configuration (READ/WRITE)*

Index	Bits	Values and Functions
2CH	7	Type of 64KB RAM at 0FC0000H
		(0) : 64KB of RAM at 0FC0000H is R/W
		1 : 64KB of RAM at 0FC0000H Is read only
	6	Type of 64KB RAM at 0FD0000H
		(0) : 64KB of RAM at 0FD0000H is R/W
		1 : 64KB of RAM at 0FD0000H is read only
	5	Type of 64KB RAM at 0FE0000H
		(0) : 64KB of RAM at 0FE0000H is R/W
		1 : 64KB of RAM at 0FE0000H is read only
	4	Type of 64KB RAM at 0FF0000H
		(0) : 64KB of RAM at 0FF0000H is R/W
		1 : 64KB of RAM at 0FF0000H is read only
	3	Enable 64KB RAM at 0FC0000H
		(0) : 64KB of ROM at 0FC0000H is disabled
		1 : 64KB of ROM at 0FC0000H is enabled
	2	Enable 64KB RAM at 0FD0000H
		(0) : 64KB of ROM at 0FD0000H is disabled
		1 : 64KB of ROM at 0FD0000H is enabled
	1	Enable 64KB RAM at 0FE0000H
		(0) : 64KB of ROM at 0FE0000H is disabled
		1 64KB of ROM at 0FE0000H is enabled
	0	Enable 64KB RAM at 0FF0000H
		0 : 64KB of ROM at 0FF0000H is disabled
		(1) : 64KB of ROM at 0FF0000H is enabled

Index 2FH *DRAM Timing Control (READ/WRITE)*

Index	Bits	Values and Functions
2FH	7:4	(0) : Reserved
	3	-RAS mode for DRAM cycles. This controls the timing for the generation of -RAS (and -CAS). When early -RAS is selected, -RAS is asserted in the middle of the first T2 state. -CAS is asserted at the end of the second T2 state. When late -RAS is selected, -RAS is asserted at the end of the first T2 state and -CAS is asserted in the middle of the third T2 state.
	2	Posted write wait states. This specifies the number of wait states for posted CPU write cycles. If a posted write is already in progress wait states are added until the current cycle is finished.
		0 : zero wait state posted writes
		(1) : one wait state posted writes
	1	Posted DRAM write enable
		0 : Enable posted writes
		(1) : Disable posted writes
	0	Reserved

Non-cacheable blocks: The 82C351 provides programmable registers for defining an area of memory as non-cacheable. For example, the BIOS ROM area needs to be non-cached. Four such areas can be declared non-cacheable. The index registers 30H to 39H are used to set the starting address and the amount of memory to be non-cached. Block 0 to block 3 are the four non-cacheable blocks. Each block has three registers which define the starting address and the amount of memory. For example, block 0 has register 30H, 31H and 38H. The register 31H contains the non-cacheable size. The non-cacheable size can vary from 4K to 4MB.

Index 30H *Block 0 Non-Cacheable Address A23 to A16 (READ/WRITE)*

Index	Bits	Values and Functions
30H	7:0	(00000000): Block 0 Non-Cacheable address A23 to A16

Index 31H *Block 0 Non-Cacheable Address A15 to A12 and Size (READ/WRITE)*

Index	Bits	Values and Functions
31H	7:4	(0000): Block 0 Non-Cacheable address A15 to A12
	3:0	Non-Cacheable Size
		(0000) : Disabled
		0001 : 4 KB
		0010 : 8 KB
		0011 : 16 KB
		0100 : 32 KB
		0101 : 64 KB
		0110 : 128 KB
		0111 : 256 KB
		1000 : 512 KB
		1001 : 1 MB
		1010 : 2 MB
		1011 : 4 MB

Index 32H *Block 1 Non-Cacheable Address A23 to A16 (READ/WRITE)*

Index	Bits	Values and Functions
32H	7:0	(00000000): Block 1 Non-Cacheable address A23 to A16

Index 33H — *Block 1 Non-Cacheable Address A15 to A12 and Size (READ/WRITE)*

Index	Bits	Values and Functions
33H	7:4	(0000): Block 1 Non-Cacheable address A15 to A12
	3:0	Non-Cacheable Size
		(0000) : Disabled
		0001 : 4 KB
		0010 : 8 KB
		0011 : 16 KB
		0100 : 32 KB
		0101 : 64 KB
		0110 : 128 KB
		0111 : 256 KB
		1000 : 512 KB
		1001 : 1 MB
		1010 : 2 MB
		1011 : 4 MB

Index 34H — *Block 2 Non-Cacheable Address A23 to A16 (READ/WRITE)*

Index	Bits	Values and Functions
34H	7:0	Block 2 Non-Cacheable address A23 to A16

Index 35H — *Block 2 Non-Cacheable Address A15 to A12 and Size (READ/WRITE)*

Index	Bits	Values and Functions
35H	7:4	(000): Block 2 Non-Cacheable address A15 to A12
	3:0	Non-Cacheable Size
		(0000) : Disabled
		0001 : 4 KB
		0010 : 8 KB
		0011 : 16 KB
		0100 : 32 KB
		0101 : 64 KB
		0110 : 128 KB
		0111 : 256 KB
		1000 : 512 KB
		1001 : 1 MB
		1010 : 2 MB
		1011 : 4 MB

Index 36H — *Block 3 Non-Cacheable Address A23 to A16 (READ/WRITE)*

Index	Bits	Values and Functions
36H	7:0	(00000000): Block 3 Non-Cacheable address A23 to A16

Index 37H *Block 3 Non-Cacheable Address A15 to A12 and Size (READ/WRITE)*

Index	Bits	Values and Functions
37H	7:4	(0000): Block 3 Non-Cacheable address A15 to A12
	3:0	Non-Cacheable Size
		(0000) : Disabled
		0001 : 4 KB
		0010 : 8 KB
		0011 : 16 KB
		0100 : 32 KB
		0101 : 64 KB
		0110 : 128 KB
		0111 : 256 KB
		1000 : 512 KB
		1001 : 1 MB
		1010 : 2 MB
		1011 : 4 MB

Index 38H *Non-cacheable Address A26 to A24 (READ/WRITE)*

Index	Bits	Values and Functions
38H	7:6	(00): Reserved
	5:3	(000): Block 1 Non-Cacheable address A26 to A24
	2:0	(000): Block 0 Non-Cacheable address A26 to A24

Index 39H *Non-Cacheable Address A26 to A24 (READ/WRITE)*

Index	Bits	Values and Functions
39H	7:6	(00): Reserved
	5:3	(000): Block 2 Non-Cacheable address A26 to A24
	2:0	(000): Block 3 Non-Cacheable address A26 to A24

Index 60H *Fast Reset Control Register*

Index	Bits	Values and Functions
60H	7:6	(00): Reserved
	5	(0): Alternate CPU reset. A low to high transition activates a CPU reset (RESET3).
	4:0	(00000): Reserved

2.2 I/O Port Registers

I/O Port 92H *Fast GATEA20 and RESET Register*

	Bits	Values and Functions
92H	7-2	Reserved
	1	Fast GATEA20
		0 : Force CPUA20 low
		(1) : Enable CPUA20
	0	Fast CPU reset
		(0) : A 0 to 1 transition causes a CPU reset

Note This port is used by OS/2 and is disabled by writing a 0 to register 2BH bit 6.

Section 3

351 Pin Descriptions

Section 3

351 Pin Descriptions

3.1 Pin Assignments

Table 3-1. Clocks

Pin #	Symbol	Type	Signal Description
CLK2IN	125	Input	Input from CMOS oscillator.
CPUCLK	124	Output	Output of processor clock select logic. 12mA drive capability.
CPUCLKIN	123	Input	Processor clock input. This is used by the CPU, cache, and DRAM control logic (everything that uses CPUCLKIN except the clock select logic). This is the same clock the CPU receives. If the internal clock select logic is used, this is connected to the CPUCLK pin of this chip. If the internal clock select logic is bypassed, this is connected to the oscillator, or whatever is the CPU clock source . This requires a CMOS level clock.
-SCLK	126	Output	CPUCLKIN/2. High during phase 1, low during phase 2 (proper phase for the 486). The 351 arbitrarily selects the phase of SCLK, and uses that phase to generate RESET3 and RESET4, etc. 4mA drive capability.
ATCLKIN	82	Input	External oscillator input for the AT clock. Input should be twice the frequency of the AT clock.
BUSCLK	79	Output	AT bus clock. May be derived from CLK2IN or ATCLKIN signals. Frequency between 6 and 8.33MHz is recommended. 4mA drive capability.

Table 3-2. Resets

Pin #	Symbol	Type	Signal Description
-RESET1	57	Input	RESET1 is an active low (schmitt triggered) input which is generated by POWERGOOD for a cold reset. It should be used to reset the keyboard controller. When low, it activates RESET3 and RESET4. -RESET1 is latched internally.
-RESET2	58	Input	RESET2 is generated from the keyboard controller. It forces a CPU reset by activating RESET3.
RESET3	2	Output	RESET3 is an active high signal which is used to reset the 386DX whenever -RESET1 or -RESET2 is active. It is also activated during a CPU shutdown cycle. RESET3 remains active for at least 64 CPUCLKIN cycles. 8mA drive capability.
RESET4	3	Output	RESET4 is an active high signal used to reset the Integrated Peripheral Controller (IPC) and the AT bus. 8mA drive capability.

Table 3-3. Arbitration

Pin #	Symbol	Type	Signal Description
HOLD	137	Output	HOLD request is an active high output to the 386DX HOLD pin. HOLD is used to request to the CPU to relinquish the use of the bus to another master (HRQ1, HRQ2 or REFREQ). 4mA drive capability.
HLDA	141	Input	HLDA is an active high input from the 386DX HLDA pin. When high, HLDA indicates that the CPU has relinquished the bus in response to an active high HOLD signal.
HRQ1	74	Input	HOLD REQUEST 1 is an active high signal and is used to indicate that a DMA/Master is requesting the use of the bus. For an AT compatible architecture, HRQ1 should be connected to the hold request signal HRQ, from the 82C356.
HLDA1	76	Output	HOLD ACKNOWLEDGE 1 is an active high output signal to the 82C355 and 82C356 and indicates that the CPU has relinquished control of the system buses in response to HRQ1 (DMA/ MASTER). 2mA drive capability.
HRQ2	136	Input	HOLD REQUEST 2 is an active high signal that indicates a DMA/Master is requesting the system buses. This pin must be tied to ground if not used.
HLDA2	35	Output	HOLD ACKNOWLEDGE 2 is an active high output signal to the 82C355 and 82C356 that indicates the CPU has relinquished control of the system buses in response to HRQ2 (LOCAL MASTER). 2mA drive capability.
-AEN8	73	Input	Address enable for 8-bit DMA transfers is an active low input signal from one of the two DMA controllers. When active, -AEN8 enables the address latches for 8-bit DMA transfers. It is inactive when an external bus master is active.
-AEN16	84	Input	Address enable for 16-bit DMA transfers is an active low input signal from one of the two DMA controllers. When active, -AEN16 enables the address latches for 16-bit DMA transfers. It is inactive when an external bus master is active.
-MASTER	64	Input	MASTER is an active low input signal from an active device on the I/O channel (AT expansion bus). After -MASTER is forced low by an I/O device, the I/O CPU must wait for one system clock period before driving its address and data lines. -MASTER must not be held low for more than 15 microseconds as this may result in memory loss due to the lack of a refresh cycle.
REFREQ	62	Input	Refresh Request is an active high input signal generated from OUT1 of the 82C356 (by the 8254 compatible timer-1 inside the 82C356). When active, REFREQ initiates a DRAM refresh sequence.
-REF	149	Input/ Output	REFRESH is a active low open drain bidirectional signal. As a input, -REF can be used to force a refresh cycle from an I/O master device. As an output, -REF initiates a refresh cycle for the DRAMs. -REF requires an external pull-up of at least 680 ohms. 18mA drive capability.

Table 3-4. CPU Control

Pin #	Symbol	Type	Signal Description
-ADS	138	Input	ADDRESS STROBE is an active low input from the 386DX -ADS pin. -ADS indicates valid bus cycle definitions (ie. W/-R, D/-C, M/-IO and -BE<0:3> and addresses (A2 to A31). A 10K pull-up resistor is recommended on this pin.
W/-R	139	Input/ Output	WRITE/READ status is a bi-directional signal from the 386DX W/-R pin. As an input, during local memory or I/O cycles. W/-R indicates a write bus cycle when high and a read bus cycle when low. A 10K pull-up resistor is recommended on this pin. During the DMA cycle, W/-R is the same as -MEMR. 4mA drive capability.
D/-C	147	Input	DATA/CONTROL status is an input from the 386DX D/-C pin. D/-C indicates a data cycle when high and a command cycle when low. A 10K pull-up resistor is recommended on this pin.
M/-IO	148	Input	MEMORY/IO status is an input from the 386DX. M/-IO indicates a memory cycle when high and an I/O cycle when low. A 10K pull-up resistor is recommended on this pin.
-READY	152	Input/ Output	READY is generated to indicate to the CPU the end of the cycle. This signal is generated for all CPU cycles except for cache hit cycles. 12mA drive capability.

Table 3-5. AT Bus

Pin #	Symbol	Type	Signal Description
BALE	75	Output	BUFFERED ADDRESS LATCH ENABLE is an active high output to the AT expansion Bus. BALE indicates a valid address on the SA bus. It is used to hold the address during an AT bus cycle. 2mA drive capability.
-MALE	54	Output	MEMORY ADDRESS LATCH ENABLE is an active low output signal to the 82C356. -MALE allows the 82C356 to latch local addresses onto the SA bus during CPU AT cycles. -MALE also indicates the start of a new CPU cycle. 2mA drive capability.
-MEMR	68	Input/ Output	MEMORY READ is an active low bi-directional signal directing memory to place valid data on the data bus. -MEMR is an output if the CPU is controlling the bus or an input if a DMA or external bus master is in control of the bus. 18mA drive capability.
-MEMW	69	Input/ Output	MEMORY WRITE is an active low bi-directional signal directing memory to accept data from the data bus. -MEMW is an output if the CPU is controlling the bus or an input if a DMA or external bus master is in control of the bus. 18mA drive capability.
-SMEMR	60	Output	SYSTEM MEMORY READ is an active low output signal to the AT expansion Bus. When active, -SMEMR indicates that the lowest 1MB memory space is being addressed for a read cycle. 18mA drive capability.
-SMEMW	61	Output	SYSTEM MEMORY WRITE is an active low output signal to the AT expansion Bus. When active, -SMEMW indicates that the lowest 1 megabyte memory space is being addressed for a write cycle. 18mA drive capability.
-IOR	71	Input/ Output	I/O READ is an active low bi-directional signal instructing an I/O device to place data on the data bus. -IOR is an output if the CPU is controlling the bus or an input if a DMA controller is in control of the bus. 18mA drive capability.
-IOW	72	Input/ Output	I/O WRITE is an active low bi-directional signal instructing an I/O device to accept data from the data bus. -IOW is an output if the CPU is controlling the bus or an input if a DMA controller is in control of the bus. 18mA drive capability.
-INTA	55	Output	INTERRUPT ACKNOWLEDGE is an active low output to the interrupt controller in the 82C356. 2mA drive capability.

Table 3-5. AT Bus (continued)

Pin #	Symbol	Type	Signal Description
IOCHRDY	63	Input	I/O CHANNEL READY is a schmitt-triggered active high input from the AT Bus. When low, IOCHRDY indicates a "not ready" condition and forces the insertion of wait states in I/O or Memory accesses. When high, it allows the completion of the current I/O or memory access.
-0WS	66	Input	ZERO WAIT STATE is a schmitt-triggered active low input from the AT Bus, causing termination of the AT bus cycle. 16-bit Memory or I/O cards residing on the AT expansion bus use this line to speed up accesses. -0WS requires a 330 ohm pull-up resistor.
-MCS16	77	Input	MEMORY CYCLE SELECT signal is an active low input signal from the AT bus. When active, -MCS16 causes a 16-bit memory access on the I/O channel. If -MCS16 is inactive then the current memory cycle is a 8-bit cycle. A 330 ohm pull-up is recommended on this pin.
-IOCS16	78	Input	I/O CYCLE SELECT 16 is an active low input signal from the AT bus. When active, -IOCS16 causes a 16-bit I/O access on the I/O Channel. If -IOCS16 is inactive, the current I/O cycle is an 8-bit cycle. A 330 ohm pull-up resistor is recommended on this pin.

Table 3-6. Address

Pin #	Symbol	Type	Signal Description
-BE<0:3>	143-146	Input/ Output	BYTE ENABLEs -BE<0:3> are active low bi-directional signals. -BE3 controls the most significant byte; while -BE0 controls the least significant byte. These signals are inputs from the 386DX -BE<0:3> pins during a CPU cycle. As outputs, -BE<0:3> are generated during DMA cycles based on the status signals XA0, XA1, and -SBHE. 2mA drive capability.
A<2:19>	32-21 19-14	Input	LOCAL ADDRESS BITS A2-A19 are inputs to the 82C351 from the 386DX and 82C356.
A20	33	Input/ Output	LOCAL ADDRESS bit 20 is a bi-directional signal. A20 is an output signal gated with GATEA20 during CPU cycles and an input during DMA cycles. 4mA drive capability.
A<21:31>	12-4 158-157	Input	LOCAL ADDRESS BITS A21-A31 are inputs to the 82C351 from the 386DX.
-GATEA20	36	Input/ Output	-GATEA20. This signal is an open collector output used to control the gating of CPUA20. When -GATEA20 is low, the 82C351 propagates CPUA20. When -GATEA20 is high, the 82C351 forces A20 to be low regardless of CPUA20 state. -GATEA20 signal is derived from inverted -GATEA20 from the 8042 keyboard controller.
CPUA20	13	Input/ Output	CPU ADDRESS bit 20 is from the 386DX. A 10K ohm pull-up is recommended.
-SBHE	37	Input/ Output	SYSTEM BUS BYTE HIGH ENABLE is an active low bi-directional signal to or from the AT expansion Bus. When active, -SBHE indicates when the high byte transfer is taking place. As an output, this signal has a 4mA drive capability.
XA<0:1>	38, 39	Input/ Output	EXPANSION ADDRESS bits 0 and 1 are bi-directional signals and are connected directly to the 82C356. XA0 is output when the CPU is the bus master and an input when 8-bit DMA is a bus master. XA1 is an output when the CPU is the bus master and an input when 8-bit or 16-bit DMA is a bus master. 2mA drive capability. 10K pull-up required.

Table 3-6. *Address (continued)*

Pin #	Symbol	Type	Signal Description
-ROMCS	83	Output	ROM CHIP SELECT is an active low signal. When active, -ROMCS enables the chip select inputs to the BIOS EPROM(s). During a ROM cycle the 82C351 directs the 82C355 to pass data from the XD bus onto the D bus. The -ROMCS out of the 82C351 is connected to the -ROMCS of the 82C356 which controls the direction of the buffer between the XD and SD bus. 4mA drive capability.
-VRAMSEL	122	Output	VIDEO RAM SELECT is an active low output indicating the current cycle is a VRAM access. -VRAMSEL is active for A0000H to BFFFFH. 2mA drive capability.

Table 3-7. *Cache*

Pin #	Symbol	Type	Signal Description
CALE	56	Output	CACHE ADDRESS LATCH ENABLE is an active high output signal for the two octal latches. CALE is used to latch the SRAM addresses A2 to A15. When CALE is high the latch is transparent and allows the 386DX addresses to flow through to the SRAMs. On the falling edge of CALE, the addresses are latched.
-CRD	107	Output	CACHE READ is an active low signal to the SRAM's output enables. During read hit cycles, the -CRD is enabled to drive the requested data on to the CPU data bus. 12 mA drive capability.
-CWE<0:3>	102-105	Output	Cache Write Enables <0:3> are active low signals to the SRAM's write enables for bytes 0 to 3, respectively. For example, -CWE0 is connected to byte 0 of SRAM. 8mA drive capability.
-TAGWR	53	Output	TAG RAM WRITE ENABLE is an active low output signal used to update the tag RAM during a DRAM read cycle or when testing the tag RAM. 8mA drive capability.
-CRDY	151	Output	CACHE READY. This is generated by the 82C351 for all cache hit cycles. It is tied to -READY of the 82C351, and to -READY of the 386DX. 8mA drive capability.
TAG<0:9>	42-49 51-52	Input/ Output	TAG BITS 0-9. These are the tag bits from the tag RAM. The tag bits provide the address which is compared by the internal comparator against the memory address request. The tag bits used for comparison are programmable for 8, 9 or 10-bits. They require pull-ups. 4mA drive capability.

Table 3-8. DRAM

Pin #	Symbol	Type	Signal Description
-RAS<0:7>	127-129 131-135	Output	ROW ADDRESS STROBEs 0-7 are active low outputs to the system DRAM. There is one -RAS signal for each bank of the memory on the mother board. -RAS0 is the strobe for bank 0, -RAS1 for bank 1 etc. These signals are used to latch the row address to the DRAMs (falling edge). These should be buffered and terminated. 6mA drive capability.
-CAS<0:3>	116-119	Output	COLUMN ADDRESS STROBEs are active low outputs signal to the system DRAMs for selecting the byte of the DRAMs. The column address is latched on the falling edge of -CAS. For example, -CAS0 is for byte 0, -CAS1 for byte 1, etc. The CAS lines are common to all banks. These should be terminated. 6mA drive capability.
-DWE	101	Output	DRAM WRITE ENABLE is an active low output to the system DRAMs. This signal needs to be buffered and terminated. 8mA drive capability.
MA<0:7>	87-88 90-93 95-96	Input/ Output	DRAM MEMORY ADDRESS/DATA lines. These are the DRAM addresses during a DRAM cycle. During an internal register access (index register access) these are required to connect to the XD bus using a 74F245. The signal -IOW is used to control the direction of the buffer and -XDEN is used to enable the buffer. The MA<0:7> lines should be buffered and terminated. 8mA drive capability.
MA<8:10>	97-99	Output	DRAM MEMORY ADDRESS lines are used to interface to the DRAM address lines. MA8, MA9 and MA10 should be connected to the DRAM address lines A8, A9 and A10. 8mA drive capability.
-MENB	159	Input	MEMORY ENABLE is an active low input signal. When active -MENB validates the address on the local bus. This signal can be used to disable the 82C351 for a predefined address range via an external decode. When inactive, the 82C351 forces all access to the AT bus.
-LPAR	67	Input	LATCHED PARITY is an active low input to the 82C351. When active, it indicates a parity error occurred during the last local memory read and caused an NMI to be generated. The failing address is latched by this signal internal to the 82C351.

Table 3-9. Buffer Control

Pin #	Symbol	Type	Signal Description
-ACEN	115	Output	ACTION CODE ENABLE is an active low output to the 82C355 and validates ACTION CODES AC<0:3>. 2mA drive capability.
AC<0:3>	111-114	Output	ACTION CODE is a four bit encoded command to the 82C355 and is used for data bus sizing and byte assembly operations. AC<0:3> are qualified by the -ACEN signal. 2mA drive capability.
-ATEN	154	Output	AT ENABLE is an active low output signal to the 82C355 and 82C356. When active, -ATEN indicates that the current cycle is an AT cycle. 4mA drive capability.
SDIR0	86	Output	SD BUS DIRECTION. This signal is used to control the buffer direction of the SD BUS for SD<0:7>. When low, the direction is from the SD to the MD bus (during AT or on board I/O read cycle) and when high the data path is from the MD to the SD bus (during write cycles). 2mA drive capability.
SDIR1	85	Output	SD BUS DIRECTION. This signal is used to control the buffer direction of the SD BUS for SD<8:15>. When low the direction is from the SD to the MD bus (during AT or on board I/O read cycles). When high the data path is from the MD to SD bus (during write cycles). 2mA drive capability.
-XDEN	65	Output	XD BUS BUFFER ENABLE is an active low output and is used to enable the buffer between the XD and MA<0:7> buses. -XDEN is asserted during I/O access to index registers (for I/O address 22H and 23H present in the 82C351). 2mA drive capability.

Table 3-9. Buffer Control (continued)

Pin #	Symbol	Type	Signal Description
-AF32	153	Input/ Output	AF32 is an active low bi-directional signal. As a output -AF32 indicates that a local cycle is in progress (active for local DRAM and cache cycles). A high indicates an AT bus cycle. As an input, it disables the AT control logic, allowing the cycle to be handled by a local device. A 10K pull up is required. 4mA drive capability.
-DRD	142	Output	DRAM READ DIRECTION is an active low output signal which is used by the 82C355 to control the direction of data transfer from the MD to the SD bus. When this signal is asserted (low) it enables data transfers from the MD bus to the SD bus. When this signal is de-asserted (high) it causes data to be transferred from the SD bus to the MD bus. 2mA drive capability.
-PEN	156	Output	PARITY ENABLE is an active low output which enables the parity circuitry in the 82C355. This bit is controlled by index register 28 bit 7. 1mA drive capability.
BWBUSY	110	Output	BUFFER WRITE BUSY is an active high output signal to indicate that the main memory is busy in a buffered write cycle. When -LDBEN is low, LDBDIR is high, and BWBUSY is high, the values of the CPU data lines are passed on to the DRAM data lines. This is connected to the 82C355. 2mA drive capability.
-LDBEN	108	Output	LOCAL DATA BUS ENABLE is an active low output. When -LDBEN and LDBDIR are low, the current values on the MD bus are passed on to the D bus. When -LDBEN is low, and LDBDIR is high, the current values of the D bus are passed on to the MD bus. -LDBEN is connected to the 82C355. 2mA drive capability.
LDBDIR	109	Output	LOCAL DATA BUS DIRECTION CONTROL is an output to the 82C355 which controls the data direction between the D bus and the MD bus. When -LDBEN and LDBDIR are low, the current values on the MD bus are passed on to the D bus. When -LDBEN is low and LDBDIR is high the current values of the D bus are passed on to the MD bus. 2mA drive capability. -LDBDIR is connected to the 82C355.
-TEST351	34	Input	TEST PIN for the 82C351 is an active low input signal and should be pulled up with a 10K resistor for proper operation.
-WTPRES	155	Input	WEITEK PRESENT is an active low input indicating the presence a WEITEK coprocessor. -WTPRES should be pulled high with a 10K resistor.
VCC	20, 41, 59, 80, 89, 100, 121, 140, 160		+5V ±5% Supply voltage.
GND	1, 40, 50, 70, 81, 94, 106, 120, 130, 150		Ground.

3.2 Numerical Listing of Pin Assignments

Table 3-10. Numerical Pin Definitions

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	Vss	31	A3	61	-SMEMW	91	MA3
2	RESET3	32	A2	62	REFREQ	92	MA4
3	RESET4	33	A20	63	IOCHRDY	93	MA5
4	A29	34	-TEST351	64	-MASTER	94	Vss
5	A28	35	HLDA2	65	-XDEN	95	MA6
6	A27	36	-GATEA20	66	-0WS	96	MA7
7	A26	37	-SBHE	67	-LPAR	97	MA8
8	A25	38	XA0	68	-MEMR	98	MA9
9	A24	39	XA1	69	-MEMW	99	MA10
10	A23	40	Vss	70	Vss	100	Vcc
11	A22	41	Vcc	71	-IOR	101	-DWE
12	A21	42	TAG0	72	-IOW	102	-CWE0
13	CPUA20	43	TAG1	73	-AEN8	103	-CWE1
14	A19	ee	TAG2	74	HRQ1	104	-CWE2
15	A18	45	TAG3	75	BALE	105	-CWE3
16	A17	46	TAG4	76	HLDA1	106	Vss
17	A16	47	TAG5	77	-MCS16	107	-CRD
18	A15	48	TAG6	78	-IOCS16	108	-LDBEN
19	A14	49	TAG7	79	BUSCLK	109	LDBDIR
20	Vcc	50	Vss	80	Vcc	110	BWBUSY
21	A13	51	TAG8	81	Vss	111	AC0
22	A12	52	TAG9	82	ATCLKIN	112	AC1
23	A11	53	-TAGWR	83	-ROMCS	113	AC2
24	A10	54	-MALE	84	-AEN16	114	AC3
25	A9	55	-INTA	85	SDIR1	115	-ACEN
26	A8	56	CALE	86	SDIR0	116	-CAS0
27	A7	57	-RESET1	87	MA0	117	-CAS1
28	A6	58	-RESET2	88	MA1	118	-CAS2
29	A5	59	Vcc	89	Vcc	119	-CAS3
30	A4	60	-SMEMR	90	MA2	120	Vss

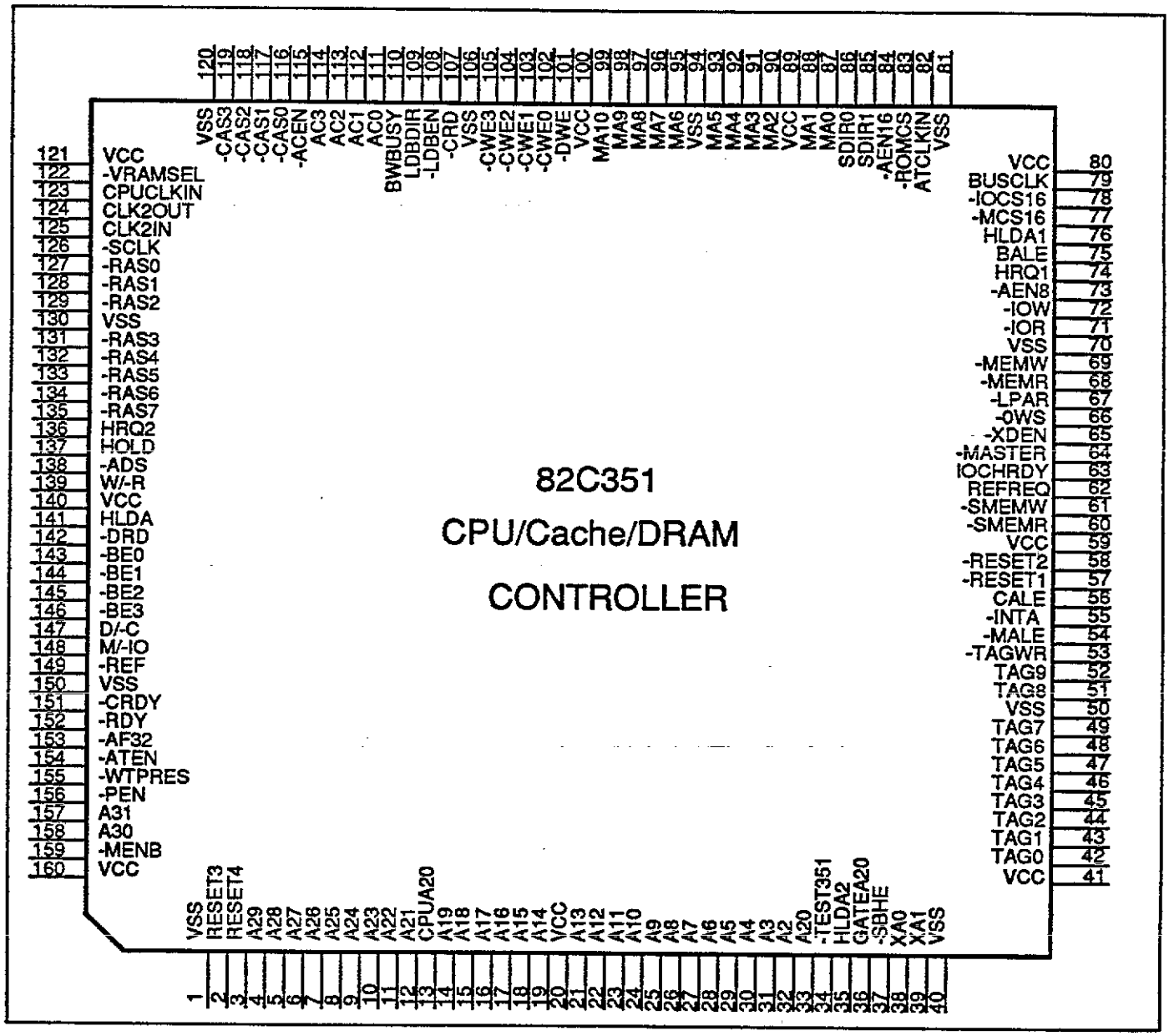
3.3 Alphabetical Listing of Pin Assignments

Table 3-11. Alphabetical Pin Definitions

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
A2	32	AC2	113	LDBDIR	109	TAG0	42
A3	31	AC3	114	-LDBEN	108	TAG1	43
A4	30	ACEN	115	-LPAR	67	TAG2	44
A5	29	-AEN8	73	MA0	87	TAG3	45
A6	28	-AEN16	84	MA1	88	TAG4	46
A7	27	ATCLKIN	82	MA2	90	TAG5	47
A8	26	BALE	75	MA3	91	TAG6	48
A9	25	BUSCLK	79	MA4	92	TAG7	49
A10	24	BWBUSY	110	MA5	93	TAG8	51
A11	23	CALE	56	MA6	95	TAG9	52
A12	22	-CAS0	116	MA7	96	TAGWR	53
A13	21	-CAS1	117	MA8	97	TEST351	34
A14	19	-CAS2	118	MA9	98	Vcc	20
A15	18	-CAS3	119	MA10	99	Vcc	41
A16	17	CPUA20	13	-MALE	54	Vcc	59
A17	16	-CRD	107	-MASTER	64	Vcc	80
A18	15	-CWE0	102	-MCS16	77	Vcc	89
A19	14	-CWE1	103	-MEMR	68	Vcc	100
A20	33	-CWE2	104	-MEMW	69	Vss	1
A21	12	-CWE3	105	REFREQ	62	Vss	40
A22	11	-DWE	101	-RESET1	57	Vss	50
A23	10	-GATEA20	36	-RESET2	58	Vss	70
A24	9	HLDA1	76	RESET3	2	Vss	81
A25	8	HLDA2	35	RESET4	3	Vss	94
A26	7	HRQ1	74	-ROMCS	83	Vss	106
A27	6	-INTA	55	-SBHE	37	Vss	120
A28	5	IOCHRDY	63	SDIR0	86	XA0	38
A29	4	-IOCS16	78	SDIR1	85	XA1	39
AC0	111	-IOR	71	-SMEMR60	-XDEN	65	
AC1	112	-IOW	72	-SMEMW	61	-0WS	66

3.4 Pin Diagram

Figure 3-1. 82C351 CPU/Cache/DRAM Controller Pin Diagram



Section 4

351 Physical Characteristics

Section 4

351 Physical Characteristics

4.1 Absolute Maximum Ratings

	Symbol	Min.	Max.	Units
Supply Voltage	Vcc		7.0	V
Input Voltage	V_I	-.5	5.5	V
Output Voltage	V_O	-.5	5.5	V
Operating Temperature	Top	-25	85	C
Storage Temperature	Tstg	-40	125	C

Note Exposure to absolute maximum ratings conditions for extended periods may affect device reliability. Exceeding the absolute maximum ratings can cause permanent damage to the device.

4.2 Operating Conditions

	Symbol	Min.	Max.	Units
Supply Voltage	Vcc	4.75	5.25	V
Ambient Temperature	T_A		70	C

This page is intentionally left blank.

Section 5

351 DC/AC Characteristics

Section 5

351 DC/AC Characteristics

5.1 DC Characteristics

	Symbol	Min.	Max.	Units
Input low voltage	V_{IL}			
TTL level (except CPUCLKIN, ATCLK, CLK2IN)			0.8	V
CMOS level (CPUCLKIN , ATCLK, CLK2IN)			1.5	V
Input high voltage	V_{IH}			
TTL level (except CPUCLKIN, ATCLK, CLK2IN)			2.0	V
CMOS level (CPUCLKIN, ATCLK, CLK2IN)			3.5	V
Output low voltage	V_{OL}	.40	V	
Output high voltage	V_{OH}		2.4	V
Input LOW current @ Vo = Vss	I_{IL}	-200	-10	µA
Input HIGH current @ Vo = Vdd	I_{IH}	-10	10	µA
3-State output OFF current LOW	I_{OZL}		-10	µA
3-State output OFF current HIGH	I_{OZH}		10	µA
Power supply current @25 MHz	I_{CC}		80	µA
Input capacitance	C_{IN}		10	pF
Output or I/O capacitance	C_{OUT}		10	pF

5.2 AC Characteristics

All timing parameters are specified under capacitive load of 50 pf and temperature of 70 degree C. All the units discussed in the following timing tables are in nanoseconds, unless otherwise specified. Also, the AC specifications mentioned in this document are subject to change.

Table 5-1. Clocks

	Clocks	25 MHz	
		Min.	Max.
t101	Operating frequency	25 MHz	
t102	CPUCLKIN period	20	
t104	CPUCLKIN high time at 3.5V	8	
t106	CPUCLKIN low time at 1.5V	8	
t107	CPUCLKIN fall time (3.5V to 1.5V)		4
t108	CPUCLKIN rise time (1.5V to 3.5V)		4
t108a	CLK2IN delay to CPUCLK	0	15
t109	CPUCLKIN high to -SCLK low	6	27
t110	CPUCLKIN high to -SCLK high	6	27

Table 5-2. Memory Cycle

	Memory Cycle	25 MHz	
		Min.	Max.
t120	Address setup to CPUCLKIN high	37	
t121	W/-R, M/-IO, D/-C setup to CPUCLKIN low	17	
t121a	-ADS setup to CPUCLKIN low	10	
t122	-BE<3:0> setup to CPUCLKIN high	30	
t123	-READY active delay from CPUCLKIN low (0 WS writes)	2	20
t125	HOLD delay from CPUCLKIN high	2	19
t126	-MALE active delay from CPUCLKIN high	0	30
t127	-MALE inactive delay from CPUCLKIN low	0	30
t128	-AF32 active delay from CPUCLKIN		23
t129	-AF32 inactive delay from CPUCLKIN low		25
t134	-READY active delay from CPUCLKIN high	6	30
t135	-READY inactive delay from CPUCLKIN high	4	35
t141	-RAS active delay from CPUCLKIN high (early RAS)	7	22
t142	-RAS active delay from CPUCLKIN high (late RAS)	7	22
t143	-RAS inactive delay from CPUCLKIN high	7	25
t145	Row address valid from CPU address active		25
t146	Row address hold from CPUCLKIN low	7	25
t146a	Column address valid from CPUCLKIN low		30
t147	Column address valid to CAS- active	8	30
t148	Column address hold from CPUCLKIN high	7	25
t150	-CAS active delay from CPUCLKIN high	7	22
t151	-CAS inactive delay from CPUCLKIN high	7	27
t152	-DRD active delay from CPUCLKIN high	10	25
t154	-DRD inactive delay from CPUCLKIN high	10	25
t159	-CWE active delay from CPUCLKIN high (read miss)	5	20
t160	-CWE inactive delay from CPUCLKIN high (read miss)	6	19
t161a	-DWE active delay from CPUCLKIN high	7	20
t162	-DWE inactive delay from CPUCLKIN high	7	30

Table 5-2. *Memory Cycle (continued)*

	Memory Cycle	25 MHz	
		Min.	Max.
t165	LDBDIR valid from CPUCLKIN low	9	45
t166	Row address valid from CPUCLKIN high	7	35
t167a	BWBUSY active delay from CPUCLKIN high (normal)	7	22
t167b	BWBUSY active delay from CPUCLKIN high (0 WS write)	7	22
t168	BWBUSY inactive delay from CPUCLKIN high (0 WS writes)		40
t169	-LBDEN active from CPUCLKIN high (normal write)		40
t169a	-LBDEN active from CPUCLKIN high (read)		40
t169b	-LBDEN active from CPUCLKIN high (0 WS write)		40
t169c	-LBDEN inactive from CPUCLKIN high		29

Table 5-3. *Cache Timing*

	Cache Timing	25 MHz	
		Min.	Max.
t132	-CRD active from CPUCLKIN high		30
t132a	-CRD active from -ADS low		30
t132b	-CRD active from W/-R low		30
t133	-CRD inactive from CPUCLKIN high		12
t138b	CALE active from CPUCLKIN high		20
t138a	CALE inactive from CPUCLKIN high		27
t156	-CWEn pulse width (0WS writes)	18	
t157	-CWEn pulse width (1WS writes)	35	
t158	CALE active to -CWEn inactive		3

Table 5-4. *Tag RAMS*

	Tag RAMS	25 MHz	
		Min.	Max.
t200	TAG<0:9> valid to -CRDY valid		22
t201	TAG<0:9> setup to CPUCLKIN high Early RAS	14	
t202	TAG<0:9> setup to CPUCLKIN high Late RAS	25	
t203	TAG<0:9> setup to CPUCLKIN high 0WS write decision	11	
t204	TAG<0:9> setup to CPUCLKIN high 1WS write decision	25	
t205	-TAGWE active to TAG<0:9> low Z	1 CPUCLKIN	
t207	TAG<0:9> float from -TAGWE inactive	0	5

Table 5-5. *DMA Cycle*

	DMA Cycle	25 MHz	
		Min.	Max.
t170	Command setup to CPUCLKIN high	20	
t171	-RAS active delay from CPUCLKIN high	0	22
t172	-RAS inactive delay from commands inactive	0	25
t173	Row address setup from address valid	0	20
t174	Row address hold from CPUCLKIN high	10	25
t175	Column address setup to -CAS active	1.5 CPUCLKIN	
t175a	Column address valid from CPUCLKIN high	12	27
t176	Column address delay to valid from CPUCLKIN low		25
t177	-CAS active delay from -RAS active (DMA memory read)	3 CPUCLKIN	
t178	-CAS active delay from -RAS active (DMA memory write)	4 CPUCLKIN	
t179	-CAS active delay from CPUCLKIN high	8	22
t180	-CAS inactive delay from commands inactive	8	20
t185	-AF32 active delay from command active	0	30
t186	-AF32 inactive delay from commands inactive	0	70
t187	-DWE active delay from CPUCLKIN high	0	17
t188	-DWE inactive delay from commands inactive	0	17
t189	-CWE active delay from CPUCLKIN high	0	20
t190	-CWE inactive delay from CPUCLKIN high	10	20

Table 5-6. *ROM Cycle*

	ROM Cycle	25 MHz	
		Min.	Max.
t191	-ROMCS active delay from CPUCLKIN high	0	25
t192	-ROMCS inactive delay from CPUCLKIN high	0	25

Table 5-7. *Local Bus Slave Cycle*

	Local Bus Slave Cycle	25 MHz	
		Min.	Max.
t193	-READY input setup to CPUCLKIN high	16	
t194	-READY input hold from CPUCLKIN high	6	
t195	-AF32 setup to CPUCLKIN high	15	
t196	-AF32 hold from CPUCLKIN high	6	

Table 5-8. *AT Bus Access*

	AT Bus Access	25 MHz	
		Min.	Max.
t1100	-ATEN active from CPUCLKIN high	0	40
t1101	-ATEN inactive from CPUCLKIN high	0	40
t1102	BALE active from BUSCLK low	0	20
t1103	BALE inactive from BUSCLK high	0	20
t1104	Command active delay from BUSCLK low	0	20
t1105	Command inactive from BUSCLK high	0	20
t1106	-ACEN active delay from BUSCLK low (read)	0	20
t1107	-ACEN inactive delay from BUSCLK high (read)	0	20
t1108	AC<0:3> active delay from BUSCLK low	0	20
t1112	-MCS16, -IOCS16 setup to BUSCLK high	25	
t1113	-CS16, -IOCS16 hold from BUSCLK high	5	
t1116	Command active delay from BUSCLK high (write)	0	20
t1117	-ACEN active delay from BUSCLK high (write)	0	20
t1118	-ACEN inactive delay from BUSCLK low (write)	0	20
t1119	-0WS setup to BUSCLK low	20	
t1120	-0WS hold from BUSCLK high	5	
t1123	XA<0:1>, -SBHE valid delay from BUSCLK low	0	18

Table 5-9. *DMA Arbitration*

	DMA Arbitration	25 MHz	
		Min.	Max.
t1140	HRQn active setup to CPUCLKIN high	15	
t1141	HRQn inactive setup to CPUCLKIN high	15	
t1142	HOLD active delay from CPUCLKIN high	2	30
t1143	HOLD inactive delay from CPUCLKIN high	5	25
t1144	HLDAn active delay from CPUCLKIN high	2	19
t1145	HLDAn inactive delay from CPUCLKIN high		30

Table 5-10. *Refresh Arbitration*

	Refresh Arbitration	25 MHz	
		Min.	Max.
t1148	REFREQ pulse width (low)	3 CPUCLKIN	
t1150	-MEMR delay from BUSCLK high	0	25

Table 5-11. *Reset Timing*

	Reset Timing	25 MHz Min.	25 MHz Max.
t1170	RESET3 active delay from CPUCLKIN high	0	25
t1171	RESET3 inactive delay from CPUCLKIN high	2	15
t1172	RESET3 pulse width	64	85 CPUCLKIN
t1173	RESET4 active delay from CPUCLKIN high	0	25
t1174	RESET4 inactive delay from CPUCLKIN high	2	15

Table 5-12. *Miscellaneous Timing*

	Miscellaneous Timing	25 MHz Min.	25 MHz Max.
t1187	BUSCLK high pulse width	programmable	
t1188	BUSCLK low pulse width	programmable	
t1189	BUSCLK rise time		6
t1190	BUSCLK fall time		6
t1191	FGA20 valid from -IOW inactive		18
t1193	A20 delay from CPUA20	0	16

Table 5-13. *Refresh Cycle*

	Refresh Cycle	25 MHz Min.	25 MHz Max.
t1200	-RASi active delay from CPUCLKIN high		25
t1201	-RASi inactive delay from CPUCLKIN high	0	15
t1202	-RASi pulse width	see Note	
t1203	-RAS(i+1) active delay from -RASi active	1 CPUCLKIN	
t1204	REFRESH address setup to -RAS0	2 CPUCLKIN	
t1205	REFRESH address hold from -RAS3	1 CPUCLKIN	

Note The RAS pulse width is programmable to 4, 5, 6, 7 CPUCLKIN

Table 5-14. *LPAR Timing*

	LPAR Timing	25 MHz Min.	25 MHz Max.
t1220	-LPAR input setup to -CAS high	10	
t1221	-LPAR input hold from -CAS high	10	

Table 5-15. *Clock Cycle*

	Clock Cycles	33 MHz Min.	33 MHz Max.
t101	Operating frequency		33 MHz
t102	CPUCLKIN period	15	
t104	CPUCLKIN high time at 3.5V	6	
t106	CPUCLKIN low time at 1.5V	6	
t107	CPUCLKIN fall time (3.7V to .8V)		4
t108	CPUCLKIN rise time (.8V to 3.7V)		4
t108a	CPUCLKIN delay to CPUCLK	0	15
t109	CPUCLKIN high to -SCLK low	5	25
t110	CPUCLKIN high to -SCLK high	5	25

Table 5-16. *Memory Cycle*

	Memory Cycle	33 MHz Min.	33 MHz Max.
t120	Address setup to CPUCLKIN high	28	
t121	W/-R, M/-IO, D/-C, setup to CPUCLKIN low	13	
t121a	-ADS setup to CPUCLKIN low	4	
t122	-BE<0:3> setup to CPUCLKIN high	25	
t123	-READY delay from CPUCLKIN low (0 WS writes)	1	14
t125	HOLD delay from CPUCLKIN high	2	25
t126	-MALE active delay from CPUCLKIN low	0	30
t127	-MALE inactive delay from CPUCLKIN low	0	30
t128	-AF32 active delay from CPUCLKIN low		17
t129	-AF32 inactive delay from CPUCLKIN low	11	20
t134	-READY active delay from CPUCLKIN high	5	23
t135	-READY inactive delay from CPUCLKIN high	5	30
t141	-RAS active delay from CPUCLKIN high (early RAS)	5	20
t142	-RAS active delay from CPUCLKIN high (late RAS)	5	20
t143	-RAS inactive delay from CPUCLKIN high	7	25
t145	Row address valid from CPU address active		25
t146	Row address hold from CPUCLKIN low	10	22
t146a	Column address valid from CPUCLKIN low		25
t147	Column address valid to -CAS active	9	30
t148	Column address hold from CPUCLKIN high	7	22
t150	-CAS active delay from CPUCLKIN high	7	20
t151	-CAS inactive delay from CPUCLKIN high	7	25
t152	-DRD active delay from CPUCLKIN high	8	30
t154	-DRD inactive delay from CPUCLKIN high	8	22
t159	-CWE active delay from CPUCLKIN high (read miss)	8	18
t160	-CWE inactive delay from CPUCLKIN high (read miss)	0	14
t161a	-DWE active delay from CPUCLKIN high	7	18
t162	-DWE inactive delay from CPUCLKIN high	7	30

Table 5-16. Memory Cycle (continued)

	Memory Cycle	33 MHz	
		Min.	Max.
t165	LDBDIR valid from CPUCLKIN low	7	45
t166	Row address valid from CPUCLKIN high	7	30
t167a	BWBUSY active delay from CPUCLKIN high (normal)	7	20
t167b	BWBUSY active delay from CPUCLKIN high (0WS writes)	7	23
t168	BWBUSY inactive delay from CPUCLKIN high (0WS writes)		40
t169	-LBDEN active from CPUCLKIN high (normal write)		40
t169a	-LBDEN active from CPUCLKIN high (read)		40
t169b	-LBDEN active from CPUCLKIN high (0 WS write)		40
t169c	-LBDEN inactive from CPUCLKIN high		29

Table 5-17. Cache Timing

	Cache Timing	33 MHz	
		Min.	Max.
t132	-CRD active from CPUCLKIN high		20
t132a	-CRD active from -ADS low		20
t132b	-CRD active from W/R low		20
t133	-CRD inactive from CPUCLKIN high		12
t138b	CALE active from CPUCLKIN high		14
t138a	CALE inactive from CPUCLKIN high		27
t156	-CWEn pulse width (0WS writes)	13	
t157	-CWEn pulse width (1WS writes)	26	
t158	CALE active to -CWEn active		3

Table 5-18. Tag RAMS

	Tag RAMS	33 MHz	
		Min.	Max.
t200	TAG<0:9> valid to -CRDY valid		20
t201	TAG<0:9> setup to CPUCLKIN high, Early RAS	11	
t202	TAG<0:9> setup to CPUCLKIN high, Late RAS	20	
t203	TAG<0:9> setup to CPUCLKIN high, 0WS write decision	11	
t204	TAG<0:9> setup to CPUCLKIN high, 1WS write decision	20	
t205	-TAGWE active to TAG<0:9> low Z	1 CPUCLKIN	
t207	TAG<0:9> float from -TAGWE inactive	0	5

Table 5-19. DMA Cycle

	DMA Cycle	33 MHz	
		Min.	Max.
t170	Command setup to CPUCLKIN high	20	
t171	-RAS active delay from CPUCLKIN high	0	20
t172	-RAS inactive delay from commands inactive	0	25
t173	Row address setup from address valid	0	20
t174	Row address valid from CPUCLKIN high	10	25
t175	Column address setup to -CAS active	1.5 CPUCLKIN	
t175a	Column address valid from CPUCLKIN high	12	27
t176	Column address delay to valid from CPUCLKIN low		25
t177	-CAS active delay from -RAS active (DMA memory read)	3 CPUCLKIN	
t178	-CAS active delay from -RAS active (DMA memory write)	4 CPUCLKIN	
t179	-CAS active delay from CPUCLKIN high	8	20
t180	-CAS inactive delay from commands inactive	8	20
t185	-AF32 active delay from command active	0	30
t186	-AF32 inactive delay from commands inactive	0	70
t187	-DWE active delay from CPUCLKIN high	0	17
t188	-DWE inactive delay from commands inactive	0	17
t189	-CWE active delay from CPUCLKIN high	0	20
t190	-CWE inactive delay from CPUCLKIN high	10	20

Table 5-20. ROM Cycle

	ROM Cycle	33 MHz	
		Min.	Max.
t191	-ROMCS active delay from CPUCLKIN high	0	25
t192	-ROMCS inactive delay from CPUCLKIN high	0	25

Table 5-21. Local Bus Slave Cycle

	Local Bus Slave Cycle	33 MHz	
		Min.	Max.
t193	-READY input setup to CPUCLKIN high	12	
t194	-READY input hold from CPUCLKIN high	6	
t195	-AF32 setup to CPUCLKIN high	10	
t196	-AF32 hold from CPUCLKIN high	6	

Table 5-22. AT Bus Access

	AT Bus Access	33 MHz	
		Min.	Max.
t1100	-ATEN active from CPUCLKIN high	0	40
t1101	-ATEN inactive from CPUCLKIN high	0	40
t1102	BALE active from BUSCLK low	0	20
t1103	BALE inactive from BUSCLK high	0	20
t1104	Command active delay from BUSCLK low	0	20
t1105	Command inactive from BUSCLK high	0	20
t1106	-ACEN active delay from BUSCLK low (read cycle)	0	20
t1107	-ACEN inactive delay from BUSCLK high (read cycle)	0	20
t1108	AC<0:3> active delay from BUSCLK low	0	20
t1112	-MCS16, -IOCS16 setup to BUSCLK high	25	
t1113	-MCS16, -IOCS16 hold from BUSCLK high	5	
t1116	Command active delay from BUSCLK high (write)	0	20
t1117	-ACEN active delay from BUSCLK high (write)	0	20
t1118	-ACEN inactive delay from BUSCLK low (write)	0	20
t1119	-0WS setup to BUSCLK low	20	
t1120	-0WS hold from BUSCLK high	5	
t1123	XA<0:1>, -SBHE valid delay from BUSCLK low	0	18

Table 5-23. DMA Arbitration

	DMA Arbitration	33 MHz	
		Min.	Max.
t1140	HRQn active setup to CPUCLKIN high	15	
t1141	HRQn inactive setup to CPUCLKIN high	15	
t1142	HOLD active delay from CPUCLKIN high	2	25
t1143	HOLD inactive delay from CPUCLKIN high	4	23
t1144	HLDAn active delay from CPUCLKIN high	2	19
t1145	HLDAn inactive delay from CPUCLKIN high		30

Table 5-24. Refresh Arbitration

	Refresh Arbitration	33 MHz	
		Min.	Max.
t1148	REFREQ pulse width (low)	3 CPUCLKIN	
t1150	-MEMR delay from BUSCLK high	0	25

Table 5-25. Reset Timing

	Reset Timing	33 MHz Min.	33 MHz Max.
t1170	RESET3 active delay from CPUCLKIN high	0	25
t1171	RESET3 inactive delay from CPUCLKIN high	2	12
t1172	RESET3 pulse width	64	85 CPUCLKIN
t1173	RESET4 active delay from CPUCLKIN high	0	25
t1174	RESET4 inactive delay from CPUCLKIN high	2	12

Table 5-26. Miscellaneous Timing

	Miscellaneous Timing	33 MHz Min.	33 MHz Max.
t1187	BUSCLK high pulse width	Programmable	
t1188	BUSCLK low pulse width	Programmable	
t1189	BUSCLK rise time		6
t1190	BUSCLK fall time	6	
t1191	FGA20 valid from -IOW inactive		18
t1193	A20 delay from CPUA20	0	16

Table 5-27. Refresh Cycle

	Refresh Cycle	33 MHz Min.	33 MHz Max.
t1200	-RASi active delay from CPUCLKIN high		25
t1201	-RASi inactive delay from CPUCLKIN high	0	15
t1202	-RASi pulse width	see Note	
t1203	-RAS(i+1) active delay from -RASi active	1 CPUCLKIN	
t1204	REFRESH address setup to -RAS0	2CPUCLKIN	
t1205	REFRESH address hold from -RAS3	1CPUCLKIN	

Note The -RAS pulse width is programmable to 4, 5, 6, 7 CPUCLKIN.

Table 5-28. LPAR Timing

	LPAR Timing	33 MHz Min.	33 MHz Max.
t1220	-LPAR input setup to -CAS inactive	10	
t1221	-LPAR input hold to -CAS inactive	10	

This page is intentionally left blank.

Section 6

351 Timing Diagrams

Section 6

351 Timing Diagrams

Figure 6-1. 82C351 AC Timing Waveforms

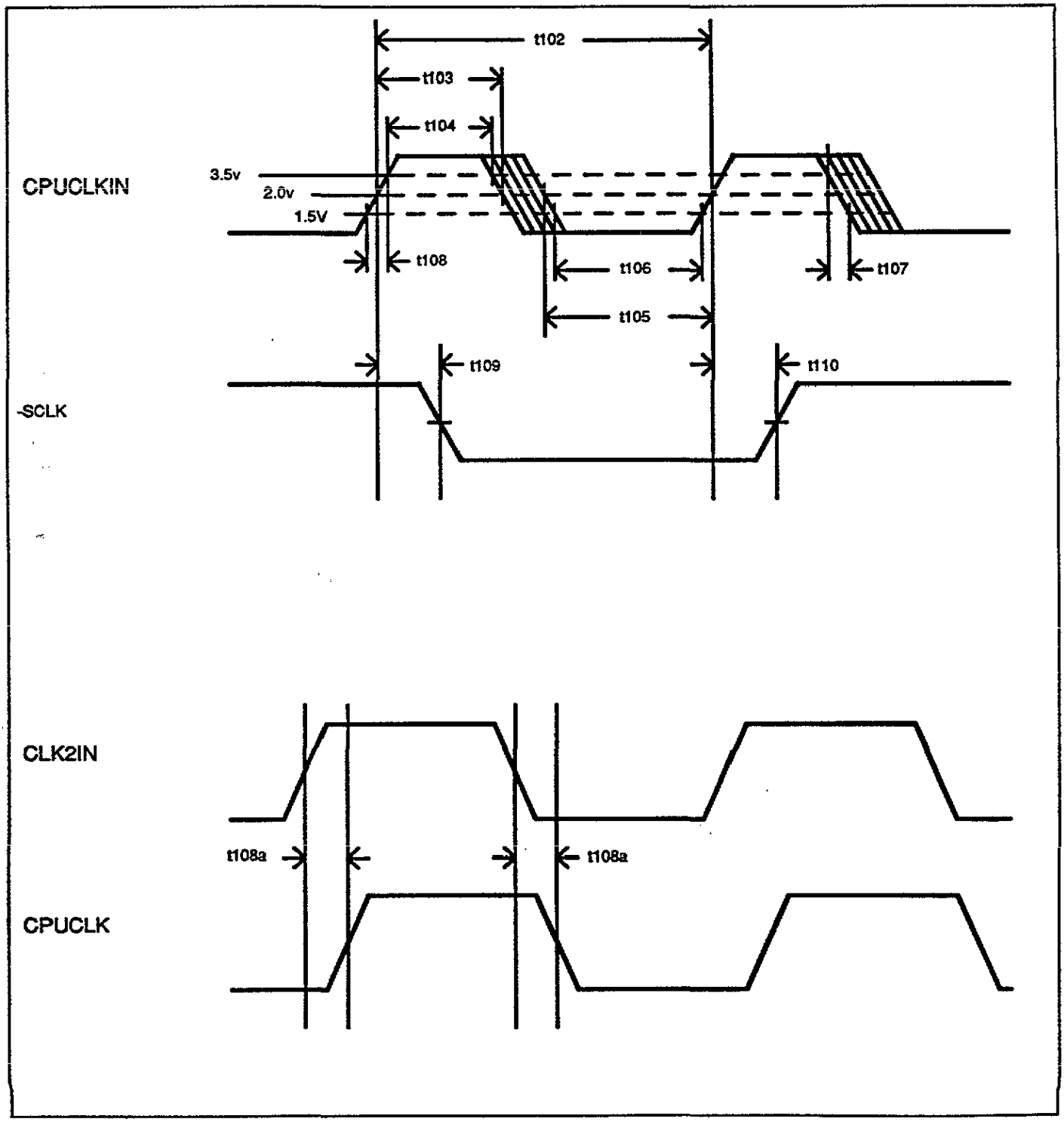


Figure 6-2. Back to Back Write Early RAS, Four Wait States

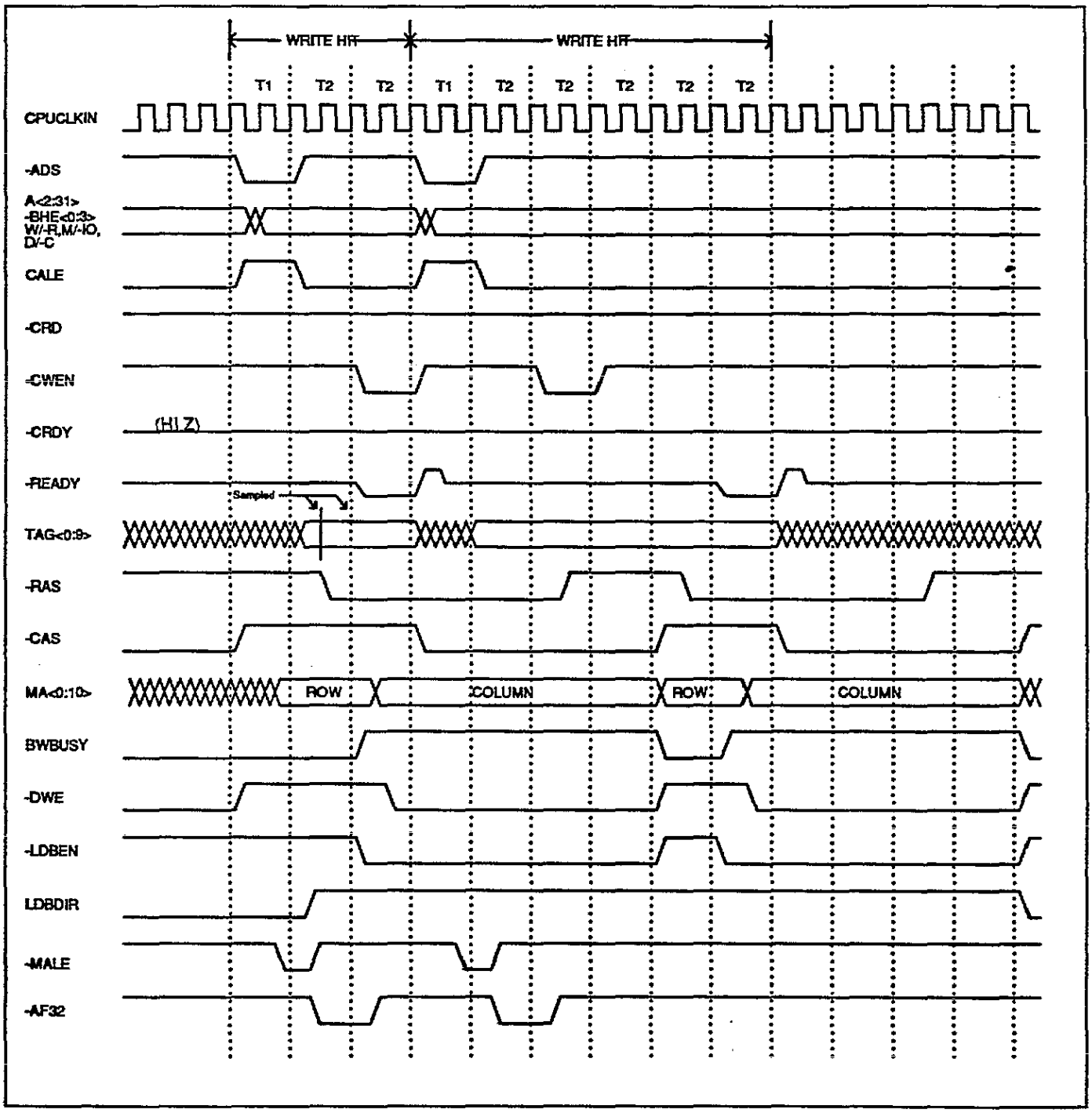


Figure 6-3. 82C351 One Wait State Write Mode, Late RAS Mode, Five Wait State DRAM

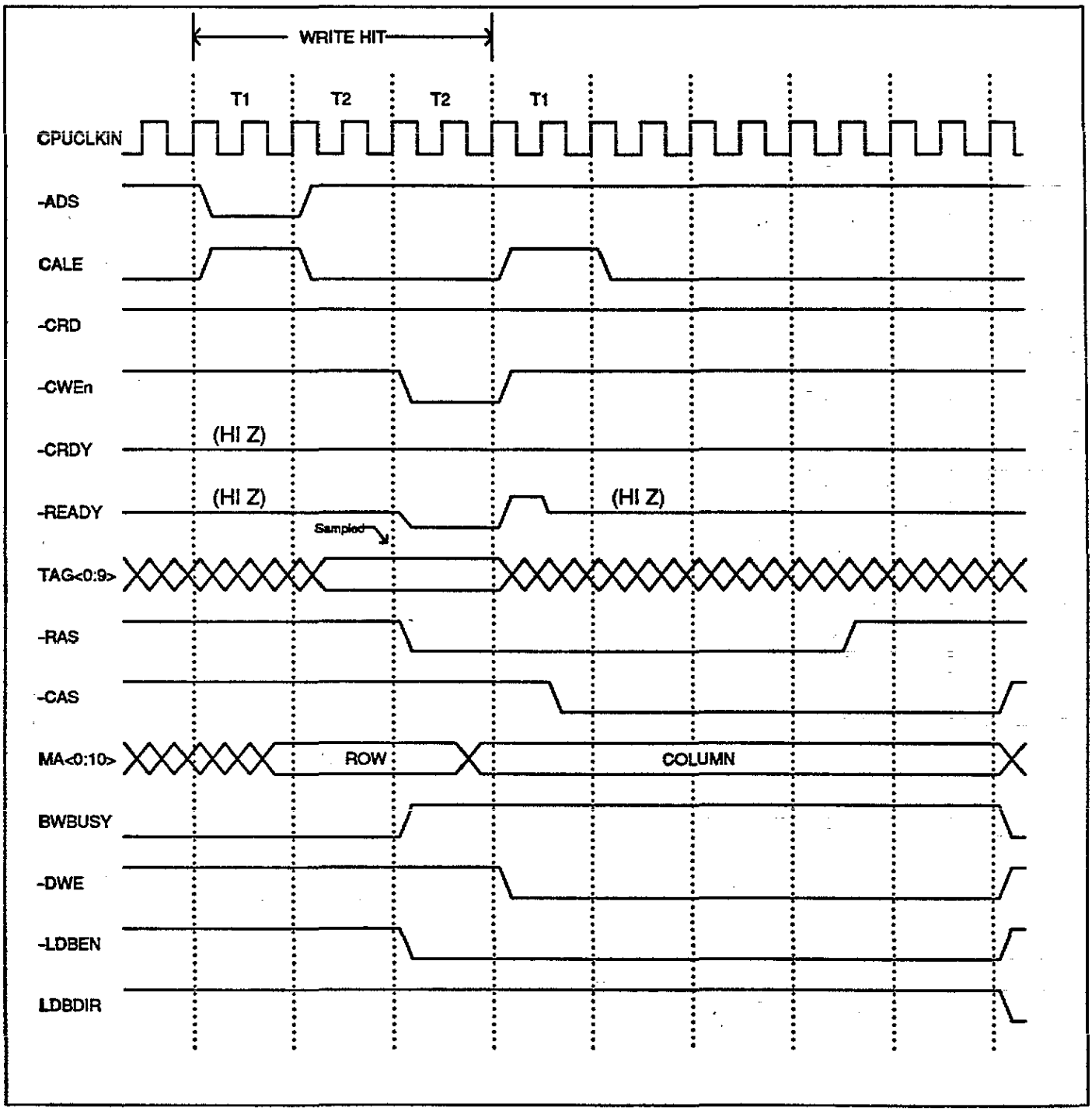


Figure 6-4. 82C351 and 386DX Interface

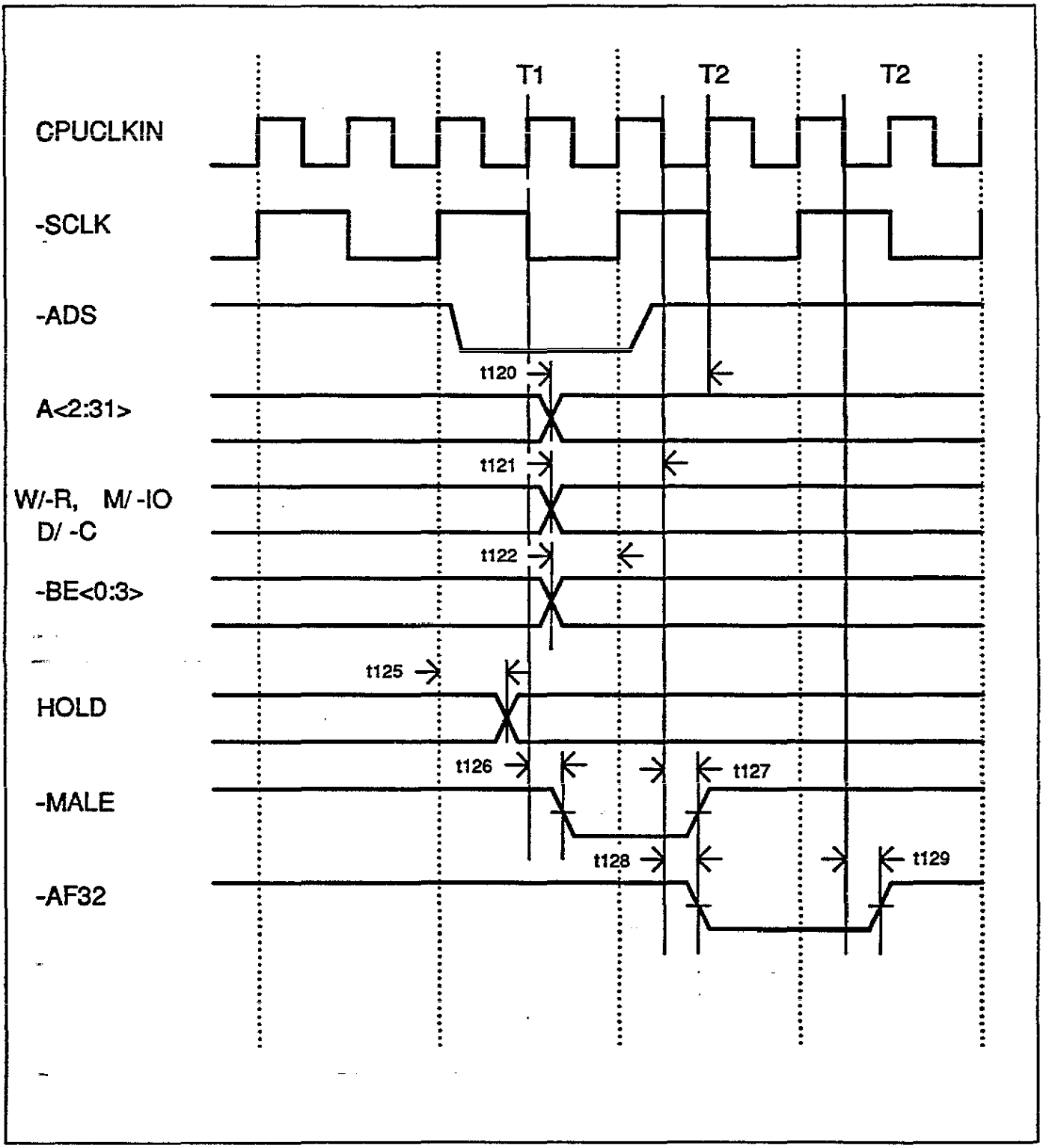


Figure 6-5. *82C351 Write Hit, One Wait State Cache, Early RAS Mode, Five Wait States DRAM*

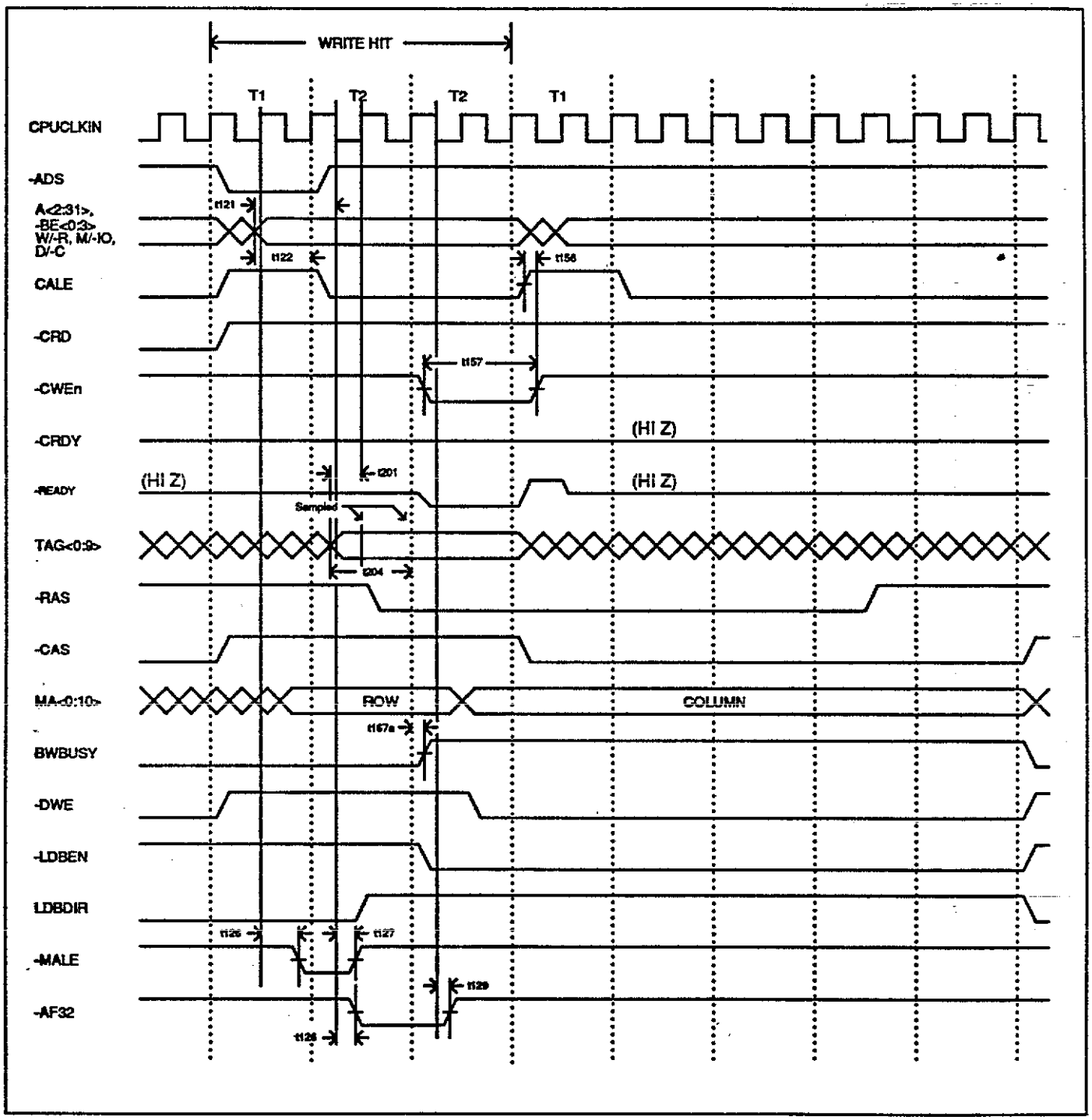


Figure 6-6. 82C351 Read Miss, Late RAS Mode, Three Wait States DRAM

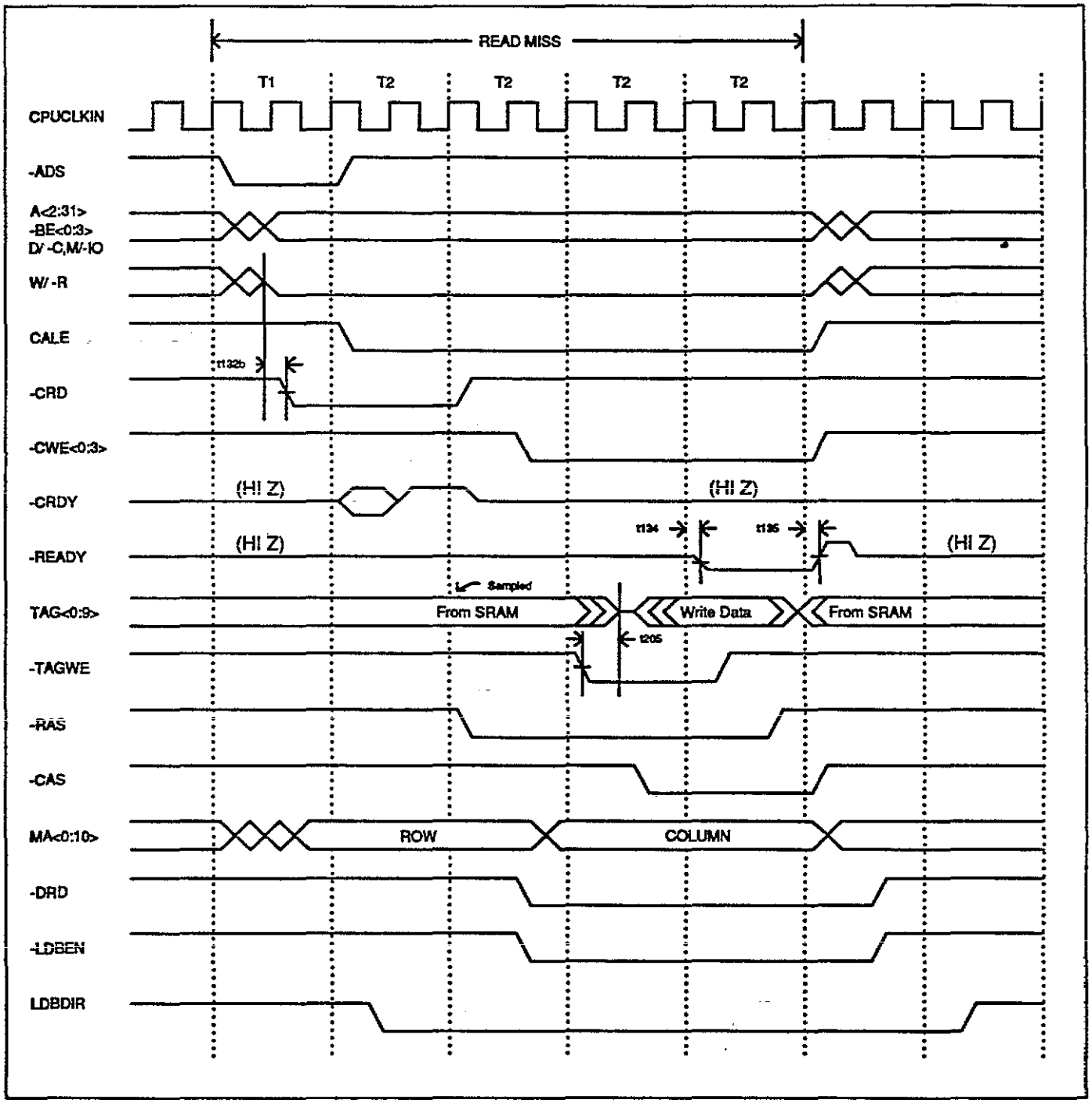


Figure 6-7. *82C351 Read Miss, Late RAS Mode, Four Wait States DRAM*

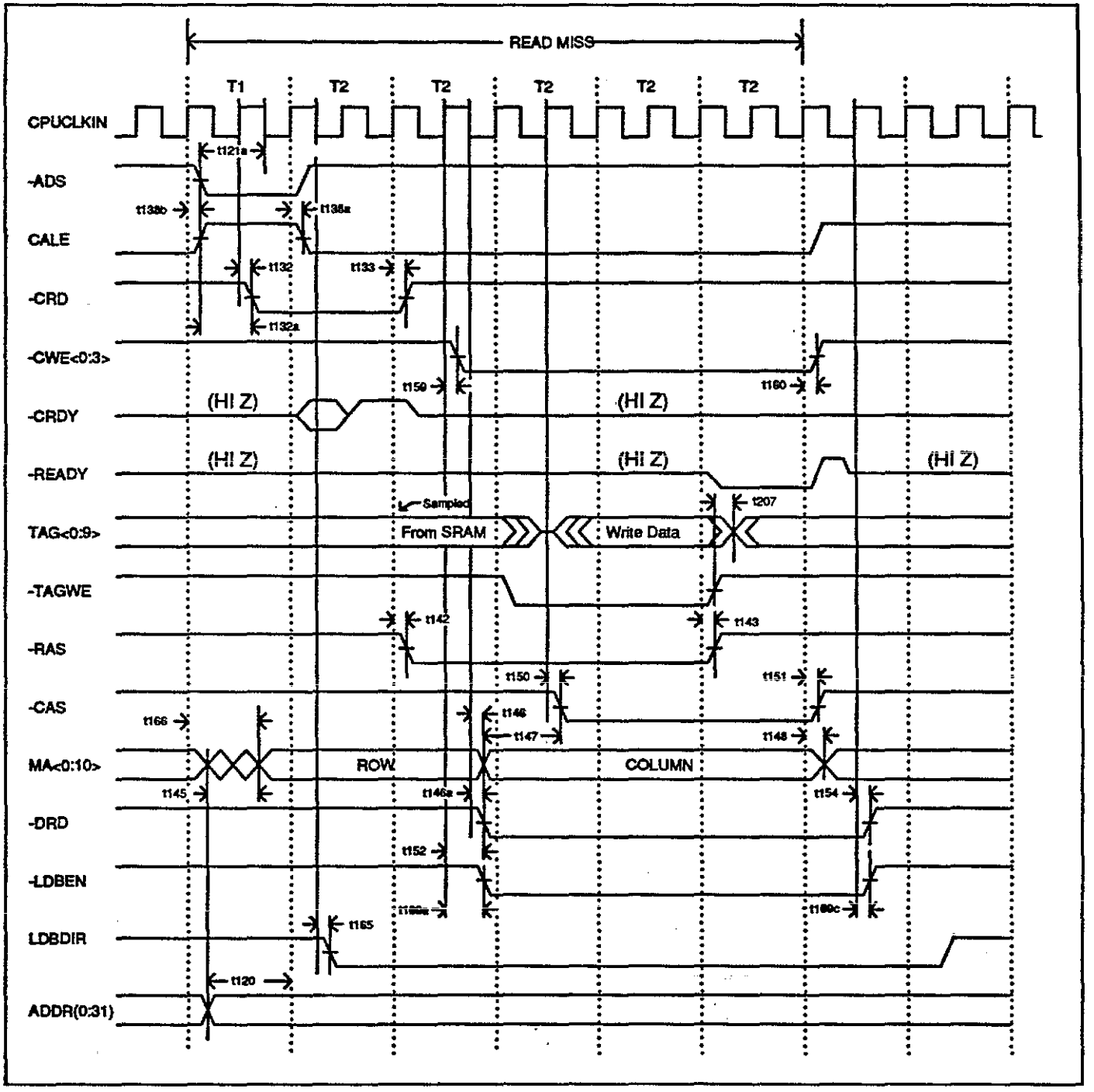


Figure 6-8. *82C351 Read Miss, Early RAS Mode, Three Wait States DRAM*

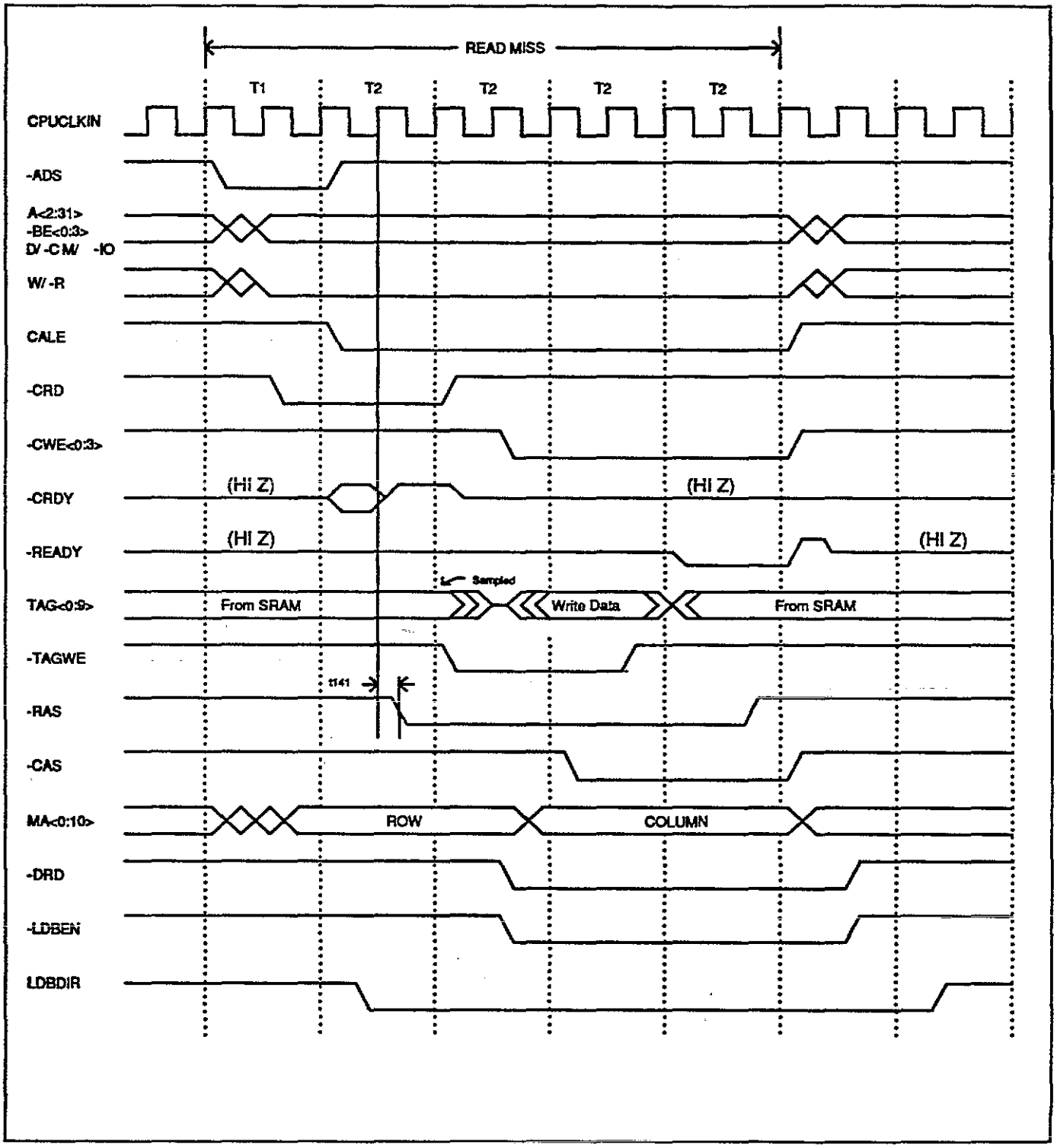


Figure 6-9. *82C351 Read Hit/Write Hit, Zero Wait State Cache, Late RAS Mode, Four Wait States DRAM*

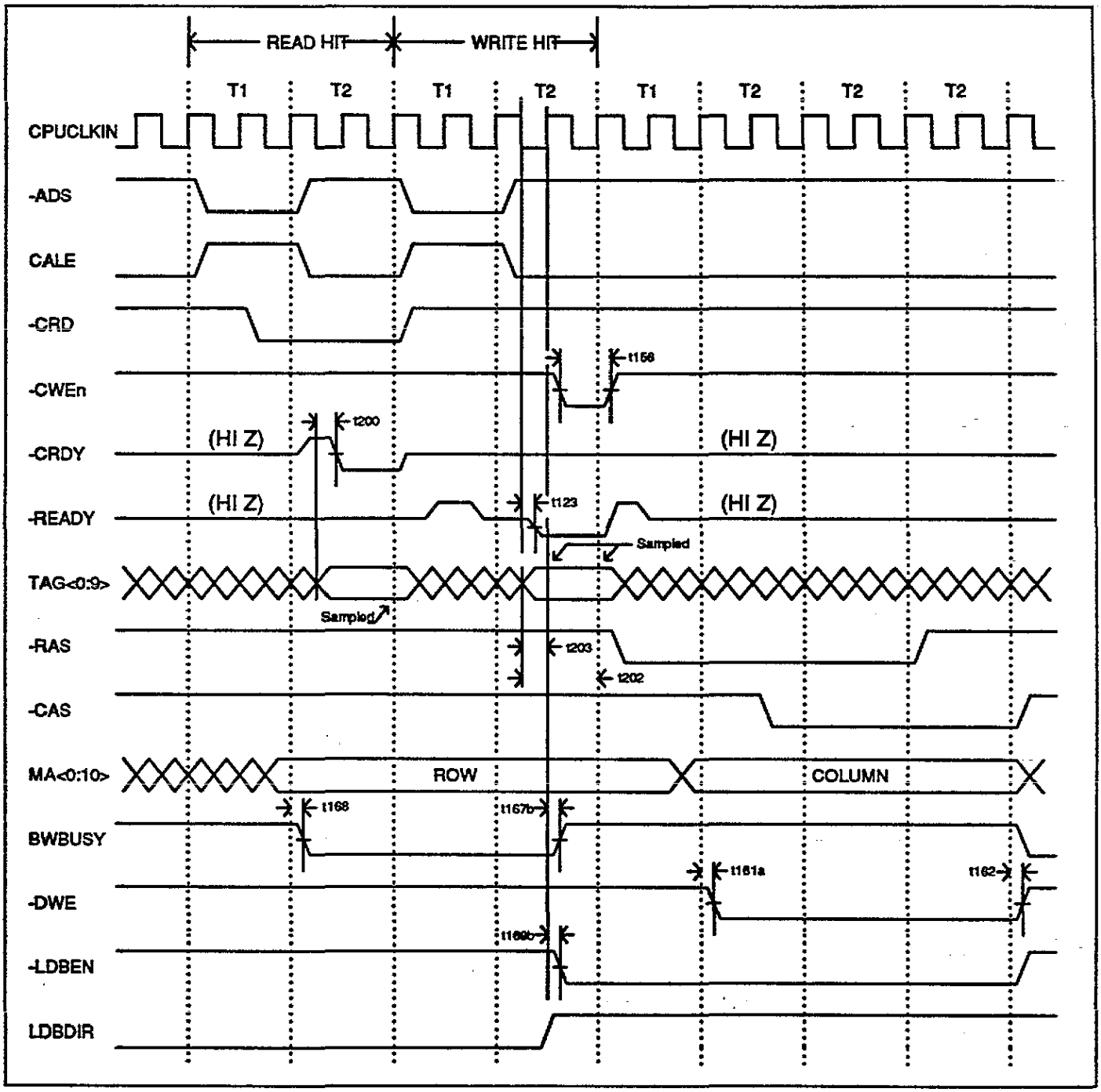
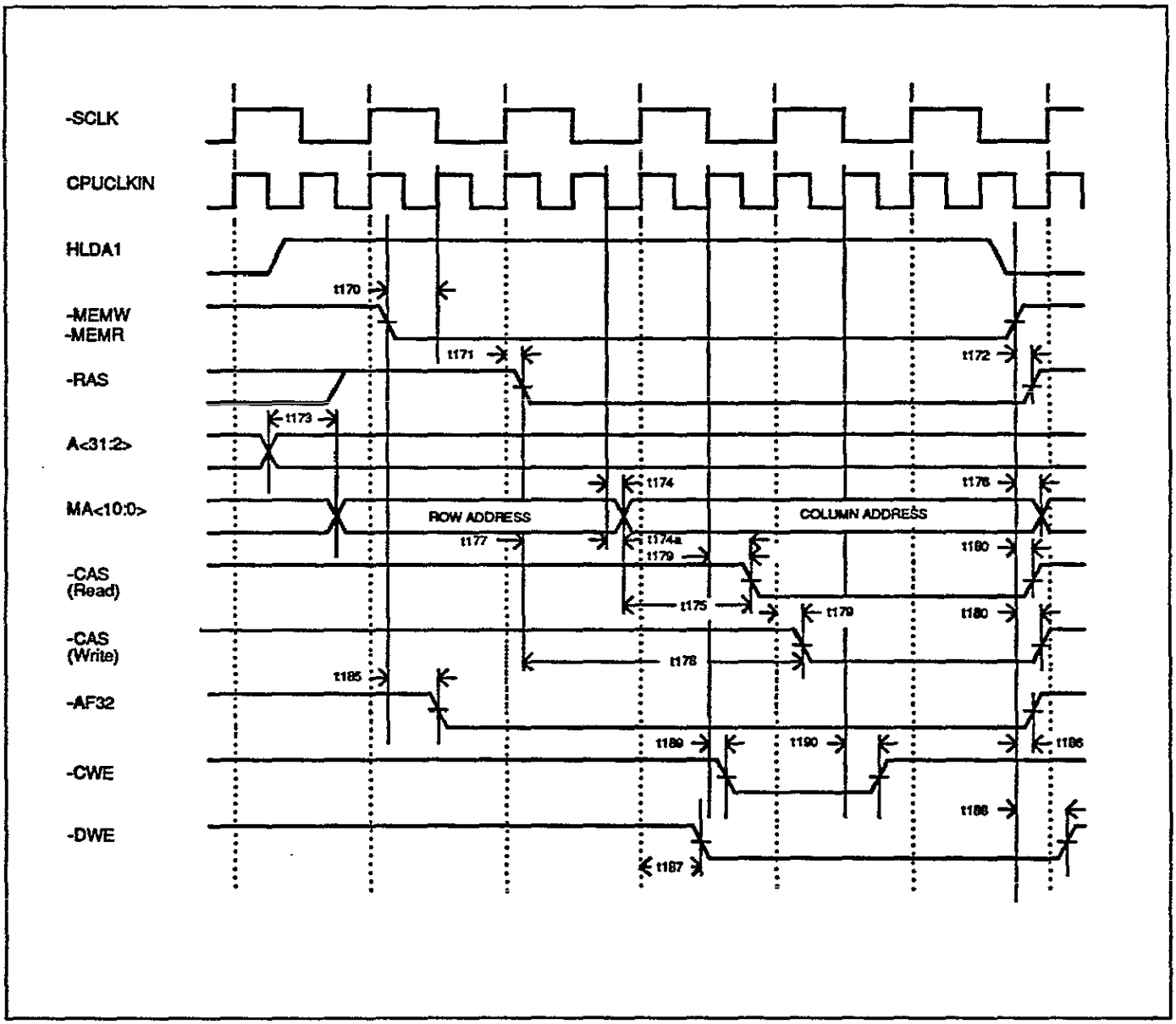


Figure 6-10. *82C351 DMA Cycle*



Note -CRD and -CWE are generated during DMA Write Hit Cycles.
-CRD is generated for chip select type SRAMs.
-AF32 is generated during DMA Cycle for 82C351 to generate the proper Action Code.

Figure 6-11. ROM Read Cycle

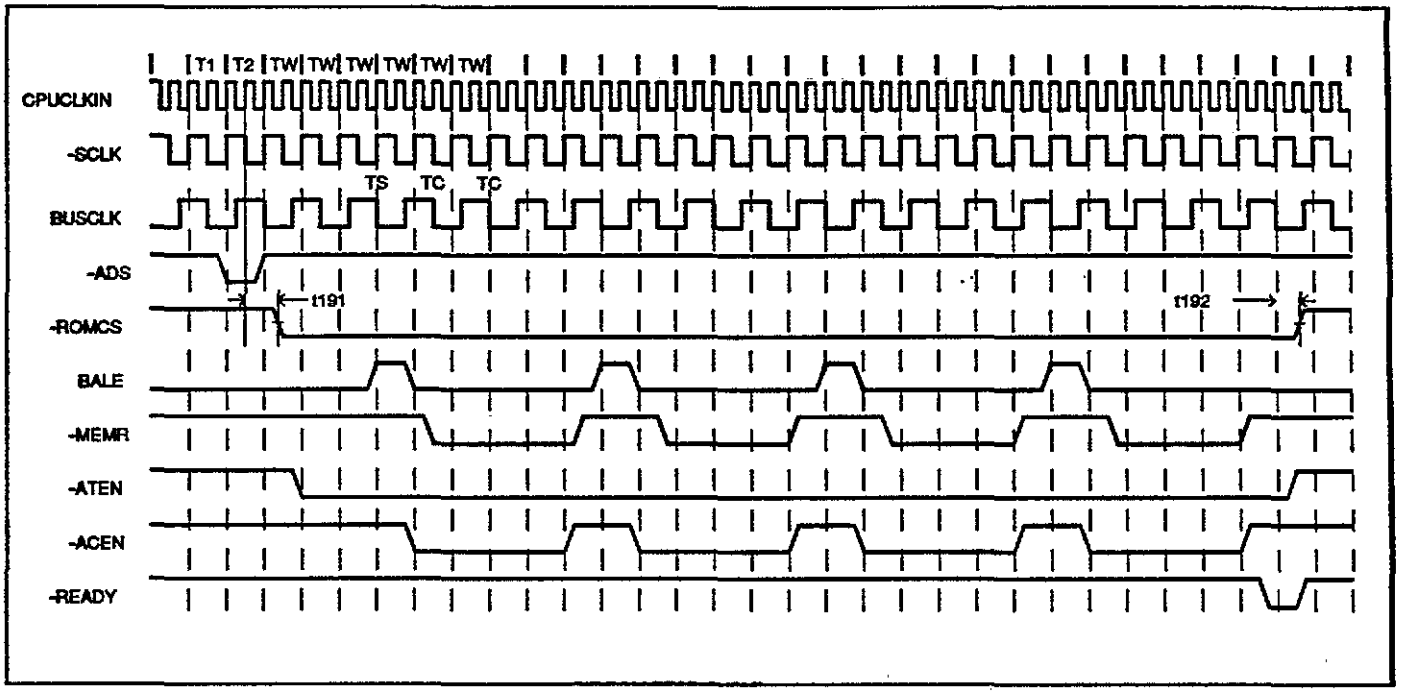


Figure 6-12. *82C351 AT Read Cycle*

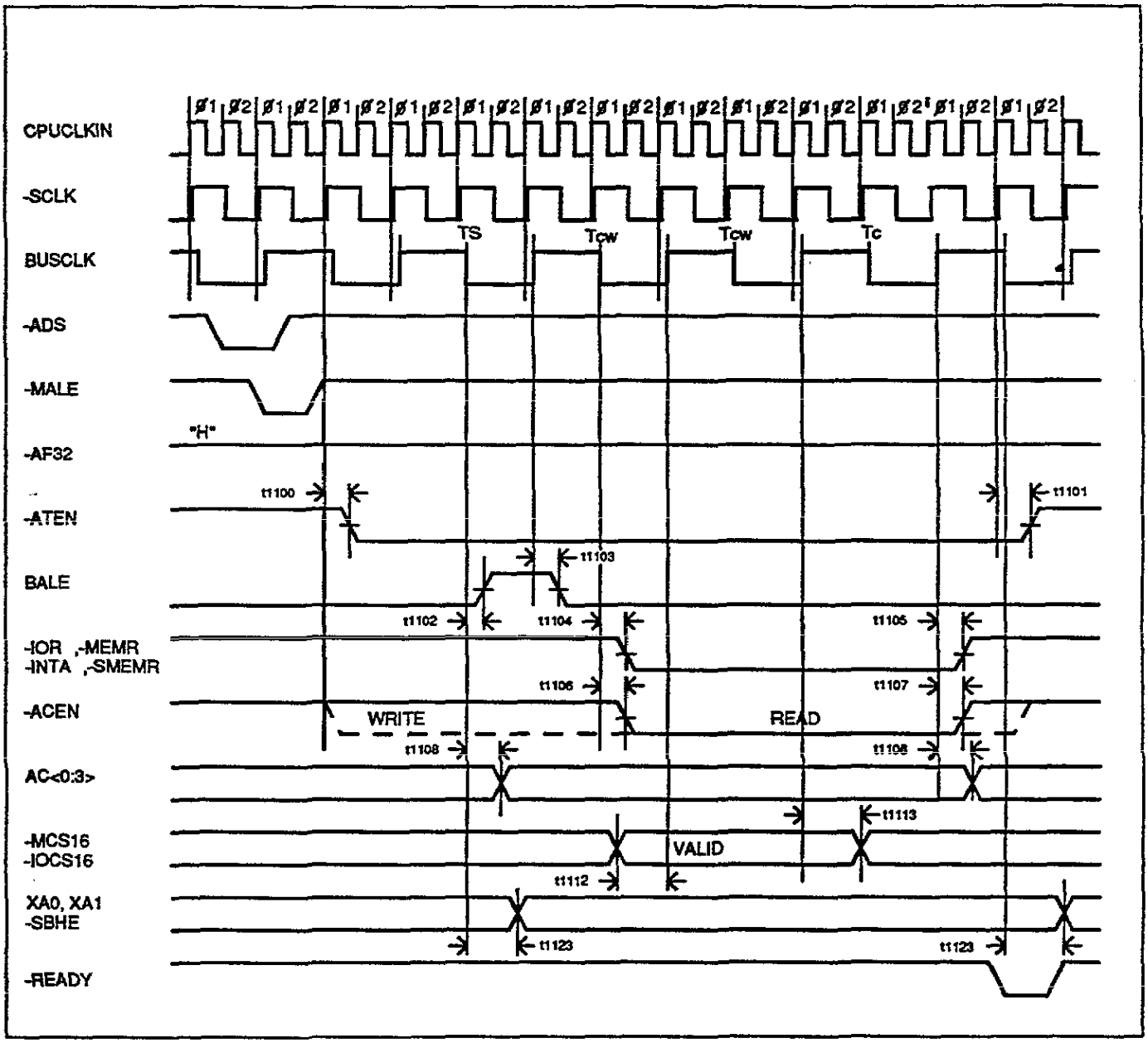
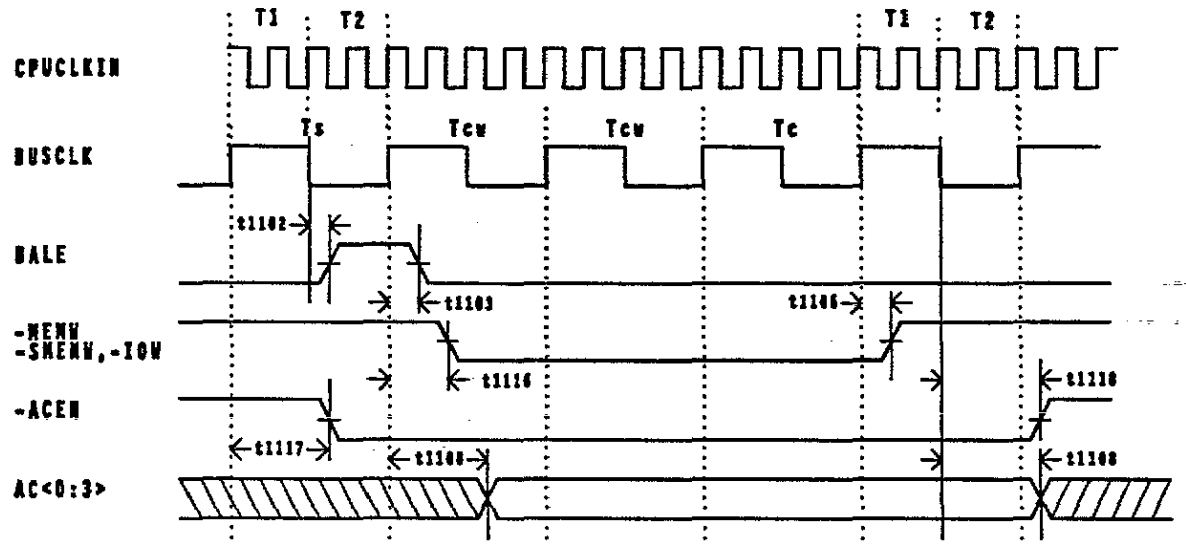
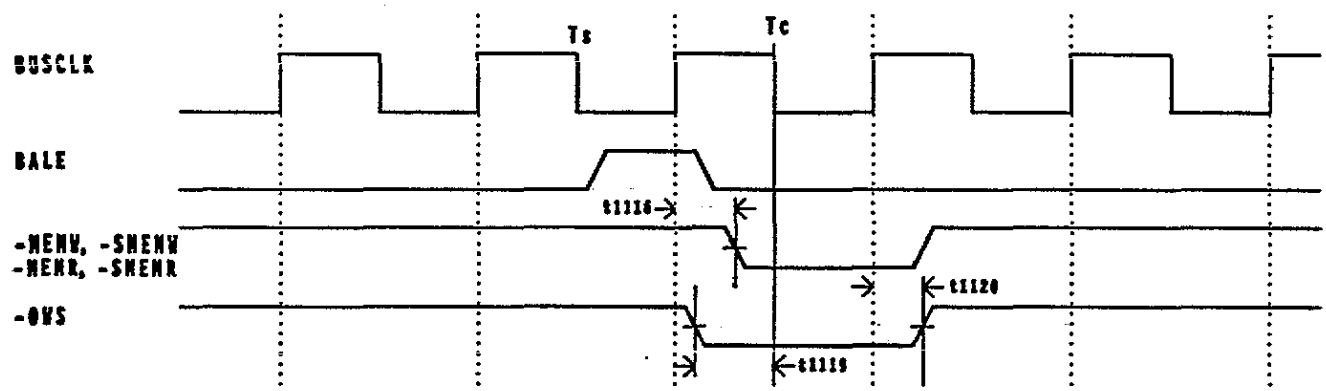


Figure 6-13. 82C351 AT Write Cycle



Two Wait States



One Wait State

Figure 6-14. *82C351 DMA Arbitration*

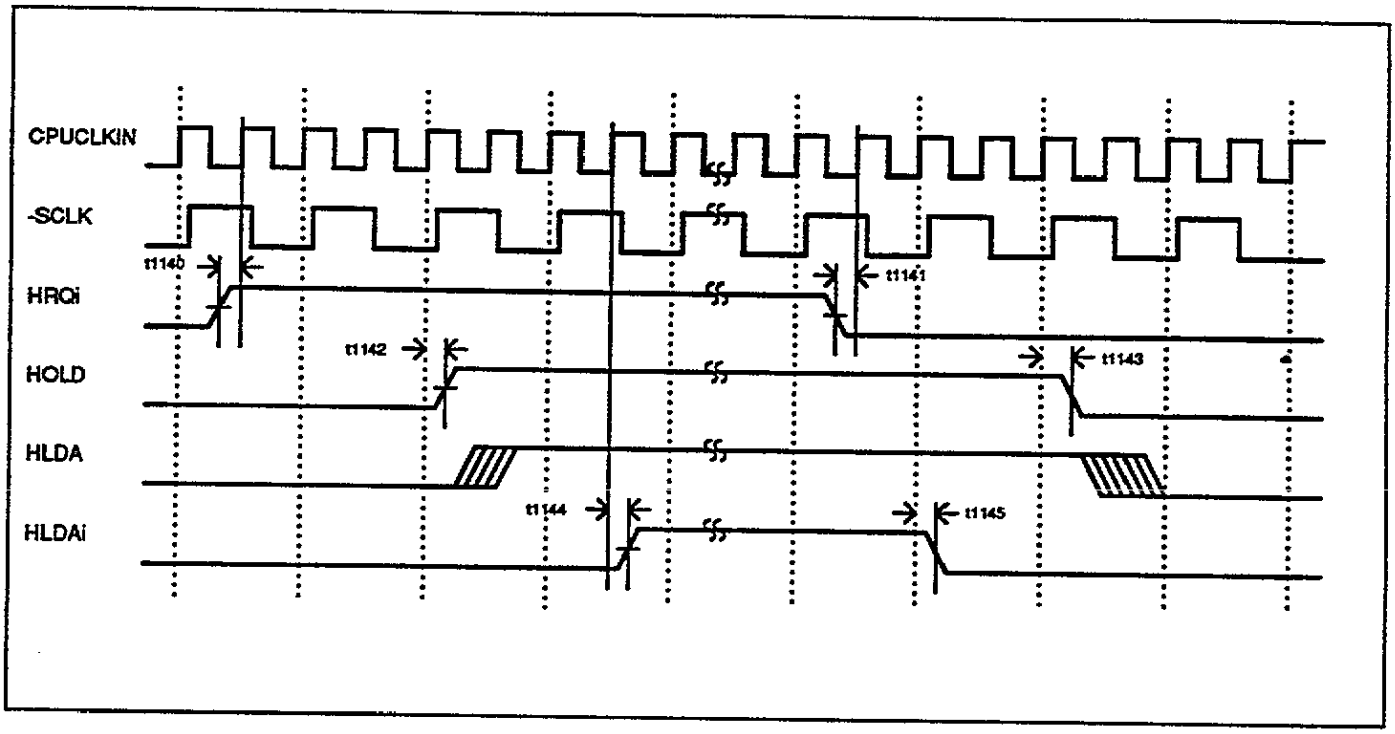


Figure 6-15. 82C351 Refresh Arbitration

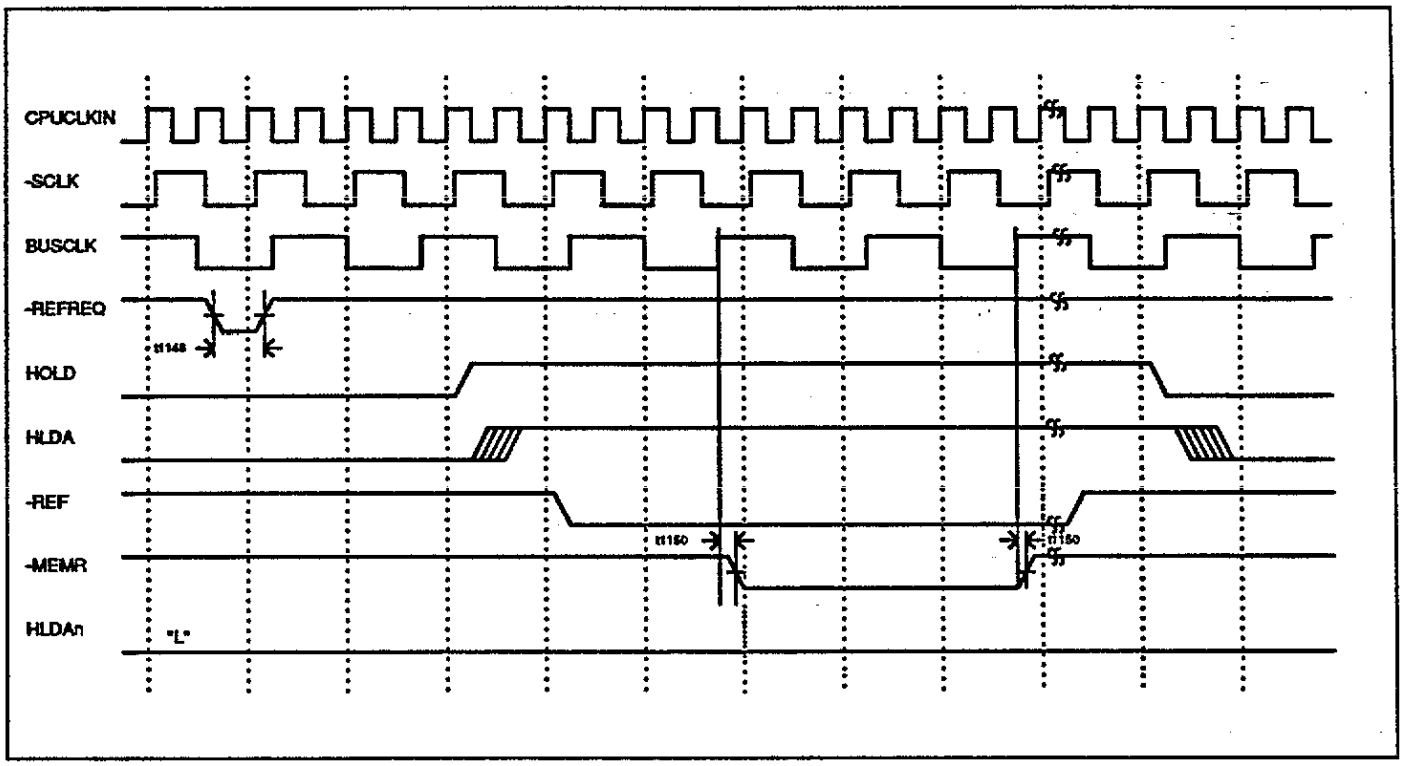


Figure 6-16. *82C351 Reset*

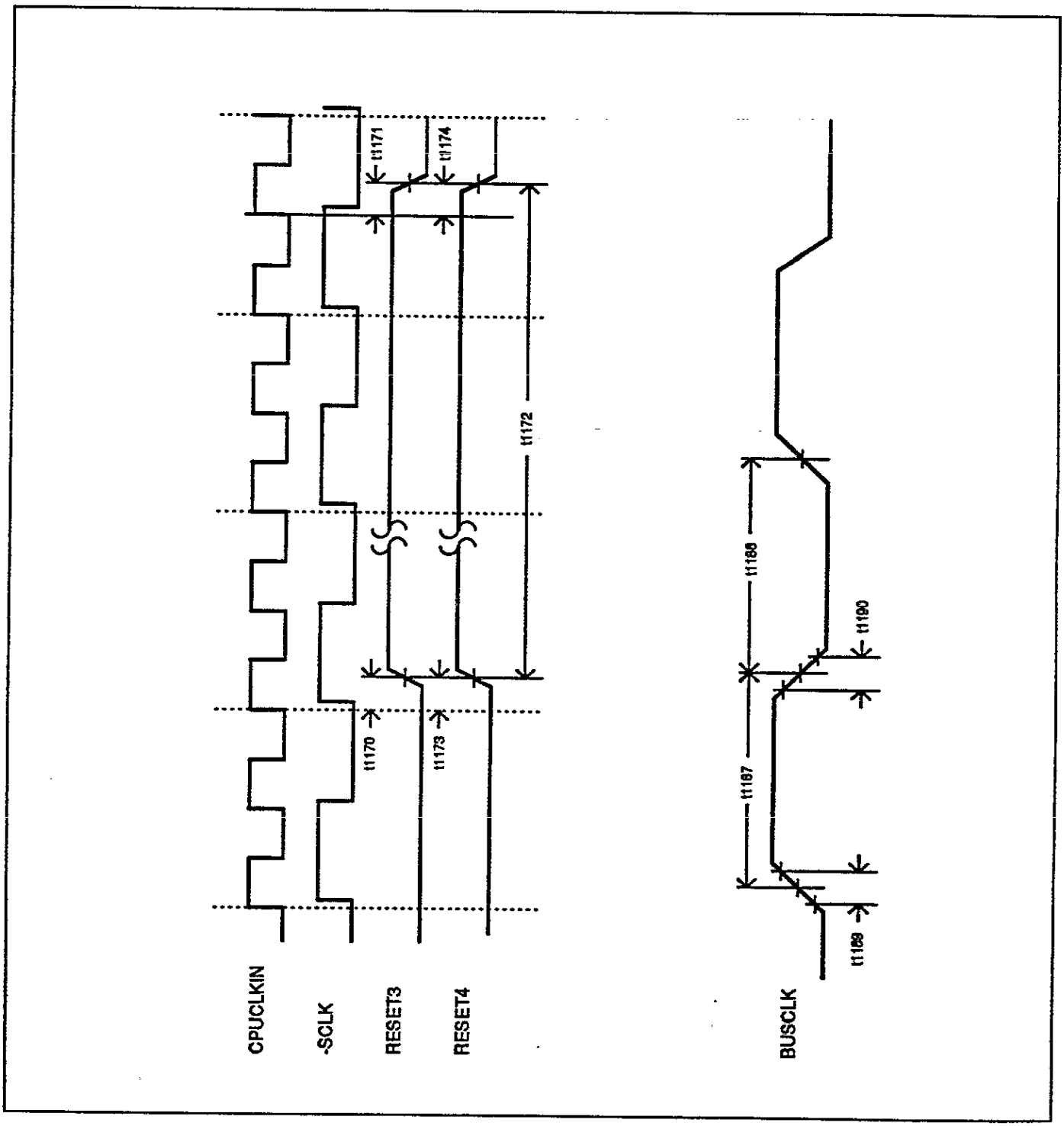


Figure 6-17. 82C351 Miscellaneous Timing

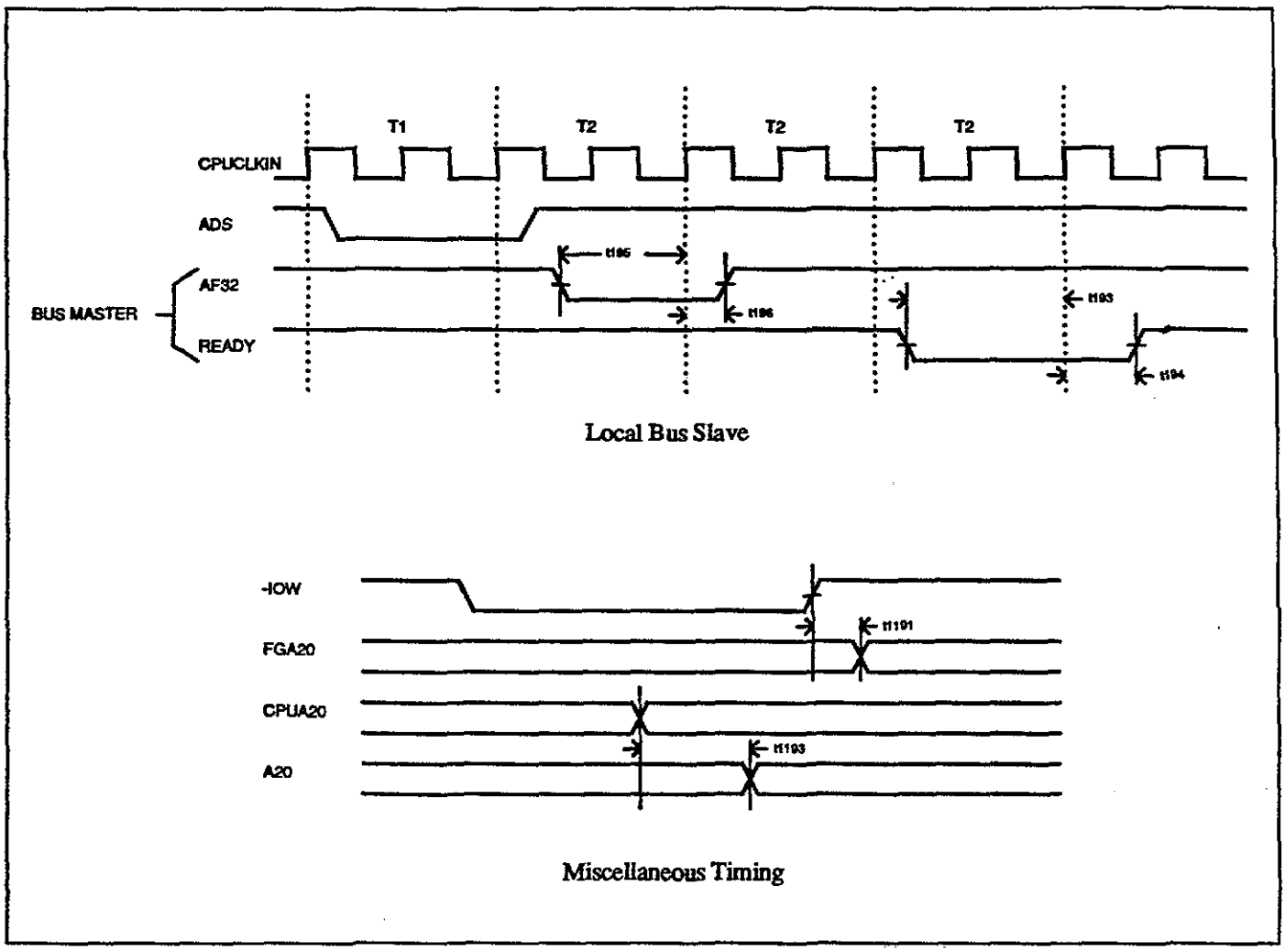
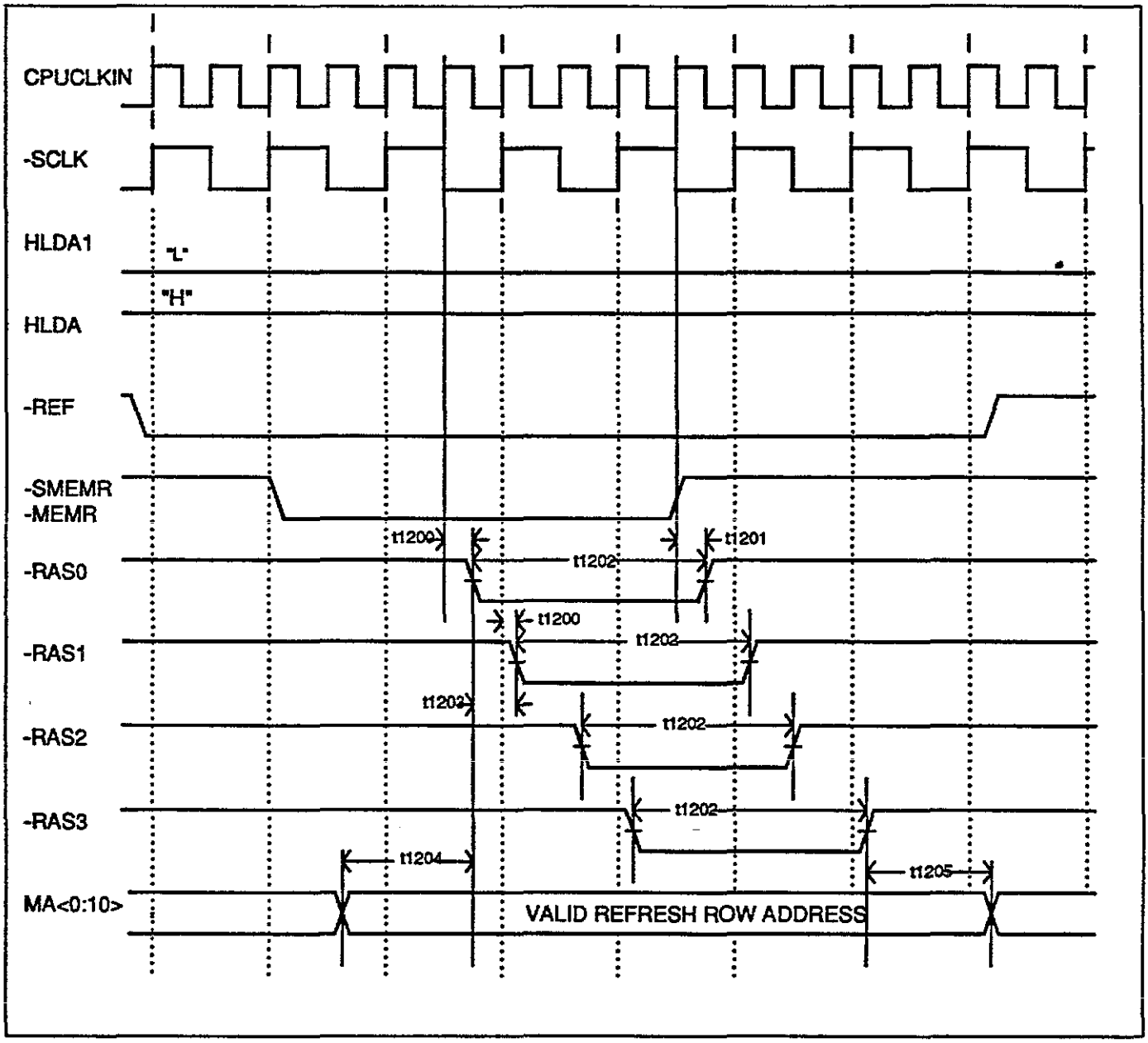
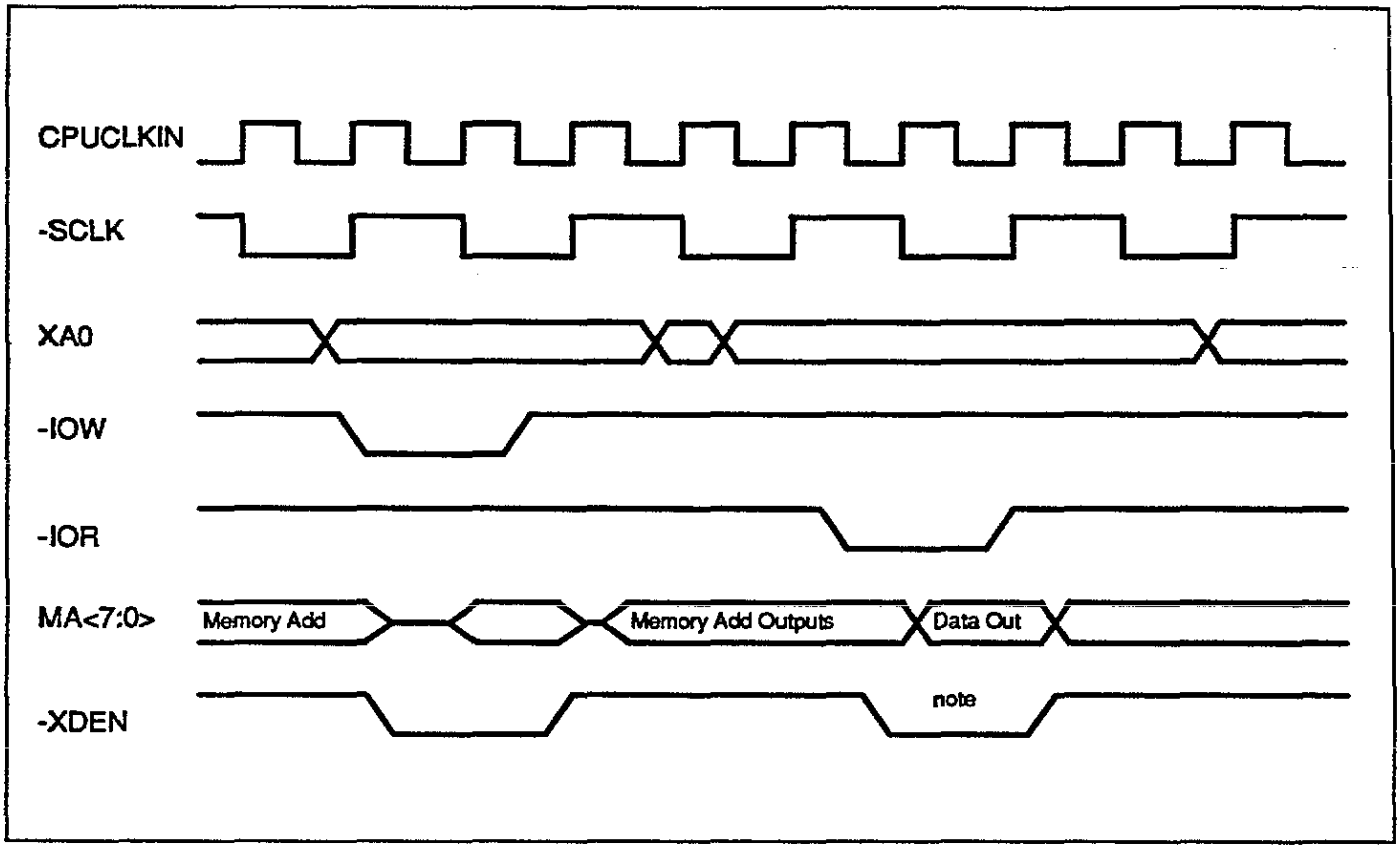


Figure 6-18. *Refresh Cycle*



Note t1202 is the -RAS pulse width during refresh and it is programmable through register 11-bits 1 and 2. IOCHRDY will not be generated during a classical refresh.

Figure 6-19. 82C351 Index Register Read/Write



Note No data is output and XDEN is inactive if the index setup by the previous I/O 22 Write does not point to a valid I/O 23 register of the 82C351.

This page is intentionally left blank.

Section 7

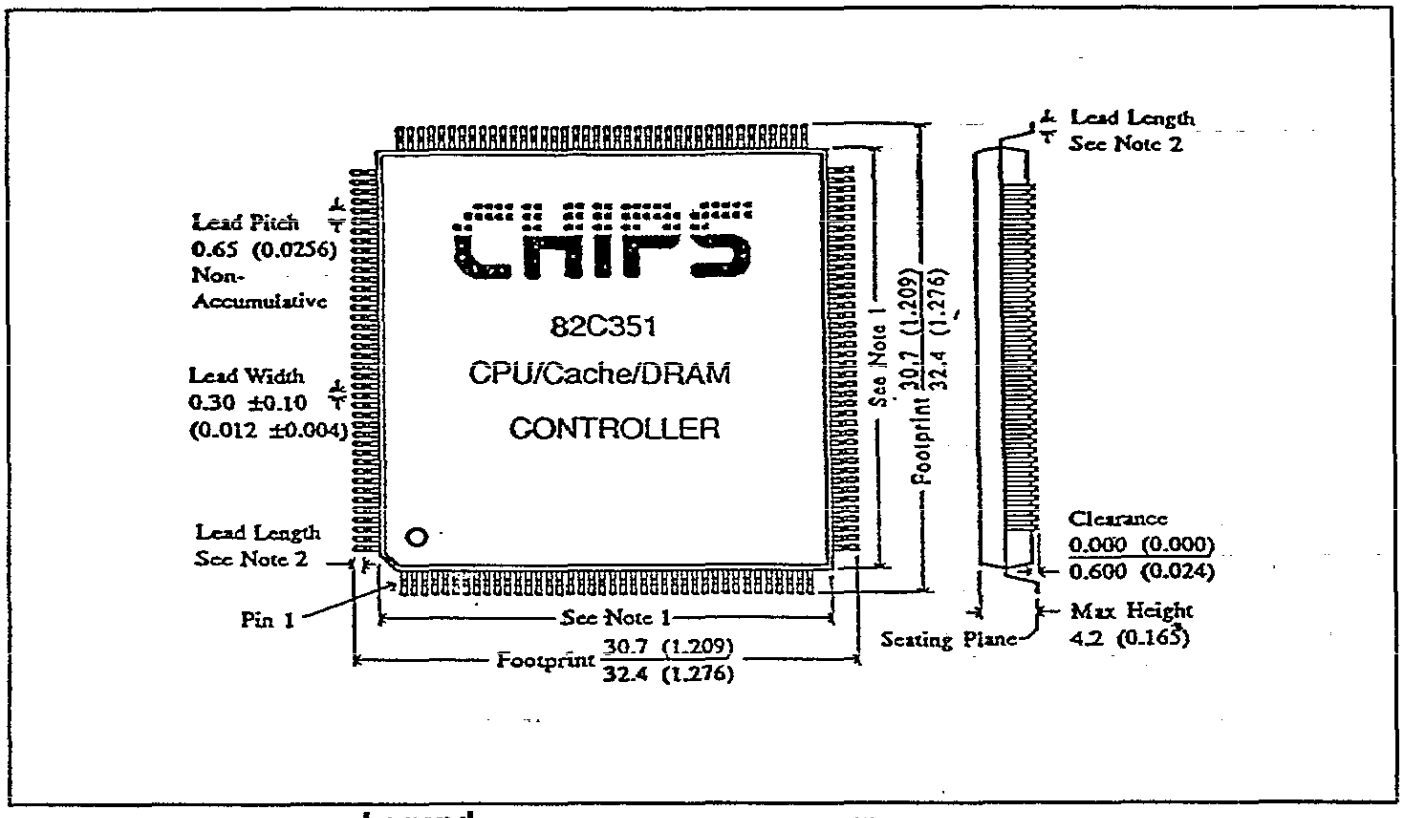
351 Physical Dimensions

Section 7

351 Physical Dimensions

7.1 82C351 CPU/Cache/DRAM/Controller

Figure 7-1. 160-Pin Plastic Flat Package



Legend minimum inches (minimum milimeters)
maximum inches (maximum milimeters)

Note 1 Package Body Size = 28 +0.2/-0.4 (1.102 + 0.008/-0.016) (Swire)
Package Body Size = 28 ±0.2 (1.102 ±0.008) (All Other Package Vendors)

Note 2 Lead Length = 0.6 ±0.3 (0.024 ±0.012) (Package Vendor = Seiko)
Lead Length = 0.7 ±0.2 (0.028 ±0.008) (Package Vendor = Yamaha)
Lead Length = 0.8 ±0.2 (0.031 ±0.008) (All Other Package Vendors)

This page is intentionally left blank.

VOLUME

82C355 Data Buffer

Section 1

355 Functional Description

Section 1

355 Functional Description

1.1 Features

The 82C355 provides all of the logic required to interface the Memory Data bus (MD<0:31>), to the Local System Data bus (D<0:31>). It also performs data alignment and size conversion, and contains parity detection and generation logic.

1.2 Functional Subsystem

Figure 1-1 shows the block diagram of the 82C355.

The 82C355 performs the following functions:

- Buffers data between the D and MD buses
- Generates and checks parity for the DRAM subsystem
- Latches the data for DRAM buffered writes
- Latches data from the AT bus during CPU AT bus reads
- Performs data steering for AT bus accesses
- Provides paths for SD and XD buses

1.3 Bus Interface

The 82C355 is the data buffer that interfaces the CPU data bus (D bus) to the memory data bus (MD bus). Figure 1-2 shows the various data paths. The MD bus interfaces to the SD Bus (AT bus) via bi-directional buffers. The XD bus, derived from the SD bus using additional bi-directional buffers, is used as a data path for on board peripherals such as the keyboard controller, etc. The signals HLDA1, -ATEN, AC<0:3>, SDIR0, and SDIR1 control the direction of the data paths. Refer to Table 1-1 for bus control definition.

Data on the D bus is latched inside the 82C355 by BWBUSY. The latches are open except for buffered writes. When BWBUSY is low the latches are open. When BWBUSY is high, the data is latched.

Figure 1-1. 82C355 Data Buffer Block Diagram

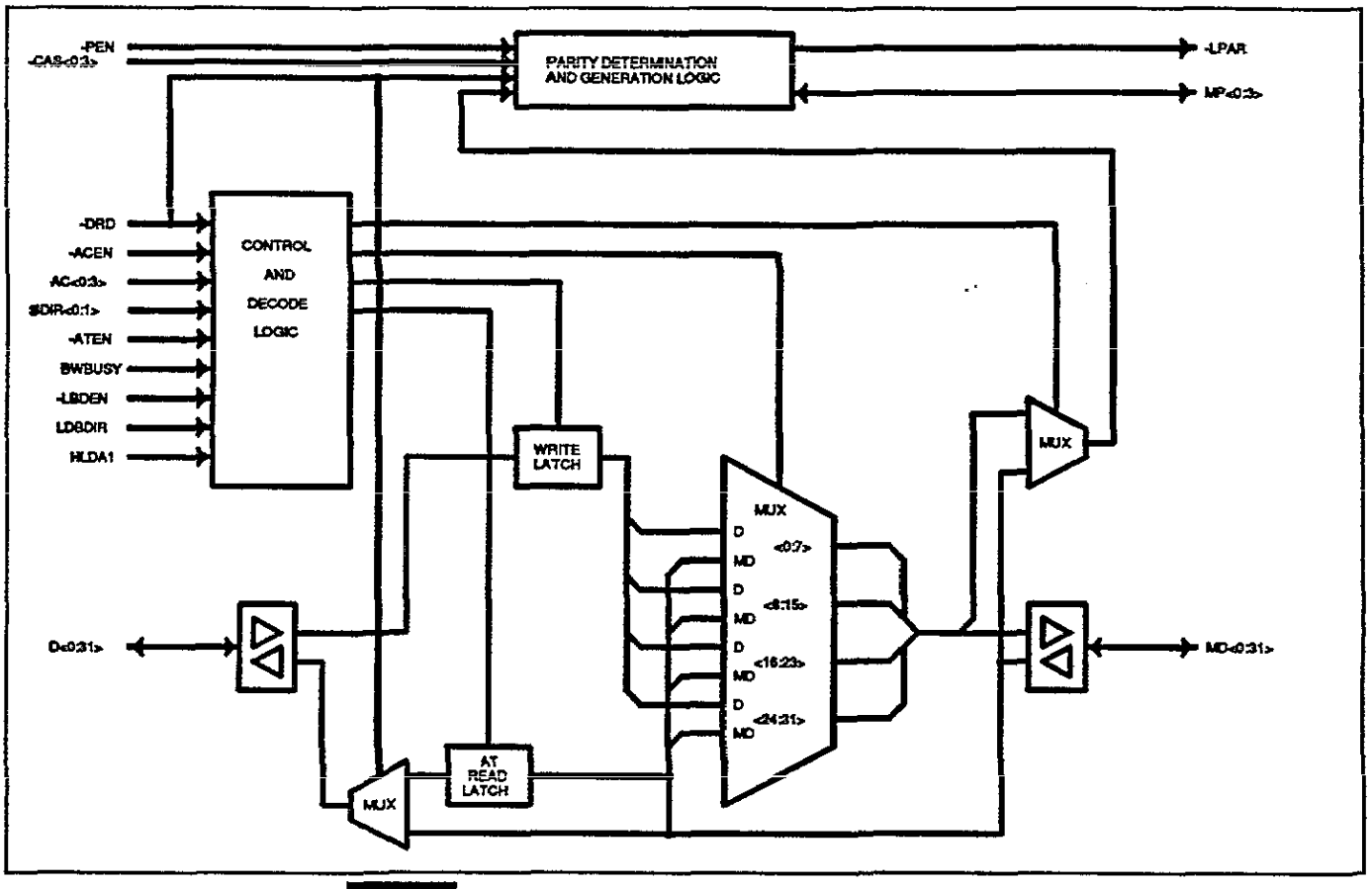


Table 1-1. Bus Control Definition

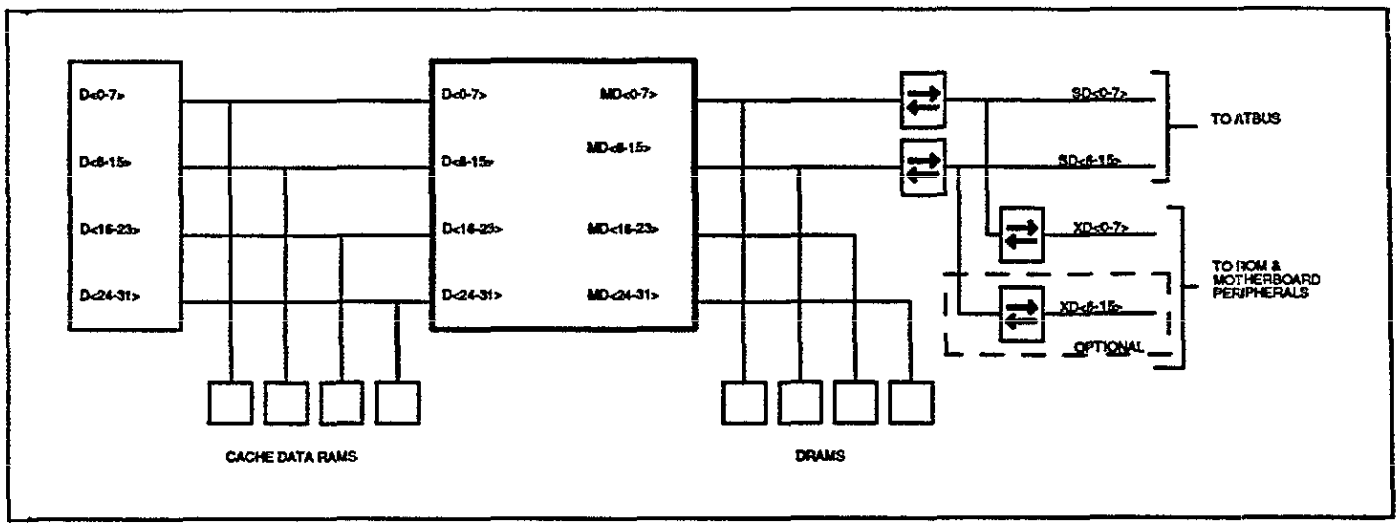
HLDA	-ATEN	SDIR0	SDIR1	-DRD	Cycle Type	MD Bus 355	SD Bus LSB	BUF MSB
0	0	0	0	X	CPU AT READ	I	I	I
0	0	0	1	X	Not a valid cycle	—	—	—
0	0	1	0	X	Not a valid cycle	—	—	—
0	0	1	1	X	CPU AT WRITE	O	O	O
1	1	1	1	0	MASTER/DMA 16-bit, 8-bit LOC MEM RD	—	I	I
1	1	0	1	0	Not a valid cycle	—	—	—
1	1	1	0	0	MASTER/DMA 16-bit LOC MEM RD	—	O	I
1	1	0	0	1	MASTER/DMA 16-bit AT MEM RD	—	O	O
1	1	0	0	1	MASTER/DMA 16-bit LOC MEM WR	—	I	I
1	1	0	1	1	MASTER/DMA 8-bit LOC MEM WR	—	I	O
1	1	1	0	1	Not a valid cycle	—	—	—
1	1	0	0	1	MASTER/DMA 16-bit AT MEM WR	—	O	O

Note The "—" indicates "don't care."

Section 2

355 Pin Descriptions

Figure 1-2. PEAK/DM Data Path



1.4 Data Conversion

The 82C355 provides the proper data conversions for CPU initiated accesses to the AT bus. The action codes AC<0:3> are used to control the flow of data between the 32-bit memory (MD) data bus and the 32-bit local (D) data bus. The action codes are generated by the 82C351 according to the signals -MCS16 and -IOCS16. The action codes are used to control the CPU accesses to the AT bus and are qualified by -ACEN.

Data Steering for AT Bus Conversions

The 82C355 handles the data bus swapping required when a master and slave use different data sizes. The action codes shown in Table 1-2 and Figure 1-3 identify the data steering for all possible cycles.

In Figure 1-3, A, B, C and D are the CPU data bytes (where A is the LSB and D is the MSB), E and F are the data bytes on the AT bus (where E is the LSB and F is the MSB); G, H, I, and J are the data bytes on the Local RAM bus (where G is the LSB and J is the MSB). For example, when the CPU is performing local DRAM cycles, the bus conversions are not necessary as both the CPU and the local memory accesses are 32-bits wide. For CPU initiated AT cycles, bus conversions are required because the AT bus is 16-bits wide. The 82C351 performs two 16-bit cycles and the 82C355 latches the data, and presents 32-bit data to the CPU.

Table 1-2. Action Code Definition

AC<0:3>	From	To
0	AB	EF
1	B	E/F
2	CD	EF
3	D	E/F
5	E	G
6	E	B/H
7	E	C/I
8	E	D/J
9	EF	AB/GH
A	EF	CD/IJ
B	F	E
D	E	F

1.5 Parity

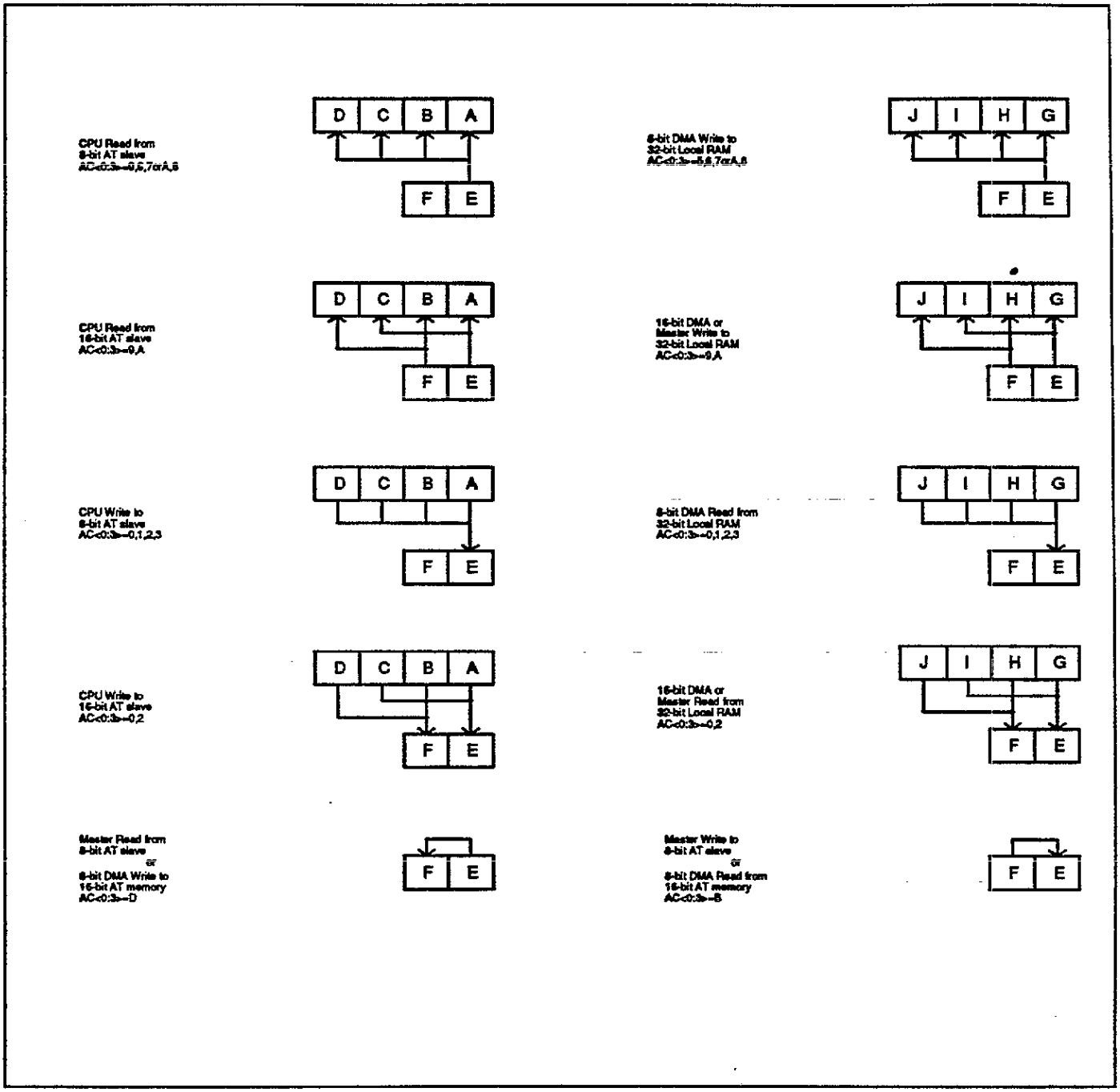
The 82C355 generates and checks parity for the DRAMs. There is one parity bit for each byte. MP0 is the parity for MD<0:7>, MP1 for MD<8:15>, MP2 for MD<16:23>, and MP3 for MD<24:31>.

During a write cycle (when -DRD is high) the parity is generated by the 82C355 and the parity bits are driven on MP<0:3>. The generation of the parity is done on a byte by byte basis.

When -DRD is low, the buffers driving MP<0:3> are tristated, disabling the parity generator and enabling the parity checker. The output of the parity checker, along with the parity bits read into the 82C355 are internally compared on a byte by byte basis using the CAS<0:3> signals. If there is a parity error, the 82C355 activates -LPAR to the 82C351. The parity enable signal -PEN should be low to enable parity checking.

The parity error signals from each byte are NORed together, then ORed with -PEN and sent out on the -LPAR signal. When an error is detected, the -LPAR signal will remain low until the next DRAM read from the offending byte with proper parity.

Figure 1-3. 82C355 Bus Conversion



Note A, B, C, and D are 4 CPU bytes with A being the least significant byte.
E and F are 2 AT bus bytes with E being the least significant byte.
G, H, I, and J are 4 local RAM bytes with G being the least significant byte.

This page is intentionally left blank.

Section 2

355 Pin Descriptions

2.1 Pin Assignments

Table 2-1. Pin Definitions

Pin #	Symbol	Type	Signal Description
D0-D3 D4-D8 D9-D10 D11-D13 D14-D17 D18 D19-D23 D24-D25 D26-D28 D29-D31	83-86 92-96 98-99 101-103 105-108 110 112-116 118-119 3-5 7-9	Input/ Output	The Data bus signals D0 through D31 are bi-directional lines used to connect the 82C355 to the CPU bus. They are driven by the 82C355 during DRAM reads, AT reads, DMA writes, and Master write cycles. 8mA drive capability.
MD0-MD4 MD5-MD9 MD10-MD11 MD12-MD14 MD15-MD19 MD20-MD24 MD25-MD26 MD27-MD31	24-28 32-36 38-39 41-43 45-48, 50 52-56 58-59 63-67	Input/ Output	Memory Data Bus. The memory banks reside on this bus. The SD bus (AT bus) is derived from the MD bus. 8mA drive capability.
MP<0:3>	77-80	Input/ Output	Memory Parity bits. MP0 through MP3 are inputs during read cycles and outputs during write cycles. During read cycles these bits are compared with internally generated parity bits to determine whether a parity error has occurred.
-CAS<0:3>	69-72	Input	These are the -CAS byte enable coded lines from the 82C351, and are used to check Parity. Parity is checked on the rising edge of -CAS for DRAM read cycles. Only the bytes for which -CAS is active have their parity checked.
-PEN	74	Input	Parity enable (input from the 82C351). When low, parity is checked for DRAM read cycles. When high, no parity checking is performed.
-LPAR	76	Output	Local Parity error. This is the OR of the 4 parity error signals, ANDed with PEN. This signal will go inactive on the next memory read with proper parity for the byte which caused the error.
HLDA1	19	Input	Hold Acknowledge to indicate a DMA cycle for control logic to control the buffer directions. When high a DMA channel or AT bus master has control of the bus. Used for data bus steering.
-ACEN	10	Input	Action Code enable is an active low input from the 82C351 used to validate action code signals AC<0:3>. This signal follows the AT bus IOR, MEMR, or INTA signals for CPU AT bus reads. For CPU AT bus writes -ACEN is driven before the IOW or MEMW signals to provide data setup time. During DMA and bus master cycles, it is always low.
AC<0:3>	11-14	Input	Action codes are four bit encoded commands from the 82C351. They are used for data bus sizing and byte assembly operation. AC<0:3> are qualified by the -ACEN signal. Table 3-1, shown in *Section 1: 355 Functional Description*, contains the action code definition.

Table 2-1. Pin Definitions (continued)

Pin #	Symbol	Type	Signal Description
-DRD	82	Input	DRAM READ is an active low signal. When active it enables data from the MD bus to the D bus. This signal determines the data direction. 0=DRAM read cycle, 1=DRAM write cycle. It is used by the parity logic. Parity is checked when -CAS goes high only if DRD is low.
-ATEN	88	Input	AT BUS ENABLE is an active low signal. When active, it indicates that the current cycle is to the AT bus.
SDIR0	16	Input	Bus direction for SD<0:7>. Used for bus steering of the MD<0:7> lines. When high, SDIR0 directs SD<0:7> away from the MD bus. When low, SDIR0 points SD<0:7> towards the MD Bus.
SDIR1	18	Input	Bus direction for SD<8:15>. Used for bus steering of the MD<8:15> lines.
BWBUSY	20	Input	Buffered Write Busy. This is an active high signal used to latch the data during write cycles. The rising edge latches the CPU write data.
LDBEN	21	Input	Local Data Bus Enable is an active low signal enabling the data to flow between the MD and D bus and vice-versa. This is active for CPU DRAM, AT, and I/O cycles.
LDBDIR	22	Input	Local Data Bus direction. When low, data flows from the MD bus to the D buses. When high, data path flows from the D to the MD bus.
VCC	17, 29, 40, 49, 60, 73, 87, 100, 109, 120		+5 ±5% Supply Voltage.
GND	6, 15, 23, 31, 37, 44, 51, 57, 62, 68, 75, 81, 91, 97, 104, 111, 117		Ground.
NC	1, 2, 30, 61, 89, 90		No connect - These pins should be left floating.

2.2 Numerical Listing of Pin Assignments

Table 2-2. Numerical Pin Definitions

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	NC	31	Vss	61	NC	91	Vss
2	NC	32	MD5	62	Vss	92	D4
3	D26	33	MD6	63	MD27	93	D5
4	D27	34	MD7	64	MD28	94	D6
5	D28	35	MD8	65	MD29	95	D7
6	Vss	36	MD9	66	MD30	96	D8
7	D29	37	Vss	67	MD31	97	Vss
8	D30	38	MD10	68	Vss	98	D9
9	D31	39	MD11	69	-CAS0	99	D10
10	-ACEN	40	Vcc	70	-CAS1	100	Vcc
11	AC0	41	MD12	71	-CAS2	101	D11
12	AC1	42	MD12	72	-CAS3	102	D12
13	AC2	43	MD14	73	-CAS3	103	D13
14	AC3	44	Vss	74	-PEN	104	Vss
15	Vss	45	MD15	75	Vss	105	D14
16	SDIR0	46	MD16	76	-LPAR	106	D15
17	Vcc	47	MD17	77	MP0	107	D16
18	SDIR1	48	MD18	78	MP1	108	D17
19	HLDA1	49	Vcc	79	MP2	109	Vcc
20	BWBUSY	50	MD19	80	MP3	110	D18
21	-LDBEN	51	Vss	81	Vss	111	Vss
22	LDBDIR	52	MD20	82	-DRD	112	D19
23	Vss	53	MD21	83	D0	113	D20
24	MD0	54	MD22	84	D1	114	D21
25	MD1	55	MD23	85	D2	115	D22
26	MD2	56	MD24	86	D3	116	D23
27	MD3	57	Vss	87	Vcc	117	Vss
28	MD4	58	MD25	88	-ATEN	118	D24
29	Vcc	59	MD26	89	NC	119	D25
30	NC	60	Vcc	90	NC	120	Vcc

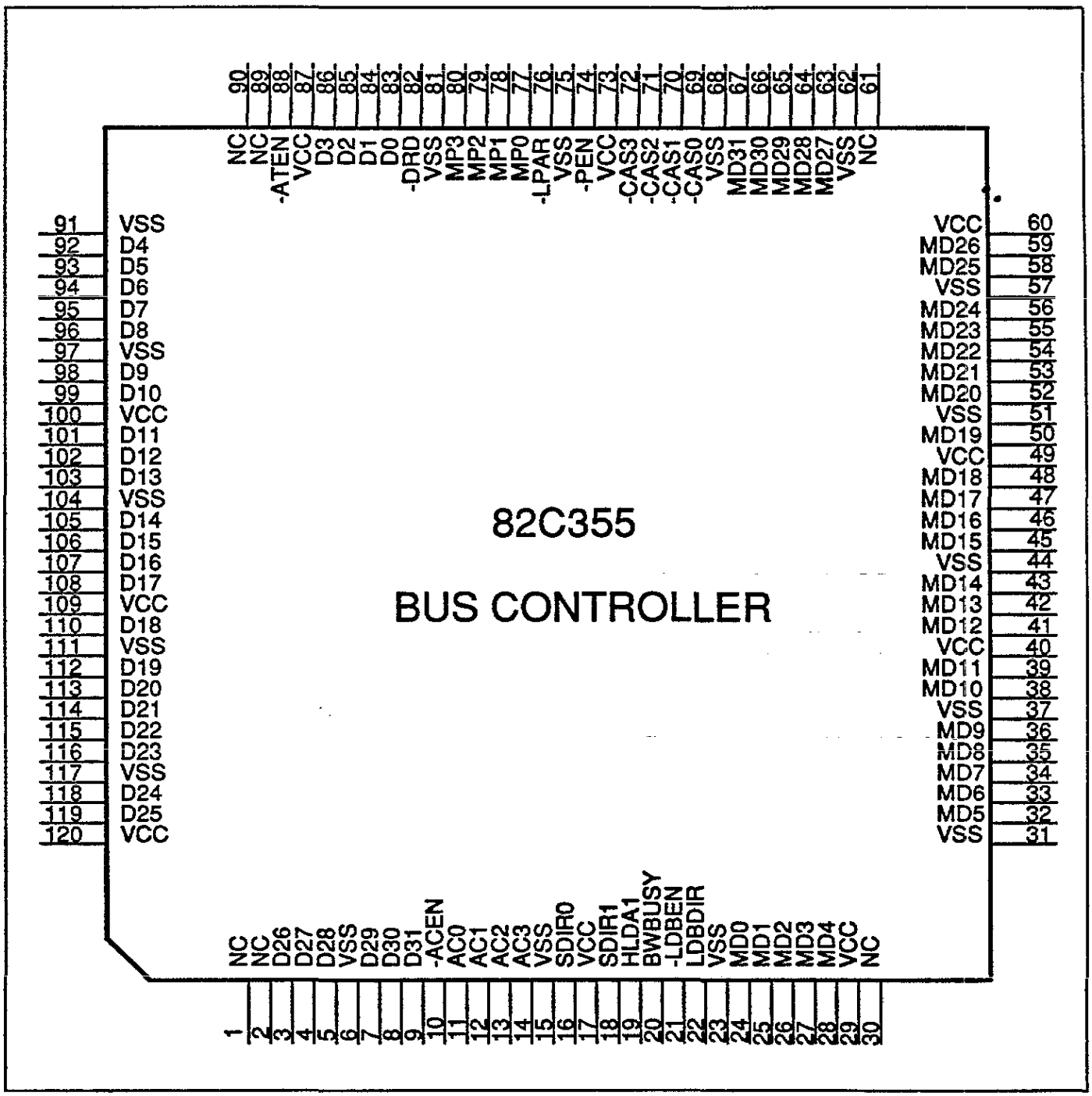
2.4 Alphabetical Listing of Pin Assignments

Table 2-3. Alphabetical Pin Definitions

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
-ACEN	10	D18	110	MD12	41	-PEN	74
AC0	11	D19	112	MD13	42	SDIR0	16
AC1	12	D20	113	MD14	43	SDIR1	18
AC2	13	D21	114	MD15	45	Vcc	17
AC3	14	D22	115	MD16	46	Vcc	29
-ATEN	88	D23	116	MD17	47	Vcc	40
BWBUSY	20	D24	118	MD18	48	Vcc	49
-CAS0	69	D25	119	MD19	50	Vcc	60
-CAS1	70	D26	3	MD20	52	Vcc	73
-CAS2	71	D27	4	MD21	53	Vcc	87
-CAS3	72	D28	5	MD22	54	Vcc	100
-DRD	82	D29	7	MD23	55	Vcc	109
D0	83	D30	8	MD24	56	Vcc	120
D1	84	D31	9	MD25	58	Vss	6
D2	85	HLDA1	19	MD26	59	Vss	15
D3	86	LDBDIR	22	MD27	63	Vss	23
D4	92	-LDBEN	21	MD28	64	Vss	31
D5	93	-LPAR	76	MD29	65	Vss	37
D6	94	MD0	24	MD30	66	Vss	44
D7	95	MD1	25	MD31	67	Vss	51
D8	96	MD2	26	MP0	77	Vss	57
D9	98	MD3	27	MP1	78	Vss	62
D10	99	MD4	28	MP2	79	Vss	68
D11	101	MD5	32	MP3	80	Vss	75
D12	102	MD6	33	NC	1	Vss	81
D13	103	MD7	34	NC	2	Vss	91
D14	105	MD8	35	NC	30	Vss	97
D15	106	MD9	36	NC	61	Vss	104
D16	107	MD10	38	NC	89	Vss	111
D17	108	MD11	39	NC	90	Vss	117

2.5 Pin Diagram

Figure 2-1. 82C355 Bus Controller Pin Diagram



This page is intentionally left blank.

Section 3

355 Physical Characteristics

Section 3

355 Physical Characteristics

3.1 Absolute Maximum Ratings

	Symbol	Min.	Max.	Units
Supply Voltage	Vcc		7.0	V
Input Voltage	V_I	-.5	5.5	V
Output Voltage	V_O	-.5	5.5	V
Operating Temperature	Top	-25	85	C
Storage Temperature	Tstg	-40	125	C

Note Exposure to absolute maximum ratings conditions for extended periods may affect device reliability. Exceeding the absolute maximum ratings can cause permanent damage to the device.

3.2 Operating Conditions

	Symbol	Min.	Max.	Units
Supply Voltage	Vcc	4.75	5.25	V
Ambient Temperature	T_A		70	C

This page is intentionally left blank.

Section 4

355 DC/AC Characteristics

Section 4

355 DC/AC Characteristics

4.1 DC Characteristics

	Symbol	Min.	Max.	Units
Input low voltage	V_{IL}			
TTL level		-0.3	.8	V
Input high voltage	V_{IH}			
TTL level		2.0	+0.3	V
Output low voltage	V_{OL}		.45	V
Output high voltage	V_{OH}	2.4		
Input LOW current @ Vo = Vss	I_{IL}	-10	+10	µA
Input HIGH current @ Vo = Vdd	I_{IH}	-10	10	µA
Output current low	I_{OL}			
(all pins except -LPAR)		-8	+8	µA
(-LPAR)	I_{OL}	-4	+4	µA
Output current high	I_{OH}			
(All pins except -LPAR)		-8	+8	µA
(-LPAR)	I_{OH}	-4	+4	µA
3-State output OFF current LOW	I_{OZL}	-10	+10	µA
3-State output OFF current HIGH	I_{OZH}	-10	+10	µA
Power supply current @25 MHz	I_{CC}		80	mA
Input capacitance	C_{IN}		10	pF
Output or I/O capacitance	C_{OUT}		10	pF

4.2 AC Characteristics

All timing parameters are specified under capacitive load of 50 pf and temperature of 70 degree C. All the units discussed in the following timing tables are in nanoseconds, unless otherwise specified. Also, the AC specifications mentioned in this document are subject to change.

Table 4-1. *Parity Timing*

	Parity Timing	25 MHz	
		Min.	Max.
t519	-LPAR inactive delay from -PEN inactive		20
t521	-LPAR active delay from -PEN active	6	20
t522	MP<0:3> setup to -CAS inactive	5	
t523	MP<0:3> valid from MD<0:31>		28
t524	MP<0:3> hold from -CAS inactive	5	

Table 4-2. *Data Paths*

	Data Paths	25 MHz	
		Min.	Max.
t555	MD bus setup to -CAS<0:3> inactive	25	
t556	MD bus hold to -CAS<0:3> inactive	10	
t570	D bus to MD bus		50
t571	-LBDEN low to DATA valid		39
t572	MD bus to D bus		25

Table 4-3. *Parity Timing*

	Parity Timing	33 MHz	
		Min.	Max.
t519	-LPAR inactive delay from -PEN inactive		18
t521	-LPAR active delay from -PEN active	6	18
t522	MP<0:3> setup to -CAS inactive	5	
t523	MP<0:3> valid from MD<0:31>		25
t524	MP<0:3> hold from -CAS inactive	4	

Table 4-4. *Data Paths*

	Data Paths	33 MHz	
		Min.	Max.
t555	MD bus setup to -CAS<0:3> inactive	25	
t556	MD bus hold to -CAS<0:3> inactive	10	
t570	D bus to MD bus		50
t571	-LBDEN low to DATA valid		39
t572	MD bus to D bus		25

Section 5

355 Timing Diagrams

Section 5
355 Timing Diagrams

Figure 5-1. 82C355 Write and -LPAR Parity

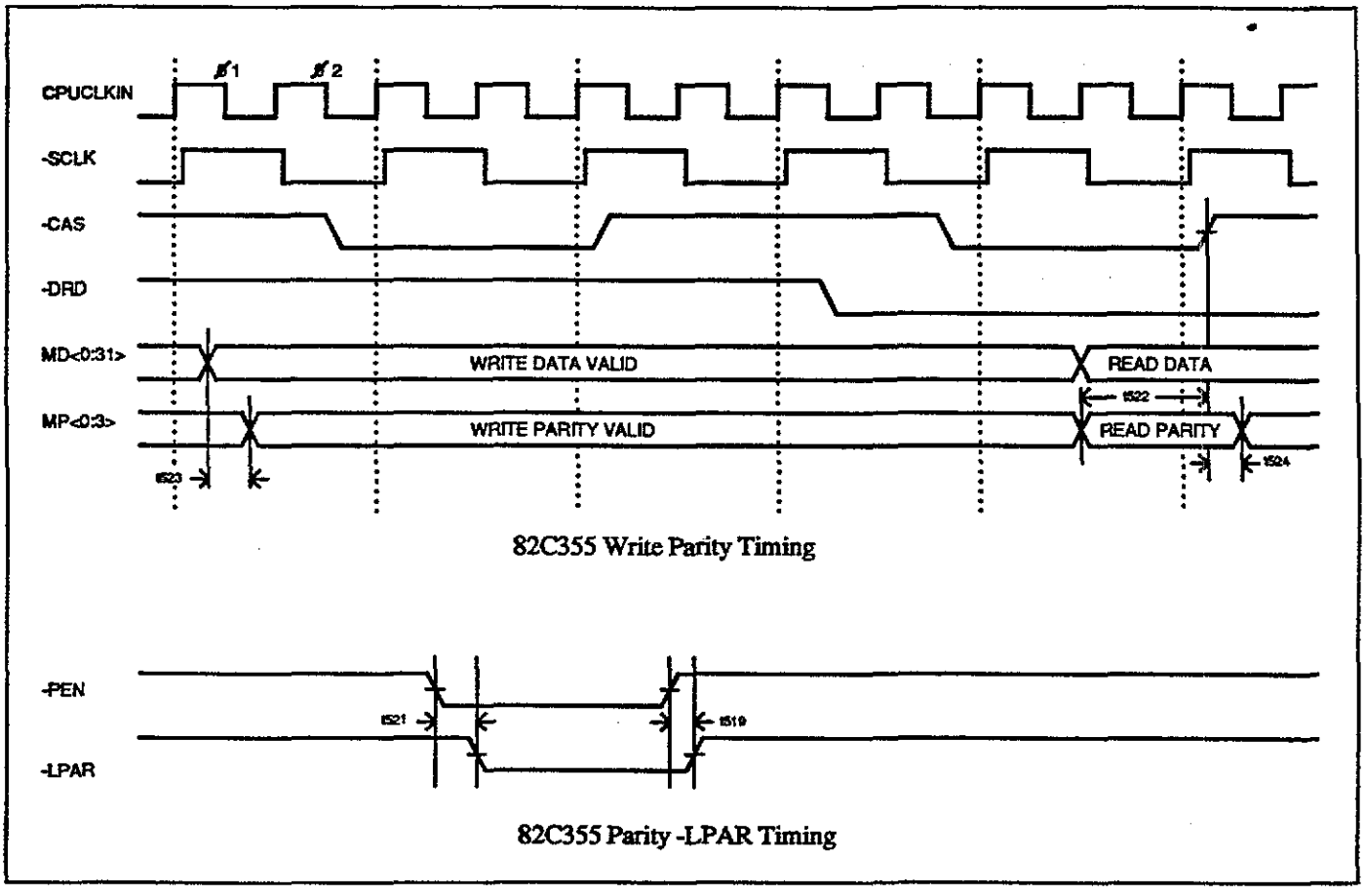


Figure 5-2. *82C355 MD to D Bus Delay*

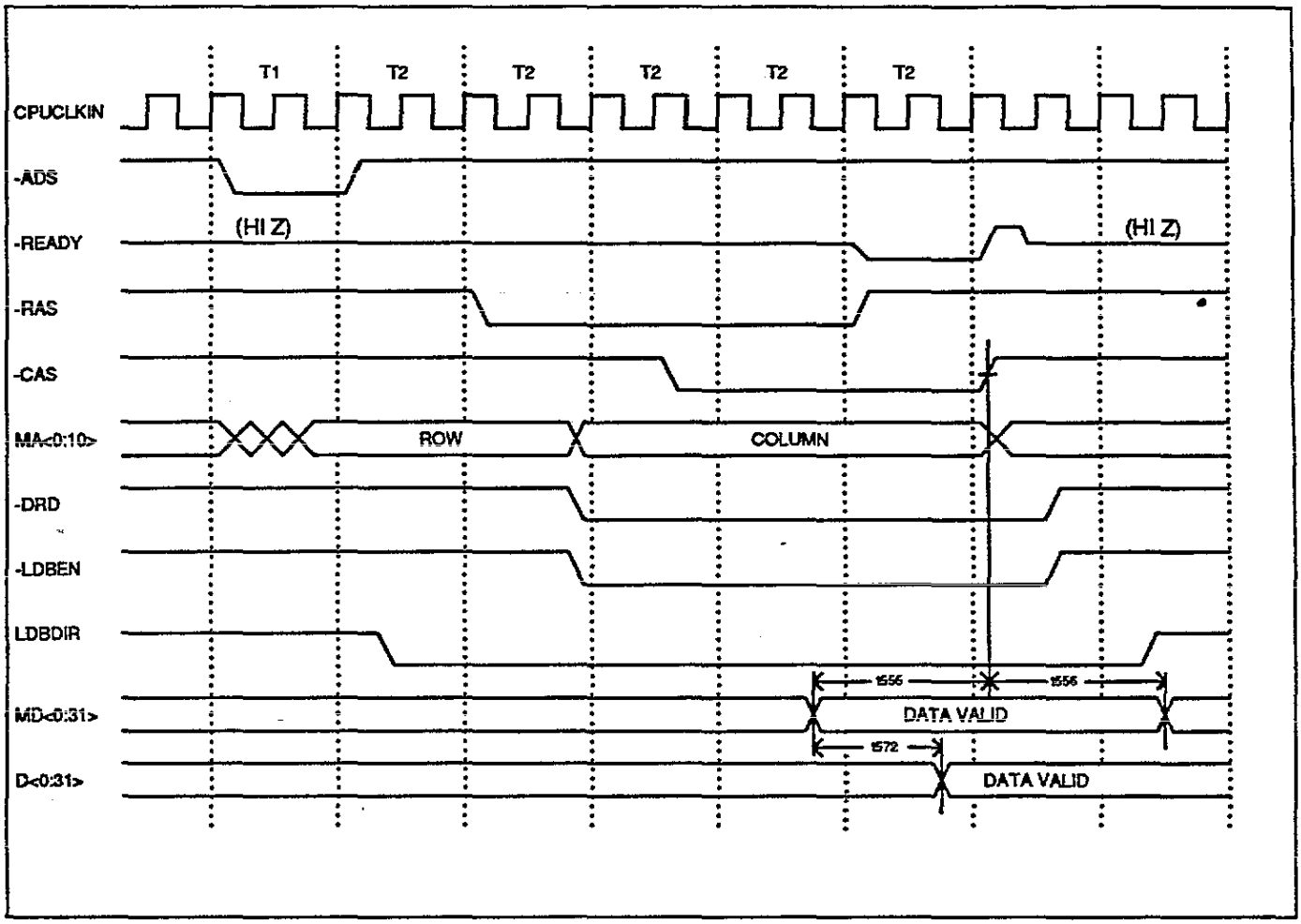
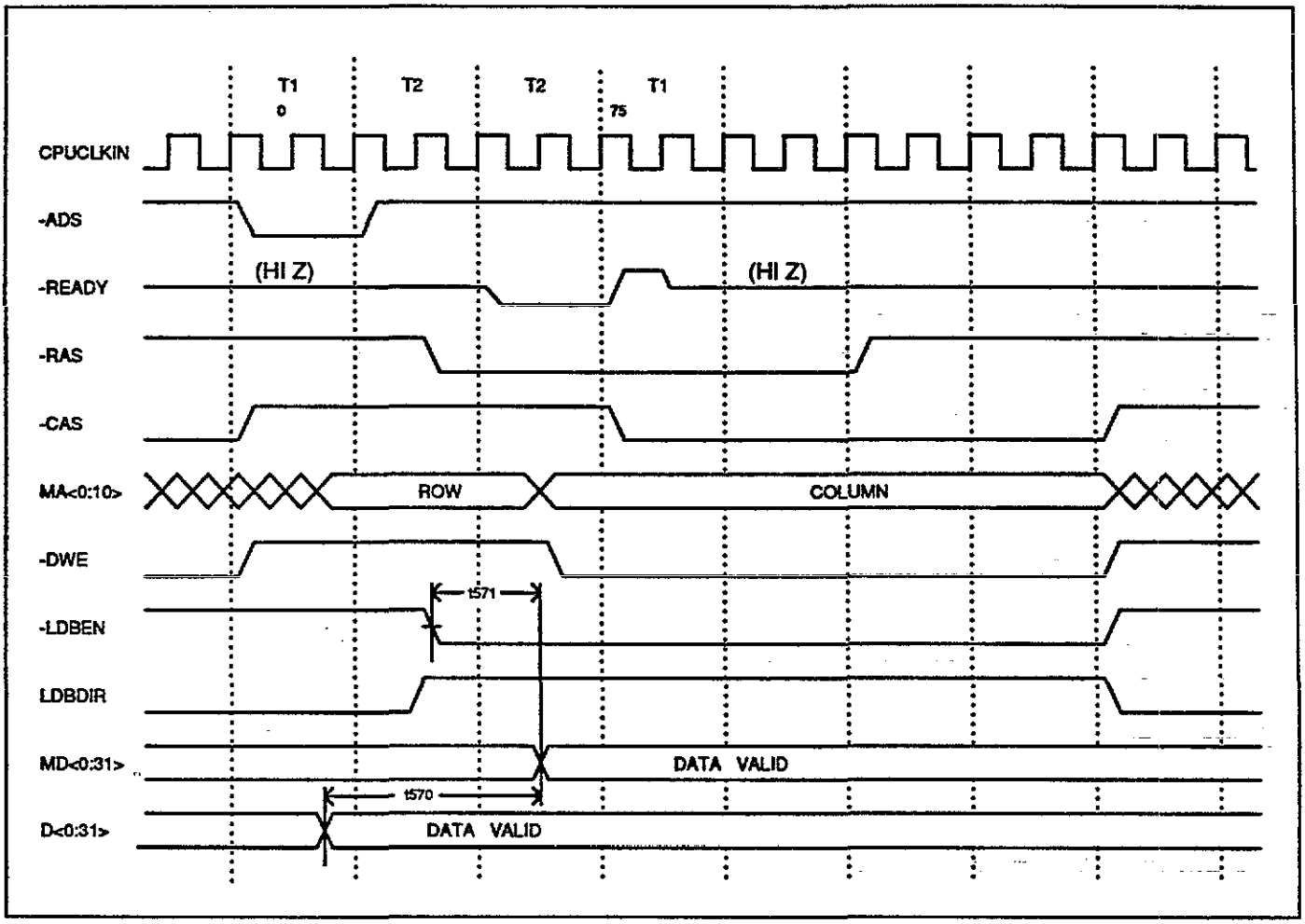


Figure 5-3. 82C355 D to MD Bus Delay and LBDEN Low to Data Valid Out



This page is intentionally left blank.

Section 6

355 Physical Dimensions

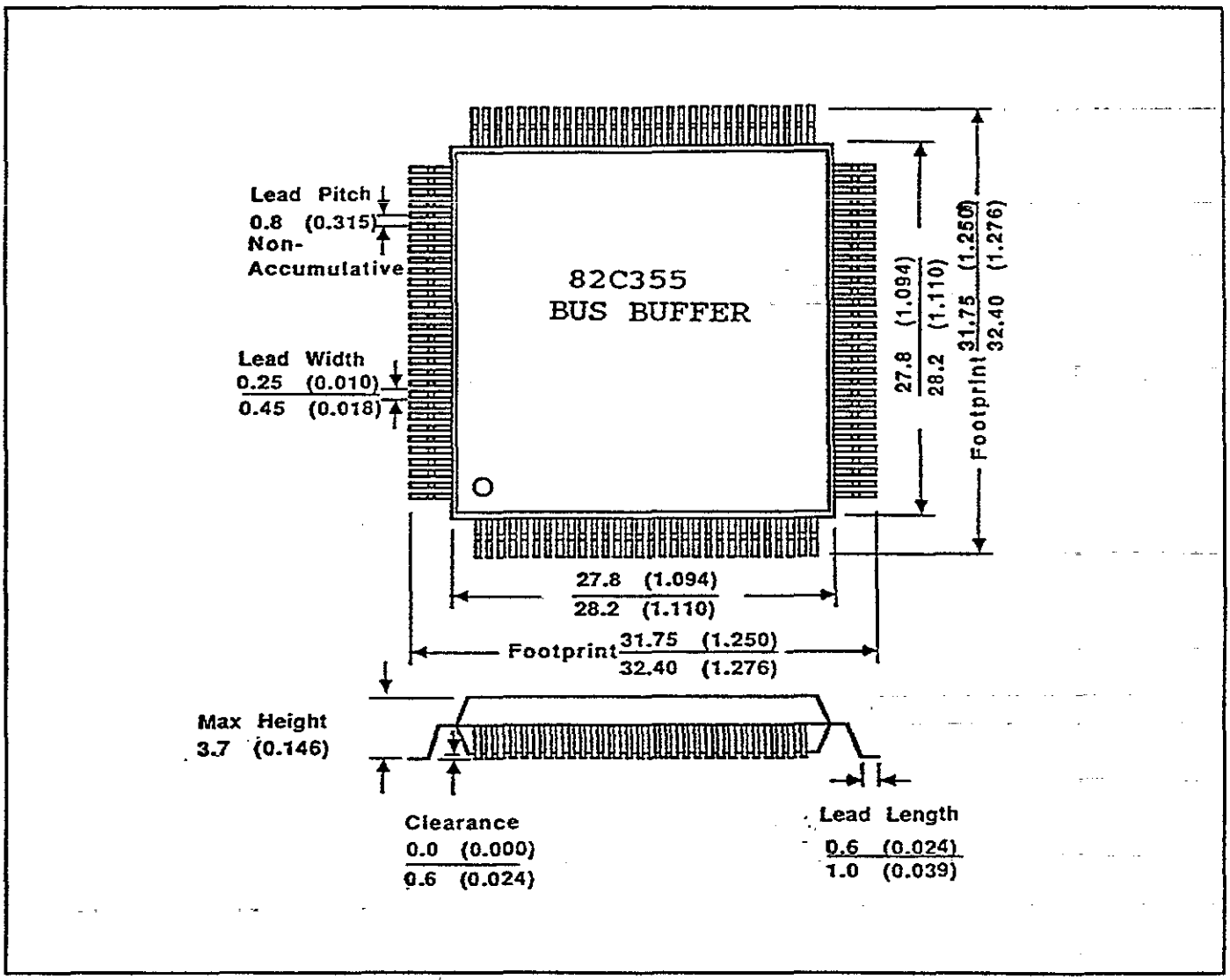
- 6.1 82C355 Data Buffer

Section 6

355 Physical Dimensions

6.1 82C355 Data Buffer

Figure 6-1. 120-Pin Plastic Flat Package



Legend minimum in inches (minimum milimeters) / maximum in inches (maximum milimeters)

This page is intentionally left blank.

VOLUME

82C356 Peripheral Controller

Section 1

356 Functional Description

Section 1

356 Functional Description

1.1 Features

The 82C356 peripheral controller contains the address buffers used to interface the local address bus A<2:23> and I/O channel address bus SA<0:19>. It also contains an equivalent Integrated Peripheral Controller (IPC) that incorporates two 8237 DMA controllers, two 8259 Interrupt Controllers, one 8254 Timer/Counter, and an MC146818 RTC with CMOS RAM, in addition to several other SSI interface logic devices.

1.2 Functional Subsystem

The 82C356 peripheral controller consists of the following functional subsystems as shown in Figure 1-1:

- Address BUS interface between the A and SA buses
- I/O Address Decodes
- 14.318 MHz oscillator crystal interface
- Port B and NMI logic
- Integrated Peripheral Controller (IPC)
- Configuration Registers

Address Bus Interface Between A and SA Buses

The 82C356 interconnects the System Address bus SA<0:19> to the local bus A<2:23>. The drivers provided are 24mA current drivers for direct connection to the I/O channel address bus.

The buffers are controlled by HLDA1, -MASTER, -REF, and -ATEN to drive the signals from the source to the target buses. Table 1-1 shows how the drivers are configured between the buses for each type of active bus request. When -REF is asserted, the refresh address is latched onto the SA bus as the refresh row address and the refresh counter for the I/O channel, residing in the 82C356, is incremented. When none of the controlling signals are active, the default buffer direction is from the A bus to the SA bus for memory accesses initiated by the CPU. For all CPU sourced accesses, the addresses are latched on the trailing edge of -MALE.

Figure 1-1. 82C356 Block Diagram

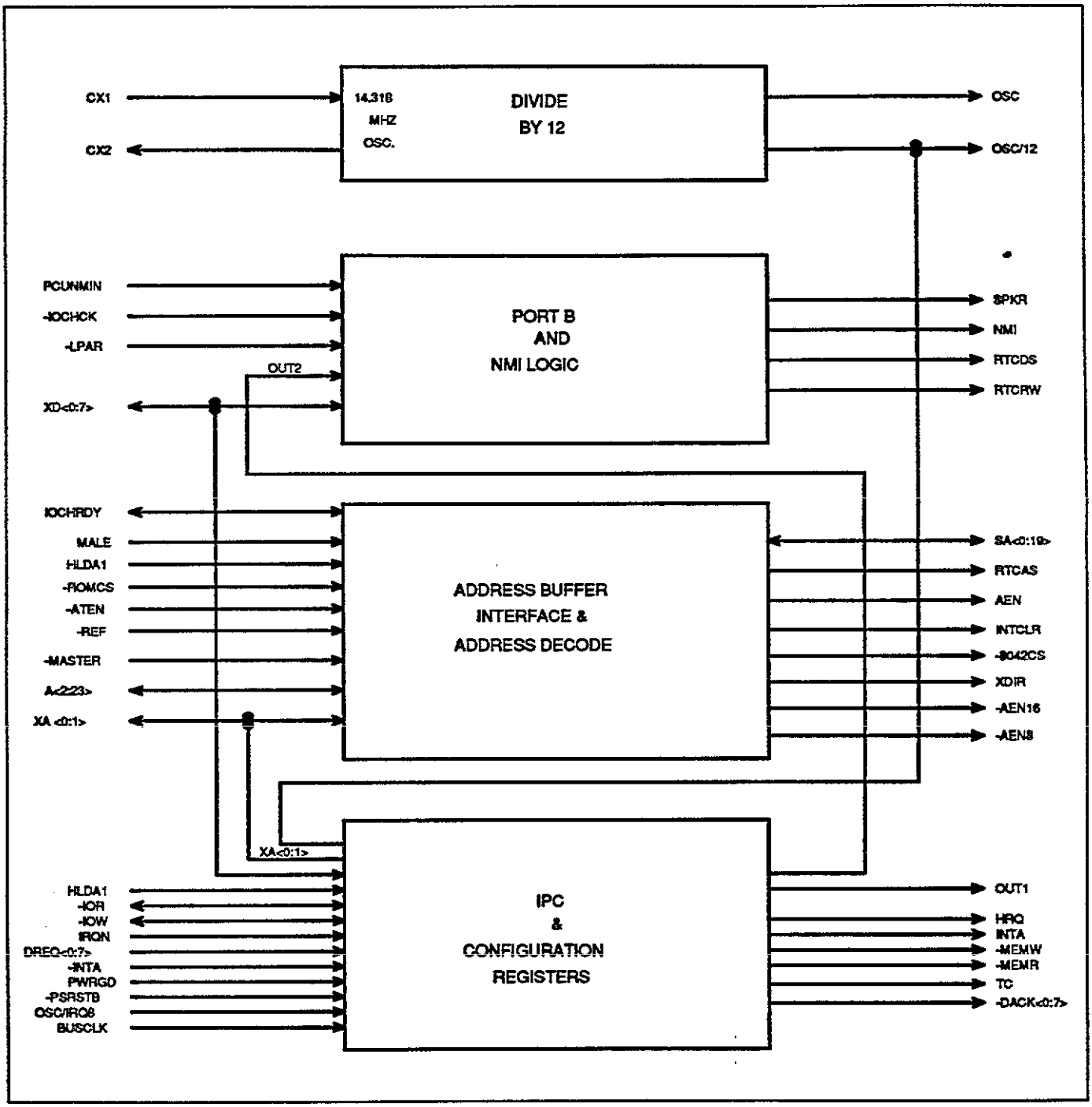


Table 1-1. Address Buffer Function

HLDA	ATEN	REF	MSTR	Mode	Action
0	1	1	1	NON-AT	All buses are tri-stated in this mode.
0	0	1	1	AT	The SA bus XA <0:1> are driven by the A bus.
0	1	0	1	REFRESH	A<2:16> are disabled.
1	1	1	0	MASTER	The SA bus drives the A bus and XA <0:1> while A<17:23> are tri-stated.
1	1	0	0	MASTER-REFRESH	The A bus is disabled.
1	1	1	1	DMA	The 82C356 drives the A bus, the SA bus, and XA<0:1>.

1.3 Address Decode

The 82C356 provides I/O decode for the keyboard controller. The signal -8042CS provides the address decoding for the keyboard controller. The -8042CS output is an I/O decode of addresses 60H and 61H.

1.4 Port B and NMI logic

The 82C356 provides access to Port B defined for the PC/AT, as shown in Table 1-2. The NMI (non-maskable interrupt) circuitry latches and enables the I/O channel check and parity error conditions. If the corresponding NMIs are enabled in port B, a non-maskable interrupt is generated to the CPU and the source of the NMI is recorded in port B. Reading port B indicates the source of the error condition. The master enable for NMI generation (as defined for PC/AT system at system I/O port 70H, bit7) is implemented in the 82C356. If this bit is set to 1, NMI generation is disabled and if set to 0, it is enabled.

Table 1-2. Port 61 Bit Definition

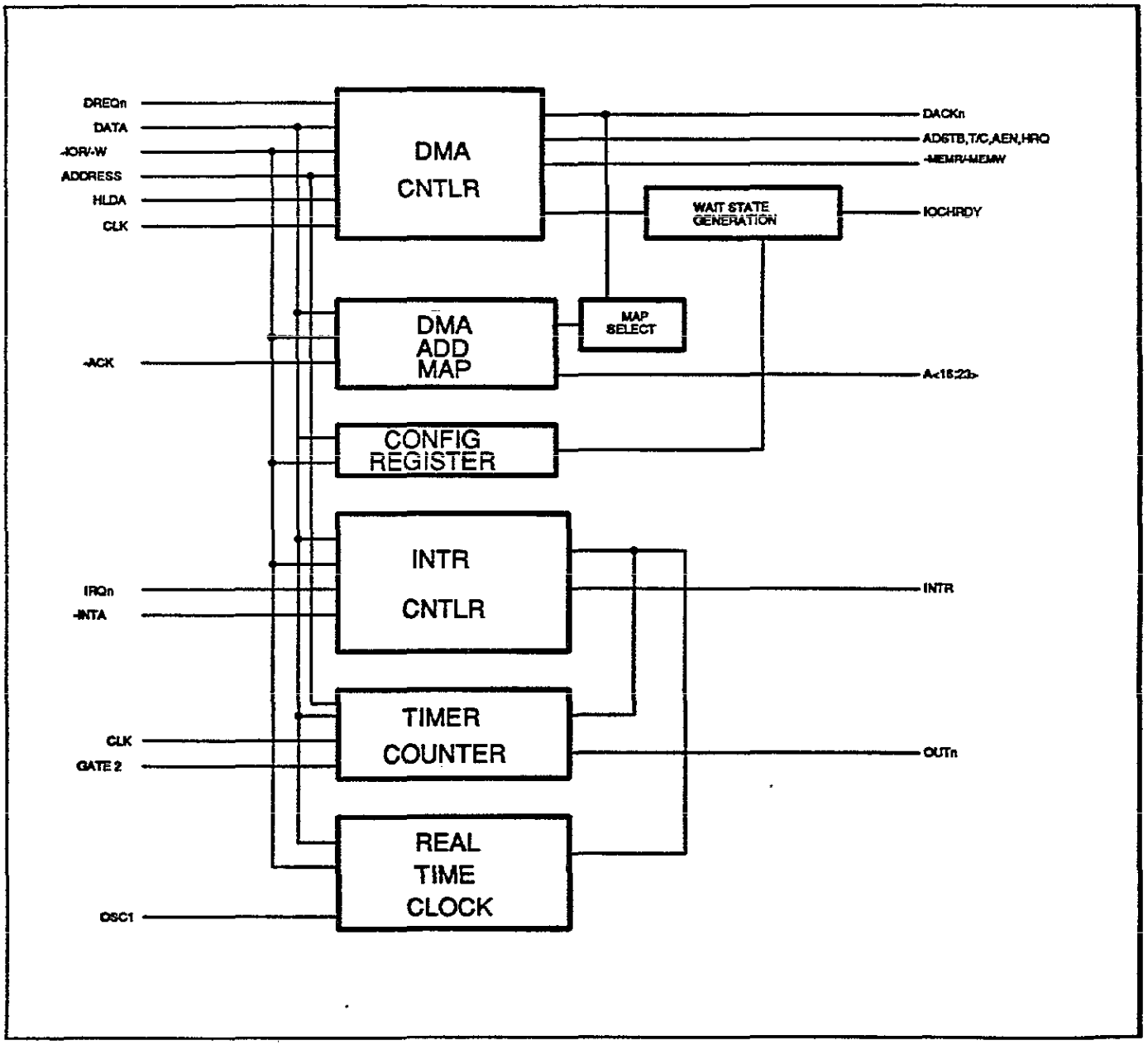
Address	Bit	Function	Description
61H			Port B Register
	7	Read only	PCK - System memory parity check
	6	Read only	CHK - I/O channel check
	5	Read only	T2O - Timer 2 out
	4	Read only	RFD - Refresh detect
	3	Read write	EIC - Enable I/O channel check
	2	Read write	ERP - Enable system memory parity check
	1	Read write	SPK - Speaker Data
	0	Read write	T2G - Timer 2 Gate Speaker

The 82C356 also accepts an NMI due to power fail, READY timeout, or from the Power Control Unit (PCU) 82C636. The PCU is used to control power for memory, serial port etc. This also provides slow DRAM refresh during standby mode when power is off. The PCU and PFAIL NMI, are combined into one pin, since the PCU can supply the PFAIL interrupt. Reg 26H<4> and Reg 26H<2> are used to enable the NMI sourced from the PCU and READY timeout, respectively. Reg 26H<3> and Reg 26H<0> indicate the source of the NMI.

1.5 Integrated Peripheral Controller

The Integrated Peripheral Controller (IPC) portion of the 82C356 is an LSI implementation of the standard peripherals required to implement an IBM PC/AT system board. It contains the equivalent of two 8237A DMA Controllers, two 8259A Interrupt Controllers, an 8254 Counter/Timer, and an MC146818 Real Time Clock with RAM. The IPC provides all the standard peripherals required for a system board implementation except the keyboard controller. Figure 1-2 illustrates the subsystems contained within the IPC.

Figure 1-2. Integrated Peripheral Controller Block Diagram



Two DMA Controllers are provided and connected in such a way as to provide the user with four channels of DMA (DMA1), for 8-bit transfers and three channels of DMA (DMA2), for 16-bit transfers (the first 16-bit DMA channel is used for cascading). Included, as part of the DMA subsystem, is the DMA Page Register used to drive the upper address lines when required.

Sixteen interrupt channels are provided in the IPC. These channels are partitioned into two cascaded controllers: INTC1 and INTC2, with 8 inputs each. Of these 16 channels, three are connected internally to various devices, allowing 13 user definable interrupt channels. The three internally connected channels are as follows:

- Channel 0 — Counter/Timer Counter 0 Interrupt
- Channel 2 — Cascade to Slave Interrupt Controller (INTC2)
- Channel 8 — Real Time Clock Interrupt

The remaining 13 channels may be defined and utilized as necessary to meet the users specific system requirements.

A Counter/Timer (CTC) subsystem is provided containing three independent counters. All three counters are driven from a clock input pin independent of all the other clock inputs. Counter 0 is connected to Interrupt 0 of INTC1. It is intended to be used as a multilevel interrupt to the system for such tasks as time keeping and task switching. Counter 1 may be programmed to generate pulses or square waves for use by external devices. The third channel (Counter 2) is a full function Counter/Timer which has a gated input for controlling the internal counter. This channel can be used as an interval counter, a timer, or a gated rate/pulse generator.

A Real Time Clock (RTC) is included in the 82C356 for maintaining the time and date. This subsystem also contains 114 bytes of CMOS RAM, in addition to the Clock/Calendar. The Clock/Calendar information and RAM are kept active by connecting the device to an external battery when the system power is turned off.

To interconnect and control all of these major subsystems, a multilevel control section is employed which is divided into subsystems for purposes of discussion.

The first section covers the Clock and Wait State Control. This subsystem controls the generation of DMA wait states and the negation of IOCHRDY (if programmed to do so) during CPU access of the device. The last subsystem is the Multilevel Decode.

1.6 Multilevel Decode Subsystem

To accommodate over 200 registers in the 82C356 and maintain I/O decode compatibility with the IBM PC/AT, a multilevel decode scheme is employed. The multilevel decode subsystem performs the function of generating enables to various subsystems. Control and direction of the XD0-XD7 data bus buffers are also handled by this subsystem. The output buffers are enabled whenever an enable is generated to an internal subsystem and the -IOR signal is asserted.

The multilevel decode subsystem provides 8 separate enables to various subsystems of the IPC. Table 1-3 contains a truth table of the internal decode.

Table 1-3. *Integrated Peripheral Controller Internal Decode*

-MSE (AEN)	XA9	XA8	XA7	XA6	XA5	XA4	XA3	XA2	XA1	XA0	Address Range	Selected Device
1	0	0	0	0	0	0	X	X	X	X	000-00F	DMA1
1	0	0	0	0	1	0	0	0	0	X	020-021	INTC1
1	0	0	0	0	1	0	0	0	1	X	022-023	CONFIG
1	0	0	0	1	0	0	0	0	X	X	040-043	CTC
1	0	0	0	1	1	1	X	X	X	X	070-071	RTC
1	0	0	1	0	0	0	X	X	X	X	080-08F	DMAPAG
1	0	0	1	0	1	X	X	0	0	X	0A0-0A1	INTC2
1	0	0	1	1	0	X	X	X	X	X	0C0-0DF	DMA2
0	X	X	X	X	X	X	X	X	X	X	Disabled	
X	1	X	X	X	X	X	X	X	X	X	Disabled	
X	X	1	X	X	X	X	X	X	X	X	Disabled	

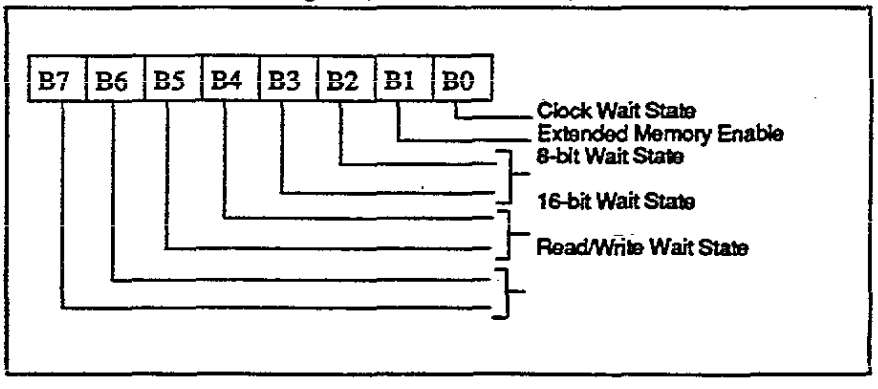
The decode is enabled by three signals: -MSE(AEN), XA9, and XA8. To enable any internal device, -MSE(AEN) must be 1 and XA9 and XA8 must be 0.

The decode scheme employed in the 82C356 is designed to comply with the IBM PC/AT requirements, and is more fully decoded. If the user wishes to take advantage of the areas unused by inserting additional peripherals in the I/O map, this may be done since the subsystems in the 82C356 do not respond to the unused address spaces established by the multilevel decode. The extra peripherals may be tied directly to the XD0-XD7 data lines since the IPC output buffers are not enabled unless an internal subsystem is enabled.

1.7 Clock and Wait State Control

The Clock and Wait Sate Control subsystem performs four functions: control of the DMA command width, control of the CPU read or write cycle length, and selection of the DMA clock rate. All of these functions are user selectable by writing to the Clock/Wait State Control Register.

Figure 1-3. *Clock/Wait State Control Register(023H) (Index 01H)*



Writing and reading this register is accomplished by first writing an 01H to location 022H to select the control register,and performing either a read or write to location 023H.

Clock Wait State: This bit allows the user to insert a divider between the DMA Controller subsystems and the SCLK input pin, or connect the two directly. When this bit position contains a 0, the SCLK input is divided by two and is used to drive both the 8-bit and 16-bit DMA subsystems. A 1 in this position bypasses the divider and uses the SCLK input directly. Whenever the state of this bit is changed, an internal synchronizer controls the actual switching of the clock to prevent a short clock pulse from causing a DMA malfunction.

Extended Memory Function Enable: This bit enables the extended -MEMR function. Normally, the assertion of -MEMR is delayed one clock cycle later than -IOW in the IBM PC/AT implementation (EMR=0). This may not be desirable in some systems. A 1 programmed into this bit position starts -MEMR at the same time as -IOW.

8-bit Wait State: Wait states may be inserted in 8-bit DMA cycles by programming these two bits.

8W1	8W0	8-bit DMA Wait States
0	0	1
0	1	2
1	0	3
1	1	4

Further control of the cycle length is available through the use of the IOCHRDY pin on the 82C356. During DMA, this pin is used as an input to the wait state generation logic to extend the cycle, if necessary. This input is driven low by the peripheral to extend the cycle. The cycle can then be completed by releasing IOCHRDY and allowing it to return high.

16-bit Wait State: Wait states can be independently controlled for both 8-bit and 16-bit DMA cycles. This allows the user to tailor the DMA cycle more closely to the application.

16W1	16W0	16-bit DMA Wait States
0	0	1
0	1	2
1	0	3
1	1	4

Read/Write Wait State: When the higher speed CPU's are accessing the IPC subsystems, the cycle can be extended by programming up to four wait states into the Configuration Register. This causes the IPC to assert a not ready condition on IOCHRDY (low) whenever a valid decode from the Top Level Decode is detected and either -IOR or -IOW is asserted. IOCHRDY remains low for the number of wait states programmed into the clock/wait state control register bits 6 and 7. Wait states are in increments of SCLK cycles and are not affected by the DMA Clock Divider.

RW1	RW0	Read/Write Cycle Wait States
0	0	1
0	1	2
1	0	3
1	1	4

The Configuration Register contents are preloaded by RESET4 to an initial value of 0C0H hex. This value establishes a default which is PC/AT compatible and corresponds to:

Read/Write cycles	4 wait states
16 bit DMA transfers	1 wait state
8 bit DMA transfers	1 wait state

-MEMR delayed 1 DMA clock cycle later than -IOW

DMA clock is equal to SCLK/2

1.8 DMA Functional Description

The equivalent of two 8237A DMA controllers is implemented in the IPC subsystem. Each controller is a four channel DMA device which generates the memory addresses and control signals necessary to transfer information between a peripheral device and memory directly. This allows high speed information transfer with little CPU intervention.

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

DMA cycle length control is provided internally in the IPC subsystem 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 65536 words. The register associated with each counter allows the channel to reinitialize without reprogramming. From this point on the description of the DMA subsystem pertains to both DMA1 and DMA2, unless otherwise specified.

DMA Operations

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

When a DMA request becomes active the device enters the Active condition and issues a hold request to the system. Once in the Active condition the DMA controller generates the necessary memory addresses and command signals to accomplish a memory-to-I/O, I/O-to-memory, or a memory-to-memory transfer. Memory-to-I/O and I/O-to-memory transfers take place in one cycle, while memory-to-memory transfers require two 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 cycle. Memory-to-memory transfers however, require the DMA to first store data from the read operation in an internal register, and then write the stored data to memory on the subsequent cycle.

During transfers between memory and I/O, two commands are activated during the same cycle. In the case of a memory-to-I/O transfer, the DMA controller asserts both -MEMR and -IOW, allowing data to be transferred directly to the requesting device from memory. Note that the 82C356 does not latch data from, or drive data out, on this type of cycle.

Idle Condition

When no device is requesting service the DMA is in an Idle Condition which maintains the state machine in the SI state. During this time, the DMA controller samples the DREQ input pins every clock cycle. The internal select from the multilevel 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 the above two situations occurs, the DMA exits the idle condition. Note that the program condition has priority over the active condition since a CPU cycle has already started.

Program Condition

The program condition is entered whenever HLDA is inactive and internal select is active. During this time, address lines XA<0:3> become inputs if DMA1 is selected, or XA<1:4> become inputs if DMA2 is selected. Note, when DMA2 is selected, XA0 is ignored. These address inputs are used to select the DMA controller registers which are to be read or written. Due to the large number of internal registers in the DMA subsystem, an internal flip-flop bit 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. This internal flip-flop can be cleared by RESET4 or a Master Clear command, and can 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 IPC subsystem 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 DMA controller can occur if a request for service occurs on an unmasked channel which is being programmed. The channel should be masked or DMA disabled to prevent the DMA controller from attempting to service a device with a channel that is partially programmed.

Active Condition

The IPC 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 DMA controller then begins a DMA transfer cycle.

In a read cycle, for example, after receiving a DREQ, the DMA controller issues an HRQ to the system. Until an HLDA is returned, the DMA controller remains in an idle condition. On the next clock cycle, the DMA controller exits Idle and enters state S0. During S0, the device resolves priority and issue -DACK on the highest priority channel requesting service. The DMA controller then proceeds to state S1 where the multiplexed addresses are output and latched. State S2 is then entered, at which time the DMA controller asserts -MEMR. The device then transitions into S3 where the -IOW command is asserted. The DMA controller remains in S3 until the Wait State counter has decremented to zero and IOCHRDY is true. Note that at least one additional S3 occurs, unless Compressed Timing is selected. Once IPCHRDY is true, the DMA controller enters S4 where both commands are deasserted. In Burst Mode and Demand Mode (discussed in the following sections), subsequent cycles begin in S2 unless the intermediate addresses require updating. In these subsequent cycles, the lower addresses are changed in S2. The DMA controller can be programmed on a channel by channel basis to operate in one of four modes:

- Single Transfer Mode
- Block Transfer Mode
- Demand Transfer Mode
- Cascade Mode

Single Transfer Mode

This mode directs the DMA controller 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 DMA controller deasserts HRQ and releases the bus once the transfer is complete. After HLDA has gone inactive, the DMA controller again asserts HRQ and executes another cycle on the same channel, unless a request from a higher priority channel has been received. In this mode the CPU is ensured of being 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 TC pulse is generated. If the autoinitialization option has been enabled, the channel reinitializes itself. If autoinitialize is not selected the DMA controller sets the DMA request bit mask and suspends transferring on the channel.

Block Transfer Mode

When Block Transfer Mode is selected, the DMA controller begins transfers in response to either a DREQ or a software request and continues until a terminal count of 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. Autoinitialization is also operational in this mode.

Demand Transfer Mode

In Demand Transfer mode, the DMA begins transfers in response to the assertion of DREQ and continues until either terminal count is reached, or DREQ becomes inactive. This mode is normally used for peripherals which have limited buffering availability. The peripheral can initiate a transfer and continue until its buffer capacity is exhausted. The peripheral may then re-establish service by again asserting DREQ. During idle periods between transfers, the CPU is released to operate and can monitor the operation by reading intermediate values from the address and word count registers. Once DREQ has been 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 autoinitialization (if enabled.)

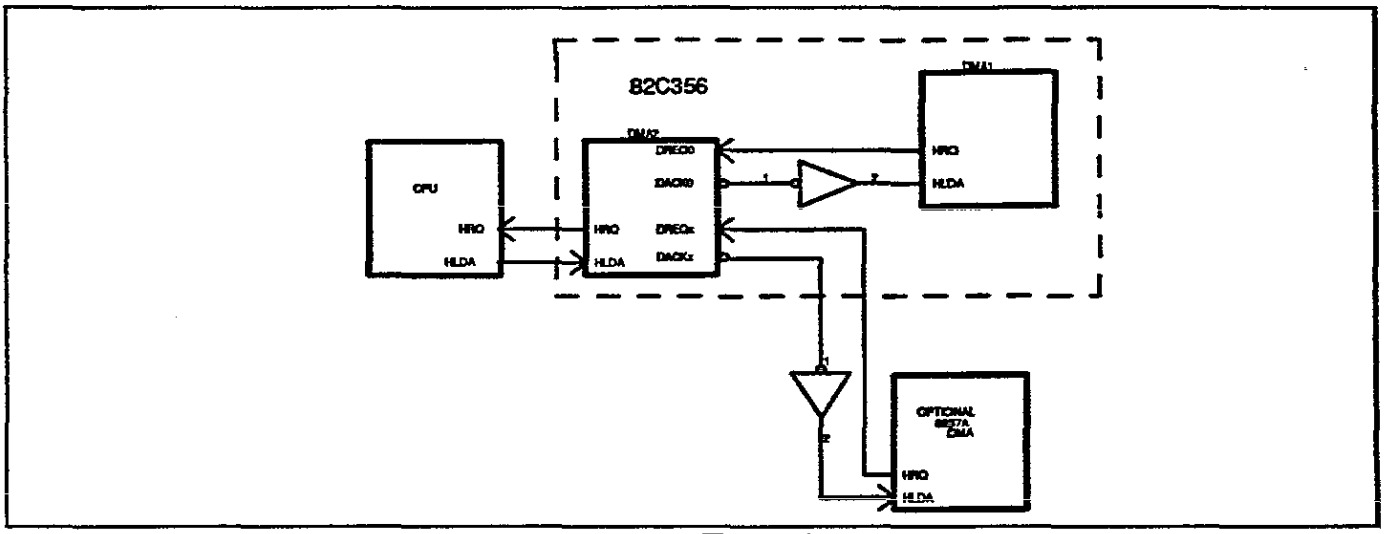
Cascade Mode

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 1-3 shows the cascade interconnection for two DMA devices. Note that Channel 0 of DMA2 is internally connected for Cascade mode to DMA1. Additional devices can be cascaded to the available channels in either DMA1 or DMA2 since cascade is not limited to two levels of DMA controllers.

When programming cascaded controllers, begin with the device that is actually generating HRQ to the system (first level device), then proceed to the second level devices. RESET4 causes the -DACK outputs to become active low and are placed in the 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 DMA controllers have an inverter between -DACK0 of DMA2 and HLDA of DMA1. The first level device's DMA request mask bit prevents second level cascaded devices from generating unwanted hold requests during the initialization process.

Figure 1-4. Cascade Mode Interconnect



1.9 DMA Transfers

Four types of transfer modes are provided in the DMA subsystem. These transfer types are: Read Transfer, Write Transfer, Memory-to-Memory Transfer, and Verify Transfer.

- **Read Transfer** — Read transfers move data from memory to an I/O device, by generating the memory address and asserting -MEMR and -IOW during the same cycle.
- **Write Transfer** — Write transfers move data from an I/O device to memory by generating the memory address and asserting -IOR and -MEMW during the same cycle.
- **Memory-to-Memory Transfer** — The memory-to-memory transfer is used to move a block of memory from one location in memory to another. DMA channels 0 and 1 may be programmed to operate as memory-to-memory channels by setting bits in the Command Register. Once programmed to perform a memory-to-memory tranfer the process can be started by generating either a software 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, a byte of data is latched in the internal Temporary Register of the 82C356. The contents of this register are then placed on the XD<0:7> data lines during the write portion of the cycle and subsequently written to memory. Channel 0 may be programmed to maintain the same source address on every cycle. This allows the CPU to initialize large blocks of memory with the same value. The DMA controller will continue performing transfer cycles until Channel 1 reaches terminal count.
- **Verify Transfer** — The verify transfer is a pseudo-transfer which is useful for diagnostics. In this type of transfer the DMA will operate as if it is performing a Read or Write Transfer by generating HRQ, addresses and -DACK but will do so without asserting a command signal. Since no transfer actually takes place IOCHRDY is ignored during Verify transfer cycles.

1.10 Autoinitialization

Each of the four DMA channel Mode Registers contains a bit which causes the channel 1 to reinitialize after reaching terminal count. During this process, referred to as autoinitialization, 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 has been programmed to autoinitialize, the request mask bit will not be 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 autoinitialization. If Channel 0 reaches terminal count before Channel 1, then Channel 0 will reload the starting address and word count and continue transferring data from the beginning of the source block. Should Channel 1 reach terminal count first, it will reload the current registers and Channel 0 will remain uninitialized.

1.11 DREQ Priority

The DMA subsystem 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 downward through the channels in order; 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. The rotating priority assignment is illustrated in Table 1-4.

Table 1-4. Rotating Priority Scheme

First Arbitration		Second Arbitration		Third Arbitration	Priority
Channel 0	Cycle Granted	Channel 2	Cycle Granted	Channel 3	Highest
Channel 1	Cycle Granted	Channel 3		Channel 0	
Channel 2		Channel 0		Channel 1	
Channel 3		Channel 1		Channel 2	Lowest

Note In the preceeding table, a box around a channel number indicates a requested channel.

In instances where multiple requests occur at the same time, the IPC issues an HRQ but does not freeze the priority logic until HLDA is returned. Once HDLA becomes active, the priority logic is frozen, and -DACK is asserted on the highest requesting channel. Priority is not re-evaluated until HLDA has been deactivated.

1.12 Address Generation

The DMA Page Register generates A<17:23>. During 16-bit DMA transfers, XA0 and XA16 remain inactive. The DMA Page Register is a set of 16, 8-bit registers in the IPC which are used to generate the high order addresses during DMA cycles. Only 8 of the registers are actually used, but all 16 are included to maintain IBM PC/AT compatibility. Each DMA channel has a register associated with it, with the exception of Channel 0 of DMA2, which is used for internal cascading to DMA1. Assignment of each of these registers is shown in Table 1-5 along with its Read/Write address.

Table 1-5. DMA Address Extension Map

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

During Demand and Block Transfers, the IPC generates multiply sequential transfers. For most of these transfers the information in the internal address latches remains the same, eliminating the need to be relatched. Since the need to update the latches occurs only when a carry or borrow from the lower 8-bits of the Address Counter exists, the IPC only updates the latch contents when necessary. The IPC therefore, only executes S1 cycles when necessary, resulting in an overall through-put improvement.

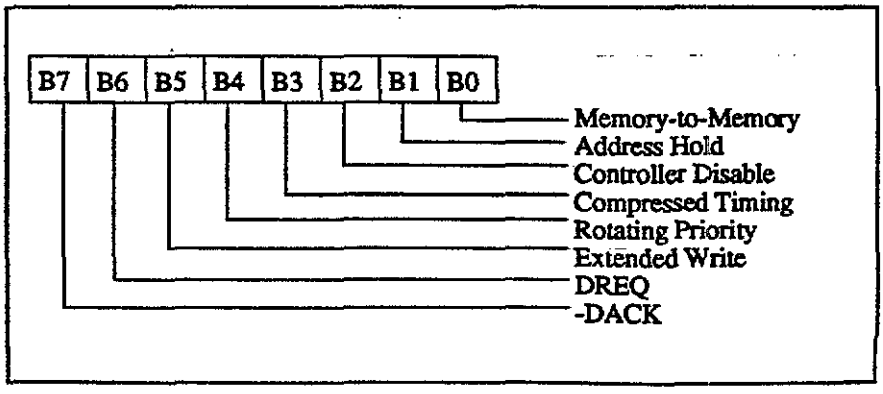
1.13 Compressed Timing

The DMA subsystem in the 82C356 can be programmed to transfer a word in as few as 3 DMA clock cycles. The normal DMA cycle consists of three states: S2, S3, and S4 (this assumes Demand or Block Transfer Mode). Normal transfers require 4 DMA clock cycles since S3 is executed twice due to the 1 wait state insertion. In systems capable of supporting high through-put, the IPC can be programmed to omit one S3 and assert both commands in S2. S2 begins the cycle by generating the address and asserting both commands. One S3 cycle is executed and the cycle terminates in S4. If Compressed Timing is selected, TC is active in S2. Note that Compressed Timing is not allowed for memory-to-memory transfers.

1.14 Register Descriptions

- **Current Address Register** — Each DMA channel has a 16-bit Current Address Register which 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 autoinitialization 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 which 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 rolls over from zero to FFFFH. When this occurs, the DMA controller generates TC and either suspend operation on that channel and set the appropriate Request Mask Bit or Autoinitialize and continue.
- **Base Word Count Register** — This register preserves the initial value of the Current Word Count Register. It is also a write only register which is loaded by writing to the Current Word Count Register. This register is loaded in the Current Word Count Register during autoinitialization.
- **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 RESET4 or a Master Clear command.

Figure 1-5. Clock/Wait State Control Register (023H) (Index 01H)



Memory-to-Memory A 1 in the bit 0 position enables Channel 0 and Channel 1 to be used for memory-to-memory transfers.

Address Hold Writing a 1 enables the address hold feature in Channel 0 when performing memory-to-memory transfers.

Controller Disable Bit 2 is the master disable for the DMA controller. Writing a 1 to this location disables the DMAsubsystem (DMA1 or DMA2). This function is normally used whenever the CPU needs to reprogram one of the channels to prevent DMA cycles from occurring.

Compressed Timing Compressed timing is enabled by writing a 1. The default 0 condition causes the DMA to operate with normal timing.

Rotating Priority Writing a 1 causes the 82C356 to utilize a rotating priority scheme for honoring DMA requests. The default condition is fixed priority.

Extended Write Extended Write is enabled by writing a 1 to 5 bit, causing the write commands to be asserted one DMA cycle earlier during a transfer. The read and write commands both begin in state S2 when enabled.

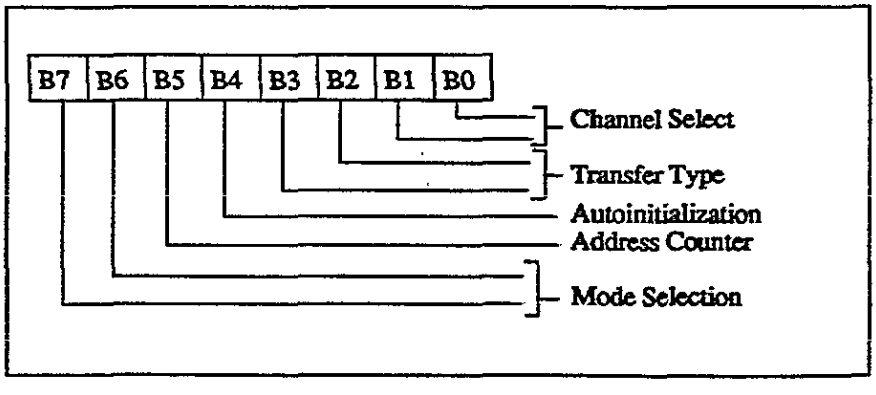
DREQ DREQ active level is determined by bit 6. Writing a 1 in this bit position causes DREQ to become active low.

-DACK -DACK active level is determined by bit 7. Programming a 1 in this bit position makes DACK an active high signal.

Mode Register

Each DMA channel has an associated Mode Register. All four Mode Registers reside at the same I/O address. Bits 0 and 1 of the Write Mode Register command determines which channel's Mode Register is written. 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 will both be 1.

Figure 1-6. Mode Register



Channel Select <0:1>: Channel Select bits 1 and 0 determine which channel's Mode Register is written.

Table 1-6. Channel Selector

CS1	CS0	Channel
0	0	Channel 0 Select
0	1	Channel 1 Select
1	0	Channel 2 Select
1	1	Channel 3 Select

Transfer Type <0:1>: Bits 2 and 3 control the type of transfer to be performed.

Table 1-7. Transfer Type

TT1	TT0	Type
0	0	Verify Transfer
0	1	Write Transfer
1	0	Read Transfer
1	1	Illegal

Autoinitialization 1: Autoinitilization function is enabled by writing a 1 in bit 4 of the Mode Register.

Address Counter: Bit 5 determines the direction of the address counter, as well as the address after the transfer. A 0 increments the address after each transfer. A 1 decrements the address after each transfer.

Mode Selection <0:1> Mode Selection for each channel is accomplished by bits 6 and 7.

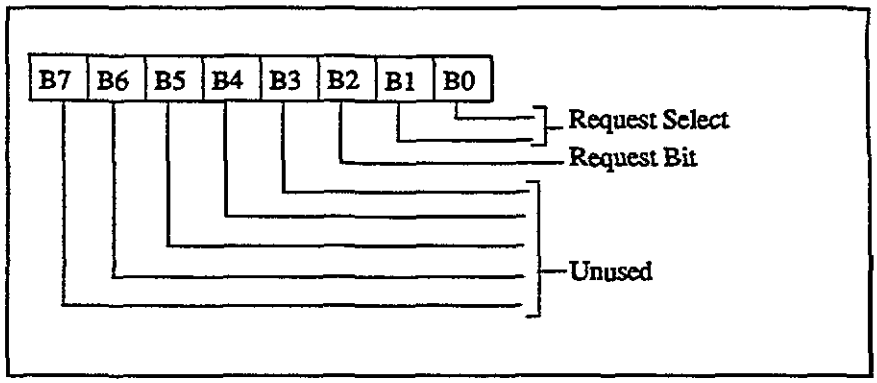
Table 1-8. Mode Selection

M1	M0	Mode
0	0	Demand Mode
0	1	Single Cycle Mode
1	0	Block Mode
1	1	Cascade Mode

Request Register

This four bit register is used to generate software requests (DMA service can be requested either externally or under software control). Request Register bits can be set or cleared 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 RESET4.

Figure 1-7. *Request Register Write Operation*



Request Select <0:1>: Channel Select 0 and 1 determines which channel's Mode Register is written. Read back for the mode register results in bits 0 and 1 being ones.

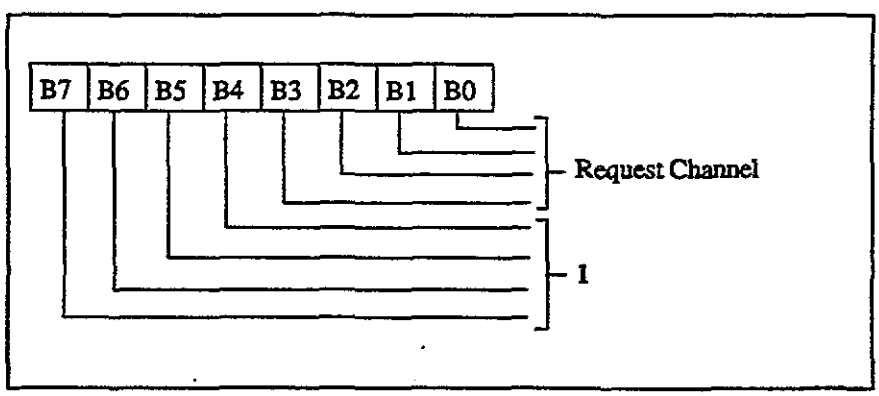
Table 1-9. *Request Channel Select*

RS1	RS0	Channel
0	0	Channel 0 Select
0	1	Channel 1 Select
1	0	Channel 2 Select
1	1	Channel 3 Select

Request Bit: The request bit is set by writing a 1 to bit 2. RS1-RS0 selects which bit (channel) is to be manipulated.

Format for the Request Register read operation is shown as follows:

Figure 1-8. *Request Register Read Operation*

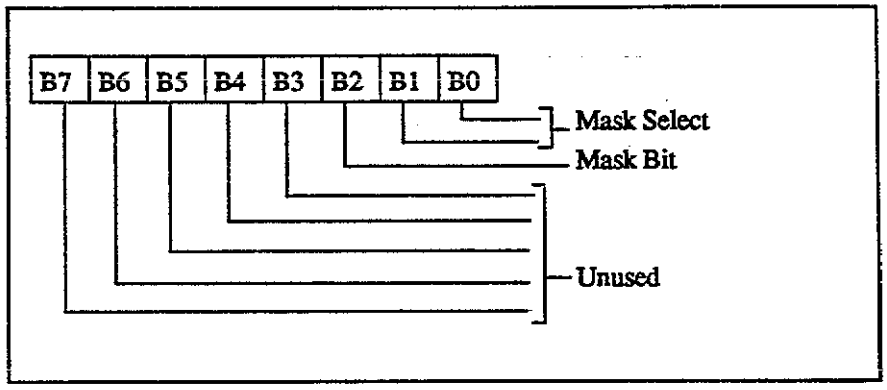


Request Channel <0:3>: During a request Register read, the state of the request bit associated with each channel is returned in bits 0 through 3 of the byte. The bit position corresponds to the channel number.

Request Mask Register

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

Figure 1-9. Register Mask Register Set/Reset Operation



Mask Select<0:1>: These two bits select the specific mask bit which is to be set or reset.

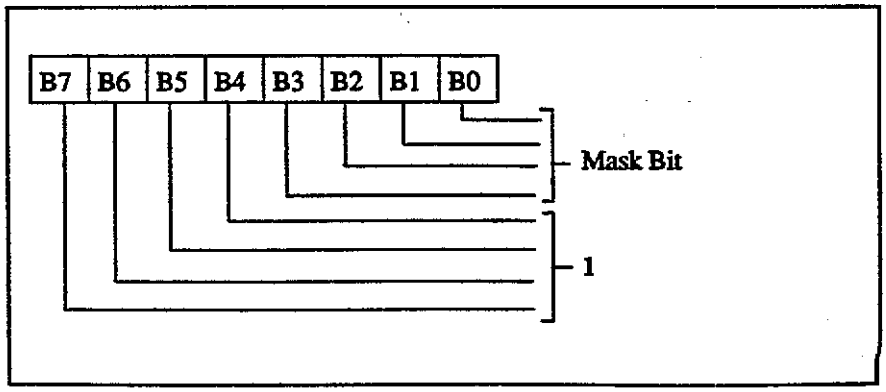
Table 1-10. Mask Channel Select

MS1	MS0	Channel
0	0	Channel 0 Select
0	1	Channel 1 Select
1	0	Channel 2 Select
1	1	Channel 3 Select

Mask Bit: Bit 2 sets or resets the request mask bit for the channel selected by MS1 and MS0). Writing a 1 in this bit position sets the mask, inhibiting external requests.

Alternately, all four mask bits can be programmed in one operation by writing to the Write All Mask Bits address. Data format for this and Read All mask Bits function is shown as follows:

Figure 1-10. Request Register Read Operation

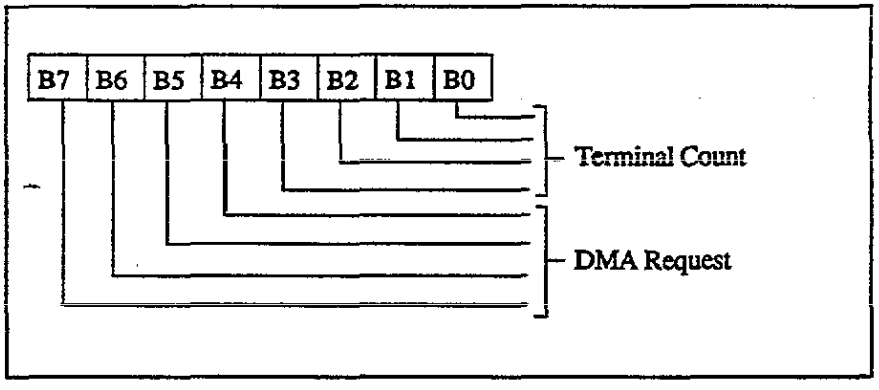


Mask Bit <0:3>: Each bit position in the field represents the mask bit of a channel. The mask bit number corresponds to the channel number associated with the mask bit. All four mask bits are set following a RESET4 or a Master Clear command. Individual channel mask bits are set as a result of terminal count being reached if Autoinitialize is disabled. The entire register can be cleared, enabling all four channels, by performing a Clear Mask Register operation.

Status Register

The status of all four channels can be determined by reading the Status Register. Information is available to determine if a channel has reached terminal count and whether an external service request is pending. Bits 0-3 of this register are cleared by RESET4, Master Clear, or each time a Status Read takes place. Bits 4-7 are cleared by RESET4, Master Clear, or the pending request being deasserted. Bits 4-7 are not affected by the state of the Mask Register Bits. The Channel number corresponds to the bit position.

Figure 1-11. Status Read Only Register



Temporary Register

The Temporary Register 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 XD<0:7>. During the second cycle of the transfer, the data in the Temporary Register is placed on the XD<0:7> pins. Data from the last memory-to-memory transfer remains in the register unless a RESET4 or Master Clear occurs.

Special Commands

Five Special Commands are provided to make the task of programming the DMA controller easier. These commands are activated as a result of a specific address and assertion of either a -IOR or -IOW. Information on the data lines is ignored by the 82C356 whenever an -IOW activated command is issued, thus data returned on -IOR activated commands is invalid.

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 flip-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 RESET4, the DMA will be 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, Channel 3 last.

1.15 Interrupt Controller Functional Description

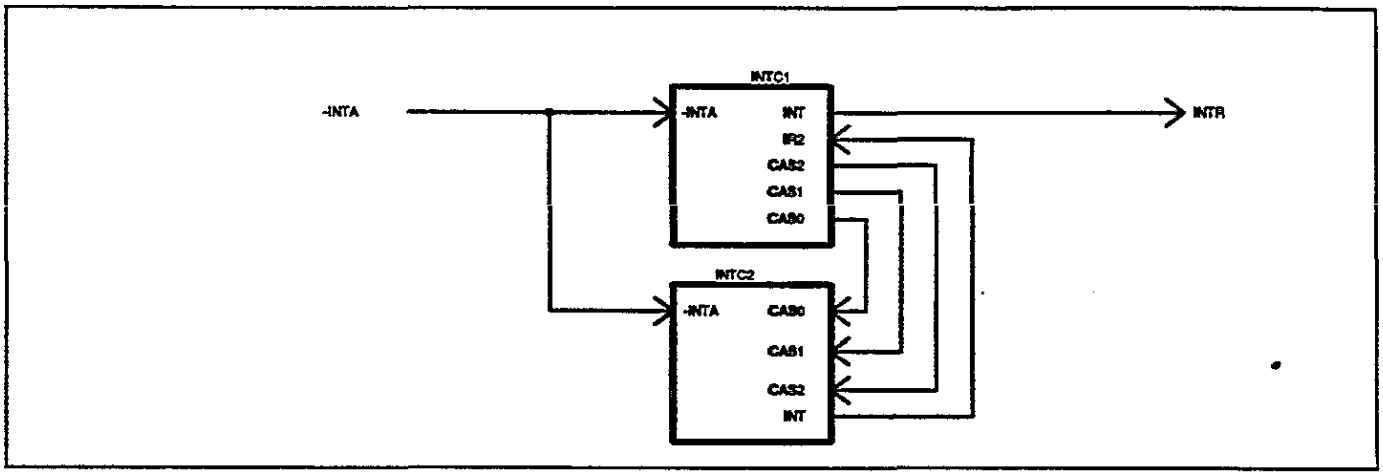
The programmable interrupt controllers in the 82C356 functions as a system wide interrupt manager in an IAPX86 system. They accept requests from peripherals, resolve priority on pending interrupts and interrupts in service, issue an interrupt request to the CPU, and provide a vector which is issued as 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.

Interrupt System

Two interrupt controllers, INTC1 and INTC2, are included in the 82C356. Each of the interrupt controllers are equivalent to an 8259A device operating in IAPX86 Mode. The two devices are interconnected and must be programmed to operate in Cascade Mode (see Figure 1-12) for proper operation of all 16 interrupt channels. INTC1 is located at addresses 020H-021H and is configured for Master operation (defined below) in Cascade Mode. INTC2 is a Slave device (defined below) and is located at 0A0h-0A1H. The Interrupt Request output signal from the INTC2 (INT) is internally connected to the interrupt request input Channel 2 (IR2) of the INTC1. The address decoding and Cascade interconnection matches that of the IBM PC/AT.

Figure 1-12. Internal Cascade Interconnect



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. Interrupt request from the Real Time Clock is connected to Channel 0 (IR0) of INTC2. Table 1-11 lists the 16 interrupt channels and their interrupt request source.

Description of the Interrupt Subsystem pertains to both INTC1 and INTC2 unless otherwise noted. Wherever register addresses are used, the address for the INTC1 register is listed first and the address for the INTC2 register follows in parenthesis. For Example, 020H (0A0H) indicates 020H is the address for INTC1 and 0A0H is for INTC2.

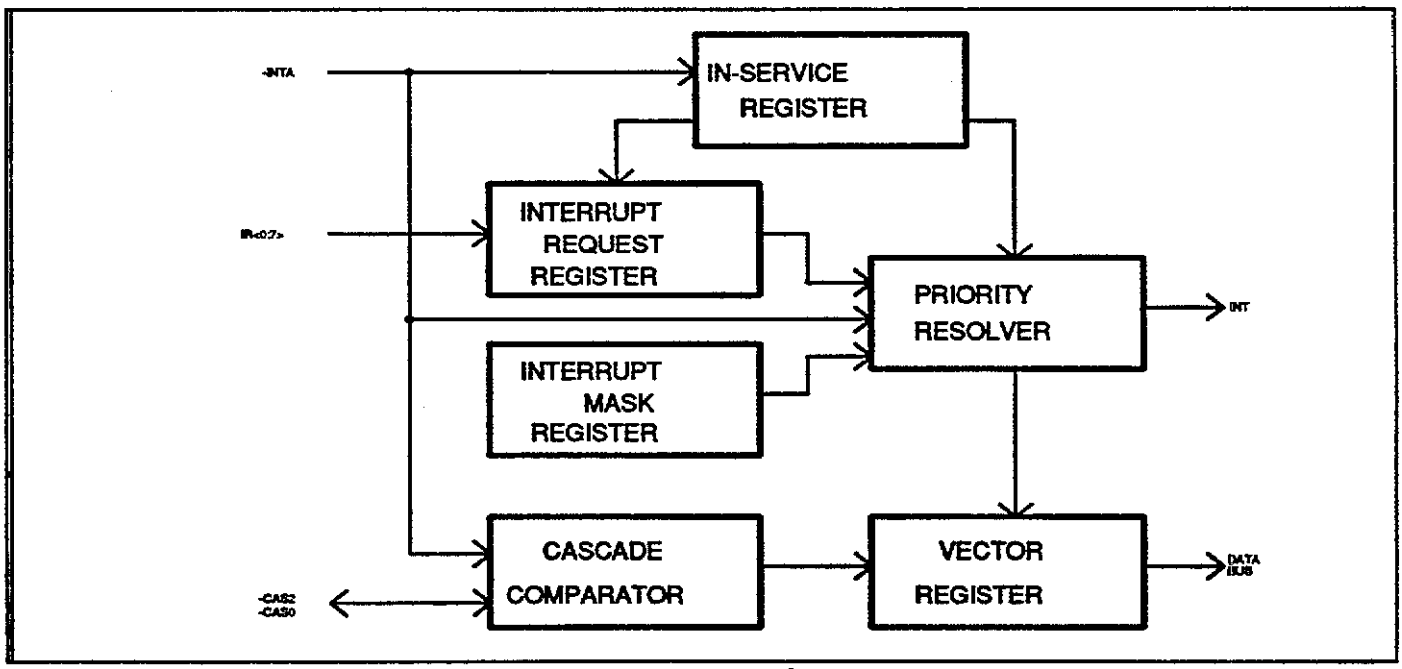
Table 1-11. Interrupt Request Source Environment

Controller Number	Channel Number	Interrupt Request Source
INTC1	IR0	Counter/Timer Out 0
INTC1	IR1	IRQ1 Input Pin
INTC1	IR2	INTC2 Cascade Interrupt
INTC1	IR3	IRQ3 Input Pin
INTC1	IR4	IRQ4 Input Pin
INTC1	IR5	IRQ5 Input Pin
INTC1	IR6	IRQ6 Input Pin
INTC1	IR7	IRQ7 Input Pin
INTC2	IR0	Real Time Clock IRQ
INTC2	IR1	IRQ9 Input Pin
INTC2	IR2	IRQ10 Input Pin
INTC2	IR3	IRQ11 Input Pin
INTC2	IR4	IRQ12 Input Pin
INTC2	IR5	IRQ13 Input Pin
INTC2	IR6	IRQ14 Input Pin
INTC2	IR7	IRQ15 Input Pin

1.16 Controller Operation

Figure 1-13 is a block diagram of the major elements in the interrupt controller. The Interrupt Request Register (IRR) is used to store requests from all of the channels that are requesting service. Interrupt Request Register bits are labeled using the Channel Name IR<0:7> The In-Service Register (ISR) contains all the channels which are currently being serviced (more than one channel can be in service at a time). In-Service Register bits are labeled IS<0:7>. The Interrupt Mask Register (IMR) allows the CPU to disable any or all of the interrupt channels. The Priority Resolver evaluates inputs form the above three register, 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 which is compared in the Cascade Buffer/Comparitor with a three bit ID code previously written. If a match occurs in the slave controller, it will generate an interrupt vector. The contents of the Vector Register are used to provide the CPU with an interrupt vector during Interrupt Acknowledge (INTA) cycles.

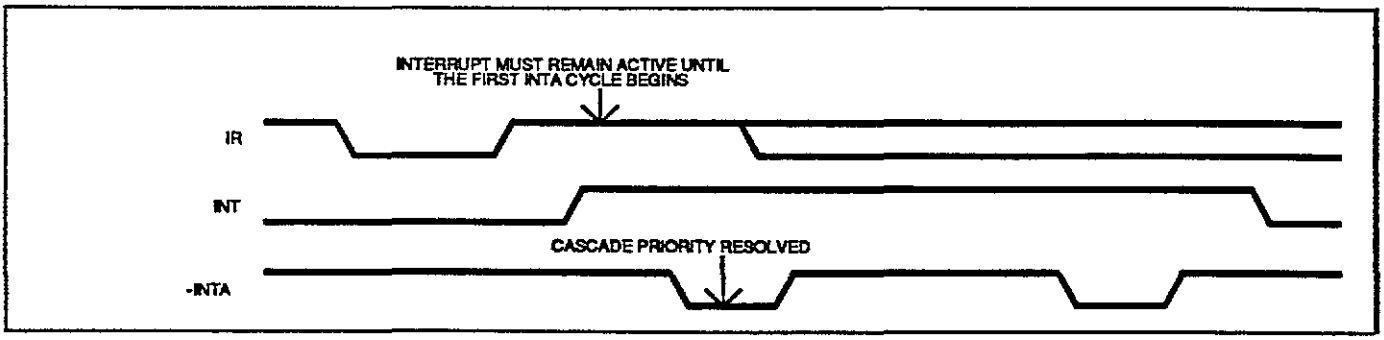
Figure 1-13. Interrupt Controller Block Diagram



1.17 Interrupt Sequence

The Interupt Controller subsystem 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 which is provided by the Interrupt Controller on the second of two CPU generated INTA cycles (the first INTA cycle is used for resolving priority and the second cycle is for transferring the vector to the CPU), as shown in Figure 1-14. The events which occur during an interrupt sequence are as follows:

Figure 1-14. Interrupt Sequence



1. One or more of the interrupt requests (IR7-IR0) become active, setting the corresponding IRR bit(s).
2. The interrupt controller resolves priority based on the state of the 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 cleared. The internal Cascade address is generated and XD<0:7> remain tri-stated.
5. The CPU will execute a second INTA cycle, during which the Interupt Controller will drive an 8-bit The format of this vector is shown in Table 1-12. Note that V<3:7> in Table 1-12 are programmable by writing to Initialization Control Word 2 (see *Section 1.19 Programming the Interrupt Controller*, Subsection titled *Initialization Command Words*).
6. At the end of the second INTA cycle, the ISR bit is cleared if the Automatic End Of Interrupt mode is selected (see *Section 1.18 End Of Interrupt*). Otherwise, the ISR bit must be cleared by an End Of Interrupt (EOI) command from the CPU at the end of the interrupt service routine.

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

Table 1-12. Interrupt Vector Byte

IRQ	D7	D6	D5	D4	D3	D2	D1	D0
IRQ7	V7	V6	V5	V4	V3	1	1	1
IRQ6	V7	V6	V5	V4	V3	1	1	0
IRQ5	V7	V6	V5	V4	V3	1	0	1
IRQ4	V7	V6	V5	V4	V3	1	0	0
IRQ3	V7	V6	V5	V4	V3	0	1	1
IRQ2	V7	V6	V5	V4	V3	0	1	0
IRQ1	V7	V6	V5	V4	V3	0	0	1
IRQ0	V7	V6	V5	V4	V3	0	0	0

1.18 End of Interrupt

End of Interrupt (EOI) is defined as the condition which causes an ISR bit to be cleared. 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 ISR bit (non-specific EOI).

The 82C356 can determine the correct ISR bit to reset when operated in modes which do not alter the fully nested structure, since the current highest priority ISR bit is the last level acknowledged and serviced. In conditions where the fully nested structure is not preserved, a specific EOI must be generated at the end of the interrupt service routine. An ISR bit that is masked, in Special Mask Mode by a IMR bit, will not be cleared by a non-specific EOI command. The interrupt controller can optionally 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 which 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:

Table 1-13. Fixed Priority Mode Interrupts

Priority Status	Lowest							Highest
	7	6	5	4	3	2	1	0

Nesting allows interrupts of a higher priority 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 which occurs during an interrupt service routine, will only be acknowledged if the CPU has internally re-enabled the interrupts.

Specific Rotation Mode: Specific Rotation allows the system software to re-assign priority levels by issuing a command which redefines the highest priority channel.

Table 1-14. Before Rotation

Priority Status	Lowest							Highest
	7	6	5	4	3	2	1	0

(specific rotation command issued with Channel 5 specified)

Table 1-15. After Rotation

Priority Status	Lowest							Highest
	5	4	3	2	1	0	7	6

Automatic Rotation Mode: In applications where 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 will be serviced at least once in 8 interrupt requests to the CPU from the controller. Automatic rotation will occur, it enabled, due to the occurrence of EOI (automatic or CPU generated).

Table 1-16. Before Rotation (IR4 is highest priority request being serviced)

ISR Status Bit	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
	0	1	0	1	0	0	0	0
Priority Status	**Lowest**							**Highest**
	7	6	5	4	3	2	1	0

Table 1-17. After Rotation (IR4 service completed)

ISR Status Bit	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
	0	1	0	0	0	0	0	0
Priority Status	**Lowest**							**Highest**
	4	3	2	1	0	7	6	5

1.19 Programming the Interrupt Controller

Two types of commands are used to control the 82C356 interrupt controllers, Initialization Command Words (ICWs)and operational Command Words (OCWs).

Initialization Command Words

The initialization process consists of writing a sequence of 4 bytes to each interrupt controller. The initialization sequence is started by writing the first Initialization Command Word (ICW1) to address 020H (0A0H) with a 1 on bit 4 of the data byte. The interrupt controller interprets this as the start of the initialization sequence and does the following:

1. Initialization Command Word Counter is reset to zero.
2. CW1 is latched into the device.
3. Fixed Priority Mode is selected
4. IR7 is assigned the highest priority
5. Interrupt Mask Register is cleared
6. Slave Mode Address is set to 7
7. Special Mask Mode is disabled
8. IRR is selected for Status Read operations

The next three I/O writes to address 021H (0A1H), will load ICW2-ICW4. See Figure 1-15 for a flow chart of the initialization sequence. The initialization sequence can be terminated at any point (all 4 bytes must be written for the controller to be properly initialized) by writing to address 020H (0A0H) with a 0 in data bit 4. Note, this causes OCW2 or OCW3 to be written.

Figure 1-15. Initialization Sequence

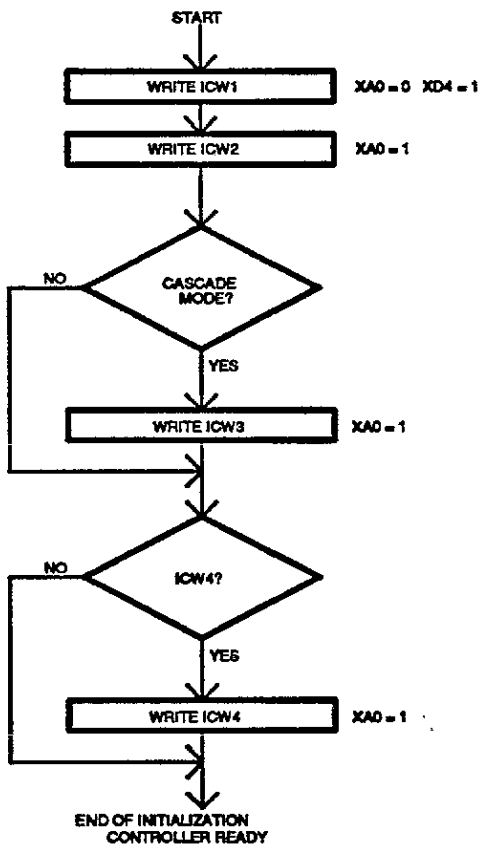
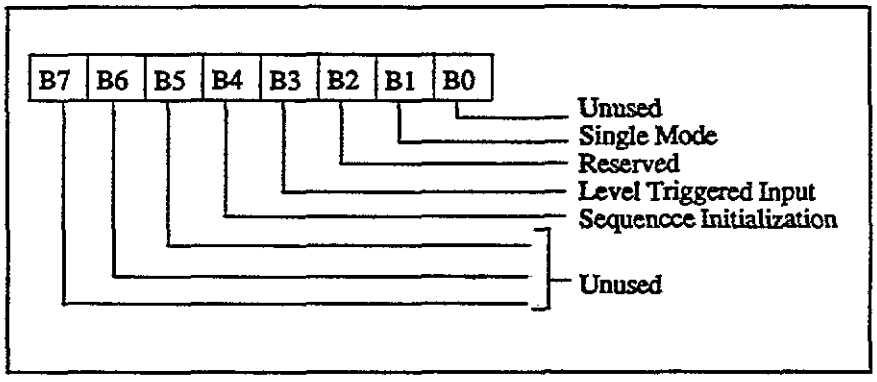


Figure 1-16. *ICW1 - Address 020H (0A0H) Write Only Register*

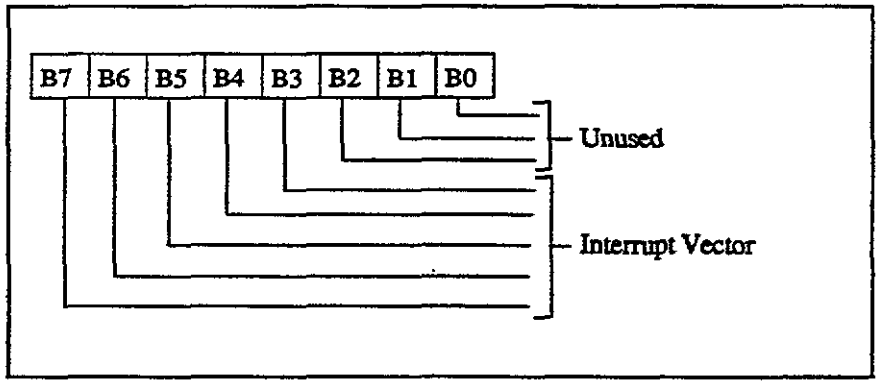


Single Mode: Bit 1 selects between Single Mode and Cascade Mode. Single Mode is used whenever only one interrupt controller (INTC1) is used. Cascade Mode allows two interrupt controllers to be connected through IR2 of INTC1. INTC1 allows INTC2 to generate its own interrupt vectors if Cascade Mode is selected, and the highest priority interrupt request pending is from an INTC2 input. INTC1 and INTC2 must be programmed for Cascade Mode for both devices to operate.

Level Triggered Input: Bit 3 selects level or edge triggered inputs to the IRR. If a 1 is written to LTM, a "high" level on the IRR input generates an interrupt request. Interrupt requests 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 deasserted early) and the interrupt request must be removed prior to EOI to prevent a second interrupt from occurring.

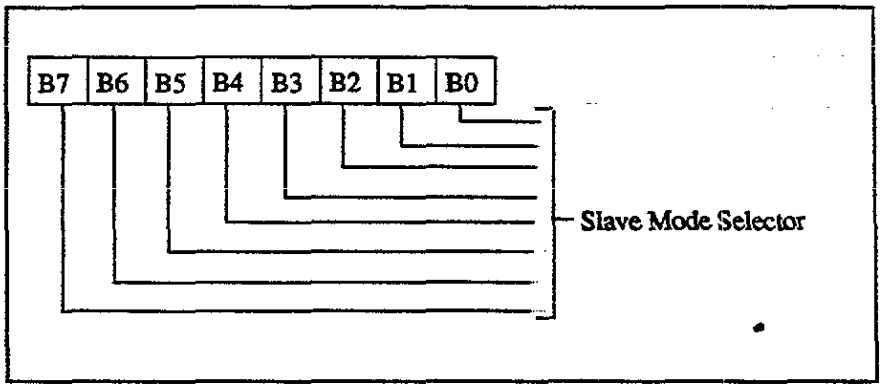
Sequence Initialization: Bit 4 indicates to the interrupt controller that an Initialization Sequence is starting and must be a 1 to write ICW1.

Figure 1-17. *ICW2 - Address 021H (0A1H) Write Only Register*



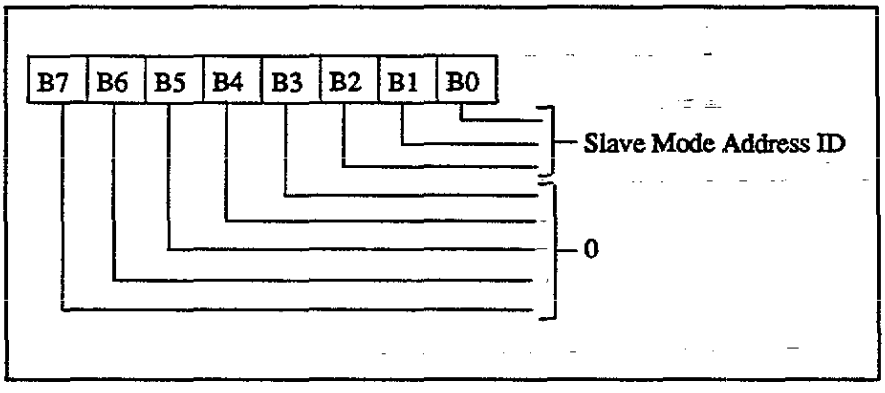
Interrupt Vector<3:7>: These bits are the upper 5 bits of the interrupt vector and are programmable by the CPU. The lower three bits of the vector are generated by the Priority Resolver during INTA (see Figure 1-18). INTC1 and INTC2 need not be programmed with the same value in ICW2.

Figure 1-18. *ICW3 Format for INTC1 - Address 021H Write Only Register*



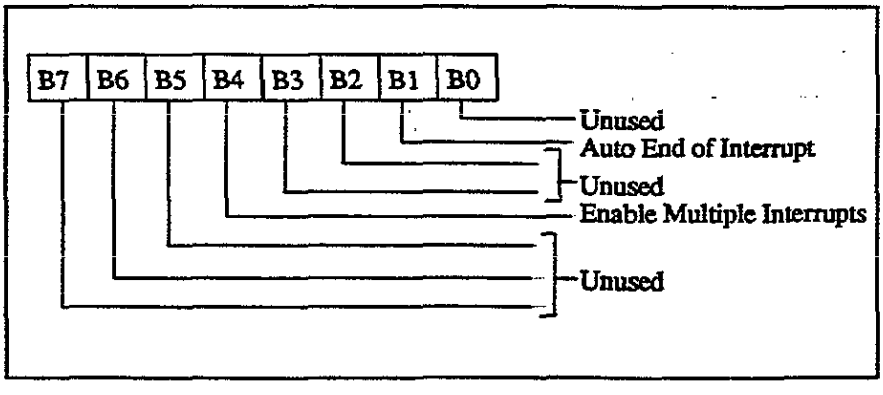
Slave Mode Selector <0:7>: Select which IR inputs have Slave Mode controllers connected. ICW3 in INTC1 must be written with a 04H for INTC2 to function.

Figure 1-19. *ICW3 Format for INTC2 - Address 0A1H*



ID0-ID2: Determine the Slave Mode address the controllers will respond to during the cascaded INTA sequence. ICW3 in INTC2 should be written with a 02H for cascade Mode operation. Note, bits <3:7> should be zeros.

Figure 1-20. *ICW4 for INTC2 - Address 0A1H Write Only Register*



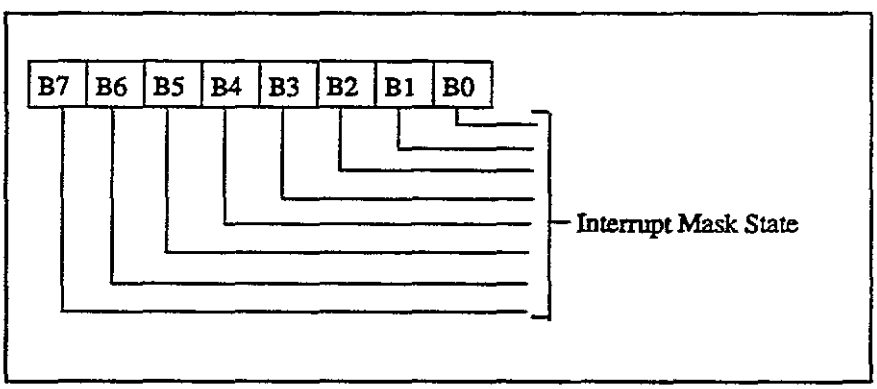
Auto End Of Interrupt: Auto End Of Interrupt is enabled when ICW4 is written with a zero in bit 1. The interrupt controller performs a non-specific EOI on the trailing edge of the second INTA cycle. Note, this function should not be used in a device with fully nested interrupts unless the device is a cascade Master.

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

Operational Command Words

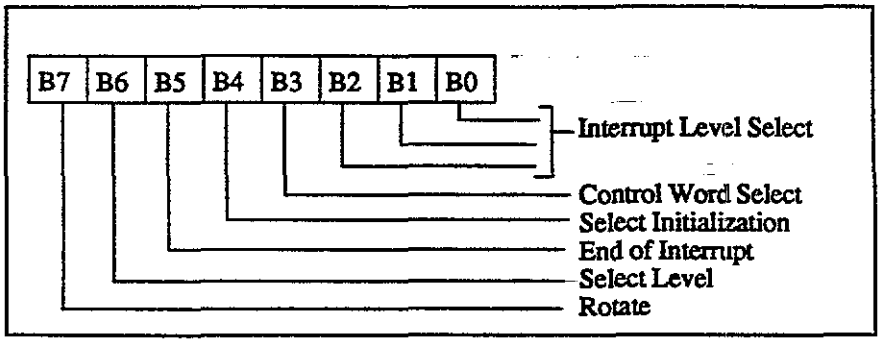
Operational Command Word 1 (OCW1) is located at address 021h (0A1h) and may be written any time the controller is in Initialization Mode. Operational Command Words 2 and 3 (OCW2, OCW3) are located at address 020H (0A0H). Writing to address 020H (0A0H) with a 0 in bit 4 places the controller in operational mode and loads OCW2 (if data bit 3 = 0), or OCW3 (if data bit 3 = 1).

Figure 1-21. *OCW1 - Address 021H (0A1H) Read/Write Register*



Interrupt Mask State<0:7>: These bits control the state of the Interrupt Mask Register. Each Interrupt Request can be masked by writing a 1 in the appropriate bit position (M0 controls IR0 etc.). Setting an IMR bit has no affect on lower priority requests. All IMR bits are cleared by writing ICW1.

Figure 1-22. *OCW2 - Address 020H (0A0H) Read/Write Register*



Interrupt Level Select<0:2>: These three bits are internally decoded to select which interrupt channel is to be affected by the Specific command. L<0:2> must be valid during three of the four specific cycles (see *Select Level* below).

Control Word Select: If the I/O write places a 0 in bit 4 (*Select Initialization*), then writing a 0 in bit 3 (*Control Word Select*) selects OCW2 and writing a 1 selects OCW3.

Select Initialization: Writing a 0 in this bit position takes the interrupt controller out of initialize mode and writes OCW2 or OCW3.

End of Interrupt: This bit, in conjunction with the Rotate and End of Interrupt, selects the operational function. Writing a 1 in this bit position causes a function related to End of Interrupt to occur.

Rotate	Select Level	End of Interrupt	Function
0	0	1	Non-specific End of Interrupt Command
0	1	1	Specific End of Interrupt Command
1	0	1	Rotate on non-specific End of Interrupt
1	1	1	Rotate on specific End of Interrupt

Select Level: This bit, in conjunction with Rotate and End of Interrupt, selects operational function. Writing a 1 in this bit position causes a specific or immediate function to occur. All specific commands require the Interrupt Channel Select 2-0 to be valid, except "No Operation."

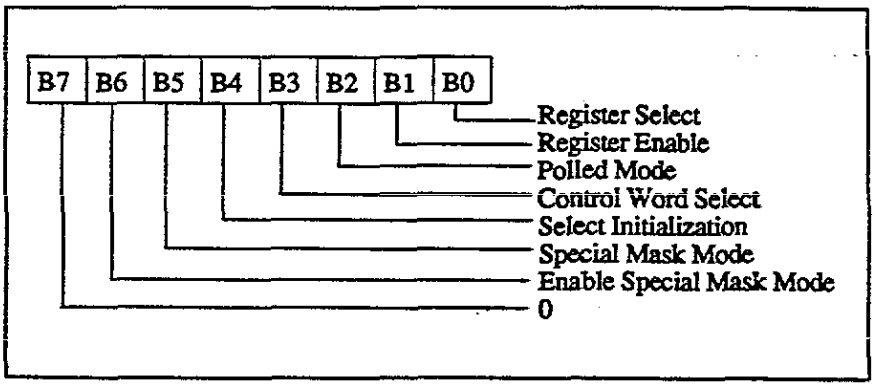
Rotate	Select Level	End of Interrupt	Function
0	1	0	No operation
0	1	1	Specific End of Interrupt Command
1	1	0	Specific Rotate Command
1	1	1	Rotate on Specific End of Interrupt

Rotate: This bit, in conjunction with Select Level and End of Interrupt, selects operational function. Writing a 1 in bit 7 causes one of the rotate functions to be selected.

Rotate	Select Level	End of Interrupt	Function
1	0	0	Rotate on auto EOI enable*
1	0	1	Rotate on non-specific EOI
1	1	0	Specific Rotate Command
1	1	1	Rotate on specific EOI

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

Figure 1-23. *OCW3 - Address 020H (0A0H) Write Only Register*



Register Select: This bit selects between the Interrupt Request Register and the In-Service Register during Status Read operations if Register Enable = 1.

Register Enable: When the Register Enable bit (bit 1) is 1, reading the Status Port at address 020H (0A0H) causes the contents of Interrupt Request Register or In-Service Register (determined by Register Select) to be placed on XD7-XD0. Asserting Polled Mode forces Register Enable to reset.

Polled Mode: Polled Mode is enabled by writing a 1 to bit 2 of -OCW causing the IPC to perform the equivalent of an INTA cycle during the next I/O read operation to the controller. The byte read during this cycle will have bit 7 set if an interrupt is pending. If bit 7 of the byte is set, the level of the highest pending request will be encoded on bits 2-0. The Interrupt Request Register remains frozen until the read cycle is completed; at which time the Polled Mode bit is reset.

Control Word Select: If the I/O write places a 0 in bit 4 (*Select Initialization*), then writing a 0 in bit 3 (*Control Word Select*) selects OCW2 and writing a 1 selects OCW3.

Select Initialization: Writing a 0 in this bit position takes the interrupt controller out of initialize mode and writes OCW2 or OCW3.

Special Mask: If Enable Special Mask Mode and Special Mask Mode are written with a 1, the Special Mask Mode is enabled. Writing a 1 to Enable Special Mask Mode, and a 0 to Special Mask Mode, disables Special Mask Mode. During Special Mask Mode, writing a 1 to any bit position inhibits interrupts and a 0 enables interrupts on the associated channel by causing the Priority Resolver to ignore the condition for the In-Service Register.

Enable Special Mask Mode: Writing a 1 in this bit position enables the Set/Reset Special Mask Mode function controlled by bit 5 (*Special Mask Mode*). Enable Special Mask Mode allows the other functions in OCW3 to be accessed and manipulated without affecting the Special Mask Mode state.

1.20 Counter/Timer Functional Description

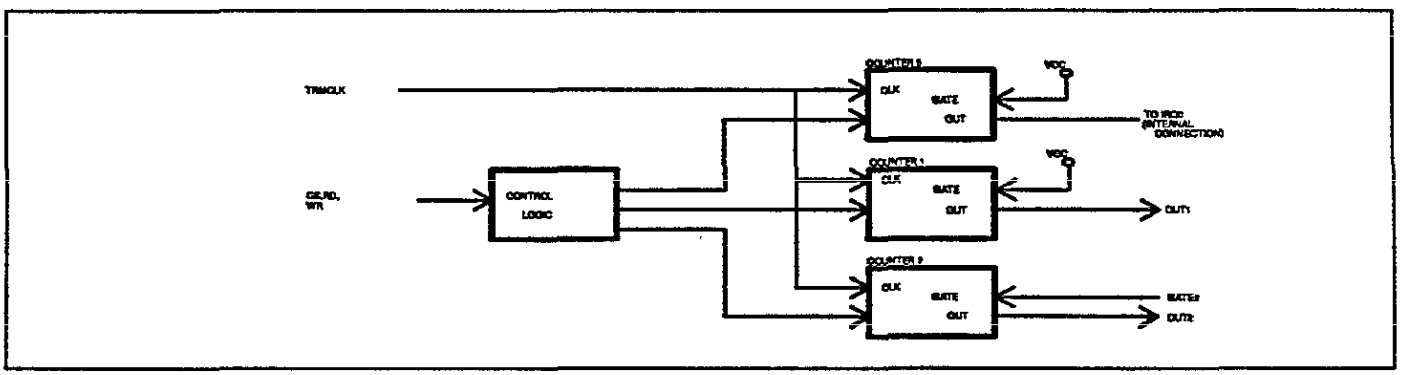
The Counter/Timer (CTC) in the 82C356 is general purpose, and can be used to generate accurate time delays under software control. The CTC contains 3

16-bit counters (Counter 0-3) which can be programmed to count in binary or binary coded decimal (BCD). Each counter operates independently of the other two and can be programmed for operation as a timer or a counter.

All three of the counters shown in Figure 1-24 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. Counters 0 and 1 can be programmed for all six modes, but Modes 1 and 5 have limited usefulness due to the lack of an external hardware trigger signal. Counter 2 can be operated in any of the six modes listed as follows:

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

Figure 1-24. Counter/Timer Block Timer



All three counters in the CTC are driven from a common clock input pin TMRCLK which is independent from other clock inputs to the IPC. Counter 0's output (Out 0) is connected to IR0 of INTC1 (see *Section 1.15 Interrupt Controller Functional Description*), and may be used as an interrupt to the system for time keeping and task switching. Counter 1 may be programmed to generate pulses or square waves for use by external devices. 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.

1.21 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,CIH), and a pair of 8-bit Counter Output Latches (COL, 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 externally accessible), and an OUT signal (OUT 0 is not externally accessible). The OUT signal 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, the type of command, and count format information.

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

The Counting Element (CE) is a loadable 16-bit synchronous down counter. CE is loaded or decremented on the falling edge of TMRCLK. CE contains the maximum count when a 0 is loaded; which is equivalent to 65536 in binary operation or 10000 in BCD. CE does not stop when it reaches 0. In Modes 2 and 3, the CE is reloaded and in all other modes it wraps around to FFFFH in binary operation or 9999 in BCD. 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 CE to be loaded or reloaded in one TMRCLK cycle.

The Counter Output Latches (COL, COH) are transparent latches which can be read while transparent or latched (see *Section 1.21 Counter Description*, subsection titled *Latch Counter Command*).

1.22 Programming the CTC

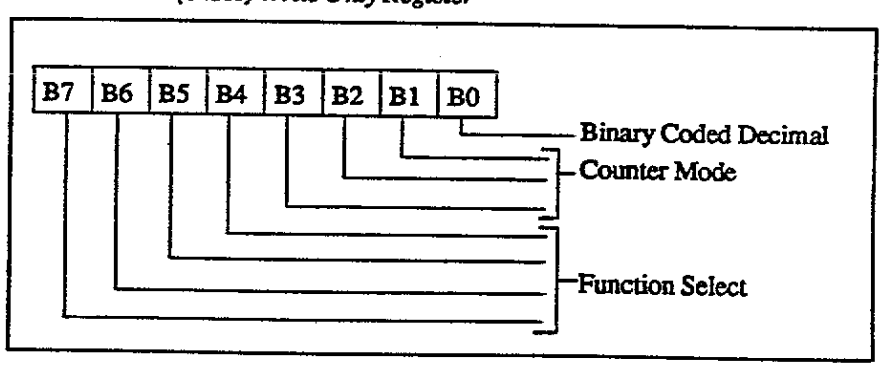
After power-Up the condition of CTC Control Registers, counter registers, CE and the output of all counters is undefined. Each counter must be programmed before it can be used.

Counters are programmed by writing a Control Word and then an initial count. The Control Register of a counter is written by writing to the Control Word address (see Figure 1-24). The Control Word is a write only location.

Table 1-18. *Counter/Timer Address Map*

Address	Function
040H	Counter 0 Read/Write
041H	Counter 1 Read/Write
042H	Counter 2 Read/Write
043H	Counter Register Write Only

Figure 1-25. *Control Word - (043H) Write Only Register*



Binary Coded Decimal: Bit 0 selects binary coded decimal counting format during Read/Write Counter Commands. When bit 0 is 0, then it is a binary count and when bit 0 is set to 1, it is a BCD count. Note, during Read-Back Command this bit must be 0.

Counter Mode<0:2>: Bits 1-3 determine the counter's mode during Read/Write Counter Commands (see Read/Write Counter Command) or select the counter during a read-back command (see Read-Back Command). Bits 1-3 become "don't care" during Latch Counter Commands.

Function Select <0:3>: Bits 4-7 determine the command to be performed as shown in the following table.

Table 1-19. Control Word 043H Bits 4-7

F3	F2	F1	F0	Command
0	0	0	0	Latch Counter 0 (see Counter Latch Command)
0	0	0	1	Read/Write Counter 0 LSB only
0	0	1	0	Read/Write Counter 0 MSB only
0	0	1	1	Read/Write Counter 0 LSB then MSB
0	1	0	0	Latch Counter 1 (see Counter Latch Command)
0	1	0	1	Read/Write Counter 1 LSB only
0	1	1	0	Read/Write Counter 1 MSB only
0	1	1	1	Read/Write Counter 1 LSB then MSB
1	0	0	0	Latch Counter 2 (see Counter Latch Command) 2
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 siginificant byte

Read/Write Counter Command

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

1. Each counters Control Word must be written before the initial count is written.
2. 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 and then most significant byte).

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

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

Table 1-20. M<0:2> Counter Command

M2	M1	M0	Function
0	0	0	Select Mode 0
0	0	1	Select Mode 1
X	1	0	Select Mode 2
X	1	1	Select Mode 3
1	1	0	Select Mode 4
1	1	1	Select Mode 5

Latch Counter Command

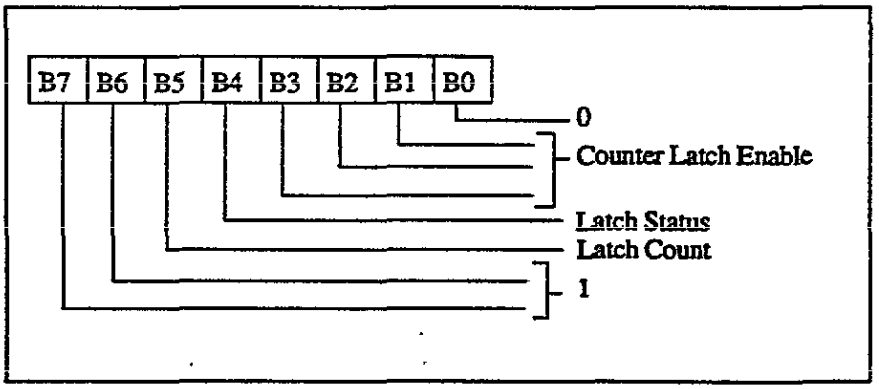
When a Latch Counter Command is issued, the counter's output latches COL and COH, latch the current state of the CE. COL and COH remain latched until read by the CPU, or the counter is reprogrammed. The output latches then return to a "transparent" condition. In this condition, the latches are enabled 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 is issued. Also, multiple Latch Counter Commands issued to the same counter without reading the counter causes all but the first command to be ignored.

Read-Back Command

The Read-Back Command allows the user to check the count value, Mode and state of the OUT signal and Null Count Flag of the selected counter(s). The format of the Read-Back Command is as follows:

Figure 1-26. Control Word - (043H) Write Only Register



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

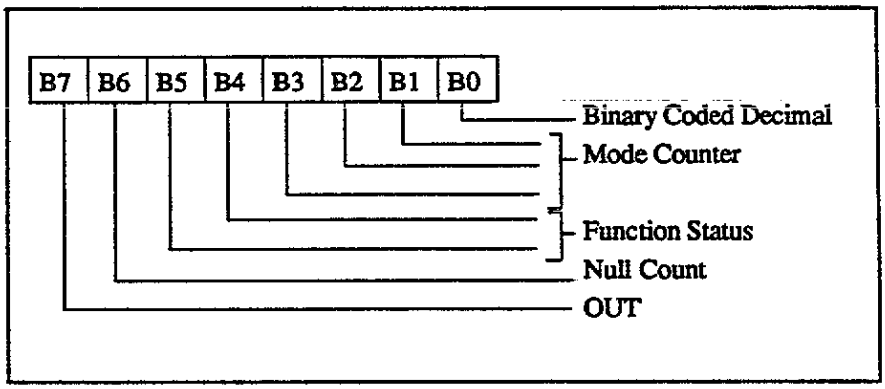
Each counter's latch remains latched until either the latch is read, or the counter is reprogrammed.

If Latch Status = Latch Count = 0, status will be 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.

Latch Status: Writing a 0 in bit 4 causes the selected counter(s) to latch the current condition of it's Control Register, Null Count and Output into the Status Register. The next read of the Counter results in the contents of the Status Register being read (see Status Read).

Latch Count: Writing a 0 in bit 5 causes the selected counter(s) to latch the state of the CE in COL and COH.

Figure 1-27. *Status Byte*



Binary Coded Decimal: Bit 0 indicates the CE is operating in BCD format.

Mode Counter <1:2>: These bits reflect the mode of the counter and are interpreted in the same manner as in Write Command operations.

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

Null Count: Bit 6 contains the condition of the Null count Flag. This flag is used to indicate that the contents of the CE are valid. NC will be set to a 1 during a write to the Control Register or the counter. NC is cleared to a 0 whenever the counter is loaded from the counter input registers.

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 will be used as the example in describing counter operation, but the description of Mode 0, 2, 3 and 4 is relevant to all counters.

The following terms are defined for describing CTC operation:

TMRCLK Pulse: A rising edge followed by a falling edge of the IPC TMRCLK input.

Trigger: The rising edge of the GATE2 input.

Counter Load: The transfer of the 16-bit value in CIL and CIH to the CE.

Initialized: A Control Word written and the Counter Input Latches loaded.

Counter 2 operates in one of the following modes.

- **Mode 0 - Interrupt on Terminal Count**

Writing the Control Word causes OUT2 to go low and remain low until the CE reaches 0, at which time it returns 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 is loaded. When both CIL and CIH are written, the CE is loaded after CIH is written (see Write Operations). This TMRCLK pulse does not decrement the count, so for an initial count of N, OUT2 doesn't go high until N+1 TMRCLK pulses after initialization. Writing a new initial count to the counter reloads the CE on the TMRCLK pulse and counting continues from the new count.

If an initial count is written with GATE2 = 0, it is still loaded on the next TMRCLK pulse, but counting does not begin until GATE2 = 1. Out2 therefore, 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 0. 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 retiggered.

- **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 1, 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 1 is illegal in Mode2.

Gate2 = 0 disables counting and forces OUT2 high immediately. A trigger reloads the CE on the TMRCLK pulse. Thus, GATE2 can be used to synchronize the counter to external events.

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

- **Mode 3 - Square Wave Generator**

Mode 3 is similar to Mode 2 in every respect except for the duty cycle of OUT2. OUT2 is set high initially and remains high for the first half of the count. When the first half of the initial count expires, OUT2 goes low for the remainder of the count.

If the counter is loaded with an even count, the duty cycle of OUT2 will be 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 high 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 will 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 will go 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 th new count. This allows the sequence to be "retiggerable" by software.

- **Mode 5 - Hardware Triggered Strobe**

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

CE is loaded during counting, the current counting sequence will not be affected unless a trigger occurs. A trigger causes the counter to be reloaded from CIL and CIH, making the counter "retiggerable".

- **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. Table 1-22 summerizes the gate functions for all modes.

Table 1-21. GATE Pin Function

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

1.23 Real Time Clock Functional Description

This section of the 82C356 combines a complete time-of-day clock with alarm, one hundred year calendar, a programmable period 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-up and power-down.

Register Access

Reading and writing to the 128 locations in the Real Time Clock is accomplished by first placing the Index Address of the location you wish to access on the data input pins XD0-XD6. The Index Address Register is then used as a pointer to the specific byte in the Real Time Clock, which may be read or written to by asserting -IOR or -IOW with an address on the XA<0:9> inputs of 071H.

Address Map

Table 1-22 illustrates the internal register/RAM organization of the Real Time Clock portion of the 82C356. The 128 addressable locations in the Real Time Clock are divided into 10 bytes which normally contains the time, calendar and alarm data, four control and status bytes and 114 general purpose RAM bytes. All 128 bytes are readable by the CPU. The CPU may also write to all locations except Registers C, D, Bit 7 of Register A and Bit 7 of the Seconds Byte which is always 0.

Table 1-22. Address Map for Real Time Clock

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

Time Calender and Alarm Bytes

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

The SET bit should then be cleared to allow updates. Once initialized and enabled, the Real Time Clock will perform Clock/Calendar updates at a 1 Hz rate.

Table 1-23 shows the format for the ten clock, calendar and alarm locations. The 24/12 bit Register determines whether the hour locations will be updated using a 1-12 or 0-23 format. After initialization the 24/12 bit cannot be changed without reinitializing the hour locations. In 12 hour format the high order bit of the hours byte in both the time and alarm bytes will indicate PM when it is a 1.

During updates, which occur once per second, the 10 bytes of time, calendar and alarm information are unavailable to be read or written by the CPU for a period of 2ms. These 10 locations cannot be written during this time. Information read while the Real Time Clock is performing updtes will be undefined. The Update Cycle section shows how to avoid Update Cycle /CPU contention problems.

The alarm bytes can be programmed to generate an interrupt at a specific time or they can be programmed to generate a periodic interrupt. To generate an interrupt at a specific time, the user need only program the time that the interrupt is to occur into the 3 alarm bytes. Alternately, a periodic interrupt can be generated by setting the high order two bits in an alarm register to a 1, which

turns that byte into a "don't care." For instance, an interrupt can be generated once a second by programming the same value into all three alarm registers.

Table 1-23. Real Time Clock Address Map

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

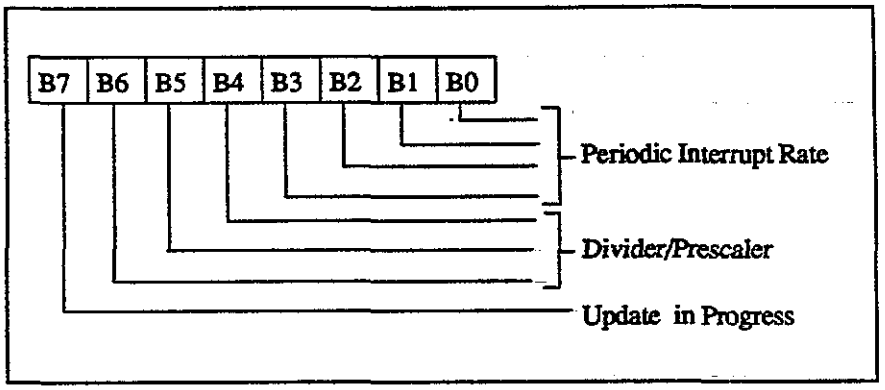
Static RAM

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

Control and Status Registers

The IPC contains four registers used to control the operation and monitor the status of the Real Time Clock. These Registers are located at Index Address 0AH-0DH and are accessible by the CPU at all times.

Figure 1-28. *Register A (0AH) Read/Write Register Except Update in Progress*



Periodic Interrupt Rate <0:3>: These four 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 do however, use the same interrupt channel in the Interrupt Controller. Use of the Periodic Interrupt allows the generation of interrupts at rates higher than once per second. Table 1-24 shows the interrupt rates for which the Real Time Clock can be programmed.

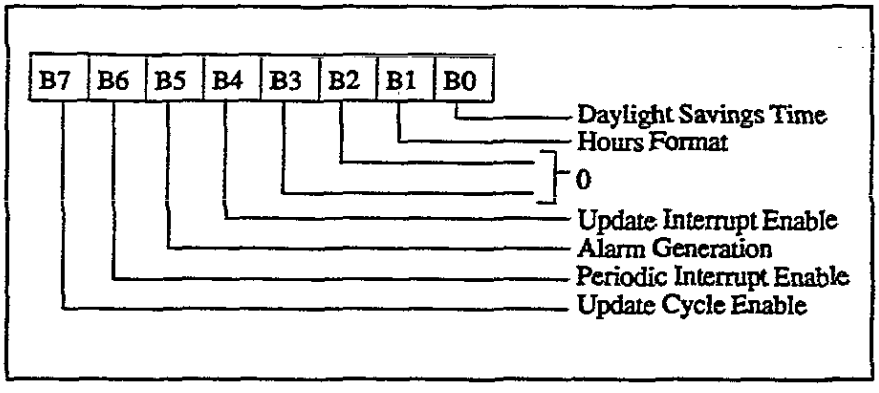
Divider/Prescaler <0:2>: These three bits are used to control the Divider/Prescaler on the Real Time clock. While the IPC can operate at frequencies higher than 32.768 Khz, this is not recommended for battery powered operation due to the increased power consumption at these higher frequencies.

Update in Progress: Update in progress flag is a status bit used to indicate when an update cycle is about to take place. A 1 indicates that an update cycle is taking place or is imminent. UIP will go active (High) 244us prior to the start of an update cycle and will remain active for an additional 2ms while the update is taking place. The UIP bit is read only and is not affected by Reset. Writing a 1 to the SET bit in register B will clear the UIP status bit.

Table 1-24. Periodic Interrupt

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

Figure 1-29. Register B (0BH) Read/Write Register



Daylight Savings Time: The Real Time Clock can be instructed to handle daylight savings time changes by setting this bit to a 1. This enables two exceptions to the normal time keeping sequence to occur on the last Sunday in April AM. Setting this bit to a 0 disables the execution of these two exceptions. -PSRSTC has no affect on this bit.

Hours Format: This control bit is used to establish the format of both the Hours and Hours Alarm bytes. If this bit is a 1, the Real Time Clock will interpret and update the information in these two bytes using the 24 hour mode. This bit can be read or written by the CPU and is not affected by Reset.

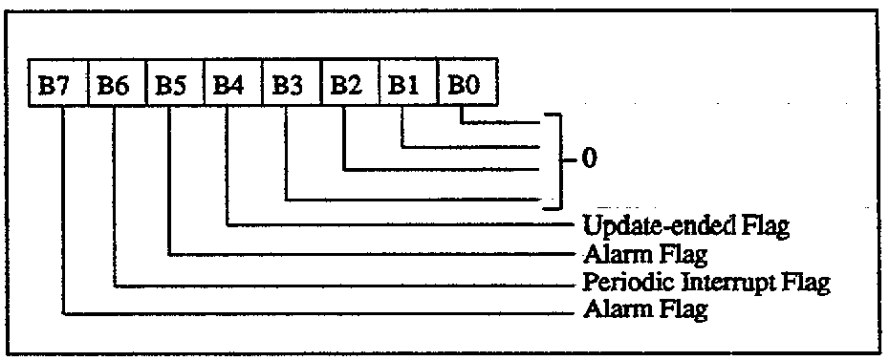
Update Interrupt Enable: Update-ended Interrupt Enable bit is a read/write bit which enables the Update-ended Interrupt Flag (UF) bit in Register C to assert IRQ. The Update-ended Interrupt Enable is cleared by RESET or by setting the Update Cycle Enable bit.

Alarm Generation: The generation of alarm interrupts is enabled by setting this bit to a 1. Once this bit is enabled the Real Time Clock will generate 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 by Reset.

Periodic Interrupt Enable: The Periodic Interrupt Enable Bit controls the generation of interrupts based on the value programmed into the RS3-RS0 bits of Register A. This allows the user to disable this function without affecting the programmed rate. Writing a 1 to this bit enables the generation of periodic interrupts. This bit is cleared to a 0 by Reset.

Update Cycle Enable: Writing a 0 to this bit enables the Update Cycle and allows the Real Time Clock to function normally. When set to a 1, the Update Cycle is inhibited and any cycle in progress is aborted. This bit is not affected by the RESET input pin.

Figure 1-30. *Register C (0CH) Read Only Register*



Update-ended Flag: Update-ended Interrupt Flag bit is set after each update cycle. When the Update-ended Interrupt Enable bit is a 1, the 1 in Update-ended Interrupt Flag (UF) causes the IRQF bit to be a 1, asserting IRQ. The Update-ended Interrupt Flag (UF) is cleared by a Register C read or a RESET.

Alarm Flag: A 1 appears in the AF bit when ever a match has occurred between the time register and alarm registers during an update cycle. This flag is also independent of it's enable (AIE) and will generate an interrupt if AIE is true.

Periodic Interrupt Flag: The Period Interrupt Flag is set to a 1 when a transition, which is selected by RS3-RS0, occurs in the divider chain. This bit will become active, independent of the condition of the PIE control bit. The PF bit will then generate an interrupt and set IRQF if PIE is a 1.

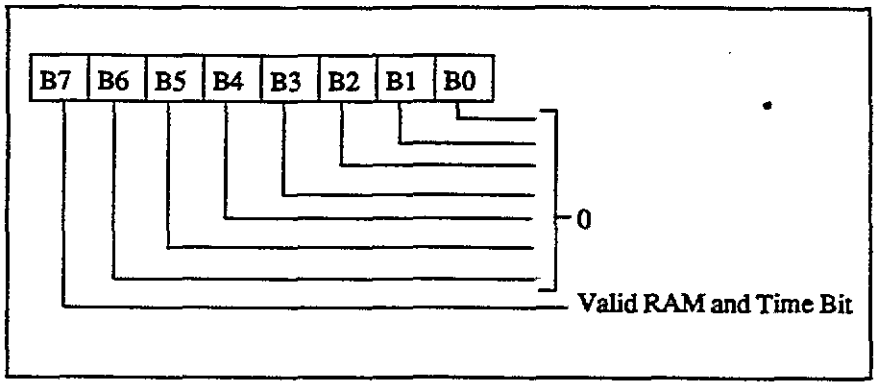
Interrupt Request Flag: The Interrupt Request Flag is set to a 1 when any of the conditions which can cause an interrupt are true and the interrupt enable for that condition is true. The condition which causes this bit to be set, also generates an interrupt. The logic expression for this flag is:

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

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

Valid RAM and Time Bit: The Valid RAM and Time Bit indicates the condition of the contents of the Real Time Clock. This bit is cleared to a 0 whenever the PS input pin is LOW. This pin is normally derived from the

Figure 1-31. Register B (0BH) Read/Write Register



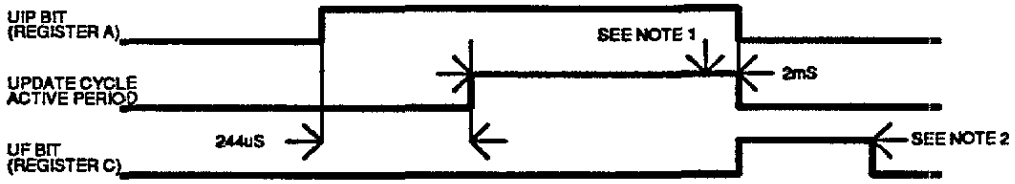
power supply which supplies Vcc to the device and will allow the user to determine whether the registers have been initialized since power was applied to the device. PSRSTB has no effect on this bit and it can only be set by reading Register D. All unused register bits will be a 0 when read and are not writeable.

Update Cycle

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

During the time that an update is taking place, the lower 10 registers are unavailable to the CPU. This is done to prevent the possible corruption of data in the registers or reading invalid data. To avoid contention problems between the Real Time Clock and the CPU, a flag is provided in Register A to alert the user of an impending update cycle. This Update In Process Bit (UIP) is asserted 244us before the actual start of the cycle and is maintained until the cycle is complete. Once the cycle is complete the UIP bit will be cleared and the Update Flag (UF) in Register C will be set. Figure 1-32 illustrates the update cycle. CPU access is always allowed to Register A through D during update cycles.

Figure 1-32. Update Cycle



Note Registers 0-9 are unavailable to be read or written during this time. UF bit is cleared by the CPU read of register C.

Two methods for reading and writing to the Real Time Clock are recommended. Both of these methods allows 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 and, if it is 0, perform the read or write operation. For this method to work successfully, the entire read or write operation (including any interrupt service routines which might occur) must not require longer than 244us to complete from the beginning of the read of Register A to the completion of the last read or write operation to the Clock Calendar Registers.

The second method of accessing the lower 10 registers is to read Register C once and disregard the contents, then subsequently continue reading this register until the UF bit is a 1. This bit becomes true immediately after an update has been completed. The user then has until the start of the next update cycle to complete a read or write operation.

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 IPC. This alternate source of power is normally provided by connecting a battery to the Vcc supply pin of the device. A means should be provided to switch from the system power supply to the battery. A circuit, such as the one shown in Figure 1-33, may be used to eliminaite power drain on the battery when the entire IPC is active. The circuit shown here allows for reliable transitions between system and battery power without undue battery power drain.

Section 2

356 Configuration Registers

2.1 Index Registers

Index 26H

RTC, NMI & Coprocessor Reset Register (READ/WRITE)
Index Register Address: 022H
Data Register Address: 023H

Index	Bits	Values and Functions
26H	7	Real Time Clock selection.
		(0): Selects internal RTC.
		1: Selects external RTC.
	6	Reserved (default = 0)
	5	Power Fail warning active during last NMI(R/O).
		(0): Power Fail warning pin not active.
		1: Power Fail warning pin was active.
	4	Power Fail Warning Enable.
		(0): PCU NMI disabled.
		1: PCU NMI enabled.
	3	PCU NMI active level.
		(0): PCU NMI disabled.
		1: PCU NMI enabled.
	2	Reserved (default = 0) (see *82C351 CPU/Cache/DRAM/ Controller, Section 2: 351 Configuration Registers*).
	1	Extended IO Decode Enable.
		(0): Disable extended I/O decode.
		1: Enable extended I/O decode.
	0	Reserved (default = 0)

Index 01H *Configuration Register (READ/WRITE)*

Index	Bits	Values and Functions
01H	7:6	Read/Write Cycle Wait States.
		00: 1 Wait State.
		01: 2 Wait States.
		10: 3 Wait States.
		(11): 4 Wait States.
	5:4	16-bit DMA Cycle Wait States.
		(00): 1 Wait State.
		01: 2 Wait States.
		10: 3 Wait States.
		11: 4 Wait States.
	3:2	8-bit DMA Cycle Wait States.
		(00): 1 Wait State.
		01: 2 Wait States.
		10: 3 Wait States.
		11: 4 Wait States.
	1	Extended DMA Memory Read enable.
		(0) Delay -MEMR by one clock cycle later than -IOW.
		1: Start -MEMR at the same time as -IOW.
	0	DMA source clock.
		(0): DMA Clock is equal to BUSCLK.
		1: DMA Clock is equal to BUSCLK.

2.2 I/O Port Registers

I/O Port 00H *DMA1 Channel 0 Address Register (READ/WRITE/FLIP-FLOP)*

Addr	Bits	-IOR	-IOW	FF	Values and Functions
00H	7:0	0	1	0	Read current low byte address.
		0	1	1	Read current high byte address.
		1	0	0	Write base & current low byte address.
		1	0	1	Write base & current high byte address.

I/O Port 01H *DMA1 Channel 0 Word Count Register (READ/WRITE/FLIP-FLOP)*

Addr	Bits	-IOR	-IOW	FF	Values and Functions
01H	7:0	0	1	0	Read current word count low byte.
		0	1	1	Read current word count high byte.
		1	0	0	Write base & current word count low byte.
		1	0	1	Write base & current word count high byte.

I/O Port 0AH DMA1 Single Bit Request Mask Register (WRITE ONLY)

	Bits	Values and Functions
0AH	7:3	(XXXXX): Don't Cares.
	2	Mask bit.
		(0): Clear Mask bit
		1: Set Mask bit
	1:0	DMA Channel Mask bit Selection.
		(00): DMA Channel 0
		01: DMA Channel 1
		10: DMA Channel 2
		11: DMA Channel 3

I/O Port 0BH DMA1 Mode Register (READ /WRITE)

	Bits	Values and Functions
0BH	7:6	Channel Mode type.
		(00): Demand Mode
		01: Single Cycle Mode
		10: Block Mode
		11: Cascade Mode
	5	Address counting direction.
		(0): Increment address
		1: Decrement address
	4	Auto-initialization enable.
		(0): Auto-initialization function disable
		1: Auto-initialization function enable
	3:2	Type of transfer.
		(00): Verify transfer
		01: Write transfer
		10: Read transfer
		11: Illegal
	1:0	DMA Channel Mode Selection.
		(00): DMA Channel 0
		01: DMA Channel 1
		10: DMA Channel 2
		11: DMA Channel 3

I/O Port 0CH DMA1 Set Byte Pointer Flip-Flop (READ ONLY)

	Bits	Values and Functions
0CH	7:0	Bits 7:0 are don't cares. The Set Byte Pointer Flip-Flop Clear command is activated as a result of the address 00CH access and the assertion of -IOR. This command allows the CPU to adjust the pointer to the high byte of an address or word count register.

I/O Port 0CH DMA1 Clear Byte Pointer Flip-Flop (WRITE ONLY)

	Bits	Values and Functions
0CH	7:0	Bits 7:0 are don't cares. The Clear Byte Pointer Flip-Flop Clear command is activated as a result of the address 00CH access and the assertion of -IOW. This command allows the CPU to adjust the pointer to the low byte of an address or word count register.

I/O Port 0DH DMA1 Temporary Register (READ ONLY)

	Bits	Values and Functions
0DH	7:0	Bits 7:0 contains values of XD7:XD0 during the first cycle of a memory-to-memory transfer. Data from the last memory-to-memory transfer will remain in the register unless a RESET or Master Clear occurs.

I/O Port 0DH DMA1 Master Clear (WRITE COMMAND ONLY)

	Bits	Values and Functions
0DH	7:0	Bits 7:0 are don't cares. The Master Clear command is activated as a result of the address 00EH access and the assertion of -IOW. This command has the same effect as a hardware RESET. The Command Register, Status Register, Request Register, and Byte Pointer Flip-Flop are cleared and the Request Mask Register is set.

I/O Port 0EH DMA1 Clear Mode Register Counter (READ COMMAND ONLY)

	Bits	Values and Functions
0EH	7:0	Bits 7:0 are don't cares. The Clear Mode Register Counter command is activated as a result of the address 00EH access and the assertion of -IOR. This command is provided to allow the CPU to restart the mode read process at a known point. 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 registers will be read is Channel 0 first, Channel 3 last.

I/O Port 0EH DMA1 Clear Request Mask Bits (WRITE COMMAND ONLY)

	Bits	Values and Functions
0EH	7:0	Bits 7:0 are don't cares. The Clear Mask Register command is activated as a result of the address 00EH access and the assertion of -IOW. This command enables all four DMA channels to accept requests by clearing the mask bits in the register.

I/O Port 0FH DMA1 Request Mask Register Bits (READ/WRITE)

	Bits	Values and Functions
0FH	7:4	(XXXX): Don't Cares.
	3	Channel 0 Mask bit.
		0: No bit mask, channel enabled
		(1): Bit mask, channel disabled
	2	Channel 1 Mask bit.
		0: No bit mask, channel enabled
		(1): Bit mask, channel disabled
	1	Channel 2 Mask bit.
		0: No bit mask, channel enabled
		(1): Bit mask, channel disabled
	0	Channel 3 Mask bit
		0: No bit mask, channel enabled
		(1): Bit mask, channel disabled

I/O Port 20H *INTC1 Initialization Command Word #1 (1ST WRITE ONLY)*

	Bits	Values and Functions
20H	7:5	(XXX): Don't Cares.
	4	Start Initialization Sequence.
		(0): Do not begin initialization sequence
		1: Begin sequence and write to ICW1
	3	Interrupt Level or Edge Trigger Select.
		0: Edge Trigger enabled; low to high transitions
		1: Level Trigger enabled; active high
	2	(X): Don't Care.
	1	Single or Cascade Mode Select.
		(0): Cascade Mode enabled; Cascade Mode allows the second interrupt controller, INTC2 to be connected through IR2 of INTC1. INTC1 will allow INTC2 to generate its own interrupt vectors if Cascade Mode is selected and the highest priority IR pending is from an INTC2 input. INTC1 and INTC2 must be programmed for Cascade Mode for both devices to operate.
		1: Single Mode Enabled; Single Mode is used whenever only one interrupt controller is used.
	0	(X): Don't Care.

I/O Port 21H *INTC1 Initialization Command Word #2 (2ND WRITE ONLY)*

	Bits	Values and Functions
21H	7:3	Upper five bits of interrupt vector byte. These bits are programmable by the CPU. The lower three bits of the vector are generated by the Priority Resolver during interrupt acknowledge.

I/O Port 21H *INTC1 Initialization Command Word #3 (3RD WRITE ONLY)*

	Bits	Values and Functions
21H	7	Slave Mode Controller connected to IR7 select.
		(0): No slave Mode Controller connected to IR7.
		1: Slave Mode Controller connected to IR7.
	6	Slave Mode Controller connected to IR6 select.
		(0): No Slave Mode Controller connected to IR6.
		1: Slave Mode Controller connected to IR6.
	5	Slave Mode Controller connected to IR5 select.
		(0): No Slave Mode Controller connected to IR5.
		1: Slave Mode Controller connected to IR5.
	4	Slave Mode Controller connected to IR4 select.
		(0): No Slave Mode Controller connected to IR4.
		1: Slave Mode Controller connected to IR4.
	3	Slave Mode Controller connected to IR3 select.
		(0): No Slave Mode Controller connected to IR3.
		1: Slave Mode Controller connected to IR3.
	2	Slave Mode Controller connected to IR2 select.
		(0): No Slave Mode Controller connected to IR2.
		1: Slave Mode Controller connected to IR2.
	1	Slave Mode Controller connected to IR1 select.
		(0): No Slave Mode Controller connected to IR1.
		1: Slave Mode Controller connected to IR1.
	0	Slave Mode Controller connected to IR0 select.
		(0): No Slave Mode Controller connected to IR0.
		1: Slave Mode Controller connected to IR0.

Note ICW3 in INTC1 must be written with a 04H for INTC2 to function.

I/O Port 21H *INTC1 Initialization Command Word #4 (4TH WRITE ONLY)*

	Bits	Values and Functions
21H	7:5	(XXX): Don't Cares.
	4	Enable 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.
	3:2	(XX): Don't Cares.
	1	Auto End of Interrupt enable. This bit is active low. The interrupt controller will perform a nonspecific EOI on the trailing edge of the second INTA cycle.
	0	(X): Don't Care.

I/O Port 21H *INTC1 Operational Command Word #1 (READ/WRITE)*

	Bits	Values and Functions
21H	7	IR7 Mask bit.
		(0): No mask.
		1: Mask IR7.
	6	IR6 Mask bit.
		(0): No mask.
		1: Mask IR6.
	5	IR5 Mask bit.
		(0): No mask.
		1: Mask IR5.
	4	IR4 Mask bit.
		(0): No mask.
		1: Mask IR4.
	3	IR3 Mask bit.
		(0): No mask.
		1: Mask IR3.
	2	IR2 Mask bit.
		(0): No mask.
		1: Mask IR2.
	1	IR1 Mask bit.
		(0): No mask.
		1: Mask IR1.
	0	IR0 Mask bit.
		(0): No mask.
		1: Mask IR0.

I/O Port 20H *INTC1 Operational Command Word #2 (2ND WRITE ONLY)*

	Bits	Values and Functions	
20H	7	Rotate function select. Bit 7 is used in conjunction with bits 6 and 5 to select operational function.	
B7	**B6**	**B5**	**Function**
1	0	0	Rotate on auto EOI enable.
1	0	1	Rotate on non-specific EOI.
1	1	0	Specific Rotate Command.
1	1	1	Rotate on specific EOI.
	6	Specific or immediate function select. Bit 6 is used in conjunction with bits 7 and 5 to select operational function.	
B7	**B6**	**B5**	**Function**
1	0	0	No operation.
1	0	1	Specific EOI Command.
1	1	0	Specific Rotate Command.
1	1	1	Rotate on specific EOI.
	5	Function related to EOI select. Bit 5 is used in conjunction with bits 7 and 6 to select operational function.	
B7	**B6**	**B5**	**Function**
1	0	0	Non-specific EOI Command.
1	0	1	Specific EOI Command.
1	1	0	Rotate on non-specific EOI.
1	1	1	Rotate on specific EOI.
	4	Interrupt controller out of initialize mode select.	
		0: Disable initialization mode and write OCW2 and OCW3.	
		1: Enable initialization mode.	
	3	Operational Command Word Select.	
		0: OCW2 selected.	
		1: OCW3 selected.	
	2:0	Bits 2:0 are internally decoded to select which interrupt channel is to be affected by the Specific Command.	
		000: IR7.	
		001: IR6.	
		010: IR5.	
		011: IR4.	
		100: IR3.	
		101: IR2.	
		110: IR1.	
		111: IR0.	

I/O Port 20H *INTC1 Operational Command Word #3 (3RD WRITE ONLY)*

	Bits	Values and Functions
20H	7	Bit 7 must be set to 0 for proper operation.
		(0): Necessary value.
	6	Enable Special Mask Mode.
		(0): Special Mask Mode bit becomes a son't care.
		1: Enable Special Mask Mode bit.
	5	Special Mask Mode.
		(0): Reset Special Mask.
		1: Set Special Mask.
	4	Interrupt controller out of initialize mode select.
		0: Disable initialization mode and write OCW2 and OCW3.
		1: Enable initialization mode.
	3	Operational Command Word Select.
		0: OCW2 selected.
		1: OCW3 selected.
	2	Enable Poll Command.
		(0): No Poll Command.
		1: Poll Command.
	1	Enable Read Register Command.
		(0): Read Register bit becomes a don't care.
		1: Enable Read Register Command bit.
	0	Read Register Command.
		(0): Read Interrupt Request Register on next -IOR pulse.
		1: Read In-Service Register on next -IOR pulse.

I/O Port 40H *Counter/Timer Counter 0 (READ/WRITE)*

	Bits	Values and Functions
40H	7:0	Counter 0 count register (Read/Write).

I/O Port 41H *Counter/Timer Counter 1 (READ/WRITE)*

	Bits	Values and Functions
41H	7:0	Counter 1 count register (Read/Write).

I/O Port 42H *Counter/Timer Counter 2 (READ/WRITE)*

	Bits	Values and Functions
42H	7:0	Counter 2 count register (Read/Write).

I/O Port 43H *Counter/Timer Control Word Register (WRITE ONLY)*

	Bits	Values and Functions
43H	7:4	Determine command to be performed.
		0000 Latch counter 0
		0001 Read/Write counter 0 LSB only.
		0010 Read/Write counter 0 MSB only.
		0011 Read/Write counter 0 LSB then MSB.
		0100 Latch counter 1.
		0101 Read/Write counter 1 LSB only.
		0110 Read/Write counter 1 MSB only.
		0111 Read/Write counter 1 LSB then MSB.
		1000 Latch counter 2.
		1001 Read/Write counter 2 LSB only.
		1010 Read/Write counter 2 MSB only.
		1011 Read/Write counter 2 LSB then MSB.
		11xx Read-Back command.
	3:1	Determine the Counter's mode
		000 Select mode 0.
		001 Select mode 1.
		x10 Select mode 2.
		x11 Select mode 3.
		110 Select mode 4.
		111 Select mode 5.
	0	Determine the Count type
		0 Binary count.
		1 Binary coded decimal count.

I/O Port 70H *Index Register Address*

	Bits	Values and Functions
70H	7:0	Index register address. The index value is placed in port 70H to access a particular register. The valid address range for the 82C356 is 0 thru 7FH.

I/O Port 71H *Data Register Address*

	Bits	Values and Functions
71H	7:0	Data register address. The index value is placed in port 70H to access a particular register and the data to be read from or written to that register is placed in port 71H.

I/O Port 80H *Unused*

	Bits	Values and Functions
80H	7:0	Not defined.

I/O Port 81H *8-bit DMA Channel 2 (READ/WRITE)*

	Bits	Values and Functions
81H	7:0	Address bits A16-A23 during 8-bit DMA (DACK2) cycles.

I/O Port 82H *8-bit DMA Channel 3 (READ/WRITE)*

	Bits	Values and Functions
82H	7:0	Address bits A16-A23 during 8-bit DMA(DACK3) cycles.

I/O Port 83H *8-bit DMA Channel 1 (READ/WRITE)*

	Bits	Values and Functions
83H	7:0	Address bits A16-A23 during 8-bit DMA(DACK1) cycles.

I/O Port 84H *Unused*

	Bits	Values and Functions
84H	7:0	Not defined.

I/O Port 85H *Unused*

	Bits	Values and Functions
85H	7:0	Not defined.

I/O Port 86H *Unused*

	Bits	Values and Functions
86H	7:0	Not defined.

I/O Port 87H *8-bit DMA Channel 0 (READ/WRITE)*

	Bits	Values and Functions
87H	7:0	Address bits A16-23 during 8-bit DMA(DACK0) cycles.

I/O Port 88H *Unused*

	Bits	Values and Functions
88H	7:0	Not defined.

I/O Port 89H *16-bit DMA Channel 2 (READ/WRITE)*

	Bits	Values and Functions
89H	7:0	Address bits A17-23 during 16-bit DMA(DACK6) cycles.

I/O Port 8AH *16-bit DMA Channel 3 (READ/WRITE)*

	Bits	Values and Functions
8AH	7:0	Address bits A17-23 during 16-bit DMA(DACK7) cycles.

I/O Port 8BH *16-bit DMA Channel 1 (READ/WRITE)*

	Bits	Values and Functions
8BH	7:0	Address bits A17-23 during 16-bit DMA(DACK5) cycles.

I/O Port 8CH *Unused*

	Bits	Values and Functions
8CH	7:0	Not defined.

I/O Port 8DH *Unused*

	Bits	Values and Functions
8DH	7:0	Not defined.

I/O Port 8EH *Unused*

	Bits	Values and Functions
8EH	7:0	Not defined.

I/O Port A0H *INTC2 Initialization Command Word #1 (1ST WRITE ONLY)*

	Bits	Values and Functions
A0H	7:5	(XXX): Don't Cares.
	4	Start Initialization Sequence.
		(0): Do not begin initialization sequence.
		1: Begin sequence and write to ICW1.
	3	Interrupt Level or Edge Trigger Select.
		0: Edge Trigger enabled; Low to high transitions.
		1: Level Trigger enabled; Active high.
	2	(X): Don't Care.

I/O Port A1H *INTC2 Initialization Command Word #2 (2ND WRITE ONLY)*

	Bits	Values and Functions
A1H	7:3	Upper five bits of interrupt vector byte. These bits are programmable by the CPU. The lower three bits of the vector are generated by the Priority Resolver during interrupt acknowledge.

I/O Port A1H *INTC2 Initialization Command Word #3 (3RD WRITE ONLY)*

	Bits	Values and Functions
A1H	7:3	Bits 7 to 3 should be zero.
	2:0	Bits 2 to 0 determines the Slave Mode address the controller will respond to during the cascaded INTA sequence.

Note: ICW3 in INTC2 should be written with a 02H for Cascade Mode operations.

I/O Port A1H *INTC2 Initialization Command Word #4 (4TH WRITE ONLY)*

	Bits	Values and Functions
A1H	7:5	(XXX): Don't Cares.
	4	Enable 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
	3:2	(XX): Don't Cares.
	1	Auto End of Interrupt enable. This bit is active low. The interrupt controller will perform a nonspecific EOI on the trailing edge of the second INTA cycle.
	0	(X): Don't Care.

I/O Port A1H *INTC2 Operational Command Word #1 (READ/WRITE)*

	Bits	Values and Functions
A1H	7	IR7 Mask bit.
		(0): No mask.
		1: Mask IR7.
	6	IR6 Mask bit.
		(0): No mask.
		1: Mask IR6.
	5	IR5 Mask bit.
		(0): No mask.
		1: Mask IR5.
	4	IR4 Mask bit.
		(0): No mask.
		1: Mask IR4.
	3	IR3 Mask bit.
		(0): No mask.
		1: Mask IR3.
	2	IR2 Mask bit.
		(0): No mask.
		1: Mask IR2.
	1	IR1 Mask bit.
		(0): No mask.
		1: Mask IR1.
	0	IR0 Mask bit.
		(0): No mask.
		1: Mask IR0.

I/O Port A0H *INTC2 Operational Command Word #2 (2ND WRITE ONLY)*

	Bits	Values and Functions	
A0H	7	Rotate function select. Bit 7 is used in conjunction with bits 6 and 5 to select operational function.	
B7	**B6**	**B5**	**Function**
1	0	0	Rotate on auto EOI enable.
1	0	1	Rotate on non-specific EOI.
1	1	0	Specific Rotate Command.
1	1	1	Rotate on specific EOI.
	6	Specific or immediate function select. Bit 6 is used in conjunction with bits 7 and 5 to select operational function.	
B7	**B6**	**B5**	**Function**
0	1	0	No operation.
0	1	1	Specific EOI Command.
1	1	0	Specific Rotate Command.
1	1	1	Rotate on specific EOI.
	5	Function related to EOI select. Bit 5 is used in conjunction with bits 7 and 6 to select operational function.	
B7	**B6**	**B5**	**Function**
0	1	0	Non-specific EOI Command.
0	1	1	Specific EOI Command.
1	1	0	Rotate on non-specific EOI.
1	1	1	Rotate on specific EOI.
	4	Interrupt controller out of initialize mode select.	
		0: Disable initialization mode and write OCW2 and OCW3.	
		1: Enable initialization mode.	
	3	Operational Command Word Select.	
		0: OCW2 selected.	
		1: OCW3 selected.	
	2:0	Bits 2:0 are internally decoded to select which interrupt channel is to be affected by the Specific Command.	
		000: IR7.	
		001: IR6.	
		010: IR5.	
		011: IR4.	
		100: IR3.	
		101: IR2.	
		110: IR1.	
		111: IR0.	

I/O Port A0H INTC2 Operational Command Word #3 (3RD WRITE ONLY)

	Bits	Values and Functions
A0H	7	Bit 7 must be set to 0 for proper operation.
		(0): Necessary value.
	6	Enable Special Mask Mode.
		(0): Special Mask Mode bit becomes a don't care.
		1: Enable Special Mask Mode bit.
	5	Special Mask Mode.
		(0): Reset Special Mask.
		1: Set Special Mask.
	4	Interrupt controller out of initialize mode select.
		0: Disable initialization mode and write OCW2 and OCW3.
		1: Enable initialization mode.
	3	Operational Command Word Select.
		0: OCW2 selected.
		1: OCW3 selected.
	2	Enable Poll Command.
		(0): No Poll Command.
		1: Poll Command.
	1	Enable Read Register Command.
		(0): Read Register bit becomes a don't care.
		1: Enable Read Read Register Command bit.
	0	Read Register Command.
		(0): Read Interrupt Request Register on next RD- pulse.
		1: Read In-Service Register on next RD- pulse.

I/O Port C0H DMA2 Channel 0 Address Register (READ/WRITE/FLIP-FLOP)

Addr	Bits	-IOR	-IOW	FF	Values and Functions
C0H	7:0	0	1	0	Read current low byte address.
		0	1	1	Read current high byte address.
		1	0	0	Write base & current low byte address.
		1	0	1	Write base & current high byte address.

I/O Port C2H DMA2 Channel 0 Word Count Register (READ/WRITE/FLIP-FLOP)

Addr	Bits	-IOR	-IOW	FF	Values and Functions
C2H	7:0	0	1	0	Read current word count low byte.
		0	1	1	Read current word count high byte.
		1	1	0	Write base & current word count low byte.
		1	0	1	Write base & current word count high byte.

I/O Port C4H DMA2 Channel 1 Address Register (READ/WRITE/FLIP-FLOP)

Addr	Bits	-IOR	-IOW	FF	Values and Functions
C4H	7:0	0	1	0	Read current low byte address.
		0	1	1	Read current high byte address.
		1	0	0	Write base & current low byte address.
		1	0	1	Write base & current high byte address.

I/O Port C6H DMA2 Channel 1 Word Count Register (READ/WRITE/FLIP-FLOP)

Addr	Bits	-IOR	-IOW	FF	Values and Functions
C6H	7:0	0	1	0	Read current word count low byte.
		0	1	1	Read current word count high byte.
		1	0	0	Write base & current word count low byte.
		1	0	1	Write base & current word count high byte.

I/O Port C8H DMA2 Channel 2 Address Register (READ/WRITE/FLIP-FLOP)

Addr	Bits	-IOR	-IOW	FF	Values and Functions
C8H	7:0	0	1	0	Read current low byte address.
		0	1	1	Read current high byte address.
		1	0	0	Write base & current low byte address.
		1	0	1	Write base & current high byte address.

I/O Port CAH DMA2 Channel 2 Word Count Register (READ/WRITE/FLIP-FLOP)

Addr	Bits	-IOR	-IOW	FF	Values and Functions
CAH	7:0	0	1	0	Read current word count low byte.
		0	1	1	Read current word count high byte.
		1	0	0	Write base & current word count low byte.
		1	0	1	Write base & current word count high byte.

I/O Port CCH DMA2 Channel 3 Address Register (READ/WRITE/FLIP-FLOP)

Addr	Bits	-IOR	-IOW	FF	Values and Functions
CCH	7:0	0	1	0	Read current low byte address.
		0	1	1	Read current high byte address.
		1	0	0	Write base & current low byte address.
		1	0	1	Write base & current high byte address.

I/O Port CEH DMA2 Channel 3 Word Count Register (READ/WRITE/FLIP-FLOP)

Addr	Bits	-IOR	-IOW	FF	Values and Functions
CEH	7:0	0	1	0	Read current word count low byte.
		0	1	1	Read current word count high byte.
		1	0	0	Write base & current word count low byte.
		1	0	1	Write base & current word count high byte.

I/O Port D0H DMA2 Status Register (READ ONLY)

	Bits	Values and Functions
D0H	7	Channel 3 pending DMA service.
		(0): DRQ3 not active.
		1: DRQ3 active.
	6	Channel 2 pending DMA service.
		(0): DRQ2 not active.
		1: DRQ2 active.
	5	Channel 1 pending DMA service.
		(0): DRQ1 not active.
		1: DRQ1 active.
	4	Channel 0 pending DMA service.
		(0): DRQ0 not active.
		1: DRQ0 active.
	3	Channel 3 Terminal Count status.
		(0): TC not reached.
		1: Channel 3 has reached TC.
	2	Channel 2 Terminal Count status.
		(0): TC not reached.
		1: Channel 2 has reached TC.
	1	Channel 1 Terminal Count status.
		(0): TC not reached.
		1: Channel 1 has reached TC.
	0	Channel 0 Terminal Count status.
		(0): TC not reached.
		1: Channel 0 has reached TC.

I/O Port D0H DMA2 Command Register (WRITE ONLY)

	Bits	Values and Functions
D0H	7	-DACK input active level.
		(0): -DACK active low.
		1: -DACK active high.
	6	DREQ input active level.
		(0): DREQ active high.
		1: DREQ active low.
	5	Extended Write enable.
		(0): Write commands are asserted late.
		1: Write commands are asserted one DMA cycle earlier during a transfer.
	4	Priority type for channel servicing.
		(0): Fixed priority.
		1: Rotating priority.
	3	Transfer compression enable.
		(0): Normal timing, three clock cycle transfer.
		1: Compressed timing, two clock cycle transfer.
	2	Master disable for the DMA controller.
		(0): DMA controller enable.
		1: DMA controller disable.
	1	Address hold during memory-to-memory transfers.
		(0): Address hold for Channel 0 disable.
		1: Address hold for Channel 0 enable.
	0	Memory-to-Memory transfer enable.
		(0): Memory-to-Memory transfers disable.
		1: Memory-to-Memory transfers for Channel 0 and 1 enable.

I/O Port D2H *DMA2 Request Register (READ ONLY)*

	Bits	Values and Functions
D2H	7:4	(1111): These bits are always set during a read.
	3	Request bit for Channel 3.
		(0): No request pending.
		1: Request pending for Channel 3.
	2	Request bit for Channel 2.
		(0): No request pending.
		1: Request pending for Channel 2.
	1	Request bit for Channel 1.
		(0): No request pending.
		1: Request pending for Channel 1.
	0	Request bit for channel 0.
		(0): No request pending.
		1: Request pending for Channel 0.

I/O Port D2H *DMA2 Request Register (WRITE ONLY)*

	Bits	Values and Functions
D2H	7:3	(XXXXX): Don't Cares.
	2	Request bit.
		(0): Reset Request bit.
		1: Set Request bit.
	1:0	DMA Channel Request bit Selection.
		(00): DMA Channel 0.
		01: DMA Channel 1.
		10: DMA Channel 2.
		11: DMA Channel 3.

I/O Port D4H *DMA2 Command Register (READ ONLY)*

	Bits	Values and Functions
D4H	7	-DACK input active level.
		(0): -DACK active low.
		1: -DACK active high.
	6	DREQ input active level.
		(0): DREQ active high.
		1: DREQ active low.
	5	Extended Write enable.
		(0): Write commands are asserted late.
		1: Write commands are asserted one DMA cycle earlier during a transfer.
	4	Priority type for channel servicing.
		(0): Fixed priority.
		1: Rotating priority.
	3	Transfer compression enable.
		(0): Normal timing, three clock cycle transfer.
		1: Compressed timing, two clock cycle transfer.
	2	Master disable for the DMA controller.
		(0): DMA controller enable.
		1: DMA controller disable.
	1	Address hold during memory-to-memory transfers.
		(0): Address hold for Channel 0 disable.
		1: Address hold for Channel 0 enable.
	0	Memory-to-Memory transfer enable.
		(0): Memory-to-Memory transfers disable.
		1: Memory-to-Memory transfers for Channel 0 and 1 enable.

I/O Port D4H *DMA1 Single Bit Request Mask Register (WRITE ONLY)*

	Bits	Values and Functions
D4H	7:3	(XXXXX): Don't Cares.
	2	Mask bit.
		(0): Clear Mask bit.
		1: Set Mask bit.
	1:0	DMA Channel Mask bit Selection.
		(00): DMA Channel 0.
		01: DMA Channel 1.
		10: DMA Channel 2.
		11: DMA Channel 3.

I/O Port D6H *DMA2 Mode Register (READ/WRITE)*

	Bits	Values and Functions
D6H	7:6	Channel Mode type.
	5	Address counting direction.
	4	Autoinitialization enable.
	3:2	Type of transfer.
	1:0	DMA Channel Mode Selection.

I/O Port D8H *DMA2 Set Byte Pointer Flip-Flop (READ ONLY)*

	Bits	Values and Functions
D8H	7:0	Bits 7:0 are don't cares. The Set Byte Pointer Flip-Flop Clear command is activated as a result of the address 0D8H access and the assertion of -IOR. This command allows the CPU to adjust the pointer to the high byte of an address or word count register.

I/O Port D8H *DMA2 Clear Byte Pointer Flip-Flop (WRITE ONLY)*

	Bits	Values and Functions
D8H	7:0	Bits 7:0 are don't cares. The Clear Byte Pointer Flip-Flop Clear command is activated as a result of the address 0D8H access and the assertion of -IOW. This command allows the CPU to adjust the pointer to the low byte of an address or word count register.

I/O Port DAH *DMA2 Temporary Register (READ ONLY)*

	Bits	Values and Functions
DAH	7:0	Bits 7:0 contains values of XD<0:7> during the first cycle of a memory-to-memory transfer. Data from the last memory-to-memory transfer will remain in the register unless a RESET or Master Clear occurs.

I/O Port DAH *DMA2 Master Clear (WRITE COMMAND ONLY)*

	Bits	Values and Functions
DAH	7:0	Bits 7:0 are don't cares. The Master Clear command is activated as a result of the address 0DAH access and the assertion of -IOW. This command has the same effect as a hardware RESET. The Command Register, Status Register, Request Register, and Byte Pointer Flip-Flop are cleared and the Request Mask Register is set.

I/O Port DCH *DMA2 Clear Mode Register Counter (READ COMMAND ONLY)*

	Bits	Values and Functions
DCH	7:0	Bits 7:0 are don't cares. The Clear Mode Register Counter command is activated as a result of the address 0DCH access and the assertion of -IOR. This command is provided to allow the CPU to restart the mode read process at a known point. 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 registers will be read is Channel 0 first, Channel 3 last.

I/O Port DCH *DMA2 Clear Request Mask Bits (WRITE COMMAND ONLY)*

	Bits	Values and Functions
DCH	7:0	Bits 7:0 are don't cares. The Clear Mask Register command is activated as a result of the address 0DCH access and the assertion of -IOW. This command enables all four DMA channels to accept requests by clearing the mask bits in the register.

I/O Port DEH *DMA2 Request Mask Register Bits (READ/WRITE)*

	Bits	Values and Functions
DEH	7:4	(XXXX): Don't Cares.
	3	Channel 0 Mask bit.
		0: No bit mask, channel enabled.
		(1): Bit mask, channel disabled.
	2	Channel 1 Mask bit.
		0: No bit mask, channel enabled.
		(1): Bit mask, channel disabled.
	1	Channel 2 Mask bit.
		0: No bit mask, channel enabled.
		(1): Bit mask, channel disabled.
	0	Channel 3 Mask bit.
		0: No bit mask, channel enabled.
		(1): Bit mask, channel disabled.

I/O Port 61H *Control Port/Status*

	Bits	Values and Functions
61H	7	PARITY CHECK (read only). This bit indicates an error has occurred on the local memory.
		0 = no error occurred.
		1 = an error occurred.
	6	IOCHCK (read only). This bit indicates an I/O channel check has occurred (usually a parity error) on the system I/O channel.
		0 = no error occurred.
		1 = an error occurred.
	5	TIMER 2 OUT (read only). This bit returns the condition of timer 2 output.
	4	REFRESH DETECT (read only). This bit toggles on each refresh cycle.
	3	IOCHCK DISABLE (read/write). This bit disables NMI generation for channel check errors.
		0 = enables NMI (default).
		1 = disables NMI.
	2	PARITY DISABLE (read/write). This bit, in conjunction with the auxiliary parity disable bit (AUX PARITY DISABLE) of internal configuration register 46H, is used to disable parity error contributions to the NMI. It is logically ORed with AUX PARITY DISABLE. Thus, system parity is disabled if either this bit (PARITY DISABLE) or the AUX PARITY DISABLE bit is set to a logical one.
		0 = Parity is enabled (default).
		1 = Parity is disabled.
	1	SPEAKER DATA (read/write). This bit gates the output of channel 2 of the timer/counter.
		0 = Output is disabled (default).
		1 = Output is enabled.
	0	TIMER2GATE (read/write)Controls operation of timer channel 2.
		0 = Channel 2 timer operation is disabled (default).
		1 = Channel 2 timer operation is enabled.

2.3 I/O Map

Table 2-1. IPC and Address Buffer's Internal Decode

Device	Address Range
DMA Controller #1	000H-00FH
Interrupt Controller #1	020H-021H
Configuration Registers	022H-023H
Counter Timer Controller	040H-043H
Real Time Clock (CMOS RAM)	070H-071H
DMA PAGE Registers	080H-08FH
Interrupt Controller #2	0A0H-0A1H
DMA Controller #2	0C0H-0DFH

This page is intentionally left blank.

Section 3

356 Pin Descriptions

Section 3

356 Pin Descriptions

3.1 Pin Assignments

Table 3-1. Clocks

Pin #	Symbol	Type	Signal Description
CX1	74	I	14.31818MHz crystal oscillator input. A crystal is connected between CX1 and CX2.
CX2	75	O	14.31818MHz oscillator output to the crystal. A crystal is connected between CX1 and CX2.
OSC	80	O	14.31818MHz output derived from the crystal. This is used to drive the AT bus directly (requires a series resistor). 24mA drive capability.
OSC/12	72	O	This is a 1.19MHz clock(TTL) derived from dividing the 14.31818MHz input crystal. Used internally for the 8254 clock. 4mA drive capability.
BUSCLK	35	I	BUS CLOCK is used to generated the timing signals that control DMA operations. This input may be driven from DC to 10MHz. The internal clock used for DMAC is either BUSCLK or BUSCLK/2 depending on the setting of DMA CLOCK SELECT bit in the configuration register 01H.

Table 3-2. RESET and Arbitration

Pin #	Symbol	Type	Signal Description
RESET4	110	I	RESET4 is an active high input from the 82C351 which effects the following registers: DMA Controllers: Clears the command, Status, DMA Request, Temporary register, First/Last flip-flop; sets the mask register. Following reset, the DMA controller is in an idle state. INTERRUPT Controller: Clears the edge sense circuit, the interrupt mask register, all ICW4 functions, IRQ0 is assigned highest priority, slave address is set to 7, special mask mose is disabled, and status read is set to IRR.
HRQ	142	0	HOLD REQUEST is an active high output and is used to request control of the system bus. HRQ is normally connected directly to the 82C351's HRQ1 pin. However, HRQ may be tied to HLDA. This will result in one S0 state before the transfer. 4mA drive capability.
HLDA1	66	I	HOLD ACKNOWLEDGE 1 is an active high input signal from the 82C351 and indicates the CPU has relinquished control of the system busses in response to HRQ.

Table 3-3. DMA Controller

Pin #	Symbol	Type	Signal Description
DREQ<0:3>	21-18	I	DMA REQUEST lines are individual asynchronous channel request inputs used by peripherals to obtain DMA service. DREQ0 through DREQ3 support 8 bit transfers between 8 bit I/O and 8 or 16 bit system memory. In fixed priority, DREQ0 has the highest priority and DREQ7 has the lowest priority. A request is generated by activating the DREQ line of a channel. -DACK will acknowledge the recognition of a DREQ. Polarity of DREQ is programmable. Unused DREQ inputs should be pulled high or low and the corresponding mask bit set. DREQ<0:3> lines have internal 10K pull-ups.
DREQ<5:7>	8-10	I	DMA REQUEST are individual asynchronous channel request inputs used by peripherals to obtain DMA service. DREQ5 through DREQ7 support transfers between 16-bit peripherals and 16-bit system memory. In fixed priority, DREQ0 has the highest priority and DREQ7 has the lowest priority. A request is generated by activating the DREQ line of a channel. -DACK will acknowledge the recognition of a DREQ. Polarity of DREQ is programmable. Unused DREQ inputs should be pulled high or low and the corresponding mask bit set. DREQ<5:7> lines have internal 10K pullups.
-DACK<0:3>	17-14	O	DMA ACKNOWLEDGE -DACK0 through -DACK3 are outputs of the 8-bit DMA controller, and are used to notify the individual peripherals when one has been granted a DMA cycle. The active polarity of these lines are programmable. 4mA drive capability.
-DACK<5:7>	11-13	O	DMA ACKNOWLEDGE -DACK5 through -DACK7 are outputs of the 16-bit DMA controller, and are used to notify the individual peripherals when one has been granted a DMA cycle. The active polarity of these lines are programmable. 4mA drive capability.
TC	2	O	TERMINAL COUNT is an active high signal and is generated by the DMA Controller when the terminal count for any channel is reached, except for channel 0 in memory to memory mode. During memory to memory transfers TC will be pulsed when the TC for channel 1 occurs. 24mA drive capability.
-AEN8	4	O	ADDRESS ENABLE for 8 bit DMA transfers is an active low signal and is the output enable for the 8 bit latch containing the upper 8 address bits (A8-A15). It is inactive when an external bus master controls the system bus. 4mA drive capability.
-AEN16	3	O	ADDRESS ENABLE for 16 bit DMA transfers is an active low signal and is the output enable for the 8 bit latch containing the upper 8 address bits (A9-A16). It is inactive when an external bus master controls the system bus. 4mA drive capability.
-MSE(AEN)	27	O	MODULE SELECT ENABLE enables the chip select function on one of the following modules: (DMA controller, Interrupt controller, Timer, RTC, DMA Page Register, or the Configuration registers). Address Enable (for the DMA controller) is active (high) during any DMA cycle, and low for CPU, Refresh, and AT bus master cycles. 24mA drive capability.
-IOR	112	I/O	I/O Read is an active low bi-directional signal. In an idle cycle, (non-DMA or non-interrupt), it is a schmitt triggered input control signal used by the CPU to read information from the 82C356's internal registers. In an active DMA cycle, it is an output control signal used by the DMA Controller to read data from a peripheral during a DMA read transfer. 24mA drive capability.
-IOW	111	I/O	I/O Write is an active low bi-directional signal. In an idle cycle, (non-DMA or non-interrupt), it is a schmitt triggered input control signal used by the CPU to write information to the 82C356's internal registers. In an active DMA cycle, it is an output control signal used by the DMA Controller to write data to a peripheral during a DMA read transfer. 24mA drive capability.

Table 3-3. DMA Controller (continued)

Pin #	Symbol	Type	Signal Description
-MEMR	6	O	MEMORY READ is an active low three-state output (active only during DMA cycles) and is used to access data from the selected memory location during a DMA read or memory to memory transfer. 24mA drive capability.
-MEMW	5	O	MEMORY WRITE is an active low three-state output (active during DMA cycles only) and is used to write data to the selected memory location during DMA write or memory to memory transfer. 24mA drive capability.
IOCHRDY	113	I/O	I/O CHANNEL READY is a bi-directional signal. During input mode, driving IOCHRDY low causes the internal DMA ready signal to go low asynchronously. When IOCHRDY goes high, one DMA Clock cycle will elapse before internal DMA Ready goes high. This signal is used to extend memory read and write pulses for the DMA controllers to accommodate slow memories or I/O devices. During output mode, this pin is an open drain output and provides an active low output whenever an 82C356 register is addressed for a read or write. IOCHRDY provides a means of introducing a programmed number of wait-states for I/O read/write cycles to the 82C356. 8mA drive capability.

Table 3-4. Interrupt

Pin #	Symbol	Type	Signal Description
IRQ1	128	I	INTERRUPT REQUEST 1 is an asynchronous input (normally from pin 35 of the 8042 in a PC/AT). An interrupt request is executed by raising an IRQ input from low to high and holding it high until it is acknowledged (edge triggered mode) or just a high level on an IRQ input (level triggered mode). This signal has a 10K internal pullup.
IRQ<3:7>	129-133	I	INTERRUPT REQUESTS 3 through 7 are asynchronous inputs from the AT bus. An interrupt request is executed by raising an IRQ input from low to high and holding it high until it is acknowledged (edge triggered mode) or just a high level on an IRQ input (level triggered mode). These signals have internal 10K pull-ups.
IRQ<9:15>	134-140	I	INTERRUPT REQUESTS 9 through 15 are asynchronous inputs and are executed by raising an IRQ input from low to high and holding it high until it is acknowledged (edge triggered mode) or just a high level on an IRQ input (level triggered mode). These signals have internal 10K pull-ups. IRQ 13 is connected to the 387DX PAL and IRQ9-12, IRQ14-15 are connected to the AT bus.
INTR	143	O	INTERRUPT goes high whenever a valid interrupt request is asserted and is connected to the 386DX to interrupt the CPU. 4mA drive capability.
-INTA	28	I	INTERRUPT ACKNOWLEDGE from the 82C351 is an active low signal and is used to enable the interrupt controllers to vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.

Table 3-5. Timer

Pin #	Symbol	Type	Signal Description
OUT1	78	O	OUTPUT 1 is the output of TIMER 1 and is programmed as a rate generator to produce a15 uS period signal used to request a refresh cycle. 4mA drive capability, connected to REFREQ of the 82C351.
SPKR	107	O	SPEAKER gates the speaker data and Timer Out 2 to drive the internal speaker. 4mA drive capability.

Table 3-6. Real Time Clock

Pin #	Symbol	Type	Signal Description
-PSRSTB	126	I	-PSRSTB is an active low input and is used to establish the condition of the control registers when power is applied to the device. In a PC/AT compatible design, this pin should be tied to the battery back-up circuit. -PSRSTB low sets a bit in RTC indicating power has been lost.
OSCI/IRQ8	141	I	OSCILLATOR INPUT is a schmitt triggred input used as a base for the time functions. External square waves of 32.768KHz may be connected to this input. When the external RTC is enabled through index register 26H, this pin is connected to IRQ8 instead of OSC.
PWRGD	127	I	POWER GOOD must be high for all bus cycles in which the CPU accesses the 82C356. When PWRGD is low, all address, data, data strobe, and R/W pins are disconnected from the processor.
RTCAS	33	O	REAL TIME CLOCK ADDRESS STROBE is an active high output and is used to de-multiplex the bus on an external MC146818 REAL-TIME CLOCK plus RAM peripheral device. The falling edge of RTCAS causes the address to be latched within the MC146818. 4mA drive capability.
RTCDS	34	O	RTC DATA STROBE is an active high output and is used to control the bi-directional bus on an external MC146818 REAL-TIME CLOCK plus RAM peripheral device. During a read cycle, (RTCRW high), the RTCDS output drives the bus with the read data. On the other hand, during a write cycle (RTCRW low), the RTCDS's trailing edge causes the MC146818 to latch the written data. 4mA drive capability.
RTCRW	32	O	RTC R/W CONTROL OUTPUT is used to indicate read or write mode for an external MC146818 REAL-TIME CLOCK plus RAM peripheral device. A high level on RTCRW indicates to the MC146818 that the current cycle is a read cycle. A low level on RTCRW indicates to the MC146818 that the current cycle is a write cycle. 4mA drive capability.

Table 3-7. NMI Logic

Pin #	Symbol	Type	Signal Description
-LPAR	29	I	LATCHED PARITY ERROR is an active low input (from the 82C355) and indicates a parity error during a DRAM read. When active it will cause an NMI to be generated.
-IOCHCK	31	I	IO CHANNEL CHECK is an active low input and is used to signal an error condition form an I/O device. When active and enabled, it causes an NMI to be generated.
PCUNMIIN	30	I	POWER CONTROL NON MASKABLE INTERRUPT INPUT is a programmable input from the power control unit. It is an active low input signal by default and can be programmed by bit 3 of index register 26H to be either active low or active high. Bit 4 of index 26 enables or disables this feature. If the PCU is not used, PCUNMIN should be tied to Vcc for the 82C356 to operate correctly.
NMI	76	O	NON-MASKABLE INTERRUPT is an active high output and is connected to the NMI of the CPU. 4mA drive capability.

Table 3-8. Address Buffers Decodes

Pin #	Symbol	Type	Signal Description
XA<0:1>	38-39	I/O	EXPANSION ADDRESS bits 0 and 1 are bidirectional signals. They are outputs for DMA and AT master cycles and Inputs for CPU cycles. These are connected to the 82C351 which converts them to and from the byte enables. 4mA drive capability.
A<2:23>	40-49, 51-53 55-63	I/O	LOCAL ADDRESS bus bits 2 through 23 are bidirectional signals, inputs during CPU/AT cycles and output during DMA/master cycles. 4mA drive capability.
SA<0:19>	82-85, 87-90, 92-95, 97-100, 102, 104-106	I/O	I/O CHANNEL (AT) address bus. Input for AT master cycles. Output at all other times. 24 mA drive.
-8042CS	25	O	8042 CHIP SELECT is a active low signal and is a I/O decode of addresses 060H or 064H. 4mA drive capability.
INTCLR	77	O	INTERRUPT CLEAR is an active high output signal and is used to clear IRQ13 from the 387DX PLD. INTCLR is generated during RESET, Out command to port F1H, or Out command to port F0. 4mA drive capability.
XDIR	26	O	X BUS DIRECTION controls the direction of data transfer within the 82C355 between the peripheral bus and the IO channel. When low, XDIR should drive the SD bus signals toward the XD bus. When high XDIR should drive the XD bus signals toward the SD bus. XDIR is an I/O decode of ports on the XD bus ANDed with -IOR. 4mA drive capability.
-ROMCS	23	I	ROM CHIP SELECT is an input to generate XDIR during ROM read cycles. This is connected to -ROMCS of the 82C351.
-REF	68	I	REFRESH is an active low input from the 82C351 or an I/O device on the AT bus which increments the address counter (in the 82C356) and controls the address buffer direction. When active, the content of the refresh address counter is gated to the SA address bus. On the rising edge of -REF, the internal refresh counter is incremented.
-MASTER	69	I	BUS MASTER is active low input signal f generated by a device active on the AT expansion Bus. It has an internal 10K Ohm pull-up resistor.
-MALE	70	I	MEMORY ADDRESS LATCH ENABLE latches local addresses into the address registers on the rising (trailing) edge.
-ATEN	67	I	AT BUS ENABLE is an active low input signal from the 82C351 and is active when the CPU is performing an AT bus access.

Table 3-9. Data Bus and Powers

Pin #	Symbol	Type	Signal Description
XD<0:7>	121-114	I/O	Data bits 0 through 7 are 3-state bi-directional Data Bus lines and are connected to the system data bus, (the XD bus), in a PC/AT design. The outputs are enabled in the program condition during the I/O READ to output the contents of the DMA controller registers, the three Interrupt Controller registers, the Timer/Counter registers, the Real Time Clock's internal registers, index register 26H, port B, and programmable chip select registers. During an I/O WRITE cycle, the outputs are disabled and the CPU can program the DMA Controller registers, the Interrupt Controller registers, the Timer/Counter registers, the DMA Page register, the Real Time Clock registers, the internal RAM, index register 26H, port B, and programmable chip select registers. During DMA cycles, the most significant 8 bits of the address are output onto the data bus to be strobed into an external latch by AEN8- or AEN16-. During memory-to-memory operations, data from the memory comes into the DMA Controller on the data bus during a read from the memory. During the interrupt sequence, the interrupt controllers output the interrupt vector byte on the data bus. Data bus XD<0:7> also acts as the multiplexed address/data bus for the Real Time Clock.
TEST	124		Tied to ground.
Vcc	1, 24, 54, 71, 79, 103, 109, 125, 144		+5V ±5% Power Supply.
Gnd	7, 22, 36, 37, 50, 64, 65, 73, 81, 86, 91, 96, 101, 108, 122, 123		Ground.

3.2 Numerical Listing of Pin Assignments

Table 3-10. Numerical Pin Assignments

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	Vcc	37	Vss	73	Vss	109	Vcc
2	TC	38	XA0	74	CX1	110	RESET4
3	-AEN16	39	XA1	75	CX2	111	-IOW
4	-AEN18	40	A2	76	NMI	112	-IOR
5	-MEMW	41	A3	77	INTCLR	113	IOCHRDY
6	-MEMR	42	A4	78	OUT1	114	XD7
7	Vss	43	A5	79	Vcc	115	XD6
8	DREQ5	44	A6	80	OSC	116	XD5
9	DREQ6	45	A7	81	Vss	117	XD4
10	DREQ7	46	A8	82	SA0	118	XD3
11	-DACK5	47	A9	83	SA1	119	XD2
12	-DACK6	48	A10	84	SA2	120	XD1
13	-DACK7	49	A11	85	SA3	121	XD0
14	-DACK3	50	Vss	86	Vss	122	Vss
15	-DACK2	51	A12	87	SA4	123	Vss
16	-DACK1	52	A13	88	SA5	124	TEST356
17	-DACK0	53	A14	89	SA6	125	Vcc
18	DREQ3	54	Vcc	90	SA7	126	-PSRSTB
19	DREQ2	55	A15	91	Vss	127	PWRGD
20	DREQ1	56	A16	92	SA8	128	IRQ1
21	DREQ0	57	A17	93	SA9	129	IRQ3
22	Vss	58	A18	94	SA10	130	IRQ4
23	-ROMCS	59	A19	95	SA11	131	IRQ5
24	Vcc	60	A20	96	Vss	132	IRQ6
25	-8042CS	61	A21	97	SA12	133	IRQ7
26	XDIR	62	A22	98	SA13	134	IRQ9
27	-MSE(AEN)	63	A23	99	SA14	135	IRQ10
28	-INTA	64	Vss	100	SA15	136	IRQ11
29	-LPAR	65	Vss	101	Vss	137	IRQ12
30	PCUNMIN	66	HLDA1	102	SA16	138	IRQ13
31	-IOCHCK	67	-ATEN	103	Vcc	139	IRQ14
32	RTCRW	68	-REF	104	SA17	140	IRQ15
33	RTCAS	69	-MASTER	105	SA18	141	OSC1/IRQ8
34	RTCDS	70	-MALE	106	SA19	142	HRQ
35	BUSCLK	71	Vcc	107	SPKR	143	INTR
36	Vss	72	OSC/12	108	Vss	144	Vcc

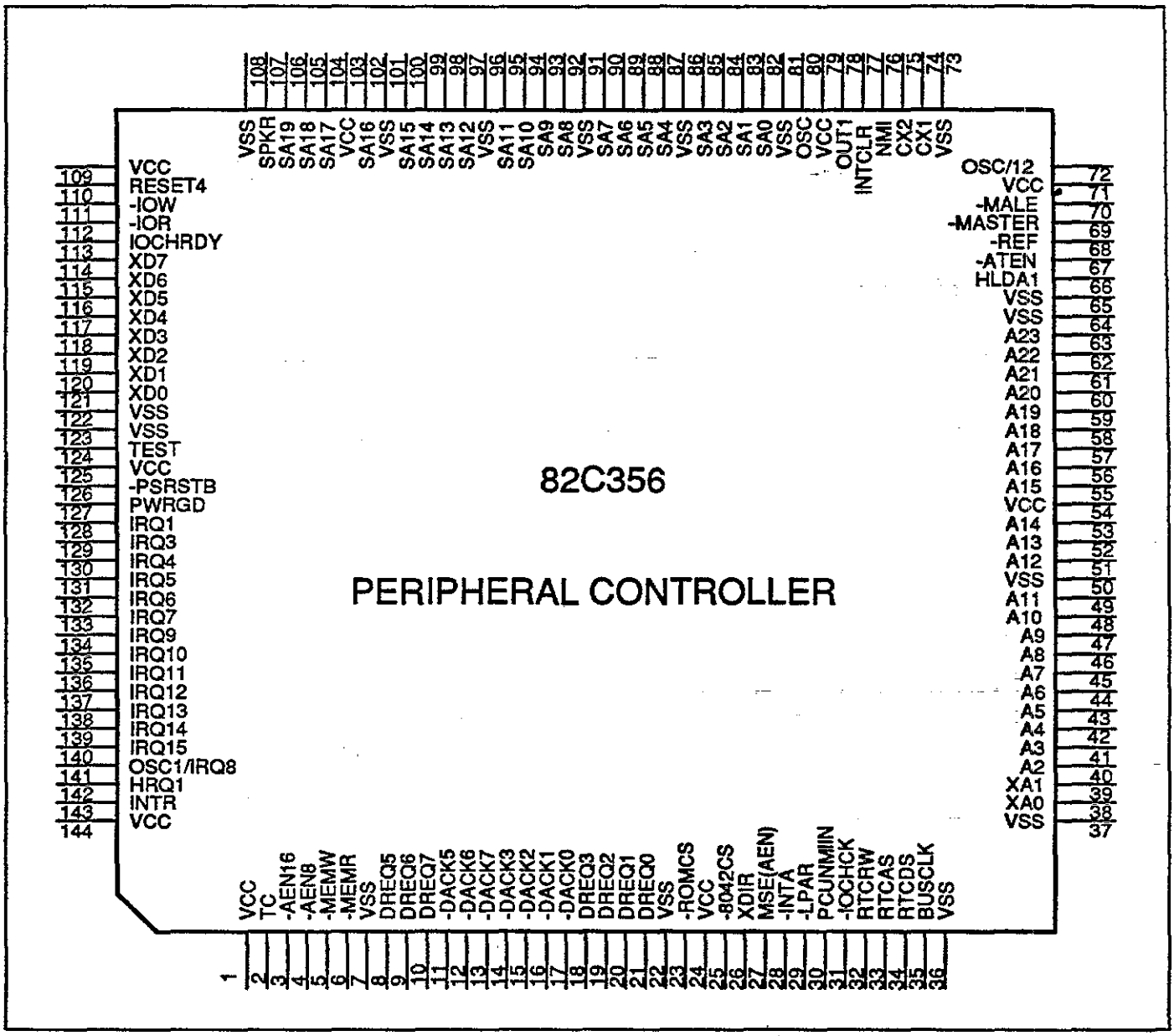
3.3 Alphabetical Listing of Pin Assignments

Table 3-11. Alphabetical Pin Assignments

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
-AEN16	3	DREQ1	20	OSC1/IRQ8	141	Vcc	24
-AEN18	4	DREQ2	19	OSC/12	72	Vcc	54
-ATEN	67	DREQ3	18	OUT1	78	Vcc	71
A2	40	DREQ5	8	PCUNMIN	30	Vcc	79
A3	41	DREQ6	9	-PSRSTB	126	Vcc	103
A4	42	DREQ7	10	PWRGD	127	Vcc	109
A5	43	HLDA1	66	-REF	68	Vcc	125
A6	44	HRQ	142	RESET4	110	Vcc	144
A7	45	-INTA	28	-ROMCS	23	Vss	7
A8	46	INTCLR	77	RTCAS	33	Vss	22
A9	47	INTR	143	RTCDS	34	Vss	36
A10	48	-IOCHCK	31	RTCRW	32	Vss	37
A11	49	IOCHRDY	113	SA0	82	Vss	50
A12	51	-IOR	112	SA1	83	Vss	64
A13	52	-IOW	111	SA2	84	Vss	65
A14	53	IRQ1	128	SA3	85	Vss	73
A15	55	IRQ3	129	SA4	87	Vss	81
A16	56	IRQ4	130	SA5	88	Vss	86
A17	57	IRQ5	131	SA6	89	Vss	91
A18	58	IRQ6	132	SA7	90	Vss	96
A19	59	IRQ7	133	SA8	92	Vss	101
A20	60	IRQ9	134	SA9	93	Vss	108
A21	61	IRQ10	135	SA10	94	Vss	122
A22	62	IRQ11	136	SA11	95	Vss	123
A23	63	IRQ12	137	SA12	97	XA0	38
BUSCLK	35	IRQ13	138	SA13	98	XA1	39
CX1	74	IRQ14	139	SA14	99	XDIR	26
CX2	75	IRQ15	140	SA15	100	XD0	121
-DACK0	17	-LPAR	29	SA16	102	XD1	120
-DACK1	16	-MALE	70	SA17	104	XD2	119
-DACK2	15	-MASTER	69	SA18	105	XD3	118
-DACK3	14	-MEMR	6	SA19	106	XD4	117
-DACK5	11	-MEMW	5	SPKR	107	XD5	116
-DACK6	12	-MSE(AEN)	27	TC	2	XD6	115
-DACK7	13	NMI	76	TEST356	124	XD7	114
DREQ0	21	OSC	80	Vcc	1	-8042CS	25

3.4 Pin Diagram

Figure 3-1. 82C356 Peripheral Controller Pin Diagram



This page is intentionally left blank.

Section 4

356 Physical Characteristics

Section 4

356 Physical Characteristics

4.1 Absolute Maximum Ratings

	Symbol	Min.	Max.	Units
Supply Voltage	Vcc		7.0	V
Input Voltage	V_I	-.5	5.5	V
Output Voltage	V_O	-.5	5.5	V
Operating Temperature	Top	-25	85	C
Storage Temperature	Tstg	-40	125	C

Note Exposure to absolute maximum ratings conditions for extended periods may affect device reliability. Exceeding the absolute maximum ratings can cause permanent damage to the device.

4.2 Operating Conditions

	Symbol	Min.	Max.	Units
Supply Voltage	Vcc	4.75	5.25	V
Ambient Temperature	T_A		70	C

Section 5

356 DC/AC Characteristics

5.1 DC Characteristics

	Symbol	Min.	Max.	Units
Input low voltage	VIL			
TTL level (All pins except BUSCLK and CX1)			.8	V
BUSCLK and CX1		-0.3	0.8	V
Input high voltage	VIH			
TTL level (All pins except BUSCLK and CX1)		2.0		V
BUSCLK and CX1	VCC	0.8	+0.3	V
Output low voltage	VOL		.45	V
Output high voltage				
All pins except IOCHRDY IOCHRDY has open drain driver	VOH	2.4 n/a		V
Input LOW current @ Vo = Vss except IR1, IR3-7, IR9-15, DREQ0-3, DREQ5-7, -IOCHCK	IIL	±10		µA
IR1, IR3-7, IR9-15, DREQ0-3, DREQ5-7, -IOCHCK			-1	mA
Input HIGH current @ Vo = Vdd				
All except PWRGD	IIH		±10	µA
PWRGD	IIH		200	µA
Output current low	IOL			
All pins except SA0-19 -IOR, -IOW, IOCHRDY, TC, -MEMR, -MEMW, MSE (AEN)			4	mA
SA0-19, -IOR, -IOW, IOCHRDY, TC, -MEMR, -MEMW, MSE (AEN)			24	mA
IOCHRDY		8		mA
Output current high	IOH			
All pins except SA0-19, -IOR, -IOW, IOCHRDY, T2, -MEMR, -MEMW, MSE (AEN)		4		mA
SA0-19, -IOR, -IOW, IOCHRDY, TC, -MEMR, -MEMW, MSE (AEN)		-3.3		mA
3-State output OFF current LOW	IOZL		±10	µA
3-State output OFF current HIGH	IOZH		±10	µA
Power supply current @8 MHz	ICC		60	mA
Input capacitance	CIN		10	pF
Output or I/O capacitance	COUT		10	pF
Stand-by Icc			10	µA

5.2 AC Characteristics

All timing parameters are specified under capacitive load of 50 pf and temperature of 70 degree C. These timing tables cover both 25MHz and 33MHz; all units are in nanoseconds, unless otherwise specified. Also, the AC specifications mentioned in this document are subject to change.

Table 5-1. CPU AT Cycle

CPU AT Cycle			
		Min.	Max.
t601	A<23:0> setup to -MALE inactive	15	
t602	A<23:0> hold from -MALE inactive	10	
t603	SA<19:0> valid from -ATEN active	0	50
t605	SA<19:0> float from -ATEN inactive	0	25
t608	-8042CS active from -MALE inactive	0	60
t611	-8042CS inactive from -MALE inactive	0	32

Table 5-2. Master Cycle

Master Cycle			
		Min.	Max.
t615	A<23:2> valid from SA valid	6	35
t616	XA<1:0> valid from SA valid	5	30
t618	-8042CS active from SA valid	10	60
t621	XA<1:0> invalid from -MASTER inactive	6	19

Table 5-3. *DMA Cycle*

DMA Cycle			
		Min.	Max
t625	BUSCLK period (1X)	125	
t625A	BUSCLK period (2X)	62	
t626	BUSCLK high time (1X)	55	
t626A	BUSCLK high time (2X)	27	
t627	BUSCLK low time (1X)	43	
t627A	BUSCLK low time (2X)	22	
t628	DREQn setup to BUSCLK low	0	
t629	HRQ delay from BUSCLK low		70
t630	HLDA setup to BUSCLK high	40	
t631	-AEN8/16 valid from BUSCLK low		95
t632	-AEN8/16 invalid from BUSCLK		75
t633	AEN active from HLDA1 active	5	38
t651	AEN inactive from HLDA1 low	5	50
t634	Address valid from BUSCLK high		100
t635	Address hold from BUSCLK high		50
t636	-DACKn valid from BUSCLK low		100
t637	Command enable delay from BUSCLK high		80
t638	Command active delay from BUSCLK high		100
t639	Write cmnd inactive delay from BUSCLK high		70
t640	Address hold from -IOW inactive	70	
t641	Address hold from -MEMR inactive	50	
t642	Command float delay from BUSCLK high		75
t643	-IOR inactive delay from BUSCLK high		115
t644	TC delay from BUSCLK		50
t649	IOCHRDY setup to BUSCLK low	25	
t650	IOCHRDY hold from BUSCLK low	15	

Table 5-4. *Internal Register Access*

Internal Register Access			
		Min.	Max
t655	Command active period	150	
t656	XD<7:0> active from -IOR active	5	35
t657	XD<7:0> valid from -IOR active		140
t658	XD<7:0> hold from -IOR inactive	8	
t659	XD<7:0> setup to -IOW inactive	130	
t660	XD<7:0> hold from -IOW inactive	20	
t661	Address setup to command active	25	
t662	Address hold from command inactive	0	
t663	IOCHRDY active delay from command	6	50
t664	IOCHRDY inactive delay from BUSCLK high	6	40
t665	NMI valid delay from -IOW inactive	20	

Table 5-5. *External Real Time Clock Access*

External Real Time Clock Access			
		Min.	Max
t666	RTCAS active delay from -IOW active	5	40
t667	RTCAS inactive delay from -IOW inactive	5	40
t672	RTCDS active delay from -IOR active	5	28
t673	RTCDS inactive delay from -IOR inactive	5	28

Table 5-6. *INTA Sequence*

INTA Sequence			
		Min.	Max
t682	IRQn pulse width (low)	100	
t685	XD<7:0> valid from -INTA active		120
t686	XD<7:0> hold from -INTA inactive	8	

Table 5-7. *Miscellaneous Signals*

Miscellaneous Signals			
		Min.	Max
t698	-PSRSTB inactive delay from Vcc	5 us	
t699	-PSRSTB active pulse width	5 us	
t6100	VRT bit valid delay (intrnl)		2 us
t6105	XDIR high from -INTA active	3	30
t6106	XDIR high from -IOR active	3	30
t6107	XDIR low from -INTA inactive	3	30
t6108	XDIR low from -IOR inactive	3	30
t6110	SA valid from -REF active	13	50
t6111	SA hold from -REF inactive	0	25
t6112	AEN active from -REF active		40
t6113	AEN inactive from -REF inactive		40
t6121	-LPAR pulse width	15	
t6122	-IOCHCK pulse width	15	
t6123	PCUNMIIN pulse width	15	
t6130	INTCLR active delay from -IOW active		30
t6134	IOCHRDY pulse width	96 SCLK	
t6135	INTCLR inactive delay from -IOW inactive		25
t6138	INTCLR active delay from RESET4 active		20
t6139	INTCLR inactive delay from RESET4 inactive		20

Section 6

356 Timing Diagrams

Section 6

356 Timing Diagrams

Figure 6-1. 82C356 Miscellaneous Signals

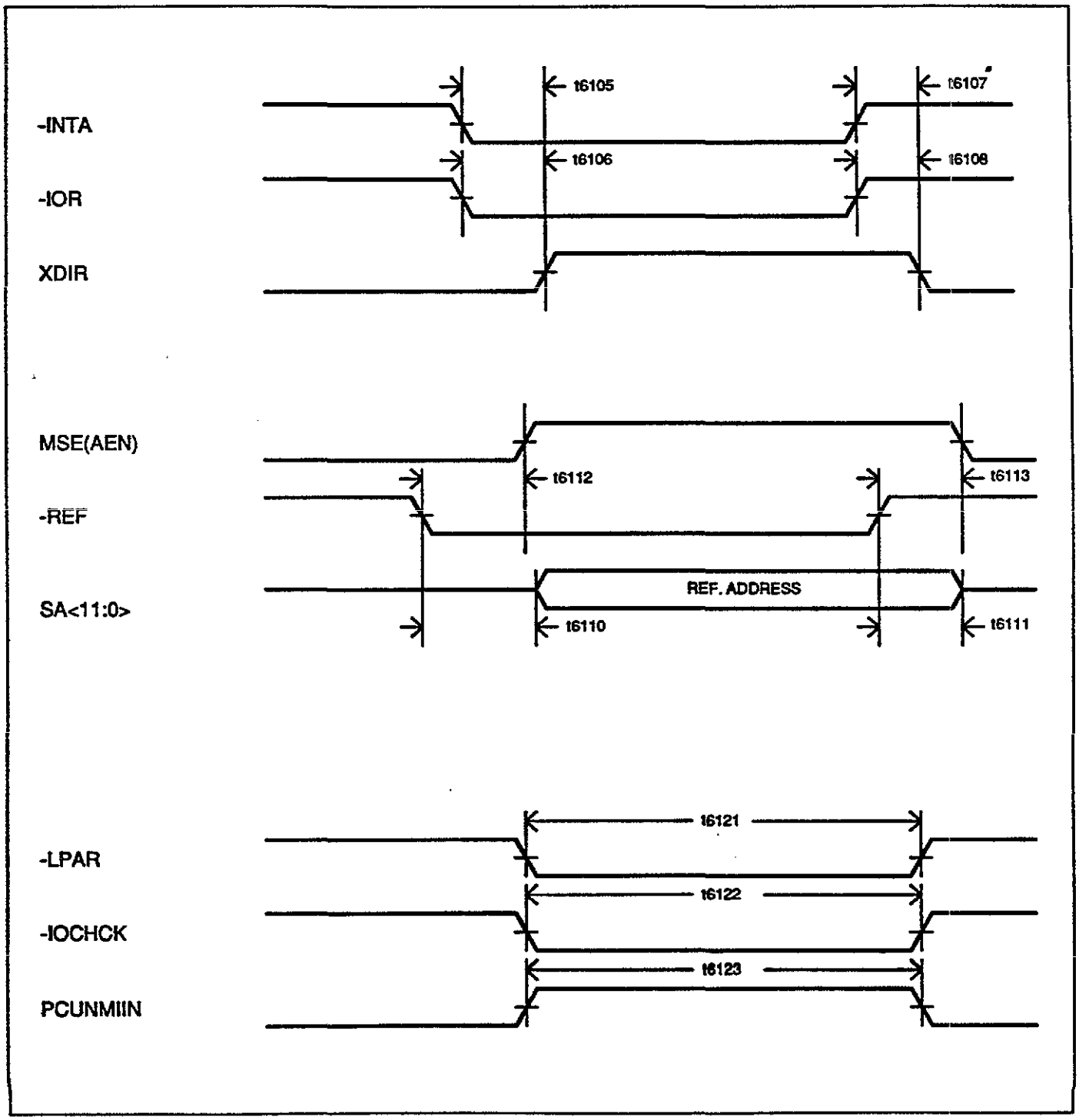


Figure 6-2. Power-ON Timing

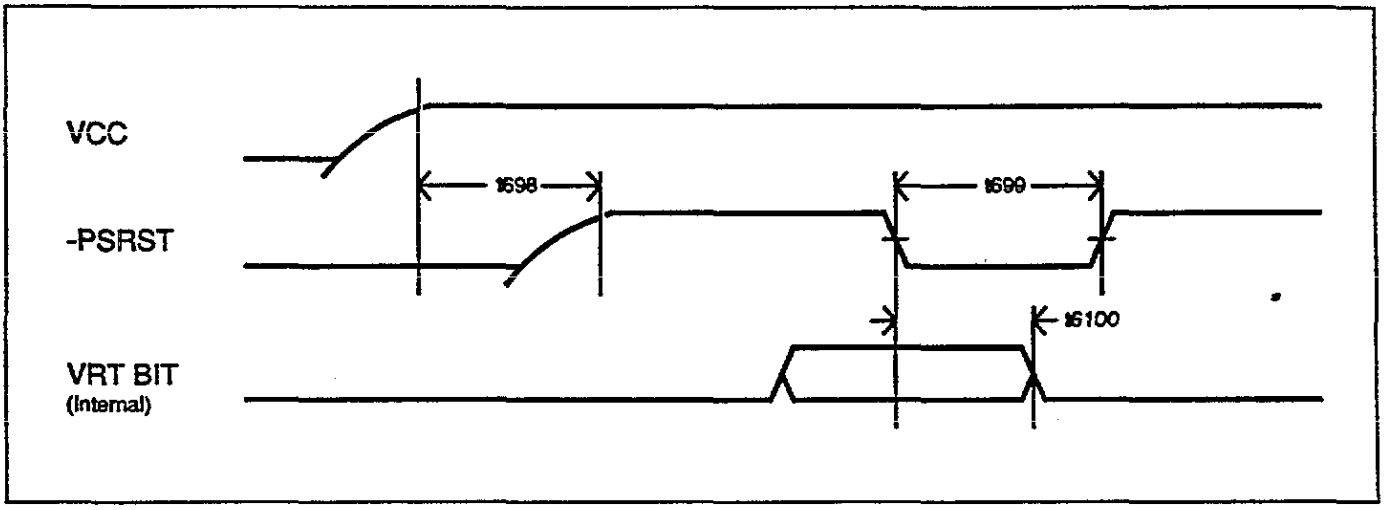


Figure 6-3. 82C356 INTA Sequence

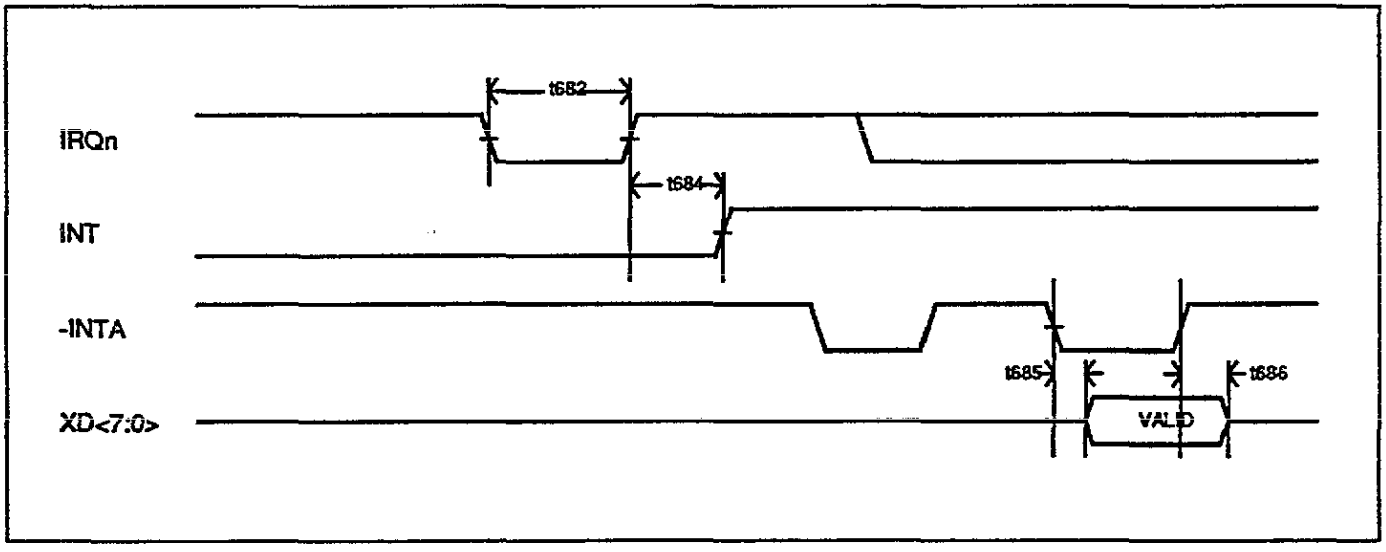
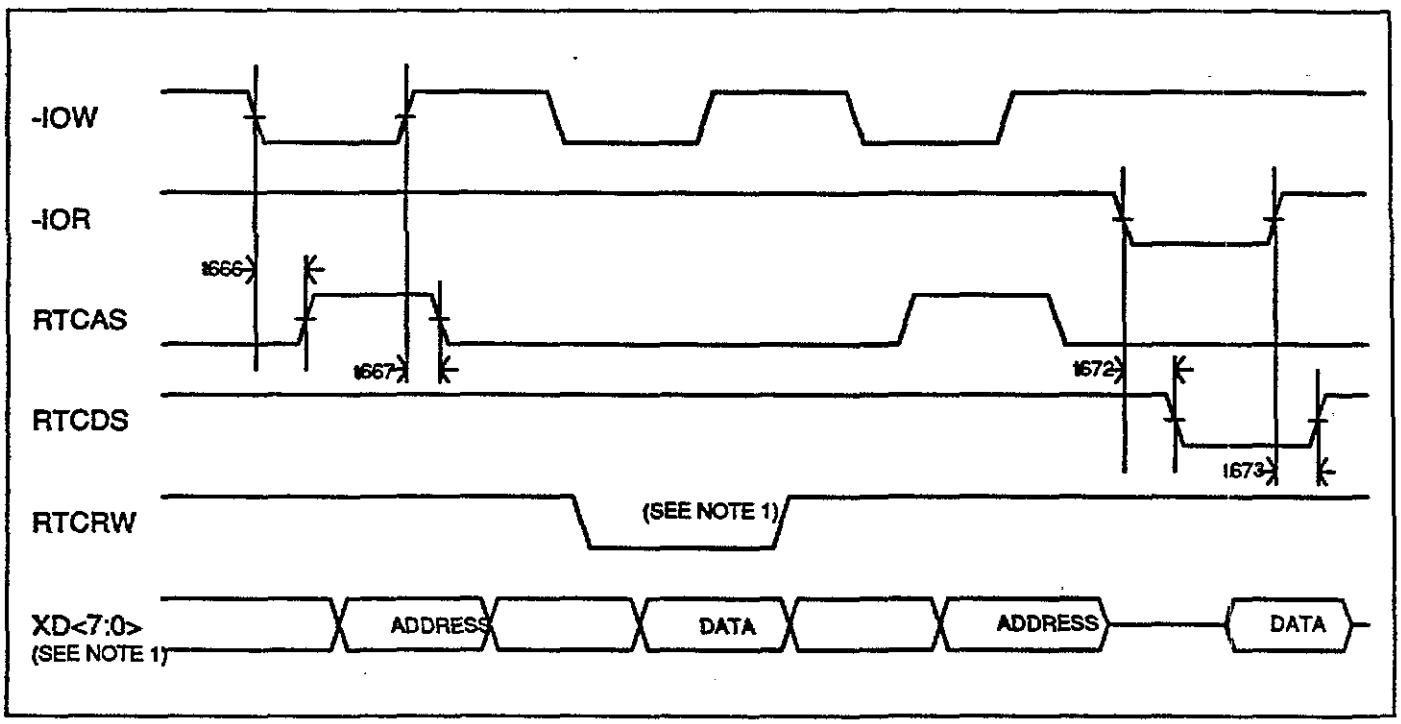
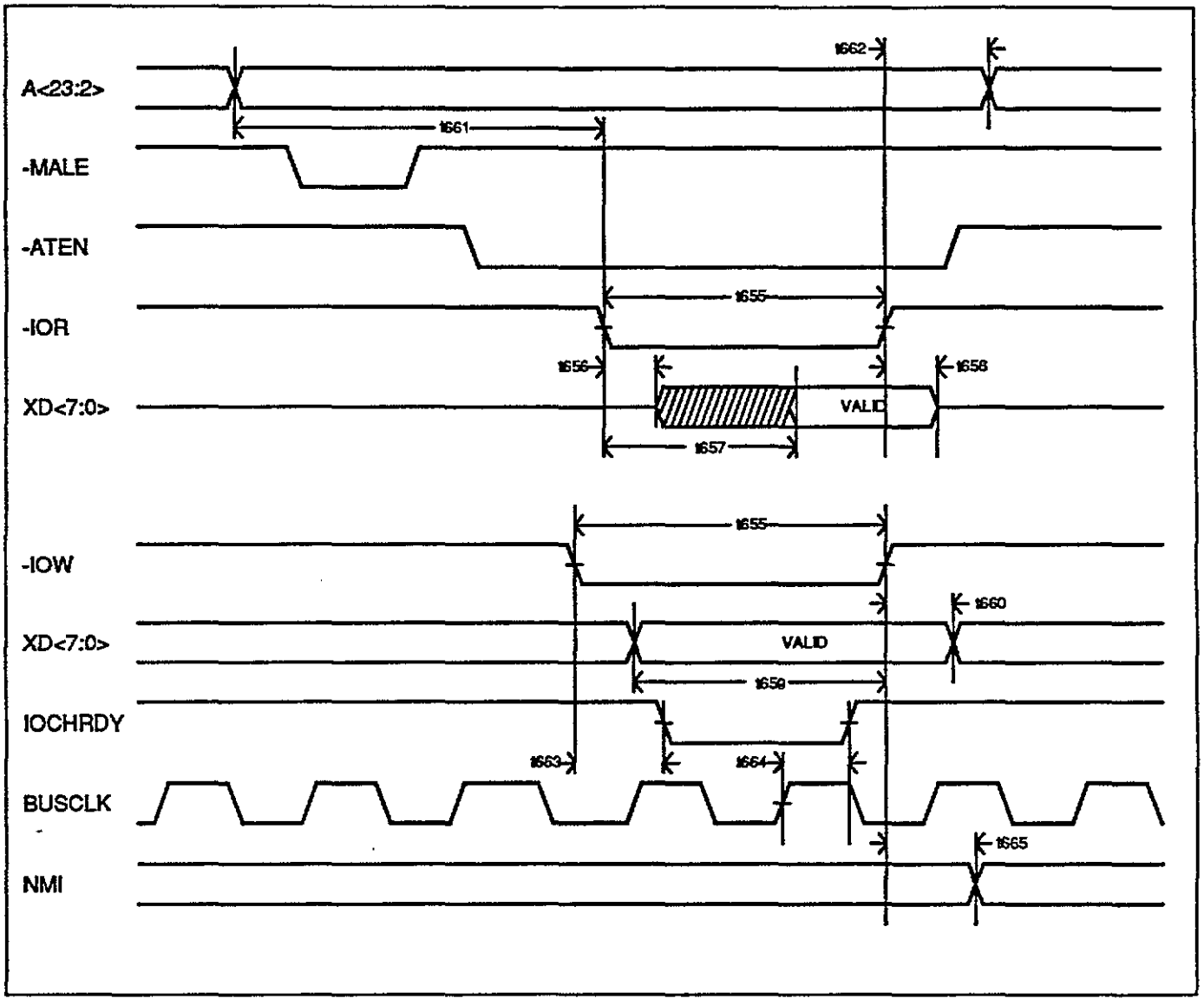


Figure 6-4. *82C356 External RTC Timing*



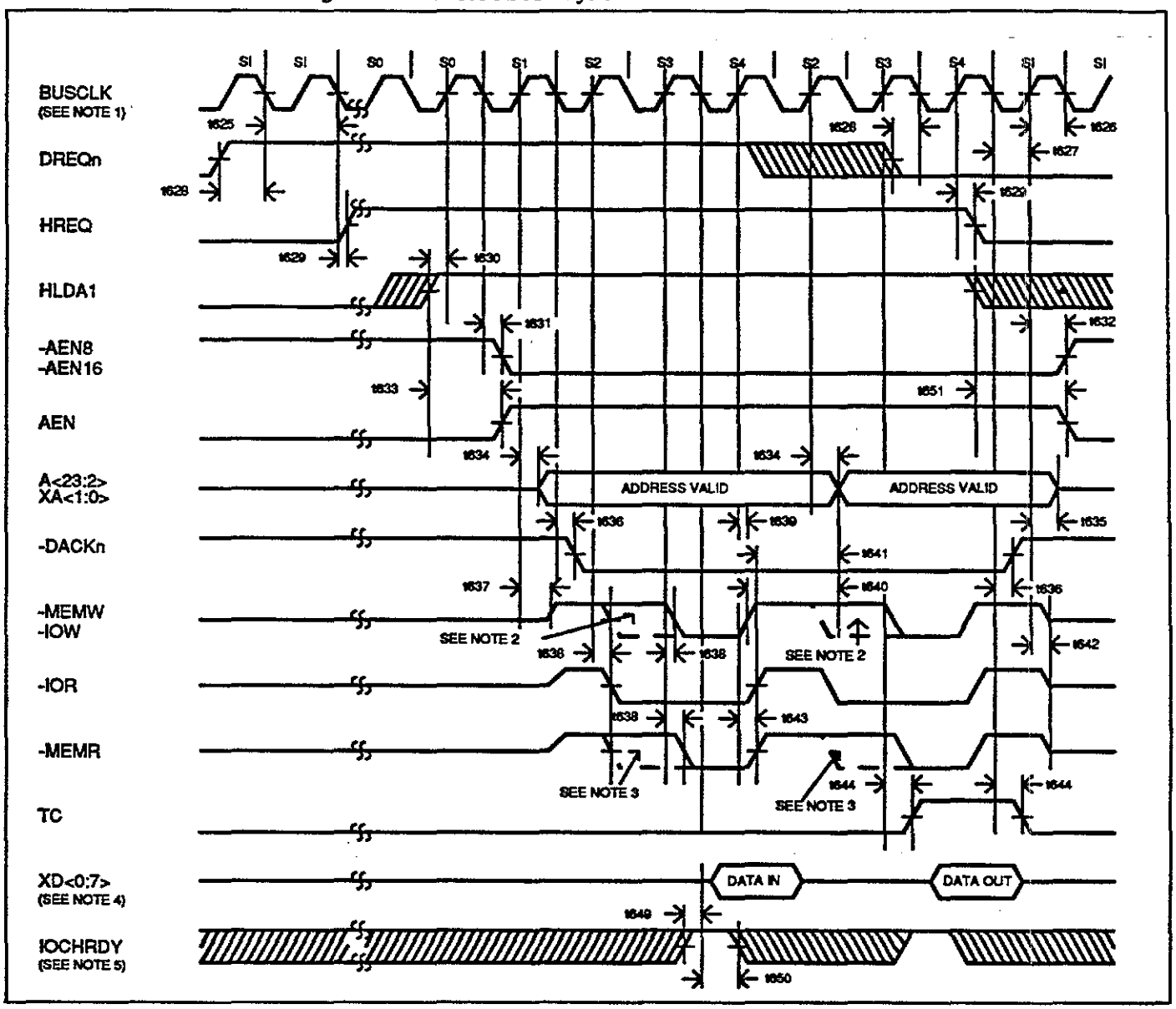
Note 1 Timing requirements for XD <7:0> are shown for completeness. They are required by the MC14681A but not the 82C356.

Figure 6-5. 82C356 Internal Register Access



Note Enabled if XD7 = 1.
Disabled if XD7 = 0.

Figure 6-6. 82C356 DMA Cycle



Note

1. All timings referenced to BUSCLK are independent of the state of the clock select bit in the configuration register. BUSCLK shown in this diagram is the undivided clock directly from the input.
2. Extended write mode selected.
3. Extended read mode selected.
4. Data bus during Memory-to-Memory transfer.
5. IOCHRDY input timing.

Figure 6-7. 82C356 Master Cycle

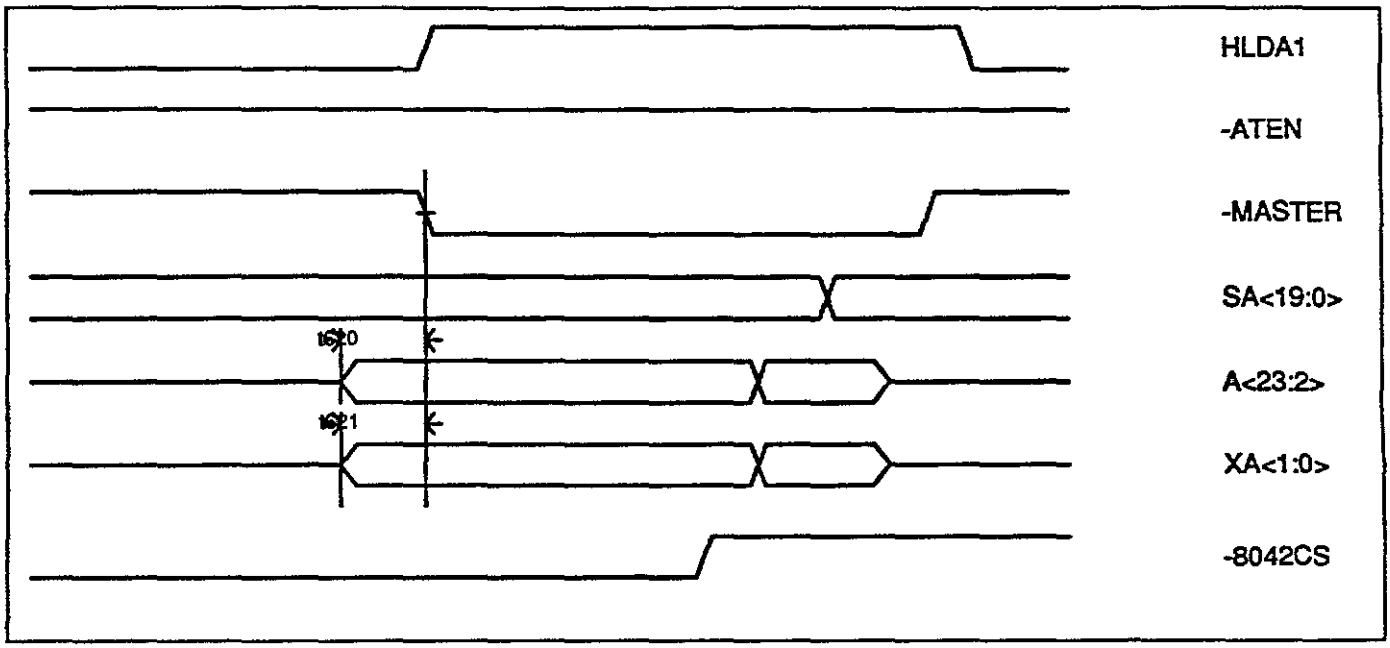
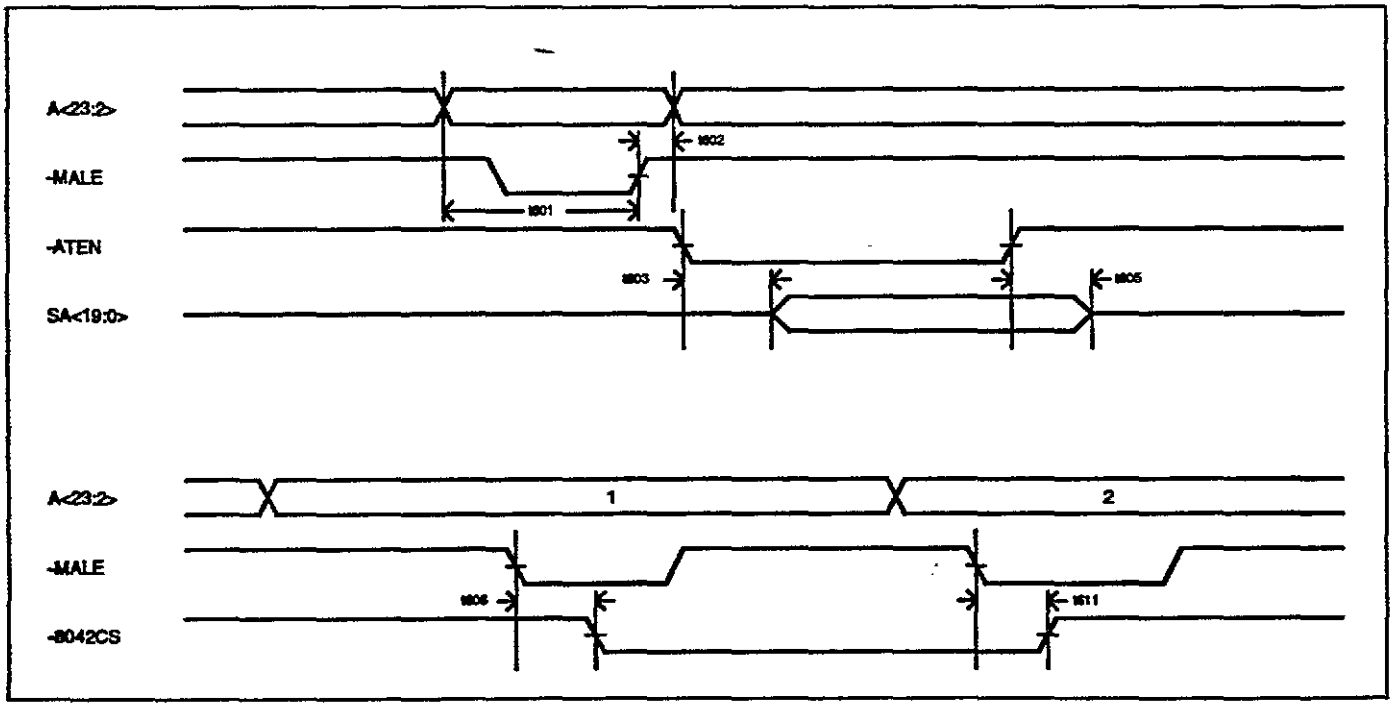


Figure 6-8. 82C356 Address Buffer Timing for CPU AT Cycles



Note HLDA, -REF, -MASTER inactive.

Section 7

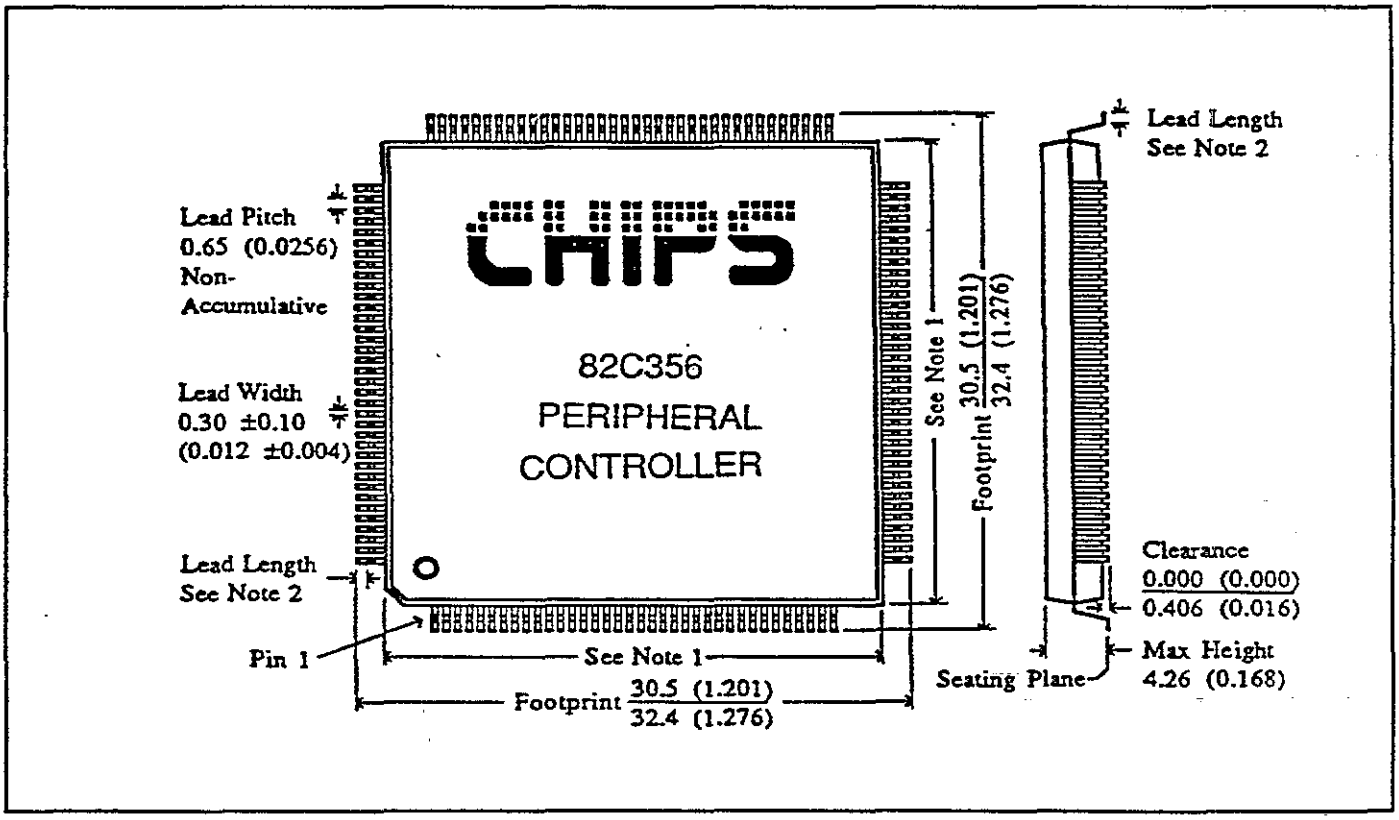
356 Physical Dimensions

Section 7

356 Physical Dimensions

7.1 82C356 Peripheral Controller

Figure 7-1. 144-Pin Plastic Flat Package

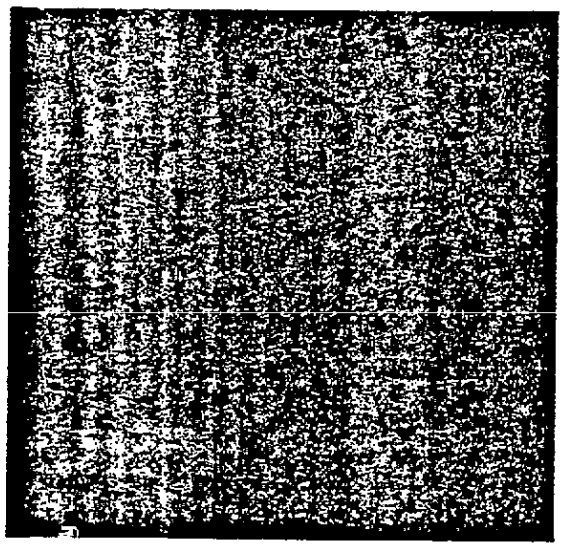


Legend minimum in inches (minimum in milimeters) / maximum in inches (maximum in milimeters)

Note 1 Package Body Size = 26 ±0.2 (1.024 ±0.008) (Package Vendor = Toshiba)
Package Body Size = 28 ±0.2 (1.102 ±0.008) (All Other Package Vendors)

Note 2 Lead Length = 1.20 ±0.2 (0.047 ±0.008) (Package Vendor = Toshiba)
Lead Length = 0.65 ±0.3 (0.026 ±0.012) (Package Vendor = Seiko)
Lead Length = 0.80 ±0.2 (0.031 ±0.008) (All Other Package Vendors)

This page is intentionally left blank.



CHIPS and Technologies, Inc.
3050 Zanker Road
San Jose, California 95134
Phone: 408-434-0600
Telex: 272929 CHIPS UR
FAX: 408-434-6452

Publication No:DB04
Stock No:012004-001
Revision No: 2.0